

Tableau Server on Linux Administrator Guide

Version 2021.2; Last Updated 4/20/2023

Copyright © 2024 Tableau Software®. [Legal & Privacy](#)



Contents

About Tableau Help	cc
Addressing Implicit Bias in Technical Language	cc
What's New and Changed in Tableau Server	1
New in version 2021.2	1
Server or site administrator	1
Tableau Server in a Container	1
Multiple analytics extensions in a site	2
New Server service - Collections	2
Configure settings for extracts in web authoring	2
Streamlined site settings for Ask Data	2
Bring Einstein Discovery predictions into your Prep flows on the web	3
Share explanations from Explain Data in published dashboards	3
Daily removal of embedded external assets without downstream connections	4
New Subscription jobs configuration key	4
Configure custom OAuth	4
Add-ons	5
Tableau Catalog - part of the Data Management Add-on	5
Resource Monitoring Tool - part of the Server Management Add-on	5
Automated Process Discovery	6
CLI support for Environment setup	6

New CLI commands	6
Node Selector	7
Agent Compatibility	7
External Repository - part of the Server Management Add-on	7
Content Migration Tool - part of the Server Management Add-on	7
Encrypted communication channel with Active Directory is required	8
Secure SAML configuration is enforced	8
Site SAML secure configuration verification	9
Tableau Prep Flow Service has been renamed to Tableau Prep Minerva Service ..	9
New configuration key for setting maximum time for single subscription job	9
Features Introduced in Previous Versions of Tableau Server	10
Added or changed in previous versions	10
Version 2021.1	10
Server or site administrator	10
Integration of Einstein Discovery Predictions	10
Server authorization-to-run (ATR) service	11
Zero downtime licensing	11
Specify ports that should not be used during dynamic port allocation	11
Block ingestion of complex content	11
Use personal access tokens for admin impersonation	12
StartTLS encryption method supported for LDAP connections to AD	12
Backup performance improvements	12

- New Server Process - Content Exploration Service 12
- Redesigned notifications center now includes notifications for sharing and comments 13
- Add-ons 13
- Azure Key Vault - part of the Server Management Add-on 13
- External File Store and External Repository- part of the Server Management Add-on 13
- Resource Monitoring Tool - part of the Server Management Add-on 14
- Tableau Catalog- part of the Data Management Add-on 14
- Change in default behavior for snapshot backups 15
- SHA-256 hash algorithm enforced for SAML assertions with IdP 15
- New SAML configuration keys 15
- Client and CA certificates used for mutual SSL must meet minimum security requirements 16
- Improvements to JVM maximum heap size scaling for Tableau Catalog ingestion and elastic server 17
- Old notifications do not appear in new notifications center 17
- Version 2020.4 17
- Server or site administrator 18
- Enable and configure Catalog or Metadata API indexing limits for content 18
- Activate and register multiple product keys 18
- Configure Tableau Server for offline maps 18
- Turn off Web Page objects 18

Improved control of extract query workloads	19
Automatic configuration of query cache size	19
Custom sign in note appears for server sign in errors	19
New processes related to Tableau Prep flows	20
Ask Data Usage and Adoption admin view	20
New tsm schedules commands	20
Checkpoint upgrades	20
Additional dynamically configurable keys for logging	20
Create extracts on the web	21
Create and interact with flows on the web	21
Add-ons	21
Tableau Resource Monitoring Tool updates- part of the Server Management Add-on	21
Tableau Server External Repository- part of the Server Management Add-on	22
Tableau Catalog updates - part of the Data Management Add-on	22
Tableau Prep Conductor updates - part of the Data Management Add-on	23
SHA1 certificates disabled for SSL mutual authentication	23
wgserver.domain.whitelist config key has been deprecated	24
CPU requirement update	25
Checkpoint Upgrade for Tableau Server	25
Upgrades from 10.5 must be done in stages	25
PostgreSQL upgrade	26

PostgreSQL database connections require updated driver	26
Visual improvements to user, group, and site dialogs	26
Changes to default JVM heap size for Tableau Catalog ingestion	26
Version 2020.3	26
New command to list dynamically configurable keys	27
Set the Site Time Zone for Extracts	28
Automatically Suspend Extract Refreshes for Inactive Workbooks	28
Manage Licenses for User Groups with Grant Role on Sign In	28
Tableau Catalog updates - part of the Data Management Add-on	29
Use admin tools to get indexing status for Catalog and Metadata API	29
Tableau Content Migration Tool updates - part of the Advanced Management ...	30
Tag Stale Content for Archiving	31
Data Acceleration for Workbooks that use the Object Model Framework	31
Version 2020.2	31
Combined What's New and What's Changed in Tableau Server	32
Applies to label for Data Management and Server Management Add-ons	32
Use Relationships to Combine Data for Multi-table Analysis	33
Create Metrics to Track Data	34
New Admin Views	34
Changes to Create and Modify Schedule	35
Data Acceleration	35
Extract Query Load Balancing	35

Dynamic Configuration	35
Tableau Catalog updates - part of the Data Management Add-on	36
Analytics extensions (external service) connections must be reconfigured	37
Oracle data source connections require updated driver	38
Version 2020.1	38
Run Now Server and Site Setting	39
New configuration options for Hyper	39
Support for Login-based License Management	39
Support for Kerberos delegation with Vertica data source	39
Support for managing site role limits	40
Updates for Permission Dialog	40
Lock nested projects	40
Tableau Server External File Store (Server Management Add-on)	40
Tableau Catalog updates - part of the Data Management Add-on	41
tableau-server-obliterate script now preserves backup and log files	42
Improvements to the tsm topology deploy-coordination-service command	42
Large statistical file extracts consume system resources	43
Version 2019.4	43
Request access	44
Configure recommendations for views	44
Enable or disable Ask Data for a site	45
Personal access tokens	45

Secure SMTP	45
Test Email Configuration	45
Create Extracts on the Web	45
New Installers for Tableau Server and tabcmd	45
Custom welcome banner	46
Limit visibility of user information	46
Manage Sandboxed Extensions	46
New tabcmd installer for Windows	47
Updates to Tableau Server Job management	47
Updates to tsm maintenance cleanup command	48
Changed behavior for Copy Link when sharing a view	48
Version 2019.3	48
Add PDF Attachments to Subscription Emails	49
Tableau Catalog features—part of the Data Management	49
Tableau Catalog on the server—part of the Data Management	50
Tableau Metadata API	50
Job Management	50
Support for Embedded Credentials in Bootstrap Files	50
Advanced Management	51
Support for Debian 9 Linux Distributions	52
Extract Encryption at Rest	52
Date-range Support for Ziplogs	53

Tableau Server Usage Data	53
Add a Note to Tableau Server Sign In Page	53
Support added for Italian	54
Default cipher suite no longer includes default support for old ciphers	55
Tableau Server Service Manager has been removed	56
Usage data is sent to Tableau	57
Upgrade without needing to provide a password	58
Some tsm command options are no longer configurable	58
Version 2019.2	58
Direct access with TSM CLI	60
Local authentication password policy improvements	60
Select from ten different languages for Tableau	60
Set a site-level start page	61
Elastic Server on multiple nodes	62
SSL connection to TabPy external service	62
Configuring the threshold for suspending data-driven alerts	62
Configuring logging levels for TSM processes	62
Minimum hardware requirements have changed	63
Upgrade requires at least 15% free disk space on each node	63
"Referrer-Policy" HTTP header added	63
Map-related internet addresses have changed	63
Version 2019.1	64

Web authoring auto recovery	65
Option to install to non-default location on RHEL-distros	65
Microservice Container Processes	65
Pluggable Authentication Modules (PAM)	66
TSM client authentication	66
Ask Data and Elastic Server Processes	66
Tableau Prep Conductor Server Process	66
Tableau Data Management Add-On	67
Node roles for Backgrounder Process	67
SSL connection to Rserve external service	67
Install to non-default locations (RHEL-like distributions only)	68
High-resolution thumbnails	68
Project card images	69
Version 2018.3	69
Materialize calculations in extracts already published to Tableau Server	70
Secure extract data at the row level	70
Improved content browsing (in-release beta)	70
Support for Content Security Policy	71
Improved support for LDAP	71
Connections to external services support SSL	71
SAML now supports PKCS#8 key files	71
Improved key and secrets refresh process	71

Backup without compression (added in version 2018.3.2)	72
Connecting to secondary trusted Active Directory domains may result in error ..	72
Floating worksheet captions become transparent	73
Changes to some Data Engine tsm configuration set options	73
SSL Off-loading and SAML	73
Version 2018.2	73
Include colleagues on data conversations with @mentions	74
Create a support case from the TSM UI or CLI	75
Manage Dashboard Extensions in Tableau Server	75
Automated daemon port configuration	75
Amazon Linux 2 now supported	76
Moving the repository in a two-node cluster not supported	76
Updates to Tableau Server Apache Logs	76
The Read Only site role is now deprecated	77
The tabcmd initialuser command prompts for password	77
New ports added	78
Automatic port assignment	78
Manual port assignment	78
Ports added in 2018.2	78
Version 2018.1	79
Resize headers on the web	79
Streamline discussion threads by deleting comments	80

- New user-based term licenses are now available 80
- New and renamed site roles are now available 80
- Improved support for managing web data connectors 80
- Removed tabadmin command options 81
- Updated tsm commands 81
- Updated upgrade-tsm script 82
- Version 10.5 82
- Optimized server performance for data alerts 83
- Nested projects 83
- Filtered search hidden by default 83
- Tableau Data Engine with Hyper 84
- Extracts use the new .hyper format 84
- Changes to the way values are computed 85
- Tableau Services Manager 85
- Hot topology support 85
- Lightweight Directory Access Protocol (LDAP) support 85
- Tableau Server configuration changes 86
- Changes to extracts 87
- Version compatibility 87
- Viewer (can publish) and Unlicensed (can publish) site roles removed 88
- Nested projects and content navigation 88
- Nested projects and tabcmd 88

Filtered search hidden by default	89
Plan Your Deployment	91
Server Administrator Overview	91
Architectural overview	92
Tableau and your data	93
User access	94
Where should I install Tableau Server in my network?	94
Sizing and scalability	95
Tableau Server management model	96
Administrative roles	97
Tableau Server administrator	97
TSM administrator	97
Tableau portal administrator	98
Management tools	98
Security	99
Tableau Services Manager Overview	99
Functionality	100
Components	100
TSM Authentication	101
Custom PAM service definition	101
TSM authorization group	102
Connecting TSM clients	102

- Infrastructure Planning 103
 - Before you install... 103
 - Hardware recommendations 104
 - Operating system requirements 107
 - Installation directory 108
 - Data directory 108
 - Tableau Prep Conductor 109
 - Additional requirements 109
 - Package requirements 113
- Disk Space Requirements 114
 - Publishing extracts to Tableau Server 114
 - Refreshing extracts 114
 - Creating many workbooks 115
 - Logging 115
 - Backup and restore processes 115
 - Restore disk space requirements 116
- Recommended Baseline Configurations 117
 - Hardware Recommendations 117
 - Estimating Disk Space 118
 - Baseline Configurations 119
 - Single Server Installations 119
 - Recommendations 119

Server Configuration	120
Multi-Node Installations	122
Two Node Installation - Specialized for extract heavy environments	122
Recommendations	122
Server Configuration	124
Two Node Installation - Specialized for flow environments	127
Server Configuration	128
High Availability Installations (HA)	131
Recommendations	131
Server Configuration	131
Virtual Machines and Public Cloud Deployments	134
Beyond Baseline Configurations	134
Disaster Recovery Considerations	134
Tableau Server Scalability	135
Identity Store	136
Local identity store	136
External identity store	136
LDAP bind	137
LDAP with GSSAPI (Kerberos) bind	138
LDAP over SSL	138
System user and groups	139
User	139

Groups	140
Authenticating clients	140
User Management in Deployments with External Identity Stores	141
Arbitrary LDAP directories	141
User binding behavior on sign in	141
Active Directory	142
Active Directory user authentication and Tableau Server	142
Active Directory user name attributes and Tableau Server	142
Adding users from Active Directory	143
Adding user groups	143
Sync behavior when removing users from Active Directory	143
Domain nicknames	144
Support for multiple domains	145
Duplicate display names	145
Sign in to Tableau Server with NetBIOS name	146
Domain Trust Requirements for Active Directory Deployments	146
Duplicate bind accounts for domain trust	147
Connecting to live data in one-way trust scenarios	148
Communicating with the Internet	148
How Tableau communicates with the internet	149
Configuring Proxies for Tableau Server	152
Who should read this article?	152

Configure a forward proxy server	152
Configuring Tableau Server on Linux to work with a forward proxy	153
Server crash reporter	156
How a reverse proxy works with Tableau Server	156
Proxy servers and SSL	157
Mobile access	157
Reverse proxy and user authentication	157
Configure Tableau Server to work with a reverse proxy server	158
Configure the reverse proxy server to work with Tableau Server	162
Validate reverse proxy setup	164
Related topics	165
Deploy	167
Install and Configure Tableau Server	167
Other installation methods	168
Before you begin	168
Installation steps	168
Before you install... ..	169
Hardware recommendations	169
Operating system requirements	172
Installation directory	173
Data directory	174
Tableau Prep Conductor	174

- Additional requirements 175
 - Package requirements 179
- Minimum Hardware Requirements and Recommendations for Tableau Server .. 179
 - Minimum Hardware Requirements for Testing and Prototyping 180
 - Minimum Hardware Recommendations 182
- Install and Initialize TSM 185
 - Prerequisites 185
 - Review optional initialization parameters 185
 - Install Tableau Server 186
 - Install the Tableau Server package 186
 - Initialize TSM 188
 - Next step 191
- Activate and Register Tableau Server 191
 - Prerequisite 191
 - Use the TSM web interface 192
 - Use the TSM CLI 195
 - Activate Tableau Server 195
 - Register Tableau Server 195
 - Next step 197
- Activate Tableau Server Using the Authorization-To-Run (ATR) Service 197
 - Select the Server ATR option 198
 - Set or change the Server ATR duration 198

Move a Server ATR license to another Tableau Server	199
How do ATR leases work?	200
Activate the same license on another Tableau Server	200
Remove product keys that were activated using Server ATR	200
Stop using Server ATR for activation	202
Offline Activation When Installing Tableau Server	202
Offline activation overview and prerequisites	203
In this article	203
Log in to Tableau Services Manager	206
Step 1 - Generate an offline activation request file	207
Step 2 - Upload the offline activation request to Tableau	208
Step 3 - Initialize or activate your license	208
First time license activation	208
Subsequent license activation	209
Deactivate Tableau Server Offline	209
Configure Initial Node Settings	210
Prerequisite	210
Use the TSM web interface	211
Identity store settings	212
Gateway port	216
Product usage data	216
Sample workbook installation	217

Initializing install	217
Use the TSM CLI	218
Configure identity store settings	218
Configure gateway settings (optional)	219
Configure product usage data (optional)	219
Configure sample workbook installation (optional)	219
Apply pending configuration changes	220
Verify LDAP configuration (Optional)	220
Initialize and start Tableau Server	221
Next Step	221
Configuration File Example	222
Entities vs keys	223
Server Usage Data	224
Disable Usage Data Sharing	224
Disabling the sharing of usage data at install	224
Disabling the sharing of usage data after install	224
Use the TSM web interface	224
Use the TSM CLI	225
Basic Product Data	225
Disabling sharing of Basic Product Data on individual computers	226
Disabling the sharing of Basic Product Data at the enterprise level	227
Add an Administrator Account	227

Prerequisites	228
Use web UI	228
Use tabcmd CLI	229
Next steps	229
Validate Installation	230
Prerequisites	230
Install PostgreSQL driver and validate installation	230
Initial Node Installation Defaults	230
Jump-start Installation	232
Before you begin	232
Step 1: Install Tableau Server package and start Tableau Services Manager ..	233
Step 2: Activate and register Tableau Server	234
Step 3: Configure local identity store	235
Step 4: Finalize installation	235
Step 5: Install PostgreSQL drivers	236
Configure Local Firewall	236
Single-node configuration	237
Multi-node cluster configuration	238
Before you begin	238
Step 1: Configure initial node.	238
Step 2: Configure additional nodes	239
Automated Installation of Tableau Server	241

- Benefits of using the automated installer 241
- Before you begin 242
- How to use the automated installer 243
 - Configure Tableau Server for a forward proxy 244
- Perform an unattended installation 245
 - Step 1: Install the automated installer 245
 - Step 2: Create files to provide additional configuration information required to run the automated install 245
 - Step 3: Run the automated install 247
- Install Tableau Server in a Disconnected (Air-Gapped) Environment 248
 - Prerequisites 248
 - Installing Tableau Server on an Air-Gapped Computer Running Linux 249
 - Activating Tableau Server in an Air-Gapped Environment 252
 - Step 1. Transcribe data from your air-gapped system into an activation request template. 252
 - Step 2. Initialize or activate your license 254
 - First time license activation 254
 - Subsequent license activation 254
- Displaying Maps in an Air-Gapped Environment 255
 - Configuring Tableau Server to use offline maps (version 2020.4.0 and later): .. 255
 - Workarounds to display maps in Tableau Server in an offline environment (versions earlier than 2020.4.0): 255
- Tableau Server in a Container 256

Introduction	256
Limitations	257
Tableau Server in a Container - Quick Start Guide	257
Before you begin	257
Tableau Server in a Container Setup Tool	261
Download the necessary files	261
Installation	262
Complete the registration form	262
How to use the tool	263
Orchestration	264
Customizing the image	265
Setting Environment Variables at Build Time	265
Environment File	265
Example Usage:	265
Environment Variables	265
Drivers, Certificates, and other files	267
Tableau Administrative Views	267
Defining a Custom Base Image	268
Base Images and Security	269
Hardening Tableau Server in a Container	270
Tableau Server in a Container Image	270
Prerequisites	270

- Run the configure-container-host script 270
- Running the Image 271
 - Basic Run Arguments Summary 271
- Automate Initial Admin 272
- Licensing 273
 - Licensing in Containers 273
 - License Environment Variable 273
 - License File 273
 - Requested license lease time 274
- Running an Uninitialized Image 274
- Interacting with the Image 275
- TSM Web UI and Remote CLI 275
 - Create a TSM Remote User 276
 - Set the password for the TSM Remote User 276
 - Secrets File (Recommended) 276
 - Environment Variable 277
- How to Rotate the TSM Remote User's Password 277
 - Start a new Tableau Server in a Container 277
 - Rotate the password manually inside a running container 278
- Initial Configuration Options 278
 - Runtime Environment Variables 278
 - Readonly Environment Variables 286

Build-Time Environment Variables	288
Tableau Server Configuration Overrides	289
Pre-initialization and Post-initialization Commands	289
Pre-initialization script	290
Post-initialization script	290
Instructions	290
User Configuration	290
Tableau Server in a Container Utilities and Tools	291
File Permission Management	291
Password Management	291
Examples	292
Example of using a secrets file	292
Example contents of a secrets file	292
Example	292
Configuring Tableau Server after it is running	293
Status	293
Liveness Check	293
Readiness Check	293
Persisting Data	294
Tableau Server Data	294
Static Hostnames	294
Complete Example	295

- Backup and Restore 295
 - Backup in Tableau Server Container 295
 - Restore inside Tableau Server Container 296
 - Prerequisites 296
- Migrating from Tableau Server to Tableau Server in a Container 297
- Upgrading Tableau Server Versions 297
 - Upgrading through Upgrade-Image method 297
 - Example: 298
 - Upgrading though Backup-Restore method 300
- Multi-node Tableau Server in a Container 300
 - Multi-Node Basic Usage 301
 - Initial node 301
 - Additional node 301
 - Exposing Licensing and TSM ports 302
 - Resolving Hostnames 302
 - Bootstrapping additional nodes 303
 - Security Considerations 303
 - Bootstrap Expiration 304
- Transferring the Bootstrap File 304
 - Transfer the file over a secure network 304
 - Use a network file mount 304
 - Other 304

Starting additional nodes	304
Configuring additional nodes	305
Tableau Server Feature Considerations	305
Active Directory	305
Set AD Domain Controller	305
Import AD certificate to Tableau Server Keystore	306
Troubleshooting	306
Installation and initialization failures	307
Debugging failing installation	307
Tableau Support and Kubernetes	308
Logs	308
Extracting All Logs	308
Ziplogs	308
Manual Tar Command	309
Navigating Logs and Debugging Tips	309
Starting The Container (initial / install)	309
Service Logs - Server Runtime	311
Stopped Container	313
Failure to set authentication properties	313
Failure during fresh startup (e.g. why isn't Tableau Server starting?)	313
Failure during restart or starting a container with existing data	314
Server Won't Start PostGRES (or other processes)	314

- Deployment Configuration Examples 315
 - Docker 315
 - Tableau Server in a Container Basic Usage 315
 - Tableau Server in a Container Basic Usage with Automated Initial Admin User 315
 - TSM only mode 315
 - Multi-Node Basic Usage 315
 - Initial Node 315
 - Additional node 316
 - Externalize Data Usage 316
 - Init-Container Basic Usage 316
 - Basic Restore from Backup Single-Node 317
- Docker-Compose 317
- Install Tableau Server in the Cloud 318
- Post Installation Tasks 319
 - Security Hardening Checklist 319
 - Installing security updates 319
 1. Update to the current version 320
 2. Configure SSL/TLS with a valid, trusted certificate 320
 3. Disable older versions of TLS 320
 4. Configure SSL encryption for internal traffic 321
 5. Enable firewall protection 321

6. Restrict access to the server computer and to important directories	322
7. Generate fresh secrets and tokens	322
8. Disable services that you're not using	323
JMX Service	323
9. Verify session lifetime configuration	323
10. Configure a server allowlist for file-based data sources	324
11. Enable HTTP Strict Transport Security for web browser clients	325
12. Disable Guest access	325
13. Set referrer-policy HTTP header to 'same-origin'	326
14. Configure TLS for SMTP connection	327
15. Configure SSL for LDAP	328
Change List	328
Configure SMTP Setup	329
Secure SMTP	329
Use the TSM web interface	330
Use the TSM CLI	331
SMTP CLI configuration reference	332
TLS ciphers	336
Files and Permissions in TSM	338
Setting permissions for individual files	340
Setting permissions for directories	340
Configure Server Event Notification	341

- Use the TSM web interface 341
- Use the TSM CLI 345
 - Set notification values individually 345
 - Set all notification values with a single json file 347
- Configure Data Cache 348
- Database Drivers 349
 - Install drivers in a cluster 349
- Server Crash Reporter 349
 - Configure Server Crash Reporter 350
 - Use the TSM web interface 351
 - Use the TSM CLI 351
 - Crash reporter settings 352
 - Configuration template 352
 - Configuration file reference 353
- Navigate the Admin Areas of the Tableau Web Environment 354
 - Access based on site role and number of sites 354
 - Server administrator 355
 - Site administrator 357
 - Server administrator tasks 358
 - Site administrator tasks 359
- Distributed and High Availability Tableau Server Installations 359
 - Prerequisite 361

Licensing	361
Creating a distributed Tableau Server installation	361
Creating a highly available (HA) Tableau Server installation	362
If an initial node fails	363
Recover from an initial node failure	364
Configure Coordination Service ensemble on additional nodes	364
Add Client File Service (CFS) to additional nodes	364
Tableau Server service license check	365
Distributed Requirements	365
Hardware	365
Hardware Guidelines for High Availability	366
Software	366
Installation location	367
Networking and Ports	367
Best Practices	367
SSL	368
Distributed Installation Recommendations	368
Recommendations for all installations	369
Install and Configure Additional Nodes	370
Use the TSM web interface	371
Generate the node bootstrap file	371
Install and initialize an additional node	372

- Configure the additional node with Cluster Controller 374
- Use the TSM CLI 376
 - Generate the node bootstrap file 376
 - Install and initialize an additional node 378
 - Configure the additional node 380
- Install drivers 382
 - Drivers and administrative views 382
- Database Drivers 383
 - Install drivers in a cluster 383
- Example: Install and Configure a Three-Node HA Cluster 383
 - A Single Server System 384
 - A Three-Node System 386
 - Configuration steps 387
 - Before you begin 388
- Use the TSM web interface 388
 - Step 1: Install the initial node 388
 - Step 2: Generate a bootstrap file for the additional nodes 388
 - Step 3: Install and initialize an additional node 389
 - Step 4: Install and initialize a second additional node 392
 - Step 5: Configure the new node with a minimum topology 395
 - Step 6: Deploy a Coordination Service ensemble 396
 - Step 7: Configure Client File Services (CFS) on additional nodes 398

Step 7: Configure processes for the second node	399
Step 8: Configure processes for third node	400
Use the TSM CLI	401
Step 1: Install the initial node	401
Step 2: Generate a bootstrap file for the additional nodes	401
Step 3: Install an additional node	403
Step 4: Install a second additional node	405
Step 5: Add a process to the additional nodes	408
Step 6: Deploy a Coordination Service ensemble	409
Step 7: Configure Client File Services (CFS) on additional nodes	411
Step 7: Configure processes for the second node	412
Step 8: Configure processes for third node	413
Step 9: Configure firewall rules (optional)	414
Add a Load Balancer	414
Add a Load Balancer	414
Configure Tableau Server to work with a load balancer	416
Load balancer configuration key-value pairs	416
Configuration template example	418
Deploy a Coordination Service Ensemble	419
Hardware requirements	419
The Coordination Service Quorum	420
Number of Coordination Service instances to use	420

- Deploy a new Coordination Service ensemble 422
- Configure Client File Service 424
 - Configure CFS on additional nodes 425
- Repository Failover 426
 - Automatic repository failover 426
 - Manual repository failover 426
 - Preferred active repository 427
- Recover from an Initial Node Failure 427
 - If an initial node fails 428
 - General requirements 429
 - Move the TSM Controller, License Service, and Activation Service to another node 429
- Recover from a Node Failure 435
 - General requirements 435
 - Removing a Failed Node 436
- Configure Nodes 439
 - Use the TSM web interface 439
 - Use the TSM CLI 441
 - Adding processes to a node 442
 - Changing the number of processes on a node 443
 - Removing all instances of a process from a node 443
 - Moving all instances of a process from one node to another node 444

Workload Management through Node Roles	445
Backgrounder Node Roles	445
Using Backgrounder Node Roles	445
Configuration Options	446
License Requirements	447
Considerations	448
File Store Node Roles	449
Guidelines to Optimize for Extract Refresh and Backup/Restore Workloads.	450
Fine Tune Extract Query Workload Management	451
Configuration Options	452
License Requirements	453
Who can do this	453
Install Tableau Server on a Two-Node Cluster	453
Restart Multi-Node Tableau Server Computers	454
Maintain a Distributed Environment	454
Move the Repository Process	455
Use the TSM web interface	456
Add a new instance of the repository.	456
Remove an instance of the repository.	457
Use the TSM CLI	457
Add a new instance of the repository.	458
Remove an instance of the repository.	458

Move the File Store Process	459
Use the TSM web interface	459
Adding a second instance of File Store	459
Decommissioning and removing an instance of file store	460
Use the TSM CLI	461
Adding a second instance of file store	461
Decommissioning and removing an instance of File Store	462
Move the Messaging Service Process	463
Use the TSM web interface	463
Moving the Messaging Service	463
Use the TSM CLI	464
Moving the Messaging Service	464
Remove a Node	465
Prerequisites for removing a node	465
Use the TSM web interface	466
Use the TSM CLI	467
Configure Tableau Server for High Availability with Coordination Service-Only Nodes	468
Prerequisite	469
Deploy an ensemble on Coordination Service-only nodes	469
Add a Load Balancer	471
Configure Tableau Server to work with a load balancer	473

Load balancer configuration key-value pairs	473
Configuration template example	475
Migrate	475
Changing the Identity Store	476
Warning	477
Methods for restoring content and permissions	477
User filters	478
User names and the Tableau Identity store	478
Method 1: Use site export and import	479
Method 2: Fresh installation—users republish content	480
Back up, remove, and then reinstall	480
Step 1: Back up Tableau Server	480
Step 2: Remove Tableau Server	480
Step 3: Reinstall Tableau Server with new authentication type	480
Migrate to New Hardware	481
Tableau Server product key activation	483
Migrate Tableau Server from Windows to Linux	483
Step 1: Plan your migration	484
Step 2: Create a backup	486
Migrate from 2018.1 or earlier	486
Migrate from 2018.2 or later	486
Step 3: Install Tableau Server on Linux and restore the Windows backup	487

- Step 4: Test Tableau Server on Linux **488**
- Step 5: Install Tableau Server on Linux in your production environment and
restore the Windows backup **490**
- Migrate from Tabadmin to the TSM CLI **490**
 - Tabadmin commands with a corresponding TSM CLI command **491**
 - Tabadmin commands with no corresponding TSM CLI command **495**
- Migrate Tableau Server from an On-Premises Computer to a VM in the Cloud ... **498**
 - Prerequisites **498**
 - To migrate Tableau Server to a VM in the cloud **498**
- Upgrade Tableau Server Overview **501**
 - Choose your upgrade path **502**
 - Preparing for Upgrade **502**
 - Server Upgrade - Minimum Hardware Recommendations **503**
 - Server Upgrade - Review What's Changed **507**
 - Encrypted communication channel with Active Directory is required **507**
 - Secure SAML configuration is enforced **508**
 - Site SAML secure configuration verification **508**
 - Tableau Prep Flow Service has been renamed to Tableau Prep Minerva
Service **509**
 - New configuration key for setting maximum time for single subscription job **509**
 - Change in default behavior for snapshot backups **509**
 - SHA-256 hash algorithm enforced for SAML assertions with IdP **510**

New SAML configuration keys	510
Client and CA certificates used for mutual SSL must meet minimum security requirements	511
Improvements to JVM maximum heap size scaling for Tableau Catalog ingestion and elastic server	511
Old notifications do not appear in new notifications center	511
SHA1 certificates disabled for SSL mutual authentication	512
wgserver.domain.whitelist config key has been deprecated	513
CPU requirement update	513
Checkpoint Upgrade for Tableau Server	513
Upgrades from 10.5 must be done in stages	514
PostgreSQL upgrade	514
PostgreSQL database connections require updated driver	514
Visual improvements to user, group, and site dialogs	515
Changes to default JVM heap size for Tableau Catalog ingestion	515
Data Acceleration for Workbooks that use the Object Model Framework ..	515
Analytics extensions (external service) connections must be reconfigured	515
Oracle data source connections require updated driver	516
tableau-server-obliterate script now preserves backup and log files	517
Improvements to the tsm topology deploy-coordination-service command	517
Large statistical file extracts consume system resources	517
New tabcmd installer for Windows	518

Updates to Tableau Server Job management	518
Updates to tsm maintenance cleanup command	519
Changed behavior for Copy Link when sharing a view	519
Default cipher suite no longer includes default support for old ciphers	520
Tableau Server Service Manager has been removed	521
Usage data is sent to Tableau	522
Upgrade without needing to provide a password	522
Some tsm command options are no longer configurable	523
Minimum hardware requirements have changed	523
Upgrade requires at least 15% free disk space on each node	523
"Referrer-Policy" HTTP header added	524
Map-related internet addresses have changed	524
Install to non-default locations (RHEL-like distributions only)	524
High-resolution thumbnails	525
Project card images	525
Connecting to secondary trusted Active Directory domains may result in error	526
Floating worksheet captions become transparent	526
Changes to some Data Engine tsm configuration set options	526
SSL Off-loading and SAML	527
Moving the repository in a two-node cluster not supported	528
Updates to Tableau Server Apache Logs	528

The Read Only site role is now deprecated	529
The tabcmd initialuser command prompts for password	529
New ports added	529
Automatic port assignment	529
Manual port assignment	529
Ports added in 2018.2	529
Removed tabadmin command options	530
Updated tsm commands	531
Updated upgrade-tsm script	531
Server Upgrade - Gather Configuration Details	531
Take screen shots	532
Record object counts	532
Record firewall configuration	532
Verify TSM Controller certificate expiry	532
Gather asset files	533
Gather custom configuration information	534
Secure SMTP	534
Analytics extensions	534
External Repository	535
Port customization	536
Server Upgrade - Verify Licensing Status	536
Server Upgrade - Verify Accounts	537

- Server Upgrade - Back Up Tableau Server538
- Server Upgrade - Download Setup539
- How Tableau Server Upgrade Works539
- What's Changed - Things to Know Before You Upgrade540
- Upgrading from 2018.1 and Later (Linux)540
 - Support and services to help with Tableau Server upgrades542
 - Server Upgrade - Disable Scheduled Tasks542
 - Single-Server Upgrade -- Run Setup542
 - Run Setup543
 - Multi-node Upgrade -- Run Setup546
 - Run Setup546
 - Multi-node Upgrade -- Run Setup on Each Node547
 - Run Setup548
 - Multi-node Upgrade -- Run Upgrade Script548
 - Run Upgrade script548
- Verify Tableau Server Upgrade550
 - Verify Tableau Service processes550
 - Verify TSM global settings551
 - Enable subscriptions and scheduling551
 - Verify user access551
 - View published workbooks551
 - Verify publishing workbooks and data sources552

Verify Tableau Prep Builder	552
Verify count of Tableau objects	552
Verify API functionality	552
REST API	552
Compatibility testing	553
Post Upgrade Cleanup	553
Uninstall previous version	553
Upgrade Tableau Server on Linux from 10.5	554
Upgrade to 10.5.x	556
Install 2018.x or later, up to 2020.3.x	557
Run TSM commands	558
Migrate 10.5.x to single user	559
Upgrade to 2018.x or later, up to 2020.3.x	559
Related topics	561
Test the Upgrade	561
Prepare a test environment	562
Upgrade the test environment	563
Confirm that everything works as expected	563
Performance and user acceptance testing	564
Test new features	564
Communicate about the upgrade	564
Troubleshoot Tableau Server Install and Upgrade	565

- General Troubleshooting Steps565
- Common Tableau Server Install Issues565
 - Installation logs location565
 - Unable to log into TSM or Tableau Server (sign in screen redisplay after entering credentials)566
 - Multiple install attempts fail566
 - Install fails due to hardware requirements567
 - Install or upgrade fails due to CPU requirements567
- Common Tableau Server Upgrade Issues568
 - Maps do not display or display incompletely after upgrading568
 - Upgrade script error: "Tableau Server Version change validation failed."568
 - Upgrade logs location568
 - Upgrade multi-node, initializing additional node fails with "Enter your credentials again" error569
 - Upgrading fails due to lack of disk space569
 - Upgrade fails on RebuildSearchIndex job570
 - Upgrade fails on 2020.4.0 or later570
- Common Settings Import Issues571
 - Import of settings file causes "not present on any node" validation error due to missing services571
 - Import of settings file causes "configuration value you specified does not match" error572
 - "You cannot directly modify instances of the Coordination Service" error574

If you see this error after importing a settings file:	574
If you see the error when setting the process count for Coordination Service manually:	575
Starting Tableau Server	576
Tableau Server cannot determine if it fully started	576
Tableau Server doesn't start	576
Reindexing Tableau Server Search & Browse	576
Problems that can be solved by rebuilding Search & Browse index	576
Activating Tableau Server	577
Tableau Server license activation fails	577
Confirm you can access the licensing server	577
Verify the date and time	579
Force the product key to be read again	579
Send the contents of trusted storage to Tableau Support	579
tabcmd Installation Problems	580
Installing tabcmd separately	580
Problems installing tabcmd on Linux	580
Java is not installed	580
Incorrect version of Java is installed	580
Uninstall Tableau Server	581
Uninstalling and completely removing Tableau Server	581
Uninstall a Tableau Server package	582

- Reinstall a Tableau Server package that was accidentally uninstalled583
- Remove Tableau Server from Your Computer 583
 - What tableau-server-obliterate does585
 - Preserving Tableau Server backup and log files586
 - Running the tableau-server-obliterate script586
 - To completely remove Tableau Server without removing server licensing 586
 - To completely remove Tableau Server and licensing587
- Help Output for tableau-server-obliterate Script588
 - Output588
- Manage Individual Sites 591**
 - What is a site 591
 - Site administrator tasks592
 - Steps for setting up your site593
 - Planning a Site594
 - Projects595
 - Users and groups595
 - Site roles and permissions596
 - Extract refresh schedules597
- Site Settings Reference597
 - Accessing site settings597
 - General section599
 - Authentication section (Tableau Online) 606

Bridge section (Tableau Online)	607
Extensions section	607
Manage Users and Groups	608
Add Users to a Site	608
Site administrator access to user management	608
Add local users to a site	609
Add Active Directory users to a site	611
Remove local users	612
Related information	612
Set Users' Site Roles	612
How user licenses, site roles, and content permissions work together	613
Change a user's site role	614
General capabilities allowed with each site role	615
What this article covers and where to find what's not covered here	615
Tableau site roles as of version 2018.1	616
Who can publish content	620
Site roles and Active Directory import and synchronization	621
View, Manage, or Remove Users	623
Set the User Authentication Type for SAML	631
Notes	632
Import Users	632
Add users from a CSV file	633

How users' site roles are assigned or maintained	634
Importing at the server level in multi-site environments	635
Importing to a single-site environment	636
Multi-site versus single-site import	636
CSV Import File Guidelines	637
CSV file format requirements	637
Required columns in the CSV file	638
Additional import file options	638
Improve performance for large CSV files passed through tabcmd	639
Notes	640
CSV settings and site roles	640
CSV import example for Tableau Server	641
Manage Site User Visibility	642
Limit user visibility	642
Best practices for limiting user visibility	644
Restore Full User Visibility	645
Guest User	645
Guest user permissions	645
Enable or disable Guest access	646
Additional Guest account characteristics	647
Groups	647
Add Users to a Group	648

Add users to a group (Users page)	648
Add users to a group (Groups page)	649
Create a Local Group	650
Create Groups via Active Directory	651
Synchronize External Directory Groups in a Site	655
Set the minimum site role for users in an external directory group	655
What happens when users are removed in the source external directory? ...	657
What happens when a user name changes in the source external directory .	657
What happens when an external directory group is removed from Tableau Server?	658
Synchronize External Directory Groups on the Server	658
Before you begin	658
Synchronize external directory groups on a schedule	659
Synchronize all external directory groups on demand	659
View synchronization activity	660
Set the minimum site role for users in an external directory group	661
What happens when users are removed in the source external directory? ...	662
Improving group synchronization performance	662
Synchronize All Active Directory Groups on a Schedule	663
1 Set a minimum site role for synchronization	664
2 Set the schedule	664
3 Run synchronization on-demand (optional)	665

4 View the status of synchronization tasks	666
Grant License on Sign In	666
Activate Grant role on sign in	667
Modifying user roles with Grant role on sign in	670
Removing users affected by Grant role on sign in	670
Delete Groups	671
Dashboard-based Custom Portals	672
Sketch out a portal design	672
Gather images for logos and navigation elements	673
Lay out text, images, and selected sheets on a dashboard	673
Link dashboard elements to content	673
Publish, test, and refine your portal	675
Manage Content Access	675
Set a Site's Web Authoring Access and Functions	675
Turn web authoring on or off for a site	676
Notes	676
See which sites allow web authoring	676
Configure cross-database join options	677
Set Web Edit, Save, and Download Access on Content	678
Why allow users to work on the site directly	678
Web authoring pros and cons	678
Managing permissions to help users avoid content proliferation	679

Coordinate edit and save capabilities with site roles for the appropriate level of access	680
Site role access	680
Configure Projects, Groups, and Permissions for Managed Self-Service	680
Create a project team and adopt a permissions strategy	681
Steps to coordinate projects and groups	682
1. Plan your permissions	682
Use a closed permissions model for managed content	683
2. Remove permissions that will cause ambiguities	683
3. Create groups	684
4. Assign permissions to the groups	685
Create permission rules	685
Example	686
5. Create projects and adjust permissions	686
Example project structure	686
Help project leaders manage content and users find it	688
6. Lock permissions in each project	689
Automate working with groups and projects	689
Next steps	689
Learn how Tableau and some of our customers address governance and self-service	690
Use Projects to Manage Content Access	691

Why use projects	692
When to create project hierarchies (example)	693
Why not use sites?	693
Project-level administration	694
Project ownership and project leader access in project hierarchies	694
Actions project-level administrators can take on projects	695
How the Default project acts as a permissions template	695
Prevent publishers from changing permissions on content they own	696
Add Projects and Move Content Into Them	696
Add a top-level or child (nested) project	697
Move a content resource to another project	698
How moving projects affect permissions	699
Delete a project	700
Required access level for moving content	700
Required site role	700
Required permissions for the project that users move content to	701
Required permissions for the project that users move content from	701
Add a Project Image	701
Set a project image	702
Let Site Users Request Access to Content	705
Default settings	706
Configure project permissions	706

Change project permissions	706
Change content permissions	708
Set permissions on content	708
Set permissions on a view	710
Permissions	710
Permissions fundamentals	712
Set permissions	713
Project-level permissions	713
Content-level permissions	716
Set permissions on content	716
Set permissions on a view	718
Set permissions at publish	718
Permission settings for specific scenarios	720
Saving, publishing, and overwriting	720
Web Editing and Web Authoring	721
Required Permission Capability Settings	722
Data access for published Tableau data sources	722
Move content	724
Metrics	725
Permissions for metrics	725
Prevent metric creation	725
Show or Hide Sheet Tabs	725

Turn off tabbed views to allow independent view permissions	727
Collections	727
Private collections	728
Permission Capabilities and Templates	728
Templates	729
Copy and paste permissions	729
Capabilities	729
Projects	730
View template	730
Publish template	730
Workbooks	730
View template	730
Explore template	731
Publish template	731
Administer template	732
Views	732
Data Sources	732
View template	732
Explore template	733
Publish template	733
Administer template	733
Ask Data Lenses	733

View template	733
Publish template	733
Administer template	734
Other content types	734
Manage Permissions with Projects	735
Project administration	736
Set a project leader	737
Lock content permissions	738
Set content permissions (lock a project)	739
Change content permissions	740
Move projects and content	741
Effective permissions	741
Evaluate permission rules	743
Evaluate permissions set at multiple levels	744
Permissions on views	745
Permissions, Site Roles, and Licenses	746
Site roles and their maximum capabilities	747
Projects	747
Workbooks	747
Data Sources	749
Data Roles	749
Flows	750

Ask Data Lenses	750
Metrics	751
Collections	751
Quick Start: Permissions	752
1 Add users to groups	752
2 Open a project's permissions settings	753
3 Create a permissions rule	753
4 View a user's effective permissions	754
Site roles	755
Permissions evaluation	755
Manage Content Ownership	757
Who can change or be given ownership, by content type	757
Considerations for changing content ownership	759
Change the owner of a content resource	760
Manage Permissions for External Assets	761
Tableau Catalog indexes content and assets	762
How does Tableau Catalog work?	763
Permissions on metadata	763
Access metadata	763
Permissions on Tableau content	764
Permissions on external assets using derived permissions	764
Set permissions on individual external assets	767

Access lineage information	770
Who can do this	774
Manage Data	780
Tableau Server Data Sources	780
Managing data sources	781
Restrictions	783
Extract Upgrade to .hyper Format	783
Changes to extracts in version 10.5	784
Tasks that cause extract upgrade	785
Impact of extract upgrade	785
Performing tasks on Tableau Server 10.5	786
Downloading from Tableau Server 10.5	787
Automating refresh and append tasks	788
Why keep an extract in .tde format?	789
How to keep an extract in .tde format	789
What to expect after extract upgrade	789
Set the Site Time Zone for Extracts	790
Create Extracts on the Web	790
Create extracts in Web Authoring	790
Extract an Embedded Data Source in Web Authoring	791
Define your Extract Settings	792
Conditions for using the Physical Tables option	794

Limitations	796
Create extracts in Content Server	797
Extract a Published Data Source on Content Server	797
Extract an Embedded Data Source on Content Server	798
Limitations	798
Keep Extracted Data Fresh	799
Monitor and Manage Extracts	799
View Data Source Attributes	799
View data sources by name	799
View a list of connections	800
Use Certification to Help Users Find Trusted Data	801
How certification helps users find trusted data	802
Create guidelines for selecting data to certify	803
Who can certify data	803
How to certify data	804
Certify data sources	804
Certify databases and tables	804
Keep Data Fresh	805
Manage Refresh Tasks	805
See also	806
Refresh Data on a Schedule	806
Quick Start: Refresh Extracts on a Schedule	808

1 Set up a schedule on the server	808
2 Enable scheduled extract refreshes and failure emails	809
3 Publish a workbook with an extract	810
4 Monitor refresh performance	811
Automate Refresh Tasks	812
Handle Extract Refresh Alerts	812
Resolving Extract Refresh Problems	814
Automatically Suspend Extract Refreshes for Inactive Workbooks	815
Configure the feature	815
Notifications	816
Resume suspended extract refreshes	816
Edit Connections on Tableau Server	816
Authentication types for Google, Salesforce, and WDC data	819
Google authentication options	820
Salesforce.com authentication options	820
Monitor progress	822
Cube Data Sources	823
Web Data Connectors in Tableau Server	823
Before you run connectors on Tableau Server	824
Manage connectors in a safe list	825
Updating WDC safe lists requires a server restart	825
Add connectors to the safe list and secondary safe list	825

Allow or disallow WDCs or WDC extract refreshes	827
Remove one or more WDCs from the safe list	827
List all WDCs on the safe list	827
Refresh the extract for a connector	827
Troubleshooting	828
Testing and Vetting Web Data Connectors	829
Examine the source	829
Test the web data connector in an isolated environment	829
Monitor the traffic created by the web data connector	830
Test the performance and resource usage of the web data connector	830
Enable Tableau Catalog	830
Before enabling Catalog	831
Required versions	831
What to expect when enabling Catalog	831
Initial ingestion	831
Initial ingestion speed	832
Disk space to store metadata	832
Memory for non-interactive microservices containers	832
Best practices for enabling Catalog	833
Summary of steps to enable Catalog	834
Enable Catalog	834
Step 1: Determine the amount of content on Tableau Server	834

Step 2: Estimate how long initial ingestion will take	835
Step 3: Decrease the time of initial ingestion	835
Step 4: Activate the Data Management	837
Step 5 (optional): Turn off Catalog capabilities for each site	837
Step 6: Run the tsm maintenance metadata-services command	838
Step 7: Monitor initial ingestion progress and validate its status	838
Step 8: Configure SMTP Setup	838
Step 9 (optional): Turn on Catalog capabilities for each site	838
Troubleshoot Catalog	839
Timeout limit and node limit exceeded messages	839
Missing content	840
Performance after initial ingestion	841
Out of memory errors	841
Disable Catalog	841
Turn off Catalog capabilities for each site	841
Stop indexing metadata on Tableau Server	842
Get Initial Ingestion Status	842
Step 1: Authenticate using the REST API	842
Step 2: Make a GET request	842
Status values from the response	843
Example response	844
Get Eventing Status	848

Step 1: Authenticate using the REST API	848
Step 2: Make a GET request	848
Status values from the response	848
Example response	848
Use Lineage for Impact Analysis	849
Navigate lineage	849
Embedded asset appears in External Assets	852
Lineage and custom SQL connections	853
Catalog doesn't support cubes	853
Mismatch between lineage count and tab count	853
Workbook count mismatch example	854
Use email to contact owners	855
Set a Data Quality Warning	855
About data quality warnings	856
Who can see the data quality warning	857
How to set a quality warning	858
How to set a monitoring quality warning	859
Set high visibility for a data quality warning	860
Remove a data quality warning	863
Who can do this	864
Manage Dashboard Extensions in Tableau Server	864
Before you run extensions on Tableau Server	864

Control dashboard extensions and access to data	866
Change the global setting enabling extensions on the server	866
Change the default settings for a site	866
Identifying an extension	867
Add extensions to the safe list and configure user prompts	868
Block specific extensions	869
Test Network-enabled extensions for security	869
Examine the source files	870
Understand data access	870
Test the extension in an isolated environment	871
Monitor traffic created by the dashboard extension	871
Configure Connections with Analytics Extensions	871
Server SSL	872
Enable analytics extensions	872
Configure analytics extensions settings	873
Edit or delete an analytics extension connection	875
Client requirement: Intermediate certificate chain for Rserve external service ..	875
Script errors	876
Determining analytics extensions usage	876
Configure Einstein Discovery Integration	876
Einstein Discovery dashboard extensions	877
Einstein Discovery analytics extensions	878

Einstein Discovery Tableau Prep extensions	878
Creators: Connect to data on the web	879
Open the Connect to Data page	879
Tableau Server	881
Connect to data On this site	882
Connect to files	882
Use connectors	882
Tableau Server connectors	883
Tableau Catalog Supported Connectors	884
Tableau Online	884
Connect to data On this site	884
Connect to files	884
Use connectors	885
Tableau Online Connectors	885
Tableau Catalog Supported Connectors	886
Use Dashboard Starters	886
Tableau Public	886
Connect to files	887
Use connectors	887
Tableau Public Connectors	887
After you connect	887
Keep data fresh in web authoring	888

Run Initial SQL	888
To use initial SQL	889
Parameters in an initial SQL statement	890
Examples	891
Defer execution to the server	892
Security and impersonation	892
Troubleshoot 'create table' for MySQL and Oracle connections	892
For MySQL connections, tables are not listed after using initial SQL to create table	892
For Oracle connections, using initial SQL to create table causes Tableau to stall	893
Create and Interact with Flows on the Web	893
Turn flow web authoring on or off for a site	894
Enable Run Now	894
Enable Tableau Prep Conductor	895
Enable Tableau Prep Extensions	895
Turn autosave off or on	896
Use Flows	896
Create, Interact with, and Embed Views on the Web	898
Alerts and subscriptions	898
Manage Saved Credentials for Data Connections	898
Remove saved credentials	899

Test connections using saved credentials	899
Update saved credentials	900
Clear all saved credentials	901
Use Relationships for Multi-table Data Analysis	901
The Tableau Data Model	904
Layers of the data model	906
Understanding the data model	907
Build a new model	908
Multi-table model	909
Single-table model	910
Single-table model that contains other tables	911
Supported data model schemas	912
Single-table	912
Star and snowflake	913
Star and snowflake with measures in more than one table	914
Multi-fact analysis	915
Unsupported models	916
Requirements for relationships in a data model	916
Factors that limit the benefits of using related tables	917
How Relationships Differ from Joins	917
Characteristics of relationships and joins	918
Relationships	918

Joins	919
Requirements for using relationships	919
Factors that limit the benefits of using related tables	920
Where did joins go?	920
Optimize Relationship Queries Using Performance Options	922
What the Cardinality and Referential Integrity settings mean	923
Cardinality options	923
Referential Integrity options	924
Where did joins go?	924
Tips on using Performance Options	926
Terms defined	927
Automatically Build Views with Ask Data	927
Navigating to Ask Data lenses	928
Ask Data from a lens page or dashboard object	930
Rephrase your question	934
Change the viz type	934
Change fields, filters, and displayed data	935
Adjust date filters	938
Compare differences over time	940
Apply simple calculations	941
Add sheets with other vizzes	942
Share links to Ask Data vizzes	942

- Tips for success 943
- Create Lenses that Focus Ask Data for Specific Audiences 943
 - Create or configure a lens page on your Tableau site 944
 - Add an Ask Data lens to a dashboard 947
 - Apply a different lens to an Ask Data dashboard object 948
 - Change a lens name, description, or project location 948
 - See how people use Ask Data with a lens 948
 - Permissions for publishing and viewing lenses 949
- Disable or Enable Ask Data for a Site 950
- Optimize Data for Ask Data 950
 - Optimize data in Ask Data 951
 - Changing settings at the data source or lens level 951
 - Add synonyms for field names and values 952
 - Exclude values of specific fields from search results 952
 - Optimize data sources 953
 - Optimize indexing for Ask Data 953
 - Use data extracts for faster performance 954
 - Ensure that users can access the data source 954
 - Be aware of unsupported data source features 955
 - Anticipate user questions 955
 - Simplify the data 955
 - Set appropriate field defaults 956

Create hierarchies for geographic and categorical fields	956
Avoid field names that Ask Data uses for analytical functions	956
Aggregation terms	957
Grouping terms	957
Filtering and limiting terms	957
Sorting terms	958
Time terms	958
Qualifying terms	958
Discover Insights Faster with Explain Data	959
Access to Explain Data for Authors vs. Viewers	960
How Explain Data helps to augment your analysis	961
Get Started with Explain Data	961
Run Explain Data on a mark	962
Tips for using Explain Data	962
Drill into explanations	962
Why am I seeing a Reselect button?	966
View analyzed fields	968
Terms and concepts in explanations	970
Explanation Types in Explain Data	973
Contributing to the value of the mark	973
Mark Attributes	974
Extreme Values	974

Number of Records	977
Average Value of Mark	979
Relevant Single Value	980
Relevant Dimensions	982
Relevant Measures	984
Other things to explore (Unique attributes of a mark)	985
Relevant Single Value	986
Relevant Dimensions	988
Analyzed Fields in Explain Data	989
View fields analyzed by Explain Data	990
To view fields used by Explain Data for statistical analysis	991
Change fields used for statistical analysis	993
To edit the fields used by Explain Data for statistical analysis	994
Fields excluded by default	996
Requirements and Considerations for Using Explain Data	997
What makes a viz a good candidate for Explain Data	998
What data works best for Explain Data	998
Situations where Explain Data is not available	999
Control Access to Explain Data	1000
Who can access Explain Data	1000
Control who can use Explain Data and what they can see	1001
Open the Explain Data Settings dialog box	1001

Enable Explain Data in viewing mode in published workbooks and dashboards	1003
Show or hide Extreme Values explanations in viewing mode	1004
To show Extreme Values explanations in viewing mode in published workbooks	1006
To hide Extreme Values explanations and record-level details in viewing mode (published workbooks)	1007
Include or exclude explanation types displayed by Explain Data	1008
Include or exclude fields used for statistical analysis	1009
How Explain Data Works	1011
What Explain Data is (and isn't)	1012
How explanations are analyzed and evaluated	1013
What is an expected range?	1013
Models used for analysis	1014
Use Dashboard Extensions	1017
Add an extension to a dashboard	1017
Configure a dashboard extension	1018
Reload a dashboard extension	1019
Data security, Network-enabled, and Sandboxed extensions	1019
Allow or deny data access to a Network-enabled extension	1020
Ensure that JavaScript is enabled in Tableau Desktop	1021
Ensure that extensions run on Tableau Online or Tableau Server	1021
Supported web browsers for Sandboxed extensions	1021

- Supported versions of Tableau Server for Sandboxed extensions 1021
- Get support for dashboard extensions1021
- Format Animations 1022
 - Understanding simultaneous and sequential animations 1023
 - Simultaneous animations 1023
 - Sequential animations 1023
 - Animate visualizations in a workbook 1024
 - Reset animation settings for a workbook 1027
 - Completely disable all animations 1027
 - Why animations won't play 1027
 - Server rendering 1027
 - Unsupported browsers and features 1028
- URL Actions 1028
 - Open a web page with a URL action 1029
 - Create an email with a URL action 1033
 - Using field and filter values in URLs 1035
 - Including aggregated fields 1036
 - Inserting parameter values 1036
- Create a Subscription to a View or Workbook 1037
 - Set up a subscription for yourself or others 1037
 - Update or unsubscribe from a subscription 1041
 - Resume or delete suspended subscriptions 1042

See also	1042
Interact with Data in Tableau	1043
Go ahead. It's safe to click around	1043
1: What is a Tableau Site?	1043
2: Search for a viz	1044
3: Interact with Content	1046
See Details and Sort Data	1046
Filter Data	1047
Undo/Revert	1048
4: Keep up	1048
Select Background Maps	1049
To select a new background map:	1050
Set a default background map in Tableau Desktop	1050
Use the Offline background map in Tableau Desktop, Online, or Server	1051
About the Tableau background map	1052
Create and Troubleshoot Metrics	1052
How metrics are defined	1053
Timeline metrics and single-number metrics	1053
Metrics with target values	1055
See connected metrics	1056
Create a metric	1058
Overwrite a metric	1061

When you can't create a metric	1062
How metrics refresh	1063
Fix failing refreshes	1063
If the connected view is still listed	1064
If there is no connected view listed	1065
Resume suspended refreshes	1065
Metrics Appear in Tableau Catalog	1066
Manage Server	1069
Security	1072
Authentication	1072
Add-on authentication compatibility	1073
Client authentication compatibility	1074
Local authentication	1075
External authentication solutions	1076
NTLM and SSPI	1076
Kerberos	1076
SAML	1076
OpenID Connect	1077
Mutual SSL	1077
Trusted authentication	1077
LDAP	1077
Other authentication scenarios	1078

Data access and source authentication	1078
Local Authentication	1078
Password storage	1079
Configure password settings	1079
Use the TSM web interface	1080
Use the TSM CLI	1080
Configuration file reference	1082
SAML	1085
SAML Requirements	1087
Certificate and identity provider (IdP) requirements	1087
SSL Off-loading	1089
Using SSL certificate and key files for SAML	1089
User management requirements	1090
SAML compatibility notes and requirements	1091
Using SAML SSO with Tableau client applications	1095
Redirecting authenticated users back to Tableau clients	1095
XML data requirements	1095
Configure Server-Wide SAML	1098
Before you begin	1099
Use the TSM web interface	1099
Use the TSM CLI	1103
Before you begin	1103

- Step 1: Configure return URL, SAML entity ID, and specify certificate and key files 1104
- Step 2: Generate Tableau Server metadata and configure the IdP1105
- Step 3: Match assertions 1105
- Optional: Disable client types from using SAML1107
- Optional: Add AuthNContextClassRef value 1107
 - Test the configuration 1108
- Configure SAML with Salesforce IdP on Tableau Server1108
 - Enable Salesforce as a SAML Identity Provider1109
 - Configure SAML on Tableau Server 1109
 - Add Tableau Server as a Connected App in Salesforce1109
 - Enable Lightning Web Component 1110
 - Embed Tableau Views into Salesforce 1110
- Configure SAML for Tableau Viz Lightning Web Component1110
 - Requirements 1111
 - Configuring the authentication workflow1111
 - Enable in-frame authentication on Tableau Server1112
- Tableau Server Versioning1112
 - Enable in-frame authentication with your SAML IdP 1113
- Salesforce safelist domains 1113
- Salesforce IdP1113
- Okta IdP 1113

Ping IdP	1113
OneLogin IdP	1113
ADFS and Azure AD IdP	1114
Salesforce Mobile App	1114
Configure SAML with Azure AD IdP on Tableau Server	1115
Prerequisites	1115
Step 1: Verify SSL connection to Azure AD	1115
Step 2: Configure SAML on Tableau Server	1115
Step 3: Configure Azure AD claim rules	1116
Step 4: Provide Azure AD metadata to Tableau Server	1116
Azure AD App Proxy	1117
Troubleshooting	1118
Configure SAML with AD FS on Tableau Server	1118
Prerequisites	1119
Step 1: Verify SSL connection to AD FS	1119
Step 2: Configure SAML on Tableau Server	1119
Step 3: Configure AD FS to accept sign-in requests from Tableau Server	1120
Step 4: Provide AD FS metadata to Tableau Server	1123
Use SAML SSO with Kerberos Database Delegation	1125
Overview of the process	1125
Configure Tableau Server for SAML with Kerberos	1126

Configure Site-Specific SAML	1126
Prerequisites for enabling site-specific SAML	1126
Server-wide settings related to site-specific SAML	1127
Configure the server to support site-specific SAML	1128
About the commands	1128
Configure SAML for a site	1129
Update SAML Certificate	1133
Update certificate for server-wide SAML	1133
Update certificate for site-specific SAML	1135
Troubleshoot SAML	1135
SAML and Enable Automatic Logon	1135
HTTP Status 500 error when configuring SAML	1136
Signing In from the Command Line	1136
Login fails: failed to find the user	1136
Login fails: SSL offloading	1137
SAML Error Log	1137
Trailing Slash	1138
Confirm Connectivity	1138
Kerberos	1138
How Kerberos works	1139
Kerberos Requirements	1140
General requirements	1140

Active Directory requirements	1141
Kerberos delegation	1141
Understanding Keytab Requirements	1142
User authentication (SSO) in Windows Active Directory	1142
Batch file: Set SPN and create keytab in Active Directory	1143
SPN and keytab batch file contents	1144
Operating system	1147
Directory service	1148
Datasource delegation	1149
Configure Kerberos	1150
Use the TSM web interface	1150
Use the TSM CLI	1152
Confirm your SSO configuration	1152
Tableau Client Support for Kerberos SSO	1153
General browser client support	1153
Tableau Desktop and browser clients	1153
Tableau Mobile app clients	1154
Operating system and browser-specific notes	1154
Note 1: Internet Explorer or Chrome on Windows desktop	1155
Note 2: Firefox on Windows or Mac OS X desktop	1156
Note 3: Chrome on Mac OS X desktop	1157
Note 4: Mobile Safari or Tableau Mobile on iOS	1158

Note 5: Android platform	1158
More information	1158
Troubleshoot Kerberos	1158
Single sign-on to Tableau Server	1159
Troubleshooting sign-in errors on the client computer	1159
Troubleshooting sign-in errors on the server	1161
Verify Kerberos configuration script	1163
Data source SSO	1163
Delegated data source access failures	1163
Kerberos delegation multi-domain configuration	1163
Cross-domain constrained delegation	1164
Web authoring	1165
Configure Mutual SSL Authentication	1165
User authentication session time limits	1165
Certificate usage	1166
Client certificate requirements	1167
Use the TSM web interface	1168
Use the TSM CLI	1169
Step 1: Require SSL for external server communication	1169
Step 2: Configure and enable mutual SSL	1170
Additional options for mutual SSL	1171
Fallback authentication	1171

User name mapping	1171
Certificate Revocation List (CRL)	1172
How Mutual SSL Authentication Works	1172
Mapping a Client Certificate to a User During Mutual Authentication	1173
User-name mapping options	1174
Change the certificate mapping	1175
Address user-name mapping ambiguity in multi-domain organizations ..	1176
OpenID Connect	1176
Authentication overview	1177
How Tableau Server Works with OpenID Connect	1179
Requirements for Using OpenID Connect	1180
IdP account	1180
Local identity store	1180
IdP claims: mapping users	1180
Default: using email claim to map users	1181
Ignoring the domain name	1181
Using custom claims to map users	1182
Changing the sub claim	1184
Authentication context	1184
Configure the Identity Provider for OpenID Connect	1184
Configure the IdP	1184
Redirect URL	1185

- Example IdP process 1185
- Configure Tableau Server for OpenID Connect 1186
 - Use the TSM web interface 1186
 - Use the TSM CLI 1188
- Signing In to Tableau Server Using OpenID Connect 1189
 - Signing in using OpenID Connect 1189
 - Restricting sign-in to server administrators for command-line tools 1190
- OpenID Connect Authentication Request Parameters 1191
 - Configure the scope value 1192
- Changing IdPs in Tableau Server for OpenID Connect 1192
 - Changing providers 1192
- Troubleshoot OpenID Connect 1193
 - Signing In from the Command Line 1193
 - Login Failed 1193
 - Error 69: "Unable to Sign In" 1194
 - OpenID Error Log 1194
- Trusted Authentication 1195
 - How Trusted Authentication Works 1195
 - How is a trusted ticket stored? 1197
 - Add Trusted IP Addresses or Host Names to Tableau Server 1198
 - Use the TSM web interface 1198
 - Use the TSM CLI 1200

Get a Ticket from Tableau Server	1201
Display the View with the Ticket	1203
Tableau Server View Examples	1203
Embedded View Examples	1203
Optional: Configure Client IP Matching	1206
Test Trusted Authentication	1206
Step 1: Add a test user	1207
Step 2: Create a test HTML page	1207
Step 3: Retrieve a trusted ticket from Tableau Server	1209
Step 4: Test access with trusted ticket	1210
Troubleshoot Trusted Authentication	1210
Ticket Value of -1 Returned from Tableau Server	1211
HTTP 401 - Not Authorized	1212
HTTP 404 - File Not Found	1213
Invalid User (SharePoint or C#)	1213
Attempting to Retrieve the Ticket from the Wrong IP Address	1213
Cookie Restriction Error	1214
An error occurred communicating with the server (403)	1214
Personal Access Tokens	1214
Understand personal access tokens	1215
Server administrator impersonation	1216

Enable Tableau Server to accept personal access tokens during impersonation sign-in requests	1216
Create tokens	1217
Token expiry	1217
Revoke users' tokens	1217
Track and monitor usage	1218
Data Connection Authentication	1218
More information	1219
Enable Kerberos Delegation	1219
Supported data sources	1219
Requirements	1219
Web authoring	1220
Configuration process	1220
See also	1223
Enable Kerberos Delegation for JDBC Connectors	1223
Supported data sources	1223
Enable Kerberos Run As Authentication for JDBC Connectors	1224
Supported data sources	1224
OAuth Connections	1224
Overview of the OAuth process	1225
Default saved credential connectors	1226
Access tokens for data connections	1228

Access tokens for authentication from approved clients	1228
Default managed keychain connectors	1229
Token limit and storage	1229
Removing unused keychain records	1229
Scenario limitations with managed keychain	1230
Convert managed keychain to saved credentials	1230
Configure custom OAuth for a site	1231
Change Salesforce.com OAuth to Saved Credentials	1231
Create a Connected Salesforce App	1232
Configure Tableau Server for Salesforce.com OAuth	1234
Configure custom OAuth for a site	1235
Step 1: Prepare the OAuth client ID, client secret, and redirect URL	1235
Step 2: Register the OAuth client ID and client secret	1236
Step 3: Validate and update saved credentials	1237
Step 4: Notify users to update their saved credentials	1238
Manage access tokens	1238
Change Google OAuth to Saved Credentials	1238
Obtain a Client ID and Enable Google APIs	1239
Configure Tableau Server for Google OAuth	1241
Configure custom OAuth for a site	1241
Step 1: Prepare the OAuth client ID, client secret, and redirect URL	1242
Step 2: Register the OAuth client ID and client secret	1242

Step 3: Validate and update saved credentials	1244
Step 4: Notify users to update their saved credentials	1244
Create and edit Google data source	1244
Managing access tokens	1245
Configure Azure AD for OAuth and Modern Authentication	1245
Step 1. Register OAuth application for Azure	1245
Step 2: Configure Tableau Server for Azure	1245
Configure default OAuth client for Azure Data Lake Storage Gen2	1245
Configure default client for Azure Synapse, Azure SQL Database, or Dat- abricks	1246
Setting multiple connectors	1247
Configure custom OAuth for a site	1248
Step 1: Prepare the OAuth client ID, client secret, and redirect URL	1248
Step 2: Register the OAuth client ID and client secret	1249
Step 3: Validate and update saved credentials	1250
Step 4: Notify users to update their saved credentials	1251
Change Snowflake OAuth to Private Link with Saved Credentials	1251
Step 1: Obtain a client ID with Snowflake	1251
Step 2: Configure Tableau Server for Snowflake OAuth	1252
Configure custom OAuth for a site	1253
Step 1: Prepare the OAuth client ID, client secret, and redirect URL	1254
Step 2: Register the OAuth client ID and client secret	1254

Step 3: Validate and update saved credentials	1255
Step 4: Notify users to update their saved credentials	1256
Set up OAuth for Intuit QuickBooks Online	1256
Create a Connected Intuit App	1256
Configure Tableau Server for Intuit QuickBooks Online	1257
Managing access tokens	1257
Allow Saved Access Tokens	1258
Managing credentials centrally	1259
See also	1260
Troubleshoot OAuth Connections	1260
Conflict error	1260
Configure SAP HANA SSO	1261
Before you begin	1261
Configure Tableau Server SAML for SAP HANA	1262
Enable Kerberos Service Account Access	1263
Data Access with the RunAs Service Account	1264
Recommendations	1264
Requirements	1265
Configuration process	1265
Enable Kerberos Run As Authentication for JDBC Connectors	1267
Supported data sources	1267
SQL Server Impersonation	1268

Impersonation Requirements	1268
How Impersonation Works	1269
Impersonate with a Run As Service Account	1270
Impersonate with Embedded SQL Credentials	1272
Configure a Custom TSM Administration Group	1274
Step 1: Create the new group	1275
Step 2: Configure Tableau Server	1275
Step 3: Add users to the new group	1276
Authorization	1276
Site roles	1276
Permissions	1276
Data access and external authorization	1277
Data Security	1278
Best Practices for Row Level Security	1280
Entitlement tables	1281
Entitlement table models	1282
Users and roles	1282
Joins	1283
Implement Row Level Security	1283
Deepest granularity	1283
Sparse entitlements	1284
Data source filter	1285

All access with deepest granularity	1285
Performance and processing order of operations	1286
Live connections	1286
Extracts	1287
Considerations with extracts	1287
Single table extracts	1288
Use built-in Row Level Security in a data source	1289
Row Level Security in the Data Source	1289
Impersonation (Microsoft SQL Server)	1290
Kerberos & constrained delegation	1290
OLAP Cubes	1291
SAML delegation & SAP HANA	1291
Initial SQL to force a user-specific session (Oracle VPD)	1291
Comparison matrix for Row Level Security Methods	1292
Manage Server Secrets	1294
Understanding how secrets storage works	1295
Who has access to the master key?	1295
Import and export configuration information	1296
Securing secrets for import and export operations	1296
Cluster nodes	1299
Secrets storage event logging	1299
Managing secrets	1299

- Updating secrets 1300
- Retrieving passwords 1300
- Extension Security - Best Practices for Deployment 1304
 - Security for extensions in Tableau 1304
 - Network-enabled extensions 1305
 - Sandboxed extensions 1305
 - Potential security risks with Network-enabled extensions 1306
 - Mitigating the security threats with Network-enabled extensions 1306
 - Manage extensions using Tableau 1307
 - Recommendations for Tableau Desktop 1307
 - Deployment scenarios 1308
 - Recommendations for Tableau Server and Tableau Online 1310
 - Trust Sandboxed extensions and the Network-enabled extensions on the safe list 1310
- Checklist for the safe list: 1311
- Add extensions to the safe list: 1311
 - Block specific extensions from running on Tableau Server 1311
 - Turn off extensions for a site 1312
 - Show or hide user prompts to run Network-enabled extensions 1312
 - Turn off Sandboxed extensions 1312
- Tableau Server Key Management System 1313
 - Tableau Server local KMS 1313

Troubleshoot configuration	1314
Multi-node misconfiguration	1314
Regenerate RMK and MEK on Tableau Server	1314
Extract Encryption at Rest	1315
Limitations	1315
Performance Overview	1315
Increase in Backgrounder Load	1315
Increase in Viz Load Time and Worker Load	1316
Impact on Backup and Restore	1317
Enforce Encryption at Rest on a Site	1317
Enable Encryption at Rest on a Site	1317
Disable Encryption at Rest on a Site	1318
View Extract Encryption Mode for All Sites	1318
Encrypt or Decrypt Extracts for a Published Workbook or Data Source	1318
Encrypt or Decrypt Multiple Items	1319
View Encryption Status for a Single Item	1319
Filter Data Sources by Encryption Status	1319
Filter Workbooks by Encryption Status	1320
View Status of Encrypt or Decrypt Extracts Background Tasks	1320
The tabcmd Utility	1320
Specify the extract encryption mode when you create a site	1320
Specify the extract encryption mode when you edit a site	1321

- Get the extract encryption mode when you list sites 1321
- Encrypt extracts when you publish a workbook, data source, or extract to the server1321
- Decrypt all extracts on a site 1321
- Encrypt all extracts on a site1321
- Reencrypt all extracts on a site with new encryption keys 1321
- Tableau Server Rest API1322
- Network Security1322
 - Client to Tableau Server 1322
 - Client access from the Internet 1323
 - Clickjack Protection 1323
 - Tableau Server to your database1324
 - Tableau Server to the internet 1324
 - Tableau Server to a SMTP server1324
 - Communication with the repository 1324
 - Server component communication in a cluster1325
 - Clickjack Protection 1325
 - Effects of clickjack protection 1326
 - Disabling clickjack protection 1327
- HTTP Response Headers1327
 - Configuring response headers1328
 - HTTP Strict Transport Security (HSTS)1328

Options	1328
Referrer-Policy	1328
Options	1329
X-Content-Type-Options	1329
Option	1329
X-XSS-Protection	1330
Option	1330
Content Security Policy	1330
Configure and enable CSP	1330
Step 1: Set default directives	1330
Step 2: Add additional directives (optional)	1333
Step 3: Specify report-only directives (optional)	1333
Step 4: Enable CSP on Tableau Server	1334
Step 5: Run tsm pending-changes apply	1335
View CSP report	1335
SSL	1335
Configure SSL for External HTTP Traffic to and from Tableau Server	1336
SSL certificate requirements	1336
Configuring SSL for a Cluster	1338
SSL with multiple gateways	1338
Additional configuration information for Tableau Server cluster environments ..	1338
Prepare the environment	1339

Configure SSL on Tableau Server	1339
Use the TSM web interface	1339
Use the TSM CLI	1340
Port redirection and logging	1341
Add SSL port to the local firewall	1342
Change or update SSL certificate	1342
Example: SSL Certificate - Generate a Key and CSR	1343
Steps to generate a key and CSR	1343
Configure a certificate for multiple domain names	1344
Generate a key	1344
Create a certificate signing request to send to a certificate authority	1344
Send the CSR to a certificate authority to obtain an SSL certificate	1345
Use the key and certificate to configure Tableau Server	1345
For SAN certificates: modify the OpenSSL configuration file	1345
Configure SSL for Internal Postgres Communication	1347
Use the TSM web interface	1348
Use the TSM CLI	1349
What the command does	1350
Option for repository-ssl enable	1350
Cluster environments	1350
Configure Postgres SSL to Allow Direct Connections from Clients	1351
Configure Mutual SSL Authentication	1352

User authentication session time limits	1353
Certificate usage	1353
Client certificate requirements	1354
Use the TSM web interface	1355
Use the TSM CLI	1357
Step 1: Require SSL for external server communication	1357
Step 2: Configure and enable mutual SSL	1357
Additional options for mutual SSL	1358
Fallback authentication	1358
User name mapping	1359
Certificate Revocation List (CRL)	1359
Mapping a Client Certificate to a User During Mutual Authentication	1359
User-name mapping options	1360
Change the certificate mapping	1361
Address user-name mapping ambiguity in multi-domain organizations ..	1362
Configure Encrypted Channel to LDAP External Identity Store	1362
Certificate requirements	1363
Importing certificate into the Tableau keystore	1364
Encryption methods	1364
Configuring encrypted channel for simple bind	1365
New installations of Tableau Server	1365
New installations in an Active Directory environment	1365

- Upgrade scenarios 1367
- Disable default encrypted LDAP channel 1367
 - Disable new installation 1367
 - Disable before upgrading 1368
- Error messages 1369
 - Setup GUI 1369
- System User, sudo Privileges, and systemd 1370
 - Privilege separation 1370
 - sudo privileges 1370
 - systemd user service 1371
 - Running systemctl commands 1371
- Security Hardening Checklist 1372
 - Installing security updates 1372
 - 1. Update to the current version 1372
 - 2. Configure SSL/TLS with a valid, trusted certificate 1373
 - 3. Disable older versions of TLS 1373
 - 4. Configure SSL encryption for internal traffic 1374
 - 5. Enable firewall protection 1374
 - 6. Restrict access to the server computer and to important directories 1375
 - 7. Generate fresh secrets and tokens 1375
 - 8. Disable services that you're not using 1375
 - JMX Service 1376

9. Verify session lifetime configuration	1376
10. Configure a server allowlist for file-based data sources	1377
11. Enable HTTP Strict Transport Security for web browser clients	1378
12. Disable Guest access	1378
13. Set referrer-policy HTTP header to 'same-origin'	1379
14. Configure TLS for SMTP connection	1380
15. Configure SSL for LDAP	1381
Change List	1381
Manage Licenses	1382
Licensing Overview	1382
Activation	1383
Online activation	1383
Offline activation	1383
Lost activation	1384
Deactivate	1384
Tableau Server licensing and virtual machines (VMs)	1385
Login-based License Management	1385
Adding users	1385
Understanding License Models and Product Keys	1386
Term licensing models	1386
Role-based license model	1387
Core-based license model	1388

- Perpetual license model (legacy) 1389
- Embedded Analytics 1390
- Feature licenses 1390
 - Data Management Add-On 1391
 - Tableau Server Management Add-on 1391
 - Login-based License Management 1391
- View Server Licenses 1392
 - Viewing licenses from the Tableau Server web UI 1392
 - Use the TSM web interface 1392
 - Use the TSM CLI 1393
- Refresh Expiration Date for the Product Key 1394
 - Before you begin 1395
 - Use the TSM web interface 1396
 - Use the TSM CLI 1396
- Add Capacity to Tableau Server 1396
 - Use the TSM web interface 1397
 - Use the TSM CLI 1400
- Activate Tableau Server Offline - Adding a License 1401
 - Offline activation overview and prerequisites 1401
 - Log in to Tableau Services Manager 1406
 - Step 1 - Generate an offline activation request file 1407
 - Step 2 - Upload the offline activation request to Tableau 1407

Step 3 - Initialize or activate your license	1407
First time license activation	1408
Subsequent license activation	1408
Deactivate Product Key	1408
Before you begin	1409
Use the TSM web interface	1409
Use the TSM CLI	1410
Automate Licensing Tasks	1410
Troubleshoot Licensing	1410
Handle an unlicensed server	1410
Unlicensed role-based server	1411
Unlicensed core-based server	1411
Unlicensed server administrator	1411
Troubleshoot role-based licensing	1412
A user or administrator is unlicensed due to license expiration	1412
Server Administrator site role is unchanged when using a Creator license ..	1414
Licenses are not immediately available	1414
A user with a Viewer license cannot open Tableau Server or Tableau Online workbooks from Tableau Desktop	1414
Migrate from Core-Based to Role-Based Licensing	1414
Prepare for migration to role-based licensing	1415
Migrate to role-based licensing	1415

- Use role-based licenses on a server with core-based licensing 1416
 - Example of completing a migration from core-based licensing 1417
- Login-based License Management 1417
 - How login-based license management works 1418
 - Login-based license management with Tableau Online 1419
 - Login-based license management with Tableau Server 1419
- Use login-based license management 1420
 - Step 1: Install Tableau Server 1420
 - (Optional) Step 2: Change login-based license management settings 1420
 - Login-based license management settings 1424
 - (Optional) Step 3: Change the authorization to run (ATR) duration 1427
 - Step 4: Activate Tableau Desktop 1430
- View login-based license usage 1431
- Troubleshooting 1432
 - Login-based license management is not enabled on Tableau Server 1432
 - Login-based license management is not enabled on Tableau Desktop 1433
 - Product key expiration date doesn't change after purchasing a year sub-
scription 1433
 - You do not have a Creator license 1433
 - You have activated the maximum number of computers 1434
 - To shorten the ATR token duration for maximum activation 1434
 - To return your computer to an unlicensed state 1435

Your Tableau credentials are invalid	1436
Your computer's clock is not synchronized to the current time	1436
Unable to activate with your credentials	1436
Zero Downtime Licensing	1437
When should you restart Tableau Server?	1437
Add Users to Tableau Server	1438
Before you begin	1438
Adding users at the server level vs. the site level	1439
Add a user to the server	1440
Sign in to the Tableau Server Admin Area	1442
Navigate the Admin Areas of the Tableau Web Environment	1444
Access based on site role and number of sites	1444
Server administrator	1445
Site administrator	1447
Server administrator tasks	1447
Site administrator tasks	1448
Sign in to Tableau Services Manager Web UI	1449
Requirements	1449
Sign in to the TSM web UI	1450
Customize Your Server	1451
Language and Locale for Tableau Server	1452
Supported Languages	1453

Default Settings	1453
How Language and Locale are Determined	1453
Use Custom Fonts in Tableau Server	1454
Manage Sites Across a Server	1455
Sites Overview	1455
What is a site	1455
Authentication and sign-in credentials	1456
The Default site	1456
Why or why not add sites	1457
Administrator-level access to sites	1458
Licensing and user limits	1459
Export or Import a Site	1460
Site Migration Options	1460
Site Migration Limitations	1460
What information is preserved in a site export	1460
What information isn't preserved in a site export	1461
Prepare the Source and Target Sites	1461
Identify the Tableau Server node	1461
Delete stale content	1461
Remove obsolete users	1461
Create or identify the target site	1462
Locate site IDs	1462

Check the identity store	1463
Create users on the target server if necessary	1463
Configure the target server to deliver subscriptions	1463
Check schedules	1463
Tips for importing to a target with fewer users or schedules	1464
Migrating a Site	1464
Step 1: Export a site	1464
Step 2: Generate the import mapping files	1465
Step 3: Verify that site settings are mapped correctly	1466
To verify mapping files	1466
Step 4: Import the correctly mapped files to the target site	1467
Mapping File Content Reference	1468
CSV file name: mappingsDomainMapperForGroups	1468
CSV file name: mappingsScheduleMapper	1469
CSV file name: mappingsSiteMapper	1470
CSV file name: mappingsSystemUserNameMapper	1470
CSV file name: MappingsScheduleRecurrenceMapperWithAutoCreation	1471
Add or Delete Sites	1471
Add a site	1471
Delete sites	1472
Site Availability	1473
To activate or suspend a site	1473

Manage Site Role Limits	1474
Create role limits on a site	1474
When site role limits are met	1475
Allow Users to Save Revision History	1475
Notes	1475
Permissions users need to work with revision history	1476
Enable revision history and set the number of revisions allowed	1476
Clear all revisions	1477
Security for previewing and restoring workbooks	1477
See also	1477
Extract Refresh Schedules	1477
Before refreshing extracts	1478
Setting up refresh schedules	1478
Refreshing extracts manually	1479
Refreshing extracts from Tableau Desktop	1479
Enable Extract Refresh Scheduling and Failure Notification	1480
Managing schedules from the server	1481
How refresh failure emails work	1482
How the last successful refresh date is determined	1482
Create or Modify a Schedule	1482
To create a new schedule	1482
To modify an existing schedule	1484

Rules for Creating or Modifying Schedules	1485
See also	1485
Enable Custom Schedules for Subscriptions	1486
Enable custom schedules	1486
How Scheduled Server Jobs are Prioritized	1487
Jobs and Tasks	1487
Priority Rules for Jobs	1487
Configure Workbook Performance after a Scheduled Refresh	1489
Determine the performance impact	1489
Turn off workbook caching for the server	1490
Turn off workbook caching for a site	1490
Configure the workbook caching threshold	1491
Ensure Access to Subscriptions and Data-Driven Alerts	1491
Set Up a Site for Subscriptions	1492
Prerequisite: Configure the server to send subscription emails	1492
Enable subscriptions	1493
Test subscriptions in a site	1494
Manage all user subscriptions	1495
Suspended Subscriptions	1495
Resume suspended subscriptions	1496
See also	1496
Set Up for Data-Driven Alerts	1497

Configure email for data-driven alerts	1497
Manage all data-driven alerts in a site	1497
Disable data-driven alerts for a site	1498
Suspend data-driven alerts	1498
Resume suspended alerts	1498
Control how often the server checks data-driven alerts	1499
Track the server's alert-checking process	1499
Identify and fix failing alerts	1500
Set Up for Metrics	1502
Ensure that users can create metrics	1502
Disable metrics for a site	1502
Configure how often metrics refresh	1503
Configure failure notifications for metric refreshes	1503
Configure when metric refreshes are suspended	1503
Manage metrics	1504
Address failing and suspended metric refreshes	1504
Resume suspended refreshes	1504
Monitor metric activity with administrative views	1505
Managing Background Jobs in Tableau Server	1506
Overview	1507
Task Types	1508
Filters	1509

Canceling Jobs	1509
Status	1510
Tableau Service Manager Jobs	1513
Canceling tsm Jobs	1514
Cancel TSM Jobs	1514
Canceling Jobs that are in progress	1515
Administrative Views	1516
Navigating to administrative views	1517
Pre-built Administrative Views	1518
Performance of Views	1519
Performance of Flow Runs	1520
Traffic to Views	1521
Traffic to Data Sources	1523
Actions by All Users	1524
Actions by Specific User	1525
Actions by Recent Users	1526
Background Tasks for Extracts	1527
Understand this view	1528
Status	1528
See details about a task	1529
Background Tasks for Non Extracts	1529
Upgrade Thumbnails Job	1531

Troubleshooting	1532
Background Task Delay	1532
Stats for Load Times	1534
Stats for Space Usage	1535
Server Disk Space	1537
Login-based License Usage	1539
Filters	1540
Which creator seats are in use in the last <nn> days?	1541
Which creator seats have not been used in the last <nn> days	1541
Desktop License Usage	1542
Who has used Tableau in the last <nn> days?	1543
What licenses have not been used in the last <nn> days	1544
Desktop License Expiration	1544
Backgrounder Dashboard	1546
Summary and Filters	1547
Details	1549
Stale Content	1552
Summary and Filters	1553
Details	1555
Archive or Delete Stale Content	1557
Ask Data Usage	1559
Explore the dashboard	1560

Data Quality Warning History	1562
See warning details	1562
Filter warning history	1563
Filter by time range	1564
Filter by content type	1564
Access data quality warning history data	1564
Who can do this	1565
Create Custom Administrative Views	1565
Performance	1566
Tableau Server Performance Overview	1567
General Performance Guidelines	1567
Hardware and Software	1567
Configuration	1568
Server Resource Manager (SRM)	1569
Performance Monitoring Overview	1569
Collect Data with the Tableau Server Repository	1570
Enable access to the Tableau Server repository	1571
Connect to the Tableau Server repository	1572
PostgreSQL Version	1574
About the Tableau Server Data Dictionary	1574
Performance Tuning	1575
Optimize for User Traffic	1575

When to optimize for user traffic	1576
Slow load times for views	1576
High resource usage corresponding to user traffic	1577
Ways to optimize for user traffic	1579
Adjust the number of VizQL server processes	1579
Adjust the number of other processes	1579
Adjust the VizQL session timeout limit	1580
Refresh the cache less often	1580
Assess view responsiveness	1580
Configure Client-Side Rendering	1581
Supported browsers	1581
Configure the complexity threshold for computers and mobile devices	1581
Disable client-side rendering	1582
Testing with the URL Parameter	1583
Optimize for Extracts	1583
When to optimize for extracts	1583
High CPU usage corresponds to extract schedules	1583
Extracts fail or run slowly	1585
Ways to optimize for extracts	1586
Adjust the extract refresh schedule	1586
Speed up specific extracts	1587
Configure the execution mode for extract refreshes	1587

Increase the number of backgrounder processes	1588
Isolate processes	1588
Optimize for Extract Query Heavy Environments	1588
When to use this configuration	1589
Benefits of using this configuration	1589
When not to use this configuration	1590
Configuration	1591
Hardware Guidance	1592
Other Performance Tuning and Optimizations:	1594
When to Add Nodes and Reconfigure	1597
Performance Tuning Examples	1598
Performance Recording	1598
Create a Performance Recording	1598
Start a Performance Recording for a View	1600
View a Performance Recording	1600
Interpret a Performance Recording	1601
Performance Summary	1601
Timeline	1601
Events	1602
Query	1604
Detailed Timeline	1604
Detailed Views	1604

- Depth 1605
- CPU and Elapsed Time 1605
- Performance Monitoring Tools 1606
- Configure Client-Side Rendering 1608
- Supported browsers 1609
- Configure the complexity threshold for computers and mobile devices 1609
- Disable client-side rendering 1610
- Testing with the URL Parameter 1610
- Data Acceleration 1611
- Supported in This Release 1611
- Not Supported in This Release 1612
- Enabling Workbooks for Data Acceleration 1612
- Scheduling Data Acceleration 1612
- User Context for Precomputation 1612
- Cost of Data Acceleration 1613
- Extract Query Load Balancing 1614
- Monitoring Tableau Server 1615
- Configure SMTP Setup 1615
- Secure SMTP 1615
- Use the TSM web interface 1615
- Use the TSM CLI 1617
- SMTP CLI configuration reference 1618

TLS ciphers	1622
Configure Server Event Notification	1624
Use the TSM web interface	1625
Use the TSM CLI	1628
Set notification values individually	1628
Set all notification values with a single json file	1630
Maintenance	1631
Backup and Restore	1631
Platform compatibility	1632
Disk Space Usage for Backup and Restore	1632
Restore disk space requirements	1634
Best Practices for Backing Up Tableau Server	1634
Protect backup file	1634
Maximize backup efficiency	1635
Perform a Full Backup and Restore of Tableau Server	1636
Backup data types	1636
Backup assets that require a manual process	1637
Backing up Tableau Server for recovery	1639
Restoring core Tableau Server functionality	1640
Restore other functionality	1644
Reencrypt Extracts After Restore	1644
Back up Tableau Server data	1645

- Disk Space Usage for Backup 1646
- Optimizing Tableau Server Backup 1647
- Create a backup using the TSM command line interface (CLI) 1648
- Create a pre-upgrade backup 1649
- Backups during upgrades 1649
- Scheduling and Managing Backups 1650
 - To schedule a backup: 1650
 - To view a scheduled backup: 1650
 - To update a scheduled backup: 1651
 - To suspend or resume a backup schedule: 1651
- Script the backup process 1652
 - Remove log files and clear temporary folders 1652
 - Run the backup 1652
 - Copy the backup file to another computer 1653
- Restore from a backup 1653
 - Limitations when restoring Tableau Server 1653
 - Restore Tableau Server from a backup file 1654
- Server Maintenance 1655
 - View Server Process Status 1656
 - Viewing process status with TSM CLI 1656
 - Viewing process status in web UI 1656
 - Tableau Services Manager (TSM) Status page 1657

Tableau Server Status page	1659
External Node	1660
Access Status Remotely	1662
Get Process Status as XML	1663
Status values in the XML	1665
Troubleshoot Server Processes	1666
Cluster Controller	1666
File Store	1667
Repository	1669
VizQL Server	1670
Clear Saved Data Connection Passwords	1670
To clear saved data connection passwords for all server users:	1671
Synchronize External Directory Groups on the Server	1672
Before you begin	1673
Synchronize external directory groups on a schedule	1673
Synchronize all external directory groups on demand	1673
View synchronization activity	1674
Set the minimum site role for users in an external directory group	1675
What happens when users are removed in the source external directory? ..	1676
Improving group synchronization performance	1676
Set the Default Start Page for All Users	1677
To set the default start page for all users	1677

User-set start pages and hierarchy	1678
Disable Automatic Client Authentication	1678
Remove Unneeded Files	1679
Monitoring disk space usage	1679
Reducing disk space usage	1680
Server Settings (General and Customization)	1680
General	1681
Customization	1691
Stop or Restart the Tableau Server Computer	1691
tsm Command Line Reference	1692
Using the tsm CLI	1693
Authenticating with tsm CLI	1694
Logging into tsm CLI locally	1694
Logging into tsm CLI remotely	1694
Viewing and adding accounts to the TSM-authorized group	1695
Scripting and automating with tsm CLI	1695
Viewing help content in the shell	1695
Synopsis	1695
Commands	1695
Categories	1696
tsm authentication	1696
tsm authentication kerberos <commands>	1697

Synopsis	1697
Options for kerberos configure	1697
tsm authentication list	1698
Synopsis	1698
Options	1698
tsm authentication mutual-ssl <commands>	1698
Synopsis	1698
Options	1699
tsm authentication openid <commands>	1700
Synopsis	1700
Options for openid configure	1700
Options for openid map-claims	1702
tsm authentication pat-impersonation <commands>	1703
Synopsis	1703
tsm authentication saml <commands>	1703
Available commands	1704
tsm authentication saml configure	1704
Synopsis	1704
Options	1704
Example	1707
tsm authentication saml enable and saml disable	1707
Synopsis	1707

tsm authentication saml export-metadata	1707
Synopsis	1707
Options	1707
tsm authentication saml map-assertions	1708
Synopsis	1708
Options	1708
Example for saml map-assertions	1709
tsm authentication sitesaml enable and sitesaml disable	1709
Synopsis	1709
tsm authentication sspi <commands>	1709
Synopsis	1710
tsm authentication trusted <commands>	1710
Synopsis	1710
Options	1710
Global options	1711
tsm configuration	1712
"Unknown key" responses	1712
tsm configuration get	1713
Synopsis	1713
Option	1713
tsm configuration list-dynamic-keys	1713
Synopsis	1713

tsm configuration set	1713
Synopsis	1714
Options	1714
Global options	1715
tsm configuration set Options	1716
Basic Use of tsm configuration keys	1716
Setting a configuration key	1716
Resetting a configuration key to default	1716
Viewing the current value of a configuration key	1717
Configuration Keys	1717
adminviews.disabled	1717
api.server.enabled	1717
auditing.enabled	1717
backgrounder.default_run_now_priority	1717
backgrounder.enable_parallel_adsync	1718
backgrounder.externalquerycachewarmup.enabled	1718
backgrounder.externalquerycachewarmup.view_threshold	1718
backgrounder.extra_timeout_in_seconds	1718
backgrounder.default_timeout.run_flow	1719
backgrounder.failure_threshold_for_run_prevention	1719
backgrounder.log.level	1719
backgrounder.querylimit	1719

backgrounder.restrict_serial_collections_to_site_level	1720
backgrounder.notifications_enabled	1720
backgrounder.sort_jobs_by_type_schedule_boundary_heuristics_mil- liSeconds	1720
backgrounder.subscription_failure_threshold_for_run_prevention	1721
backgrounder.subscription_image_caching	1721
backgrounder.timeout_tasks	1721
backgrounder.timeout.sync_ad_group	1722
backgrounder.timeout.single_subscription_notify	1722
backup.zstd.thread_count	1722
basefilepath.backuprestore	1723
basefilepath.log_archive	1723
basefilepath.site_export.exports	1723
basefilepath.site_import.exports	1723
clustercontroller.log.level	1724
clustercontroller.zk_session_timeout_ms	1724
dataAlerts.checkIntervalInMinutes	1724
dataAlerts.retryFailedAlertsAfterCheckInterval	1724
dataAlerts.SuspendFailureThreshold	1725
databaseservice.max_database_deletes_per_run	1725
dataserver.log.level	1725
elasticserver.vmopts	1725

excel.shadow_copy_all_remote.enabled	1726
features.ActiveMQ	1726
features.DeleteOrphanedEmbeddedDatabaseAsset	1726
features.DesktopReporting	1727
features.MessageBusEnabled	1727
features.PasswordlessBootstrapInit	1727
features.PasswordReset	1728
filestore.empty_folders_reaper.enabled	1728
filestore_empty_folders_reap.frequency	1728
filestore.log.level	1728
filestore.reapemptyfoldersholdoffms	1729
floweditor.max_datafile_upload_size_in_kb	1729
gateway.http.cachecontrol.updated	1729
gateway.http.hsts	1729
gateway.http.hsts_options	1729
gateway.http.request_size_limit	1730
gateway.http.x_content_type_nosniff	1730
gateway.http.x_xss_protection	1730
gateway.log.level	1731
gateway.public.host	1731
gateway.public.port	1731
gateway.slow_post_protection.enabled	1731

gateway.slow_post_protection.request_read_timeout	1732
gateway.timeout	1732
gateway.trusted	1732
gateway.trusted_hosts	1732
hyper.file_partition_size_limit	1732
hyper.global_file_partition_size_limit	1733
hyper.log_queries	1733
hyper.log_query_cpu	1734
hyper.log_timing	1734
hyper.log_troublesome_query_plans	1734
hyper.memory_limit	1734
hyper.memtracker_hard_reclaim_threshold	1735
hyper.memtracker_soft_reclaim_threshold	1735
hyper.network_threads	1735
hyper.objectstore_validate_checksums	1736
hyper.query_total_time_limit	1736
hyper.session_memory_limit	1737
hyper_standalone.consistent_hashing.enabled	1737
hyper_standalone.health.enabled	1737
hyper.temp_disk_space_limit	1738
hyper.hard_concurrent_query_thread_limit	1738
hyper.soft_concurrent_query_thread_limit	1738

hyper.use_spooling_fallback	1739
java.heap.size	1740
licensing.login_based_license_management.default_requested_duration_seconds	1740
licensing.login_based_license_management.enabled	1740
licensing.login_based_license_management.max_requested_duration_seconds	1741
maestro.app_settings.sampling_max_row_limit	1741
maestro.input.allowed_paths	1741
maestro.output.allowed_paths	1742
maestro.sessionmanagement.maxConcurrentSessionPerUser	1743
metadata.ingestor.blocklist	1744
metadata.ingestor.pipeline.throttleEventsEnable	1745
metadata.ingestor.pipeline.throttleLimit	1745
metadata.ingestor.pipeline.throttlePeriodLength	1745
metadata.query.limits.time	1746
metadata.query.limits.count	1746
metricsservices.checkIntervalInMinutes	1747
metricsservices.failureCountToWarnUser	1747
metricsservices.maxFailedRefreshAttempts	1747
mobile.deep_linking.on_prem.enabled	1747
monitoring.dataengine.connection_timeout	1748

native_api.connection.limit.<connection class>	1748
native_api.connection.globallimit	1748
native_api.ExplainDataEnabled	1748
native_api.force_alternative_federation_engine	1748
native_api.ProtocolTransitionLegacyFormat	1748
native_api.unc_mountpoints	1749
native_api.InitializeQueryCacheSizeBasedOnWeights	1749
native_api.QueryCacheMaxAllowedMB	1750
native_api.LogicalQueryCacheMaxAllowedWeight	1750
native_api.MetadataQueryCachMaxAllowedWeight	1750
native_api.NativeQueryCacheMaxAllowedWeight	1750
native_api.QueryCacheEntryMaxAllowedInPercent	1751
nlp.concepts_shards_count	1751
nlp.values_shards_count	1752
nlp.defaultNewSiteAskDataMode	1752
noninteractivecontainer.vmopts	1753
pgsql.port	1753
pgsql.preferred_host	1753
pgsql.verify_restore.port	1754
ports.blocklist	1754
recommendations.enabled	1754
recommendations.vizrecs.enabled	1754

redis.max_memory_in_mb	1755
refresh_token.absolute_expiry_in_seconds	1755
refresh_token.idle_expiry_in_seconds	1755
refresh_token.max_count_per_user	1755
rsync.timeout	1756
schedules.display_schedule_description_as_name	1756
schedules.display_schedules_in_client_timezone	1756
schedules.ignore_extract_task_priority	1756
searchserver.connection_timeout_milliseconds	1757
searchserver.javamemopts	1757
searchserver.startup.zookeeper_healthcheck_timeout_ms	1757
searchserver.zookeeper_session_timeout_milliseconds	1758
ServerExportCSVMaxRowsByCols	1758
service.jmx_enabled	1758
service.max_procs	1758
service.port_remapping.enabled	1759
session.ipsticky	1759
sheet_image.enabled	1759
ssl.ciphersuite	1760
ssl.client_certificate_login.blocklisted_signature_algorithms	1760
ssl.client_certificate_login.min_allowed.elliptic_curve_size	1760
ssl.client_certificate_login.min_allowed.rsa_key_size	1761

storage.monitoring.email_enabled	1761
storage.monitoring.warning_percent	1761
storage.monitoring.critical_percent	1761
storage.monitoring.email_interval_min	1762
storage.monitoring.record_history_enabled	1762
subscriptions.enabled	1762
subscriptions.timeout	1762
svcmonitor.notification.smtp.enabled	1762
svcmonitor.notification.smtp.mime_use_multipart_mixed	1763
tabadmincontroller.auth.expiration.minutes	1763
tdsservice.log.level	1763
tomcat.http.maxrequestsize	1763
tomcat.https.port	1764
tomcat.server.port	1764
tomcatcontainer.log.level	1764
tsm.log.level	1764
tsm.controlapp.log.level	1765
usernotifications.reap_after_days	1765
vizportal.adsync.update_system_user	1765
vizportal.commenting.delete_enabled	1765
vizportal.csv_user_mgmt.index_site_users	1766
vizportal.log.level	1766

vizportal.openid.client_authentication	1766
vizportal.openid.essential_acr_values	1766
vizportal.openid.voluntary_acr_values	1767
vizportal.rest_api.cors.allow_origin	1767
vizportal.rest_api.cors.enabled	1768
vizqlserver.allow_insecure_scripts	1768
vizqlserver.browser.render	1768
vizqlserver.browser.render_threshold	1769
vizqlserver.browser.render_threshold_mobile	1769
vizqlserver.clear_session_on_unload	1769
vizqlserver.force_maps_to_offline	1769
vizqlserver.geosearch_cache_size	1770
vizqlserver.initialsql.disabled	1770
vizqlserver.log.level	1770
vizqlserver.NumberOfWorkbookChangesBetweenAutoSaves	1771
vizqlserver_<n>.port	1771
vizqlserver.protect_sessions	1771
vizqlserver.querylimit	1771
vizqlserver.RecoveryAttemptLimitPerSession	1771
vizqlserver.session.expiry.minimum	1772
vizqlserver.session.expiry.timeout	1772
vizqlserver.sheet_image_api.max_age_floor	1772

vizqlserver.showdownload	1772
vizqlserver.showshare	1772
vizqlserver.url_scheme_whitelist	1773
vizqlserver.web_page_objects_enabled	1773
vizqlserver.WorkbookTooLargeToCheckpointSizeKiB	1773
webdataconnector.refresh.enabled	1774
webdataconnector.whitelist.fixed	1774
webdataconnector.enabled	1774
webdataconnector.whitelist.mode	1775
wgserver.audit_history_expiration_days	1775
wgserver.change_owner.enabled	1775
wgserver.clickjack_defense.enabled	1775
wgserver.domain.accept_list	1776
wgserver.domain.ldap.domain_custom_ports	1777
wgserver.domain.whitelist	1777
wgserver.extended_trusted_ip_checking	1777
wgserver.restrict_options_method	1778
wgserver.saml.blocklisted_digest_algorithms	1778
wgserver.saml.forceauthn	1778
wgserver.saml.idpattribute.username	1779
wgserver.saml.iframe_idp.enabled	1779
wgserver.saml.maxassertiontime	1779

wgserver.saml.min.allowed.elliptic_curve_size	1780
wgserver.saml.min.allowed.rsa_key_size	1780
wgserver.saml.responseskew	1781
wgserver.saml.sha256	1781
wgserver.session.apply_lifetime_limit	1781
wgserver.session.idle_limit	1781
wgserver.session.lifetime_limit	1782
wgserver.unrestricted_ticket	1782
workerX.gateway.port	1782
workerX.vizqlserver.procs	1782
zookeeper.config.snapCount	1782
tsm customize	1783
Synopsis	1784
Options	1785
Global options	1786
tsm data-access	1787
tsm data-access caching list	1787
Synopsis	1787
tsm data-access caching set	1788
Synopsis	1788
Options	1788
tsm data-access repository-access disable	1788

Synopsis	1788
Options	1788
tsm data-access repository-access enable	1789
Synopsis	1789
Options	1789
tsm data-access repository-access list	1790
Synopsis	1790
tsm data-access set-saml-delegation configure	1790
Synopsis	1790
Options	1790
tsm data-access set-saml-delegation disable	1791
Synopsis	1791
tsm data-access set-saml-delegation enable	1791
Synopsis	1791
tsm data-access web-data-connectors add	1792
Synopsis	1792
Options	1792
tsm data-access web-data-connectors allow	1793
Synopsis	1793
Options	1793
tsm data-access web-data-connectors delete	1793
Synopsis	1794

Options	1794
tsm data-access web-data-connectors list	1794
Synopsis	1794
Options	1794
Global options	1794
tsm email	1795
tsm email test-smtp-connection	1796
Synopsis	1796
Global options	1796
tsm initialize	1797
Synopsis	1797
Options	1798
Global options	1798
tsm jobs	1799
tsm jobs cancel	1799
Synopsis	1799
Options	1799
tsm jobs list	1800
Synopsis	1800
Options	1800
tsm jobs reconnect	1800
Synopsis	1800

Options	1800
Global options	1800
tsm licenses	1802
tsm licenses activate	1802
Synopsis	1802
Options	1802
tsm licenses atr-configuration get	1803
Synopsis	1803
Options	1803
tsm licenses atr-configuration set	1803
Synopsis	1803
Options	1803
tsm licenses deactivate	1804
Synopsis	1804
Options	1804
tsm licenses get-offline-activation-file	1804
Synopsis	1804
Options	1804
tsm licenses get-offline-deactivation-file	1805
Synopsis	1805
Options	1805
tsm licenses list	1805

Synopsis	1807
tsm licenses refresh	1807
Synopsis	1807
Global options	1807
tsm login	1808
Synopsis	1809
Global options	1809
tsm logout	1810
Synopsis	1810
Global options	1810
tsm maintenance	1811
tsm maintenance backup	1812
Synopsis	1813
Options	1813
Examples	1816
tsm maintenance cleanup	1816
Synopsis	1817
Options	1817
Examples	1819
tsm maintenance metadata-services disable	1819
Synopsis	1820
Option	1820

tsm maintenance metadata-services enable	1820
Synopsis	1821
Option	1821
tsm maintenance metadata-services get-status	1821
Synopsis	1821
tsm maintenance reindex-search	1821
Synopsis	1821
Option	1822
tsm maintenance reset-searchserver	1822
Synopsis	1822
Option	1822
tsm maintenance restore	1822
Synopsis	1822
Options	1823
tsm maintenance send-logs	1824
Synopsis	1824
Options	1824
tsm maintenance snapshot-backup complete	1825
Synopsis	1825
Options	1825
tsm maintenance snapshot-backup prepare	1826
Synopsis	1826

Options	1826
tsm maintenance snapshot-backup restore	1827
Synopsis	1827
Options	1827
tsm maintenance validate-resources	1827
Synopsis	1827
Options	1827
tsm maintenance ziplogs	1828
Synopsis	1828
Options	1828
Global options	1831
tsm pending-changes	1832
tsm pending-changes apply	1832
Synopsis	1833
Options	1833
tsm pending-changes discard	1833
Synopsis	1834
Options	1834
tsm pending-changes list	1834
Synopsis	1834
Options	1834
Global options	1835

tsm register	1836
Synopsis	1836
Options	1836
Global options	1837
tsm reset	1838
Synopsis	1838
Option	1838
Global options	1838
tsm restart	1840
Synopsis	1840
Option	1840
Global options	1840
tsm schedules	1841
tsm schedules delete	1842
Synopsis	1842
Options	1842
tsm schedules list	1842
Synopsis	1842
Options	1843
tsm schedules resume	1843
Synopsis	1843
Options	1843

tsm schedules suspend	1844
Synopsis	1844
Options	1844
tsm schedules update	1844
Synopsis	1844
Options	1845
Global options	1845
tsm security	1846
Prerequisites	1848
tsm security custom-cert add	1848
Synopsis	1848
Options	1849
tsm security custom-cert delete	1849
Synopsis	1849
tsm security custom-cert list	1849
Synopsis	1849
tsm security external-ssl disable	1849
Synopsis	1849
tsm security external-ssl enable	1849
Synopsis	1850
Options	1850
tsm security external-ssl list	1851

Synopsis	1851
tsm security kms set-mode aws	1852
Synopsis	1852
Options	1852
Example	1852
tsm security kms set-mode azure	1853
Synopsis	1853
Options	1853
Example	1853
tsm security kms set-mode local	1854
Synopsis	1854
tsm security kms status	1854
Synopsis	1855
tsm security maestro-rserve-ssl disable	1855
tsm security maestro-rserve-ssl enable	1855
Synopsis	1855
Options	1856
tsm security maestro-tabpy-ssl disable	1856
tsm security maestro-tabpy-ssl enable	1857
Synopsis	1857
Options	1857
tsm security regenerate-internal-tokens	1858

Synopsis	1859
Options	1859
tsm security repository-ssl disable	1859
Synopsis	1859
tsm security repository-ssl enable	1859
Synopsis	1859
Options	1859
tsm security repository-ssl get-certificate-file	1860
Synopsis	1860
Options	1860
tsm security repository-ssl list	1860
Synopsis	1861
Global options	1861
tsm settings	1862
tsm settings export	1863
Synopsis	1863
Options	1863
tsm settings import	1863
Synopsis	1863
Options	1863
Global options	1864
tsm sites	1865

tsm sites export	1866
Synopsis	1866
Options	1866
tsm sites import	1867
Synopsis	1868
Options	1868
tsm sites import-verified	1869
Synopsis	1869
Options	1869
tsm sites unlock	1870
Options	1870
Global options	1871
tsm start	1872
Synopsis	1872
Option	1872
Global options	1872
tsm status	1873
Synopsis	1873
Options	1874
Global options	1875
tsm stop	1876
Synopsis	1876

Options	1876
Global options	1877
tsm topology	1878
tsm topology cleanup-coordination-service	1879
Synopsis	1879
Option	1880
tsm topology deploy-coordination-service	1880
Synopsis	1880
Options	1880
tsm topology external-services storage disable	1881
Synopsis	1881
Options	1881
tsm topology external-services storage enable	1881
Synopsis	1882
Options	1882
tsm topology external-services storage switch-share	1882
Synopsis	1882
Option	1882
tsm topology external-services list	1882
Synopsis	1883
Option	1883
tsm topology external-services repository disable -n nodeN	1883

Synopsis	1883
Option	1883
tsm topology external-services repository enable	1883
Synopsis	1884
Options	1884
tsm topology external-services repository replace-host	1885
Synopsis	1885
Options	1885
tsm topology failover-repository	1886
Synopsis	1886
Options	1887
tsm topology filestore decommission	1887
Synopsis	1887
Options	1888
tsm topology filestore recommission	1888
Synopsis	1889
Options	1889
tsm topology list-nodes	1889
Synopsis	1889
Options	1889
tsm topology list-ports	1889
Synopsis	1889

Options	1890
tsm topology nodes get-bootstrap-file	1890
Synopsis	1891
Options	1891
tsm topology remove-nodes	1892
Synopsis	1892
Options	1892
tsm topology set-node-role	1893
Synopsis	1893
Options	1893
tsm topology set-ports	1894
Synopsis	1894
Options	1895
tsm topology set-process	1896
Synopsis	1896
Options	1896
tsm topology toggle-coordination-service	1897
Synopsis	1897
Option	1897
Global options	1898
tsm user-identity-store	1899
tsm user-identity-store get-group-mappings [options]	1899

Synopsis	1899
tsm user-identity-store get-user-mappings [options]	1899
Synopsis	1900
tsm user-identity-store list [options]	1900
Synopsis	1900
Options	1900
tsm user-identity-store set-connection [options]	1900
Synopsis	1900
Options	1900
tsm user-identity-store set-group-mappings [options]	1902
Synopsis	1902
Options	1902
tsm user-identity-store set-user-mappings [options]	1903
Synopsis	1903
Options	1903
tsm user-identity-store verify-group-mappings [options]	1905
Synopsis	1905
Options	1905
tsm user-identity-store verify-user-mappings [options]	1905
Synopsis	1905
Options	1906
Global options	1906

tsm version	1907
Synopsis	1907
Global options	1907
tsm File Paths	1908
Default locations for files	1908
Get the current file location	1910
Change the current file location	1910
Migrate from Tabadmin to the TSM CLI	1913
Tabadmin commands with a corresponding TSM CLI command	1913
Tabadmin commands with no corresponding TSM CLI command	1918
Entity Definitions and Templates	1920
Configuration File Example	1920
Entities vs keys	1921
gatewaySettings Entity	1922
Gateway settings	1922
Configuration template	1923
Configuration file reference	1923
identityStore Entity	1925
Before you begin	1925
Configuration templates	1926
Local	1927
Important	1927

LDAP - Active Directory	1927
OpenLDAP - GSSAPI bind	1929
OpenLDAP - Simple bind	1931
Configuration template reference	1932
Shared identity store options	1932
LDAP GSSAPI bind options	1932
LDAP simple bind options	1933
Shared LDAP options	1934
identityStoreSchemaType options	1935
Importing the JSON file	1937
kerberosSettings Entity	1938
Configuration template	1938
Configuration file reference	1939
mutualSSLSettings Entity	1940
Configuration template	1940
Configuration file reference	1941
openIDSettings Entity	1944
Configuration template	1944
Configuration file reference	1945
samlSettings Entity	1948
Template categories and definitions	1949
samlSettings configuration template	1949

SAML configuration entity reference	1950
Pass the configuration file to Tableau Server	1955
See also	1955
sapHanaSettings Entity	1956
SAP HANA SAML settings	1956
Configuration template	1956
Configuration file reference	1957
shareProductUsageDataSettings Entity	1958
Configuration template	1959
Configuration file reference	1959
trustedAuthenticationSettings Entity	1959
Configuration template	1960
Configuration file reference	1961
web-data-connector-settings Entity	1963
Web data connector settings	1964
Configuration template	1964
Single WDC	1964
Multiple WDCs	1965
Configuration file reference	1965
tabcmd	1967
Install tabcmd	1967
How to use tabcmd	1971

Examples	1972
Status messages and logs	1973
tabcmd Commands	1973
addusers group-name	1974
Options	1975
Global options	1975
createextracts	1978
Options	1978
Global options	1979
creategroup group-name	1981
Global options	1981
createproject project-name	1984
Options	1984
Global options	1984
createsite site-name	1987
Options	1988
Global options	1988
createsiteusers filename.csv	1991
Local authentication	1992
Active Directory authentication	1992
Options	1992
Global options	1994

createusers filename.csv	1996
Local authentication	1997
Active Directory authentication	1997
Options	1997
Global options	1999
decryptextracts	2001
Global options	2001
delete workbook-name or datasource-name	2004
Options	2004
Global options	2005
deleteextracts	2007
Options	2007
Global options	2008
deletegroup group-name	2011
Global options	2011
deleteproject project-name	2013
Option	2014
Global options	2014
deletesite site-name	2016
Global options	2016
deletesiteusers filename.csv	2019
Global options	2020

deleteusers filename.csv	2022
Options	2022
Global options	2023
editdomain	2025
Options	2026
Global options	2026
editsite site-name	2029
Options	2029
Global options	2030
encryptextracts	2033
Global options	2033
export	2035
Options	2039
Global options	2040
get url	2042
Global options	2045
initialuser	2047
Options	2048
Global options	2048
listdomains	2051
Global options	2051
listsites	2053

Options	2053
Global options	2054
login	2056
Options	2058
Global options	2059
logout	2062
publish filename.twb(x), filename.tds(x), or filename.hyper	2062
Options	2063
Global options	2066
publishsamples	2068
Description	2068
Syntax	2068
Options	2069
Global options	2069
reencryptextracts	2072
Global options	2072
refreshextracts workbook-name or datasource-name	2074
Options	2075
Global options	2077
reset_openid_sub	2080
Options	2080
Global options	2080

removeusers group-name	2083
Options	2083
Global options	2083
runschedule schedule-name	2086
Global options	2086
set setting	2089
Global options	2089
syncgroup group-name	2092
Options	2092
Global options	2093
upgradethumbnails	2096
Options	2096
Global options	2096
validateidpmetadata	2099
Options	2099
Global options	2100
version	2102
Global options	2102
Install Switches and Properties for tabcmd (Windows)	2105
Troubleshooting	2108
Troubleshoot Tableau Server on Linux	2108
General Troubleshooting Steps	2109

Clean install	2109
Disk space	2109
Remove old log files	2109
Manually gather logs	2111
Restart server	2111
Edit installation and configuration files using Linux	2111
Check systemd logs	2112
Installing Tableau Server	2112
Install fails due to hardware requirements	2112
Install fails due to timeouts	2112
Install fails with "Failed to initialize the instance of the temporary database"	2112
Installation fails on a virtual machine in Parallels	2113
Tableau Server doesn't start	2113
Cannot start Tableau Server after installation	2113
Cannot create initial administrator account with multiple Active Directory (AD) domains	2114
Fonts	2115
Support for Asian character sets	2115
Initializing Tableau Server	2115
TSM initialization fails because the tableau user account exists but is not a member of the group tableau	2115
Error initializing Tableau Server on unsupported system locale	2116

- Error initializing Tableau Server when en_US.utf8 is not included in locale list2116
- Error: status 10 - initializing Tableau Server when data directory path includes a period2117
- Error initializing Tableau Server after reinstallation2117
- Activating Tableau Server2118
 - Tableau Server license activation fails2118
- Reindexing Tableau Server Search & Browse2118
 - Problems that can be solved by reindexing Search & Browse2118
- Restarting Tableau Server2118
 - Restarting Tableau Server or applying changes fails2118
 - Error restarting Tableau Server after adding or configuring a node2119
- Backup/Restore2119
- File locations2121
 - Changing basefilepath does not change the location of an existing file2121
- TSM commands2121
 - TSM command line does not show progress for long-running tasks2121
- Opening Firewall ports2121
 - Manually opening firewall ports on Ubuntu2121
- OpenID fails on first sign-in attempt2122
- Administrative views do not display2122
- Changing locale on view2122

Work with Log Files	2123
Contents of Tableau Server Logs	2123
Investigating Tableau Server Issues	2124
Tableau Server Logs and Log File Locations	2126
Tableau Server log files on an active cluster	2127
Primary log locations on a working Tableau Server installation	2127
Configuration file locations on a working Tableau Server installation	2128
Logs that are not written in the primary location	2128
Server Log Files in a zipped archive	2128
Log File Snapshots (Archive Logs)	2129
Use the TSM web interface	2129
Uploading log snapshots for Tableau Support	2131
Use the TSM CLI	2132
Sending log archives to Tableau Support	2134
Change Logging Levels	2134
Logging Levels	2134
Change Logging Levels	2135
Dynamic log level configuration	2135
Configuration Keys for Changing Logging Levels	2136
Reset Logging Levels	2139
Troubleshoot Tableau Server Install and Upgrade	2139
General Troubleshooting Steps	2139

- Common Tableau Server Install Issues 2140
 - Installation logs location 2140
 - Unable to log into TSM or Tableau Server (sign in screen redisplay after entering credentials) 2140
 - Multiple install attempts fail 2141
 - Install fails due to hardware requirements 2142
 - Install or upgrade fails due to CPU requirements 2142
- Common Tableau Server Upgrade Issues 2143
 - Maps do not display or display incompletely after upgrading 2143
 - Upgrade script error: "Tableau Server Version change validation failed." 2143
 - Upgrade logs location 2143
 - Upgrade multi-node, initializing additional node fails with "Enter your credentials again" error 2143
 - Upgrading fails due to lack of disk space 2144
 - Upgrade fails on RebuildSearchIndex job 2144
 - Upgrade fails on 2020.4.0 or later 2145
- Common Settings Import Issues 2146
 - Import of settings file causes "not present on any node" validation error due to missing services 2146
 - Import of settings file causes "configuration value you specified does not match" error 2147
 - "You cannot directly modify instances of the Coordination Service" error 2148
 - If you see this error after importing a settings file: 2148

If you see the error when setting the process count for Coordination Service manually:	2150
Starting Tableau Server	2150
Tableau Server cannot determine if it fully started	2150
Tableau Server doesn't start	2150
Reindexing Tableau Server Search & Browse	2151
Problems that can be solved by rebuilding Search & Browse index	2151
Activating Tableau Server	2151
Tableau Server license activation fails	2151
Confirm you can access the licensing server	2151
Verify the date and time	2153
Force the product key to be read again	2153
Send the contents of trusted storage to Tableau Support	2154
tabcmd Installation Problems	2154
Installing tabcmd separately	2154
Problems installing tabcmd on Linux	2154
Java is not installed	2155
Incorrect version of Java is installed	2155
systemd User Service Failures	2155
Background	2155
Upgrading from Tableau Server on Linux 10.5	2156
Fresh installation error troubleshooting	2156

- Example 2157
- Troubleshoot Server Sign in Problems 2158
- Troubleshooting scenarios 2159
- Unable to log into TSM or Tableau Server (sign in screen redisplay after entering credentials) 2159
- Troubleshoot Licensing 2159
- Handle an unlicensed server 2159
- Unlicensed role-based server 2160
- Unlicensed core-based server 2160
- Unlicensed server administrator 2160
- Troubleshoot role-based licensing 2161
- A user or administrator is unlicensed due to license expiration 2161
- Server Administrator site role is unchanged when using a Creator license .. 2163
- Licenses are not immediately available 2163
- A user with a Viewer license cannot open Tableau Server or Tableau Online workbooks from Tableau Desktop 2163
- Handle an Unlicensed Server Process 2163
- Tableau Services Manager (TSM) Command Timeout 2165
- Troubleshooting Tableau Services Manager (TSM) Backup 2165
- Backup fails to start because services do not start 2165
- Cookie Restriction Error 2166
- Troubleshoot Subscriptions 2166

"The view snapshot in this email could not be properly rendered."	2166
Can't see images in email	2167
Can't subscribe	2167
No subscription icon	2167
Receiving invalid or "broken" subscriptions	2168
Missing PDF attachment	2168
Suspended Subscriptions	2168
Resume suspended subscriptions	2169
Can't set subscription frequency to "When Data Refreshes"	2169
Subscriptions not arriving ("Error sending email. Can't send command to SMTP host.")	2170
Custom scripts not working after upgrade to 8.1	2170
Custom scripts not working after upgrade to 9.0	2170
Server Administrator Reference	2171
Tableau Server Processes	2171
Licensed processes	2172
Process workflow	2193
Tableau Server Administration Agent	2194
Tableau Server Administration Controller	2195
What happens when the Administration Controller process fails?	2196
Moving the Administration Controller	2196
Restarting the Administration Controller	2196

Tableau Server Application Server	2197
Tableau Server Backgrounder Process	2197
Managing Backgrounder Resources	2198
Related content	2199
Tableau Server Cache Server	2199
Tableau Server Client File Service	2200
Tableau Server Collections Service	2202
Server Configuration	2202
Multi-Node Configuration	2202
Impact if the Collections Service is not running properly	2202
One instance of Collections service	2203
Multiple instances of Collections service	2203
Log Files	2203
Tableau Server Content Exploration Service	2203
Server Configuration	2203
Multi-Node Configuration	2204
Performance Tuning	2204
Log Files	2205
Search Accuracy	2205
Re-indexing	2205
Tableau Server Coordination Service	2206
Configuration for the Coordination Service	2207

The Coordination Service Quorum	2207
Number of Coordination Service instances to use	2207
If you reduce the number of nodes	2209
Viewing Coordination Service Status	2209
Tableau Server Data Engine	2210
Memory and CPU usage	2211
CPU usage	2211
Memory usage	2212
Server configuration, Scalability, and Performance	2212
Scalability:	2213
Performance:	2214
Performance benefits	2214
Tableau Server Data Server	2215
Tableau Server Data Source Properties Service	2216
Log files for the Data Source Properties service	2216
Data Source Properties service in a multi-node cluster	2217
Tableau Server File Store	2217
The decommission Command	2217
Decommissioning File Store when co-located with the Administrative Controller	2218
Tableau Server Gateway Process	2218
Port assignment	2218

- Log files for the gateway process 2219
- Gateway processes in a cluster 2219
- Additional information 2220
- Tableau Server Internal Data Source Properties Service 2220
 - Log files for the Internal Data Source Properties service 2221
 - Internal Data Source Properties service in a multi-node cluster 2221
- Tableau Server Messaging Service 2221
 - Impact if Messaging Service is not running properly 2222
 - Multiple instances of Messaging Service (multi-node, version 2020.1 and later) 2222
 - One instance of Messaging Service 2222
 - Disabling a failed Messaging Service instance 2223
 - Messaging Service in a multi-node cluster 2223
- Tableau Server Metrics Service 2223
 - Impact if the Metrics Service fails 2224
 - Metrics Service in a multi-node cluster 2224
 - Log files for the Metrics service 2224
- Tableau Server Microservice Containers 2225
 - Viewing Microservice Container Status 2225
 - Microservice Container Status 2226
 - Use the TSM web interface 2226
 - Use the TSM CLI 2227

Tableau Server Repository	2229
Preferred active repository	2229
The failoverrepository Command	2230
Tableau Server SAML Service	2230
Tableau Server Search and Browse	2231
Tuning the Search & Browse Process	2232
Search & Browse Max Heap Memory	2232
Default maximum heap memory allocations	2232
Client session timeouts	2233
Zookeeper connection health check timeout at startup	2234
Tableau Server TSM Maintenance Services	2234
Tableau Server VizQL Server	2235
Tableau Prep Conductor	2236
Performance and Scale Recommendations	2237
Topology and Configuration	2237
Tableau Prep Flow Authoring	2238
Performance and Scale Recommendations	2238
Isolate flow authoring to a separate node	2239
Add resources	2239
Topology and Configuration	2239
Tableau Server Dynamic Topology Changes	2244
Example Scenarios	2245

Making dynamic topology changes	2246
Impact of dynamic topology changes	2246
Best practices	2246
Automating dynamic topology changes	2246
Additional information	2247
Server Process Limits	2247
Tableau Services Manager Ports	2248
Firewall requirements	2248
Port assignment	2249
Dynamic port assignment	2250
Changing the port range	2250
Blocking specific ports within the range	2251
Disabling dynamic port assignment	2251
Manual port assignment	2252
Configuring ports during installation	2252
Configuring ports after installation	2254
Ports that are not dynamically mapped	2255
Controlling port remapping with initialize-tsm	2256
Dynamically mapped ports	2257
Enable the JMX Ports	2263
How the JMX Ports Are Determined	2264
ATRDdiag.exe Command Line Reference	2264

Synopsis	2265
Options	2265
Global Options	2268
Help Output for initialize-tsm Script	2268
Output	2268
Related topics	2272
Help Output for upgrade-tsm Script	2272
Output	2272
View Server Version	2274
Viewing the server version from the Tableau Server web UI	2275
Viewing the server version and TSM version from the TSM command line ...	2275
Short version, long version, and version_code	2276
Finding the long version number	2276
Configure Einstein Discovery Integration	2277
Einstein Discovery dashboard extensions	2277
Einstein Discovery analytics extensions	2278
Einstein Discovery Tableau Prep extensions	2279
Configure CORS in Salesforce.com for Einstein Discover Integration in Tableau Server	2279
Configure CORS for Einstein Discovery.	2280
Configure Connections with Analytics Extensions	2281
Server SSL	2282

- Enable analytics extensions2282
- Configure analytics extensions settings2283
 - Edit or delete an analytics extension connection2285
- Client requirement: Intermediate certificate chain for Rserve external service .2285
- Script errors2286
- Determining analytics extensions usage2286
- Changing the Identity Store2286
 - Warning2287
 - Methods for restoring content and permissions2288
 - User filters2289
 - User names and the Tableau Identity store2289
 - Method 1: Use site export and import2290
 - Method 2: Fresh installation—users republish content2290
 - Back up, remove, and then reinstall2291
 - Step 1: Back up Tableau Server2291
 - Step 2: Remove Tableau Server2291
 - Step 3: Reinstall Tableau Server with new authentication type2291
- External Identity Store Configuration Reference2291
 - Configuration methods2292
 - Configuring Active Directory2294
 - Configuration reference table2294
 - Calculated configKeys2310

Unsupported configKeys	2311
Basic Product Data	2311
Disabling sharing of Basic Product Data on individual computers	2312
Disabling the sharing of Basic Product Data at the enterprise level	2313
About Tableau Advanced Management	2313
Advanced Management Licensing Requirements	2313
Activating the Advanced Management license	2317
Who can do this	2319
About Tableau Resource Monitoring Tool	2319
What is Resource Monitoring Tool?	2319
	2320
Get Started with Tableau Resource Monitoring Tool	2320
Pre-Installation	2320
Product Compatibility with Tableau Server	2320
Resource Monitoring Tool Master Server and Agent Compatibility	2320
Concepts and Terms	2321
Setup Architecture	2321
Minimum Hardware Requirements and Recommendations for Resource Monitoring Tool	2322
Pre-Installation Checklist for Resource Monitoring Tool	2323
Troubleshoot	2323
Concepts	2323

Agent	2323
Master Server	2323
Environment	2323
Environment Status	2324
OK	2324
Warning	2324
Critical	2324
Server	2325
Pre-Installation Checklist - Tableau Resource Monitoring Tool	2325
Machine, Network, and Account Requirements	2325
Who can do this	2328
Minimum Hardware Requirements and Recommendations for Tableau Resource Monitoring Tool	2328
Resource Monitoring Tool Agent - Resource Utilization	2329
Installing in a Cloud Environment	2330
Who can do this	2330
Install the Tableau Resource Monitoring Tool	2330
Before Installing the Resource Monitoring Tool	2331
Setup Architecture	2332
Installation Overview	2334
Installation on Linux	2334
HTTPS	2335

SSL Certificate Mode and Requirements	2335
Default File and Directory locations:	2336
Master:	2336
Agent:	2336
Who can do this	2337
Next Step	2337
Install the Master Server Using Web Interface	2337
Installing to a Non-Default Location	2342
Master Server	2342
Who can do this	2342
Next Step	2342
Install the Agent Using the Web Interface	2342
Steps to download the Agent bootstrap file	2343
Steps to install Agent	2343
Installing Agent on Multi-Node Tableau Server	2346
Installing to a Non-Default Location	2347
Who can do this	2347
Next Steps	2347
Install the Master Server Using Command Line	2348
Install on Linux	2348
Install on Windows	2356
Windows install properties and switches	2365

Who can do this	2368
Next Step	2369
Install the Agent Using Command Line	2369
Install on Linux	2369
Install on Windows	2371
Windows install properties and switches	2373
Installing Agent on Multi-Node Tableau Server	2375
Who can do this	2375
Next Step	2376
Tableau Resource Monitoring Tool Prerequisites - Licenses	2376
Upgrading Resource Monitoring Tool	2376
Upgrade Notes	2376
How to Upgrade the Resource Monitoring Tool	2377
Who can do this	2380
Uninstalling Resource Monitoring Tool	2380
Who can do this	2382
Configure Tableau Resource Monitoring Tool	2383
Resource Monitoring Tool Master Server Configuration	2383
Post Install Setup Configurations	2383
Notifications	2384
Incident Thresholds	2385
Security	2385

Data	2385
Advanced	2386
Who can do this	2386
Master Server Configuration File	2386
Data Retention	2386
SMTP Configuration	2387
Notification Configuration	2388
Histogram Configuration	2389
Incident Configuration	2392
Master Logging	2392
Agent	2392
Tableau Server Detection	2393
Agent Logging	2394
Common	2394
Encrypted Messaging	2394
Who can do this	2396
Tableau Resource Monitoring Tool Communication Ports	2396
Master Server	2396
Master Server Communications	2396
RabbitMQ	2397
PostgreSQL Database	2397
Agent	2397

Server Roles in Tableau Resource Monitoring Tool	2398
Create a New User	2398
Tableau Resource Monitoring Tool - Incidents	2399
Incidents	2399
Incident Severity Levels	2400
Environment Down Incidents	2401
Who can do this	2402
Agent Incidents	2403
Agents Unlicensed	2403
Incompatible Agent Version	2404
Upgrading Agents:	2404
Upgrading Master	2406
Agent Down	2407
Agent polling and incident creation times	2408
Who can do this	2408
Extract Failure Incidents	2408
Who can do this	2409
Hardware Incidents	2409
Use the Master Server web interface	2409
Use the configuration file (config.json)	2412
Who can do this	2414
Slow Query Incidents	2414

Use the Master Server web interface	2415
Use the configuration file (config.json)	2415
Who can do this	2417
Slow Views Incidents	2417
Use the Master Server web interface	2417
Use the configuration file (config.json)	2418
Per-User Slow View Triggers	2420
Who can do this	2420
Encrypted Data Collection	2420
RabbitMQ Setup	2421
Tableau Resource Monitoring Tool Setup	2421
Tableau Resource Monitoring Tool Setup	2421
Tableau Repository SSL Configuration	2422
Who can do this	2422
Hardware Changes to Master Server - Tuning PostgreSQL Database	2423
rmtadmin Command Line Utility	2423
rmtadmin agents	2425
Synopsis	2425
Options	2425
rmtadmin bootstrap-file	2426
Synopsis	2426
Options	2426

rmtadmin cleanup	2426
Synopsis	2427
Option	2427
rmtadmin create-env	2427
Synopsis	2427
Options	2428
rmtadmin delete-env	2431
Synopsis	2431
Positional Parameter	2432
Options	2432
rmtadmin delete-env-data	2432
Synopsis	2433
Positional Parameter	2433
Option	2433
rmtadmin delete-server	2433
Synopsis	2433
Options	2434
rmtadmin delete-server-data	2435
Synopsis	2435
Options	2435
rmtadmin deregister-agent	2436
Synopsis	2436

Options	2436
rmtadmin environments	2437
Synopsis	2438
rmtadmin get	2438
Synopsis	2438
Positional Parameter	2438
Supported configuration keys	2438
rmtadmin help	2440
Synopsis	2440
rmtadmin master-setup	2440
Synopsis	2441
rmtadmin passwd	2447
Synopsis	2448
Positional Parameter	2448
rmtadmin query	2448
Synopsis	2448
Positional Parameter	2448
Options	2448
rmtadmin register	2449
Synopsis	2449
Positional Parameter	2450
Options	2450

rmtadmin restart	2450
Synopsis	2450
Options	2450
rmtadmin servers	2451
Synopsis	2452
Positional Parameter	2452
rmtadmin set	2452
Synopsis	2452
Positional Parameters	2452
Supported configuration keys	2453
Options	2453
rmtadmin start	2454
Synopsis	2454
Options	2454
rmtadmin status	2455
Synopsis	2455
rmtadmin stop	2455
Synopsis	2456
Options	2456
rmtadmin test-env	2457
Synopsis	2457
Positional Parameter	2457

rmtadmin update-env	2457
Synopsis	2458
Options	2458
rmtadmin users	2461
Synopsis	2461
rmtadmin version	2461
Synopsis	2461
rmtadmin ziplogs	2461
Synopsis	2461
Positional Parameter	2462
Option	2462
Global Option	2462
Tableau Server Topology Changes	2462
Adding a Node	2462
Removing a Node	2463
Adding or Removing a Tableau Server Process	2464
Re-registering an Agent	2467
Who can do this	2468
Tableau Resource Monitoring Tool Log Files	2468
Log Files	2468
Components	2468
Log Level Configuration	2470

Sending Log Files to Tableau Customer Support	2472
Who can do this	2472
Tableau Log Files	2472
Upgrading Tableau	2473
Ensure Resource Monitoring Tool supports the new Tableau version	2473
Stop Agents	2473
Upgrade Tableau	2473
Update Tableau Version in Resource Monitoring Tool	2474
Restart Agents	2474
Who can do this	2474
Monitor Tableau Server Performance	2474
Charts	2475
Line Chart Resolution	2475
Line Chart Date Range	2475
Environment Overview	2476
Environment Tab	2476
Performance	2477
Tableau Processes	2478
Background Tasks	2478
Concurrent Users	2479
Load Times	2480
View Load Times	2480

External Data Requests	2481
Background Tasks	2481
Server Tab	2481
Insights Tab	2482
Slowest Views	2482
Longest Extract Refreshes	2483
VizQL Session Environment Activity	2484
View Load Request Details	2485
Historic Load Times	2485
View Load Environment Activity	2485
Who can do this	2485
Tools used in Data Collection	2485
Explore Monitoring Data Using Tableau Data Source Files	2486
Requirements	2487
Downloading TDS files from a remote computer	2487
Who can do this	2488
Chargeback Reports	2488
Security	2490
Data Generation	2490
Who can do this	2491
Troubleshoot Tableau Resource Monitoring Tool Issues	2491
Troubleshoot Missing Hardware Performance Data	2491

Step 1: Check the Agent connection status	2492
Step 2: Ensure the Agent is running	2492
Step 3: Ensure the Agent is configured correctly	2492
Step 4: Restart the Agent	2493
Step 5: Contact Support	2493
Who can do this	2493
Troubleshoot Unknown Status of Tableau Server Processes	2493
Step 1: Check Tableau Server Environment Settings	2494
Step 2: Update Tableau Server Machines and Processes	2494
Step 3: Update Machine Name	2495
Step 4: Update Processes and Ports	2495
Step 5: Contact Support	2495
VizQL Session details page says the VizQL process is unknown	2495
Who can do this	2496
Troubleshoot Web Interface Timeouts	2496
Who can do this	2497
Upgrade Power Tools for Server to Tableau Resource Monitoring Tool	2498
Tableau Resource Monitoring Tool Legacy License Key Activation	2499
About Tableau Content Migration Tool	2500
What is Content Migration Tool?	2500
Help and Support	2500
Getting Started with Tableau Content Migration Tool	2500

Pre-Installation	2501
Installation requirements	2501
Compatibility with Tableau Server	2501
Compatibility with Tableau Online	2501
Compatibility with Tableau content	2502
Post-Installation	2503
Limitations when migrating content	2503
Create a migration plan	2503
Upgrading Content Migration Tool	2503
Install Tableau Content Migration Tool	2503
Installation requirements	2503
Installing Content Migration Tool	2504
Who can do this	2504
Upgrade Tableau Content Migration Tool	2504
Before Upgrading	2505
Upgrading Content Migration Tool	2505
Who can do this	2505
Using Tableau Content Migration Tool	2505
Tableau Content Migration Tool Use Cases	2506
Content Promotion	2506
Tailoring Content for Customers	2508
Content Migration	2509

External Content Sharing	2510
Validating Database Migrations	2512
Geographical Content Migration	2513
Maintenance Tasks	2513
Tagging stale content	2514
Restoring content	2514
Partial backup	2514
Migration Plan Overview	2515
Limitations when migrating content	2516
Encryption keys	2516
Migration process	2517
Step 1: Start	2517
Step 2: Planning	2517
Step 3: Migration	2518
Published workbooks	2519
Published data sources	2520
Output	2520
Errors and warnings	2521
Optional: Roll back	2521
Who can do this	2521
Migration Limitations	2521
Server-level configurations	2522

Workbooks with user configurations	2522
Compatibility with Tableau content	2522
Tableau Online migrations	2522
Embedded credentials	2522
Content metrics	2523
Descriptions for projects, workbooks, and data sources	2523
Thumbnails for workbooks and views	2523
Tableau Prep flows	2523
Migration Plans: Servers	2523
Required permissions and licenses	2524
Step 1: Source	2525
Sign in to the source server	2525
Step 2: Destination	2526
Saved connections	2526
Add or edit saved connections	2527
Add saved connections with personal access tokens	2528
Step 3: Continue to the next step	2529
Who can do this	2529
Migration Plans: Source Projects	2530
Step 1: Select your source project	2530
Step 2: Continue to the next step	2531
Who can do this	2531

Migration Plans: Workbooks	2531
Step 1: Workbook Selection	2531
Specific Workbooks Selection	2532
Select All	2533
Display:	2533
Thumbnails	2533
List	2533
Rule Based Selection	2533
Workbooks in projects	2534
Workbooks tagged with	2534
Workbooks published by	2534
All Workbooks Selection	2534
Step 2: Workbook Mapping	2535
Rename Workbook	2536
Change Project	2536
Add Project	2536
Change Prefix	2537
Change Suffix	2538
Step 3: Workbook Transformations	2538
Action URL Replacement	2540
Example:	2540
Set Parameter Value	2541

Remove Images	2541
Remove Tooltip Commands	2541
Replace Images	2542
Example:	2542
Zoom Control Visibility	2542
Web Page URL Replacement	2542
Example:	2543
Step 4: Data Source Transformations	2543
Set Calculation Formula	2546
Set Connection Info	2546
Set Custom SQL	2547
Remove Extract	2547
Step 5: Publish Options	2547
Reset Dashboard Selections	2548
Overwrite Newer Workbooks	2548
Copy Workbook Permissions	2548
Copy Extract Refresh Schedules	2548
Add Option	2548
Add Tags	2550
Remove Tags	2551
Apply Extract Refresh Schedules	2552
Set Permissions	2553

Set Generate Thumbnail As	2554
Step 6: Continue to Next Step	2555
Who can do this	2555
Migration Plans: Published Data Sources	2556
Step 1: Selection	2556
Step 2: Mapping	2557
Delete	2557
Name	2557
Project	2557
Destination Name	2558
Destination Project	2558
Step 3: Data Source Transformations	2559
Replace Table/Schema Name	2560
Set Calculation Formula	2560
Set Connection Info	2561
Set Custom SQL	2561
Remove Extract	2561
Use Tableau Bridge	2562
Step 4: Publish Options	2562
Remove Tags	2564
Add Tags	2564
Apply Extract Refresh Schedules	2565

Set Permissions	2566
Step 5: Continue to Next Step	2567
Who can do this	2567
Migration Plans: User Permissions Mapping	2567
Mapping Limitations	2568
Step 1: Add Mapping	2568
Domain Mapping	2568
User or Group Mapping	2569
Step 2: Change Mapping Order	2570
Step 3: Continue to Next Step	2570
Who can do this	2570
Migration Plans: Migration Scripts	2570
Step 1: Pre-Migration	2570
Working Directory	2571
Run	2571
Command Executable	2571
Command Parameters	2571
Script	2572
Step 2: Post-Migration	2572
Working Directory	2572
Run	2572
Command Executable	2573

- Command Parameters 2573
- Script 2573
 - Step 3: Continue to Next Step 2573
 - Who can do this 2573
- Migration Plans: Plan Options 2573
 - Step 1: Options 2574
- Exclude Extract Refreshes 2575
 - Step 2: Version Control 2576
 - Step 3: Save Plan 2577
 - Step 4: Continue to Next Step 2577
 - Who can do this 2578
- Migrating Workbooks and Data Sources that use Extracts 2578
- Changing data connections that use extracts 2579
 - Option 1: Use Published Data Sources 2579
 - Option 2: Remove the Extract During Migration 2580
 - Option 3: Refresh the Extract After Migration 2580
 - Who can do this 2581
- Consolidate Sites 2581
 - Consolidating sites using the Content Migration Tool 2581
 - Who can do this 2584
- Migration Rollback 2584
 - Step 1: Server 2584

Step 2: Migration	2586
Step 3: Workbooks	2586
Step 4: Published Data Sources	2587
Step 5: Run Rollback	2588
Who can do this	2590
Using the Tableau Content Migration Tool Console Runner	2590
Run Plan	2591
Available options:	2591
Exit codes:	2591
Show Plan Summary	2591
help	2592
version	2592
encryption	2592
improvement	2592
Examples	2592
license	2593
Examples	2593
Who can do this	2593
Example: Scripting Migration Plans	2594
Who can do this	2595
Using the Tableau Content Migration Tool Command Line Interface	2596
migrate	2596

- help 2596
 - Examples 2597
- license 2597
 - Examples 2597
- update 2597
 - Examples 2597
- version 2598
- Who can do this 2598
- Tableau Content Migration Tool Settings 2598
 - Who can do this 2600
- Tableau Content Migration Tool Log Files 2600
 - Content Migration Tool Log File Location 2600
 - Who can do this 2601
- Tableau Server Key Management System 2601
 - Tableau Server local KMS 2602
 - Troubleshoot configuration 2603
 - Multi-node misconfiguration 2603
 - Regenerate RMK and MEK on Tableau Server 2603
- AWS Key Management System 2603
 - AWS KMS for encryption at rest 2604
 - Configure AWS KMS for Tableau Server encrypted extracts 2605
 - Step 1: Create CMK and set key policy for Tableau Server in AWS 2605

Step 2: Collect AWS configuration parameters	2605
Step 3: Configure Tableau Server for AWS KMS	2606
Step 4: Enable encryption at rest	2606
Step 5: Validate installation	2606
Troubleshoot configuration	2608
Multi-node misconfiguration	2608
Refresh AWS CMK	2608
Regenerate RMK and MEK on Tableau Server	2609
Back up and restore with AWS KMS	2609
Azure Key Vault	2609
Azure Key Vault for encryption at rest	2610
Configure Azure Key Vault for Tableau Server encrypted extracts	2611
Step 1: Create a key vault and key for Tableau Server in Azure	2611
Step 2: Collect Azure configuration parameters	2611
Step 3: Configure Tableau Server for Azure Key Vault	2612
Step 4: Enable encryption at rest	2612
Step 5: Validate installation	2612
Troubleshoot configuration	2613
Multi-node misconfiguration	2613
Refresh Azure Key	2613
Back up and restore with Azure Key Vault	2614
Tableau Server External File Store	2614

Why use External File Store?	2614
Managing External File Store	2615
License Management	2615
Supported Migration Scenarios	2615
Backup and Restore	2615
High Availability Considerations	2616
Topology	2616
	2618
Next	2618
Install Tableau Server with External File Store	2618
Prerequisites	2619
Install Tableau Server with External File Store	2620
Step 1: Configure a network share	2620
Step 2: Download and install TSM	2620
Step 3: Initialize TSM	2621
Step 4: Activate and register Tableau Server	2622
Step 5. Enable External File Store	2622
Step 6: Configure the initial node settings	2623
Step 7: Complete the install	2623
Step 8: Post-installation tasks	2623
Who can do this	2624
Next	2624

Reconfigure File Store	2624
Reconfigure Tableau Server with External File Store	2624
Prerequisites	2624
Step 1: Upgrade Tableau Server	2625
Step 2: Activate the Advanced Management license	2625
Step 3: Configure File Store to use an external storage	2626
Reconfigure Tableau Server to use local File Store	2627
Configure Tableau Server to use a different external storage	2628
Who can do this	2629
Backup and Restore with External File Store	2629
Backup Strategies:	2629
Tableau Server with External File Store	2630
Creating a Snapshot Backup	2630
Restoring a snapshot backup	2631
Tableau Server with External File Store and External Repository	2632
Option 1: Include Repository in the Backup	2633
Create a Snapshot Backup	2633
Restoring a Snapshot Backup	2634
Option 2: Backup Repository Separately	2635
Create Snapshot Backups	2635
Restoring a Snapshot Backup	2637
Who can do this	2638

- Performance Considerations for External File Store 2638
 - Who can do this 2639
- Tableau Server External Repository 2639
 - External Repository Considerations 2640
 - Requirements 2641
 - Topology 2641
 - Managing the External Repository 2643
 - License Management 2643
 - Supported Migration Scenarios 2644
 - Backup and Restore 2644
 - SSL Connections 2645
 - Updating the SSL Certificate 2645
 - High Availability Considerations 2646
 - Monitoring the Status of the Repository 2646
 - Getting Logs 2648
 - Next Steps 2648
- Create a PostgreSQL DB Instance on AWS Relational Database Service (RDS) 2648
 - Requirements and Recommendations 2649
 - Create a PostgreSQL DB instance on Amazon RDS 2649
 - Step 1: Create a parameter group 2649
 - Step 2: Create a PostgreSQL DB instance on Amazon RDS 2652

Step 3: Get the PostgreSQL DB Instance Endpoint	2654
Step 4: Download the SSL certificate file (.pem file)	2654
Configuring High Availability for your PostgreSQL DB	2654
Disaster Recovery for your PostgreSQL DB	2654
Who can do this	2655
Create a Azure Database PostgreSQL Instance on Azure	2655
Requirements and Recommendations	2655
Create a Database PostgreSQL instance on Azure	2657
Step 1: Create a delegated subnet for the Azure Database for Post- greSQL instance	2657
Step 2: Create an Azure Database for PostgreSQL instance	2657
Step 3: Configure a server-level firewall rule	2658
Step 4: Configure the Azure Database for PostgreSQL Instance.	2658
Step 5: Get the PostgreSQL DB Instance Endpoint	2660
Step 6: Download the SSL certificate file	2660
Configuring High Availability for your PostgreSQL DB	2661
Disaster Recovery for your PostgreSQL DB	2661
Who can do this	2662
Create a PostgreSQL Database as a Stand-alone Installation	2662
Requirements and Recommendations	2662
Create a stand-alone PostgreSQL Database Instance	2663
Step 1: Install and initialize PostgreSQL	2663

- Step 1: Configure your PostgreSQL Instance 2663
- Super User Settings 2663
- Network and Security 2664
- Database Options 2664
- Update Parameters 2664
- Configure remote connections 2666
 - Configure SSL 2666
 - High Availability and Disaster Recovery 2667
 - Who can do this 2668
- Install Tableau Server with External PostgreSQL Repository 2668
 - Before you install 2668
 - Install and Configure Tableau Server 2669
 - Step 1: Create a configuration file 2669
 - Step 2: Install Tableau Server and Configure the External Repository 2670
 - Step 3: Complete tsm Initialize 2671
 - Step 4: Complete the install 2672
 - Who can do this 2672
- Re-Configure Tableau Server Repository 2672
 - Move a local Tableau Server Repository to external 2672
 - Move the external Tableau Server Repository to the local installation of
Tableau Server 2674
 - Who can do this 2675

Upgrade Tableau Server with External Repository for a New Major Version of PostgreSQL	2675
Before you upgrade	2675
Tableau Server Upgrade	2677
Product Compatibility	2678
Who can do this	2678
Upgrading your RDS Instance	2678
Who can do this	2679
Workload Management through Node Roles	2679
Backgrounder Node Roles	2680
Using Backgrounder Node Roles	2680
Configuration Options	2681
License Requirements	2682
Considerations	2683
File Store Node Roles	2684
Guidelines to Optimize for Extract Refresh and Backup/Restore Workloads.	2685
Fine Tune Extract Query Workload Management	2686
Configuration Options	2687
License Requirements	2688
Who can do this	2688
About Data Management Add-on	2688

Data Management Add-on Features	2689
Tableau Catalog	2689
Tableau Prep Conductor	2691
License the Data Management Add-on	2691
Tableau Prep Conductor	2692
Tableau Catalog	2692
How Data Management Add-on licensing works	2692
User-Based	2693
Core-Based	2693
Tableau Prep Conductor	2697
Overview	2697
Enabling Tableau Prep Conductor on Tableau Server	2697
Enable and Configure Tableau Prep Conductor on Tableau Server	2697
Server Topology	2698
Next step:	2700
Who can do this	2700
Step 1 (New Install): Install Tableau Server with Tableau Prep Conductor ..	2700
Before you install	2701
Install Tableau Server and enable Tableau Prep Conductor	2701
Configure public gateway settings	2701
Enable Tableau Prep Conductor	2702
Verify Tableau Prep Conductor is enabled and running	2703

Dedicate a node for Tableau Prep Conductor	2705
Multi-node installations	2706
Next step	2706
Who can do this	2706
Step 1 (Existing Install): Enable Tableau Prep Conductor	2706
Before you upgrade	2707
Prepare for upgrade:	2707
Configure public gateway settings	2707
Tableau Server Installations using User-Based licenses	2708
Tableau Server single-node installations	2708
Tableau Server multi-node installations	2711
Tableau Server Installations using Core-Based licenses	2714
Tableau Server single-node installations	2715
Tableau Server multi-node installations	2717
Next step	2721
Who can do this	2721
Step 2: Configure Flow Settings for your Tableau Server	2721
Publishing, Scheduling, and Credential Settings	2722
Implication of disabling Tableau Prep Conductor	2723
Configure notifications for flow failures	2723
To enable the server-wide email notification	2724
Set notification values	2725

To configure email notification for a site:	2725
Next step	2726
Who can do this	2726
Step 3: Create Schedules for Flow Tasks	2726
Create a new schedule:	2726
Next step	2727
Who can do this	2727
Step 4: Safe list Input and Output locations	2728
How to safe list input and output locations	2728
Next step	2731
Who can do this	2731
Step 5: Optional Server Configurations	2731
Set the timeout period for flows	2731
Set the threshold for suspended flow tasks	2732
Who can do this	2732
About Tableau Catalog	2732
How Tableau Catalog works	2733
Key Tableau Catalog terms	2734
License Tableau Catalog	2734
Enable Tableau Catalog	2734
Features and functionality	2734
Data discovery	2735

Curation and trust	2735
Lineage and impact analysis	2735
Developer resources	2735

About Tableau Help

Addressing Implicit Bias in Technical Language

In an effort to align with one of our core company values, equality, we have changed terminology to be more inclusive where possible. Because changing terms in code can break current implementations, we maintain the current terminology in the following places:

- Tableau APIs: methods, parameters, and variables
- Tableau CLIs: commands and options
- Installers, installation directories, and terms in configuration files
- Tableau Resource Monitoring Tool (we plan to make changes to non-inclusive terminology in the web interface, error messages, and related documentation soon.)
- Third-party systems documentation

For more information about our ongoing effort to address implicit bias, see [Salesforce Updates Technical Language in Ongoing Effort to Address Implicit Bias](#) on the Salesforce website.

What's New and Changed in Tableau Server

Looking for Tableau Server on Windows? See [What's New in Tableau Server](#).

[Tableau Server Release Notes](#) | [Known Issues](#)

This topic lists all the new features and product changes for the latest release.

New in version 2021.2

User

New features and enhancements related to web authoring and interacting with views on the web are listed in [What's New Tableau Desktop and Web Authoring](#). For topics related to web authoring and interacting with views, see [Using Tableau on the Web](#).

Server or site administrator

Tableau Server in a Container

Beginning with version 2021.2, we introduce Tableau Server in a Container. Tableau Server in a Container is an all-in-one Tableau Server instance running inside of a Linux Docker container.

For more information, see [Tableau Server in a Container](#).

Multiple analytics extensions in a site

Beginning with version 2021.2, you can now create multiple analytics extensions connections for each site. This means that organizations with different language or connection requirements can use analytics extensions on the same Tableau Server site, with different workbooks using different connections. For more information on analytics extensions, see [Configure Connections with Analytics Extensions](#). **Note:** As of 2021.2, Einstein Discovery analytics extensions are still limited to one per site.

New Server service - Collections

Beginning with version 2021.2, a new service has been added to support the Collections feature. The new process is called the Collections service. For more information on what this service does, and guidance on how to configure this process for multi-node installations, see [Tableau Server Collections Service](#).

Collections are like playlists that let users organize content in a way that is meaningful to them. For more information on how to create and manage collections, see [Organize Items in a Collection](#) in the Tableau Desktop and Web Authoring Help.

Configure settings for extracts in web authoring

Beginning with version 2021.2, you can configure settings for extracts that you create in Web Authoring. Optionally, configure options to tell Tableau how to store, define filters for, and limit the amount of data in your extract. For more information, see [Create Extracts on the Web](#).

Streamlined site settings for Ask Data

Streamlined site settings for Ask Data now disable or enable the feature site-wide for all data sources. If it's enabled site-wide, indexing frequency remains customizable for each data source. For more information, see [Disable or Enable Ask Data for a Site](#).

Bring Einstein Discovery predictions into your Prep flows on the web

Beginning with version 2021.2, you can bring prediction models built in Einstein Discovery directly into your Prep flows on the web.

Note: You must have a Salesforce license and user account that is configured to access Einstein Discovery to use this feature.

With the power of machine learning and artificial intelligence, you can bulk score your flow data and generate new fields for predicted outcomes at the row level, as well as add top predictors and recommended improvement fields to include data about which fields contributed to the predicted outcome. Add a Prediction step to your flow, sign into Einstein Discovery and choose from deployed models, and apply them to your flow data. Then generate your flow output and use the new data source to analyze the predicted outcomes in Tableau. This feature was first introduced in Tableau Prep Builder in version 2021.1.3. For more information, see [Add Einstein Discovery Predictions to your flow](#) in the Tableau Prep help and [Create and Interact with Flows on the Web](#).

Share explanations from Explain Data in published dashboards

Explain Data has a reimagined user interface optimized for a broader audience of business users. Authors (Creators or Explorers with editing permissions) can now enable Explain Data for viewers of dashboards and sheets in published workbooks. Viewers can select a mark of interest in the view and run Explain Data to explore their data more deeply than before.

Access to Explain Data in viewing mode is not enabled by default, so authors will need to enable this option. For information on how to enable Explain Data in viewing mode, see [Enable Explain Data in viewing mode in published workbooks and dashboards](#).

For more information on Explain Data, see [Discover Insights Faster with Explain Data](#), [Get Started with Explain Data](#), and [Control who can use Explain Data and what they can see](#).

Daily removal of embedded external assets without downstream connections

Beginning with version 2021.2, a backgrounder job for Tableau Catalog (or Tableau Metadata API) runs daily by default to delete embedded external assets without downstream connections to Tableau content. For more information, see [features.DeleteOrphanedEmbeddedDatabaseAsset](#) and `databaseservice.max_database_deletes_per_run`.

New Subscription jobs configuration key

Beginning in version 2021.2, a new configuration key, `backgrounder.timeout.single_subscription_notify`, changes how you configure the maximum time to complete for single subscription jobs. For more information, see "New configuration key for setting maximum time for single subscription job" in [What's Changed in 2021.2](#).

Configure custom OAuth

You can now configure custom OAuth clients for a site to 1) override server-wide OAuth client settings if configured for the server or 2) enable support for securely connecting to data that requires unique OAuth clients. For more information, see [Configure custom OAuth for a site](#).

Add-ons

Tableau Catalog - part of the Data Management Add-on

Add Field Descriptions - You can add descriptions for fields in a published data source that then display in the Description column on the data source page Lineage tab. You can also see the description on the Data tab when you build a visualization. For more information, see [Describe fields in a published data source](#) in the Tableau Desktop and Web Authoring Help.

Data Quality Warning Updates

- **Formatted text support for warning messages** - When you create or edit a data quality warning message, you can format the text with bold, underline, and italics, and include a link or an image. For more information, see [Set a Data Quality Warning](#).
- **Message required for warnings** - When you create a data quality warning, you're now required to enter a message to display to users. (This doesn't apply to warnings generated when you set Tableau to monitor for extract refresh or flow failures.) For more information, see [Set a Data Quality Warning](#).

Inherited descriptions appear in Tableau Desktop - When a description for a field exists upstream from the field, it now appears in Tableau Desktop with information about where the description is inherited from. For more information, see [Describe fields in a published data source](#) in the Tableau Desktop and Web Authoring Help.

Full table name appears - The table's full name has been added to the table page.

Filter by tags - You can now filter databases, files, and tables by tags in the Connect To dialog box.

Filter by embedded or non-embedded - On the External Assets page, you can filter databases, files, and tables by asset category: embedded or non-embedded.

Resource Monitoring Tool - part of the Server Management Add-on

Automated Process Discovery

Resource Monitoring Tool now supports hot topology changes on Tableau Server and automatically updates the processes information. You no longer have to configure the process information manually.

CLI support for Environment setup

New CLI commands are available to allow you to configure a new environment during install or updating an existing environment. Previously, you had to use the web interface to configure the Environment. Here are the new commands related to Environment setup:

- `rmtadmin create-env`: Creates a new Environment
- `rmtadmin update-env`: Updates an existing Environment:
- `rmtadmin bootstrap-file`: Downloads the bootstrap file to register Agents.

For more information, see [Install the Master Server Using Command Line and rmtadmin Command Line Utility](#) .

New CLI commands

Several new CLI commands have been added to support environment setup and management. Here is a list of all the new commands added in this release:

- `rmtadmin agents`: Lists all the registered Agents on Tableau Server nodes for all environments.
- `rmtadmin create-env`: Creates a new Environment
- `rmtadmin bootstrap-file`: Downloads the bootstrap file to register Agents.
- `rmtadmin cleanup`: Deletes the log files that are older than a certain number of days.
- `rmtadmin delete-env`: Deletes a specific environment and all data that has been collected for that environment.
- `rmtadmin delete-server`: Deletes the configuration information of the Tableau Server node from the environment, deregisters the Agent on that node, and deletes all monitoring data collected specific to that node.

Tableau Server on Linux Administrator Guide

- `rmtadmin delete-server-data`: Deletes the data collected from one or more Tableau Server nodes.
- `rmtadmin deregister-agent`: Deregisters the Agent from the environment.
- `rmtadmin environments`: Lists all the environments.
- `rmtadmin servers` : Lists all the Tableau Server nodes across all environments or a specific environment.
- `rmtadmin test-env`: Tests the various connection points between and Tableau Server.
- `rmtadmin update-env`: Updates the environment settings.

For more information, see [rmtadmin Command Line Utility](#) .

Node Selector

If you have more than 5 Tableau Server nodes, you will now be able to navigate and filter by nodes. Previously on the Tableau Processes chart only showed information for 5 nodes.

Agent Compatibility

Resource Monitoring Tool Agent should be on the same version as the Master Server or a compatible version. If you have one or more Agent on a non compatible version, an incident will be logged and reported. For more information, see [Agent Incidents](#).

External Repository - part of the Server Management Add-on

You can now configure a stand-alone PostgreSQL database instance as the External Repository for Tableau Server. For more information, see [Create a PostgreSQL Database as a Stand-alone Installation](#)

Content Migration Tool - part of the Server Management Add-on

Content Migration Tool now supports personal access tokens as an authentication method for your source and destination servers. For more information, see [Migration Plans: Servers](#).

Changed in 2021.2

Version 2021.2 includes the following changes you should know about before upgrading.

Encrypted communication channel with Active Directory is required

Beginning with version 2021.2, Tableau Server on Linux requires an encrypted LDAP channel when you use Active Directory as an identity store. You must install a valid SSL/TSL certificate before upgrading to 2021.2. Although not recommended, you can also disable the default encrypted LDAP channel. For more information about installing certificate and enabling or disabling encryption for Active Directory communication, see [Configure Encrypted Channel to LDAP External Identity Store](#).

Secure SAML configuration is enforced

Beginning with version 2021.2, Tableau Server enforces secure SAML configuration for Server-wide and Site SAML.

Signature algorithm The certificate that you configure for SAML must use a secure signature algorithm, for example SHA-256. If you attempt to upgrade Tableau Server with a SAML certificate that uses SHA-1 signature hash, upgrade will fail.

We recommend that you upload a new certificate with SHA-256. See [Update SAML Certificate](#).

Alternatively, you can configure Tableau Server to accept the less-secure SHA-1 hash by setting the `tsm wgserver.saml.blocklisted_digest_algorithms` configuration key.

RSA key and ECDSA curve sizes The certificate used for SAML must either have an RSA key strength of 2048, or ECDSA curve size of 256. If you attempt to upgrade Tableau Server with a SAML certificate that uses less-secure key or curve sizes, upgrade will fail.

We recommend that you upload a new certificate with more secure key or curve size. See [Update SAML Certificate](#).

Alternatively you can configure Tableau Server to accept the less-secure sizes by setting the respective configuration keys, `wgserver.saml.min.allowed.rsa_key_size` and `wgserver.saml.min.allowed.elliptic_curve_size`.

Site SAML secure configuration verification

This release includes an updated `tabcmd` utility with the `validateidpmetadata` command. The `validateidpmetadata` command will scan your existing SAML sites and identify the signature algorithms and key sizes that are configured for each site. See `validateidpmetadata`.

Tableau Prep Flow Service has been renamed to Tableau Prep Minerva Service

Beginning with version 2021.2, the Tableau Prep Flow Service (`flowqueryservice`) has been renamed to the Tableau Prep Minerva Service (`flowminerva`). This process appears on the TSM Status page and in the output of the TSM command `tsm status -v`. For more information, see [Tableau Server Processes](#).

New configuration key for setting maximum time for single subscription job

Beginning in version 2021.2, you now set the maximum time to complete a single subscription job using the new `backgrounder.timeout.single_subscription_notify` configuration key. This key replaces use of the `backgrounder.timeout_tasks` key for configuring timeouts on single subscription jobs. As of version 2021.2 the `backgrounder.timeout_tasks` key no longer uses the `single_subscription_notify` option. For more information, see `backgrounder.timeout.single_subscription_notify`.

Features Introduced in Previous Versions of Tableau Server

Looking for Tableau Server on Windows? See [What's New in Tableau Server](#).

[All Tableau Server Release Notes](#) | [All Known Issues](#)

Added or changed in previous versions

Version 2021.1

Added in 2021.1

User

New features and enhancements related to web authoring and interacting with views on the web are listed in [What's New Tableau Desktop and Web Authoring](#). For topics related to web authoring and interacting with views, see [Using Tableau on the Web](#).

Server or site administrator

Integration of Einstein Discovery Predictions

Beginning with version 2021.1, you can integrate predictions directly into Tableau views and dashboards using models deployed by Einstein Discovery in Salesforce. For an introduction to Einstein Discovery in Tableau, see [Integrate predictions from Einstein Discovery into Tableau](#) in What's New in Tableau Desktop and Web Authoring help. Also see the [Einstein Discovery in Tableau](#) page and [demo](#) on [tableau.com](#).

To use Einstein Discovery predictions on Tableau Server, some server configuration is required. For more details, see [Configure Einstein Discovery Integration](#). For information

on required licenses, access, and permissions in Salesforce and Tableau, see [Requirements for access](#).

For information on how to configure and use Einstein Discovery predictions in workbooks and dashboards, see [Integrate Einstein Discovery Predictions in Tableau](#) in Tableau Desktop and Web Authoring help.

Server authorization-to-run (ATR) service

Server authorization-to-run (ATR) enables you to activate Tableau Server in any cloud, container, or virtual environment without being impacted by underlying hardware changes that lead to maximum activation errors. For more information, see [Activate Tableau Server using the authorization-to-run \(ATR\) service](#).

Zero downtime licensing

With zero downtime licensing, you can apply most licensing changes to Tableau Server without needing to restart when license end dates, capacity, or installed features change. For more information, see [Zero downtime licensing](#).

Specify ports that should not be used during dynamic port allocation

Use the new configuration key `ports.blocklist` to specify ports in the dynamically assigned range that should **not** be used by Tableau. Setting this key with specific ports will cause Tableau Server to not use the ports when dynamically assigning ports on start up. For more information, see [Blocking specific ports within the range](#).

Block ingestion of complex content

Use the `tsm configuration set` command option `metadata.ingestor.blocklist` to troubleshoot content ingestion issues with Catalog or Metadata API. For more information, see [metadata.ingestor.blocklist](#).

Use personal access tokens for admin impersonation

Beginning with version 2021.1, you can enable Tableau Server personal access token impersonation. In this scenario, access tokens that are created by server administrators can be used for user impersonation when using the Tableau Server REST API. See [Server administrator impersonation](#) .

StartTLS encryption method supported for LDAP connections to AD

Beginning with Tableau Server on Linux 2021.1, StartTLS is the default method for simple bind LDAP connections to Active directory. The StartTLS method works by establishing an insecure connection with the Active Directory server. After a client-server negotiation, the connection is upgraded to a TLS encrypted connection. This scenario requires a valid TLS certificate on Tableau Server. See [Configure Encrypted Channel to LDAP External Identity Store](#).

Backup performance improvements

Beginning with version 2021.1, you can increase the number of threads the `tsm maintenance backup` command uses. This can significantly improve performance if your hardware supports it. Our testing suggests that the default option provides the best results, however should you choose to go beyond default multithreading option, you should limit the number of threads to the number of virtual processors on the computer, up to a recommended maximum of four. To increase the number of threads, set the `backup.z-std.thread_count` configuration key, and use the `--multithreaded` option with the `tsm maintenance backup` command. For more information about backing up Tableau Server, see [Back up Tableau Server data](#).

New Server Process - Content Exploration Service

Beginning with version 2021.1, a new server process has been added to enhance the searching and navigation capabilities in Tableau Server. The new process is called the Content Exploration Service. For more information on what this service does, and guidance on

how to configure this process for multi-node installations, see [Tableau Server Content Exploration Service](#).

Redesigned notifications center now includes notifications for sharing and comments

Users can now see notifications for new comments and shares by selecting the bell icon in the top right corner of their Tableau site. The redesigned notification center also displays updates for flows and extract jobs. Users can also control where they receive notifications in their Account Settings—directly in Tableau, by email, or both. To configure notifications for all site users, turn on notifications in Settings under Manage Notifications. For settings information, see [Site Settings Reference](#). For more information on this change, see [Changed in 2021.1](#).

Add-ons

Azure Key Vault - part of the Server Management Add-on

Tableau Server can use an Azure Key Vault as a Key Management System (KMS) beginning with version 2021.1. This requires the separately licensed Server Management Add-on for Tableau Server. For more information, see [Azure Key Vault](#).

External File Store and External Repository- part of the Server Management Add-on

Enhanced snapshot backup - If you have both External File Store and External Repository configured for your Tableau Server, you now have additional options to manage backup and restore of your Tableau Server data. For more information, see [Backup and Restore with External File Store](#).

Note: Beginning with Tableau Sever 2021.1, if you have both External File Store and External Repository enabled, Tableau Server Backup process no longer includes

creating a backup file of the repository by default. If you are upgrading from a version earlier than 2021.1, and you have scripts to run or schedule your backups, and want to continue to include the repository backup, see Option 1: Include Repository in the Backup .

Resource Monitoring Tool - part of the Server Management Add-on

1. **Performance improvements**- The Resource Monitoring Tool can now provide real-time performance reporting for a Tableau Server deployment serving up to **5,500** views per hour.
2. **Filters** - New filter options are available in the web interface to improve navigation and help explore event monitoring.
3. **SSL configuration for Tableau Server Repository** - You can now configure secure encrypted communications between Resource Monitoring Tool and Tableau Server Repository database. For more information, see Encrypted Data Collection.

Tableau Catalog- part of the Data Management Add-on

Automated data quality warning - You can set Tableau to monitor for extract refresh failures and for flow run failures. When a failure occurs, Tableau generates a data quality warning that displays to users. For more information, see Set a Data Quality Warning.

Quick search results expanded - Quick search results now include tags on external assets.

Changed in 2021.1

Version 2021.1 includes the following changes you should know about before upgrading.

Change in default behavior for snapshot backups

Note: If you have both External File Store and External Repository enabled and use scripts to do backups, this can potentially result in not having a backup of the repository.

Beginning with Tableau Server 2021.1, if you have both External File Store and External Repository enabled, Tableau Server backup process no longer includes creating a backup file of the repository by default. If you are upgrading from a version earlier than 2021.1, and you have scripts to run or schedule your backups, and want to continue to include the repository backup, see Option 1: Include Repository in the Backup .

SHA-256 hash algorithm enforced for SAML assertions with IdP

Beginning with Tableau Server 2021.1, Tableau Server will hash message signatures and digests with SHA-256 in SAML assertions to the IdP. To change this behavior, see `wgserver.saml.sha256`.

New SAML configuration keys

Beginning with Tableau Server 2021.1, Tableau Server includes three new SAML configuration keys:

- `wgserver.saml.blocklisted_digest_algorithms`: Specifies the digest algorithms that are not allowed as part of certificates used in SAML configuration or for signing SAML assertions. When set, assertions that are signed with a blocklisted algorithm will fail and certificates containing a blocklisted digest algorithm will fail. We recommend blocking the SHA-1(SHA1) algorithm. Before blocking SHA-1, confirm that your SAML certificates digest algorithms are not SHA-1, and that your SAML idP does not sign messages with SHA-1.
- `wgserver.saml.min.allowed.elliptic_curve_size`: Specifies the minimum elliptic curve size required for ECDSA certificates that uploaded and configured

for Tableau Server. If Tableau Server is configured with an ECDSA certificate that does not satisfy this minimum curve size, Tableau Server may go into a degraded state. If you attempt to upload a certificate that does not meet the minimum curve size, Tableau Server will reject the certificate. We recommend a minimum curve size of 256.

- `wgserver.saml.min.allowed.rsa_key_size`: Specifies the minimum key size that required for RSA certificates that are uploaded and configured for Tableau Server. If Tableau Server is configured with an RSA certificate that does not satisfy this minimum key size, Tableau Server may go into a degraded state. If you attempt to upload a certificate that does not meet the minimum key size, Tableau Server will reject the certificate. . We recommend a minimum key size of 2048.

In future releases of Tableau Server, these keys will be configured with secure minimum defaults, as specified in the recommendations above. In future releases, if your SAML configuration is not configured with the minimum requirements, SAML functionality or upgrade may fail. In the meantime, we recommend that you update your SAML configuration to be in compliance with the recommendations above.

Client and CA certificates used for mutual SSL must meet minimum security requirements

Beginning with Tableau Server 2021.1, Tableau Server configured for mutual SSL authentication will enforce secure certificate requirements.

Client and certificate authority (CA) certificates used for mutual SSL must either have an RSA key strength of 2048, or ECDSA curve size of 256.

Tableau Server will fail mutual authentication requests from client certificates that do not meet these requirements.

Additionally, upgrade of Tableau Server will fail if the certificate authority (CA) certificate that is configured for mutual SSL does not meet these requirements.

Tableau Server on Linux Administrator Guide

- To update the CA certificate file for Tableau Server mutual SSL, upload the certificate as described in [Configure Mutual SSL Authentication](#).
- To disable the certificate security enforcement, see the Tableau KB topic, [Tableau Server Using Mutual SSL Authentication Fails to Start or Rejects Login After Upgrade to Tableau Server 2021.1 because Certificates Use Insecure Key Attribute](#).

Improvements to JVM maximum heap size scaling for Tableau Catalog ingestion and elastic server

Beginning with Tableau Server 2021.1, JVM maximum heap size scales automatically for Catalog ingestion and elastic server. For more information, see `non-interactivecontainer.vmopts` and `elasticserver.vmopts`.

Old notifications do not appear in new notifications center

When you upgrade to Tableau Server 2021.1, old notifications will not appear in the new notifications center. All existing email notifications will continue. To continue tracking older notifications, make sure the server setting for Notifications for Extract Jobs and Flow Runs is enabled before upgrading. For more information, see [What's New and Changed in Tableau Server](#).

Version 2020.4

Added in 2020.4

User

New features and enhancements related to web authoring and interacting with views on the web are listed in [What's New in Tableau for Users](#) and [Web Authoring and Tableau Desktop Feature Comparison](#). For topics related to web authoring and interacting with views, see [Using Tableau on the Web](#).

Server or site administrator

Enable and configure Catalog or Metadata API indexing limits for content

Use the following `tsm configuration set` command options to enable and configure Catalog or Metadata API indexing limits, also called event throttling, for new and updated content across all sites:

- `metadata.ingestor.pipeline.throttleEventsEnable`
- `metadata.ingestor.pipeline.throttleLimit`
- `metadata.ingestor.pipeline.throttlePeriodLength`

Activate and register multiple product keys

Beginning with version 2020.4.0, you can activate and register multiple product keys during installation. Tableau Services Manager checks licensing levels (cores versus product keys) and license types (creators versus explorers and viewers) to ensure the installation can be completed due to licensing. You can apply additional product key(s) to Tableau Server if necessary, to achieve the desired number of creator, explorer, and viewer seats, cores, and features to accommodate user capacity. For more information, see [Activate and Register Tableau Server](#).

Configure Tableau Server for offline maps

Beginning with version 2020.4.0, you can configure Tableau Server to use offline maps that are installed with Tableau. To do this, set the `vizqlserver.force_maps_to_offline` configuration key to `true` and apply pending changes. For more information, see [Displaying Maps in an Air-Gapped Environment](#).

Turn off Web Page objects

To prevent Web Page objects in dashboards from displaying target URLs, do either of the following:

- For a specific site, deselect **Enable Web Page Objects** in the [site settings](#).
- For an entire server, set the `vizqlserver.web_page_objects_enabled` [configuration key](#) to `false`.

Improved control of extract query workloads

You can now fine tune extract query workload management in a multi-node deployment by using the new node role, `extract-queries-interactive`. For more information, see [Workload Management through Node Roles](#). To use this feature, you must have a valid Server Management Add-on license activated on your Tableau Server.

Automatic configuration of query cache size

The size of the query cache is now automatically set based on the amount of available system memory. This improves the number of query results that will fit into the cache by default. For more information, see [General Performance Guidelines](#).

The default query cache settings are suitable for most situations but it is possible to manually configure them using the TSM command line interface. The TSM settings are: `native_api.InitializeQueryCacheSizeBasedOnWeights`, `native_api.QueryCacheMaxAllowedMB`, `native_api.LogicalQueryCacheMaxAllowedWeight`, `native_api.MetadataQueryCacheMaxAllowedWeight`, `native_api.NativeQueryCacheMaxAllowedWeight`, and `native_api.QueryCacheEntryMaxAllowedInPercent`. For more information, see `native_api.InitializeQueryCacheSizeBasedOnWeights`.

Custom sign in note appears for server sign in errors

In 2020.4, sign in page notes added to the Sign In Customization setting will also appear when a user receives a sign in error, such as when a user attempts to log in with SSO (SAML, Site SAML, and Open ID). Server administrators can use this note to help users understand how to get a license or why they can't access a site. For more information, see [Customize Your Server](#). For more information, see [Customize Your Server](#).

New processes related to Tableau Prep flows

The 2020.4 release adds some new processes to Tableau Server. The Tableau Prep Flow Authoring (`floweditor`) and Tableau Prep Flow Service (`flowqueryservice`) processes now appear on the TSM Status page and in the output of the TSM command `tsm status -v`. For more information, see [Tableau Server Processes](#).

Ask Data Usage and Adoption admin view

You can now see how users engage with Ask Data on a site, using the new Ask Data Usage admin view. The dashboard highlights the top Ask Data users, data sources, and data source owners, along with headline value metrics. For more information, see [Ask Data Usage](#).

New tsm schedules commands

Version 2020.4.0 introduces new commands for managing scheduled jobs. For this first release you are limited to using the `tsm schedules` commands with backup jobs, but our intent is to expand the scope of these commands in future releases of Tableau Server. For more information, see [Scheduling and Managing Backups](#), [tsm schedules](#), and [tsm maintenance backup](#).

Checkpoint upgrades

Enhancements to the `upgrade-tsm` script give server administrators more options when faced with a failed upgrade. For more information, see "Changed in 2020.4" below.

Additional dynamically configurable keys for logging

Dynamic configuration was introduced in version 2020.2, and version 2020.3 added dynamically configurable logging options to those made available in 2020.2.0. Version 2020.4 continues to build on this with two new keys. When you make configuration changes with these keys, or configuration changes with these keys in combination with Tableau Server Dynamic Topology Changes, you do not have to restart Tableau Server.

New dynamically configurable keys in 2020.4:

Tableau Server on Linux Administrator Guide

- tomcatcontainer.log.level
- vizportal.log.level

For details viewing dynamically configurable keys, see [tsm configuration list-dynamic-keys](#). For more information about the introduction of dynamic configuration in version 2020.2, see [Dynamic Configuration](#). For more on changing logging levels, see [Change Logging Levels](#).

Create extracts on the web

Now you can create extracts directly in web authoring, without using Tableau Desktop. For more information, see [Create Extracts on the Web](#). You can extract new workbooks in addition to existing workbooks. Optionally, the extract creation can run in the background and you can be notified when it's completed. This will let you close your authoring session while the extract is being created, which can be helpful with large extracts that might take a long time.

Create and interact with flows on the web

You can now create flows to clean and prepare your data using either Tableau Prep Builder, Tableau Server, or Tableau Online. Connect to your data, build a new flow, or edit an existing flow and your work is automatically saved every few seconds as you go. Create draft flows that are only available to you or publish your flow to make it available for others. Run your individual flows right from the web or run your flows automatically on a schedule using Tableau Prep Conductor if you have the Data Management Add-on. For more information, see [Create and Interact with Flows on the Web](#).

Add-ons

Tableau Resource Monitoring Tool updates- part of the Server Management Add-on

Resource Monitoring Tool is included in the separately licensed Server Management Add-on for Tableau Server. For more information, see [Get Started with Tableau Resource Monitoring Tool](#).

The following features are included in this release:

Linux Support - Resource Monitoring Tool **Master Server and Agents can now be installed on Linux platform.** For more information, see [Install the Tableau Resource Monitoring Tool](#).

SSL Certificates - You can now use a locally imported SSL Certificates.

- Windows: Install the Tableau Resource Monitoring Tool.
- Linux: SSL Certificate Mode and Requirements.
- Command Line installation are supported for both Linux and Windows installations.
 - Master: Install the Master Server Using Command Line
 - Agent: Install the Agent Using Command Line

Agent OS Detection - Master Server can detect the OS that each Agent is installed on and will list it in the web interface. For more information, see [Install the Tableau Resource Monitoring Tool](#).

Tableau Server External Repository- part of the Server Management Add-on

Tableau External Repository is included in a separately licensed Server Management Add-on for Tableau Server. For more information, see [Tableau Server External Repository](#).

Azure Support - You can now configure Tableau Server External Repository on Azure Database. For more information, see [Create a Azure Database PostgreSQL Instance on Azure](#).

Upgrade - If you have External Repository configured, before upgrading your Tableau Server to 2020.4, you must make sure you are using PostgreSQL version 12 or later. For more information, see [Upgrade Tableau Server with External Repository for a New Major Version of PostgreSQL](#).

Tableau Catalog updates - part of the Data Management Add-on

Tableau Catalog is included in the separately licensed Data Management Add-on for Tableau Server and Tableau Online. For more information, see [About Tableau Catalog](#).

The following features are included in this release:

Data Quality Warning History admin view - Site administrators can see how data quality warnings are being used on the site using the pre-built admin view, Data Quality Warning History. For more information see Data Quality Warning History.

See quality warning details in Tableau Desktop - Tableau Desktop users can hover over the warning icon on the Data tab to see details about the data quality warnings affecting the data in their workbook. For more information, see Set a Data Quality Warning.

Metrics appear in lineage and show data quality warnings - Data quality warnings now appear on metrics affected by a warning. Metrics also appear in the Lineage tool, enabling you to see how a metric might be affected by data changes when performing impact analysis. For more information, see Create and Troubleshoot Metrics.

Tableau Prep Conductor updates - part of the Data Management Add-on

Now, the Data Management Add-on is only needed to schedule and monitor flows with Tableau Prep Conductor. Other flow-related tasks no longer require you to license the add-on. For more information, see Tableau Prep Conductor.

Changed in 2020.4

Version 2020.4 includes the following changes you should know about before upgrading.

SHA1 certificates disabled for SSL mutual authentication

Beginning with Tableau Server 2020.4, Tableau Server configured for mutual SSL authentication will block authentication of users with client certificates that use the SHA-1 signing algorithm.

Users who attempt to log in with SHA-1 client certificates encounter an "Unable to sign in" error, and the following error will be visible in the VizPortal logs:

```
Unsupported client certificate signature detected: [certificate  
Signature Algorithm name]
```

If the clients in your organization are connecting to Tableau Server using SSL mutual authentication and the clients are using certificates with the SHA-1 signing algorithm, then you must perform one of the following actions before upgrading:

- Update the certificates used by your clients. The certificates must use the SHA-256 (or stronger) signing algorithm, or
- Disable the SHA-1 blocklist before upgrading to Tableau Server 2020.4. To disable the blocking functionality, run the following TSM commands before you upgrade:

```
tsm configuration set -k "ssl.client_certificate_login.blocklisted_signature_algorithms" -v ""
```

Note that the value (`-v`) includes an empty set of quotes to specify a null value.

For more information about how to reset this value after you have updated client certificates, see the Knowledge Base article, [Mutual SSL Fails After Upgrading if Certificates Signed with SHA-1](#).

For more information about why this change has been implemented, see the Tableau Community post, [Retiring Default Support for SHA-1 Signed Certificates Used During Mutual SSL Authentication in Customer-Managed Tableau Server 2020.4](#), and the National Institute of Standards and Technology policy statement, [NIST Policy on Hash Functions](#).

wgserver.domain.whitelist config key has been deprecated

Beginning with version 2020.4, the `wgserver.domain.whitelist` configuration key has been deprecated. Use the new key, `wgserver.domain.accept_list` to specify allowed domains.

All functionality of the older key has been transferred to the new key. As of 2020.4.0, existing installations configured with the old key will not be impacted, and the key can still be used, but you must include the `--force-keys` option when setting the key, and the key will be permanently removed in a future release. For all new configurations, use `wgserver.domain.accept_list` instead. For more information about the new configuration key, see `wgserver.domain.accept_list`.

CPU requirement update

Beginning with version 2020.4, Tableau Server requires all nodes to have CPUs that support SSE4.2 and POPCNT instruction sets. These CPUs have been common for more than 10 years and are installed on most newer computers. If you attempt to install or upgrade Tableau on a computer that does not support these instruction sets, the installer will not allow you to continue.

Checkpoint Upgrade for Tableau Server

Version 2020.4 of Tableau Server includes updates to the `upgrade-tsm` script that give you the ability to rerun the script when an upgrade fails. The script keeps track of successful progress, and when you rerun the script after an upgrade failure, restarts at the latest successful "checkpoint." This saves time because successful steps do not need to be repeated on rerunning the script. As a first troubleshooting step, rerun the `upgrade-tsm` script. Doing this can result in a successful upgrade if a momentary environmental problem or timing issue caused the initial failure.

Checkpoint upgrades also give experienced IT professionals the option to further troubleshoot the failure, and make corrections (for example, if there is a problem with disk space, or a permissions issue on a particular file or directory) before rerunning the upgrade script. For troubleshooting details, see [Upgrade fails on 2020.4.0 or later](#).

Upgrades from 10.5 must be done in stages

Beginning with version 2020.4, version 10.5 of Tableau Server on Linux is no longer officially supported. This means you cannot upgrade directly from 10.5.x to version 2020.4.0 or later. To upgrade from version 10.5.x you must first upgrade to a version between 2018.1 and 2020.3, and then upgrade to the newest version. For instructions on upgrading from 10.5, see [Upgrade Tableau Server on Linux from 10.5](#).

PostgreSQL upgrade

Version 2020.4 of Tableau Server includes a major-version update of the PostgreSQL database used for the Tableau repository. During the upgrade a temporary pgsq-ly-only backup is created and this increases the length of time the upgrade takes as well as the amount of temporary disk space used.

If you have External Repository configured, before upgrading your Tableau Server to 2020.4, you must make sure you are using PostgreSQL version 12 or later. For more information, see [Upgrade Tableau Server with External Repository for a New Major Version of PostgreSQL](#)

PostgreSQL database connections require updated driver

Beginning with version 2020.4, PostgreSQL connections use a new JDBC driver. You must download and install this driver on Tableau Server on Linux to enable existing content relying on Postgres connections to properly display. This includes Administrative Custom Views. Follow the instructions on the [Driver Download](#) page to get and install the new driver.

Visual improvements to user, group, and site dialogs

The 2020.4 release adds visual updates to administrator menus and dialogs to make managing users, groups, and sites easier in Tableau Server.

Changes to default JVM heap size for Tableau Catalog ingestion

Beginning with Tableau Server 2020.4, the default JVM heap size for Catalog ingestion is determined by and then scales based on system memory. To review the default values for Catalog ingestion, see `noninteractivecontainer.vmopts`.

Version 2020.3

Added in 2020.3

User

New features and enhancements related to web authoring and interacting with views on the web are listed in [What's New in Tableau for Users](#) and [Web Authoring and Tableau Desktop Feature Comparison](#). For topics related to web authoring and interacting with views, see [Using Tableau on the Web](#).

Server or site administrator

- New command to list dynamically configurable keys
- Set the Site Time Zone for Extracts
- Automatically Suspend Extract Refreshes for Inactive Workbooks
- Manage Licenses for User Groups with Grant Role on Sign In
- Use admin tools to get indexing status for Catalog and Metadata API
- [Tag stale content for archiving](#)

Add-ons

- Tableau Catalog updates - part of the Data Management Add-on
- Tableau Content Migration Tool updates - part of the Advanced Management

New command to list dynamically configurable keys

Version 2020.3 introduces a new TSM command to view all dynamically configurable keys. The command, `tsm configuration list-dynamic-keys`, displays a list of all configuration options you can change without needing to restart Tableau Server. This command helps you distinguish dynamically configurable keys from those that require a server restart, allowing you to decide when to set each.

Dynamic configuration was introduced in version 2020.2, and version 2020.3 adds additional dynamically configurable logging options to those in 2020.2.0. When you make configuration changes with these keys, or configuration changes with these keys in combination with Tableau Server Dynamic Topology Changes, you do not have to restart Tableau Server.

New dynamically configurable keys include:

- backgrounder.log.level
- clustercontroller.log.level
- dataserver.log.level
- filestore.log.level
- gateway.log.level
- tdsservice.log.level
- vizqlserver.log.level

For details on the new command, see [tsm configuration list-dynamic-keys](#). For more information about the introduction of dynamic configuration in version 2020.2, see [Dynamic Configuration](#). For more on changing logging levels, see [Change Logging Levels](#).

Set the Site Time Zone for Extracts

The default time zone for extract-based data sources in a site is Coordinated Universal Time (UTC). Site administrators can now choose to set a different time zone.

For more information, see [Set the Site Time Zone for Extracts](#).

Automatically Suspend Extract Refreshes for Inactive Workbooks

To save resources, Tableau can automatically suspend extract refresh tasks for inactive workbooks. This feature applies to full extract refreshes only, not to incremental extract refreshes. This feature only applies to refresh schedules that run weekly or more often.

For more information, see [Automatically Suspend Extract Refreshes for Inactive Workbooks](#).

Manage Licenses for User Groups with Grant Role on Sign In

Administrators can set a minimum site role for user groups and choose **Grant role on sign in** when creating or modifying that user group. Enabling Grant role on sign in reduces manual work to manage licenses, since administrators can designate site role capabilities by group, and provision licenses when a user actually needs it. By waiting to grant licenses until users sign in, administrators can streamline license provisioning, without granting licenses to inactive users. For more information, see [Grant License on Sign In](#).

Tableau Catalog updates - part of the Data Management Add-on

Tableau Catalog is included in the separately licensed Data Management Add-on for Tableau Server and Tableau Online. For more information about Tableau Catalog, see [About Tableau Catalog](#).

The following features are included in this release:

See data quality warnings in Tableau Desktop - When you open a workbook in Tableau Desktop that has a data quality warning, the warning icon (either a blue circle or a yellow triangle with an exclamation point) appears on the Data tab on the sheet page of the workbook. For more information, see [Set a Data Quality Warning](#)

'Sensitive data' warning type added - You can use this new warning type to set a data quality warning for sensitive data in your organization. For more information, see [Set a Data Quality Warning](#)

Embedded assets are listed in External Assets - You can now see embedded assets listed in the External Assets page. For more information, see [Use Lineage for Impact Analysis](#).

Tag external assets - Add tags to databases, tables, and columns, which enables you to, among other things, tag tables and columns with personally identifiable information (PII). For more information, see [Tag Items](#) in Tableau Desktop and Web Authoring Help.

Use admin tools to get indexing status for Catalog and Metadata API

Tableau Server 2020.3 includes two admin tools to make it easier to get indexing status of Tableau Catalog and Tableau Metadata API:

- Monitor initial ingestion progress and validate its status using the Backfill API. For more information, see [Get Initial Ingestion Status](#).
- After initial ingestion, ensure indexing continues to work as expected using the Eventing API. For more information, see [Get Eventing Status](#).

Tableau Content Migration Tool updates - part of the Advanced Management

Tableau Content Migration Tool is included in the separately licensed Advanced Management for Tableau Server. For more information about Advanced Management, see [About Tableau Advanced Management](#).

The following features are included in this release:

- **Support for Tableau Online** - The Content Migration Tool now includes both Tableau Online and Tableau Server sign-in options when connecting to source and destination servers. To learn about the new sign-in experience, see [Migration Plans: Servers](#).
- **SAML and Single Sign-On** - As of version 2020.3, you can authenticate using SAML and Single Sign-On authentication types for non-scripted migration plans.
- **Tableau Bridge data sources** - Published data sources that are within a private network (inaccessible to the public internet) can be migrated and allowed to refresh using Tableau Bridge. For more information, see [Migration Plans: Published Data Sources](#).
- **Tableau Prep Builder and Tableau Prep Conductor** - Content produced by Tableau Prep Builder or Tableau Prep Conductor (version 2020.2.x and later) can now be included in your migration plans.
- **Asynchronous publishing** - The Content Migration Tool now supports asynchronous publishing of data sources.
- **Resource optimization** - You can specify how many objects to process in memory in parallel with the new **Maximum Processing Concurrency** setting. The setting allows you to optimize resource usage of CPU and memory when performing migration operations, ultimately reducing your memory footprint. For more information, see [Tableau Content Migration Tool Settings](#).
- **Migrate more content** - The Content Migration Tool editor and console runner can

now use up to 4 GB of RAM (increased from 2 GB), allowing you to migrate larger workbooks and data sources.

Tag Stale Content for Archiving

The Stale Content Admin view that was introduced in 2020.2, now has an added functionality that allows you to tag content as stale. You can then choose to archive the tagged content. For more information, see [Stale Content](#).

Changed in 2020.3

Version 2020.3 includes the following changes you should know about before upgrading.

Data Acceleration for Workbooks that use the Object Model Framework

Administrators can now also enable data acceleration for workbooks that use the Object Model framework. See [Data Acceleration](#).

Version 2020.2

Added in 2020.2

Help Design

- Combined What's New and What's Changed in Tableau Server
- Applies to label for Data Management and Server Management Add-ons

User

- Use Relationships to Combine Data for Multi-table Analysis
- Create Metrics to Track Data

Other new features and enhancements related to web authoring and interacting with views on the web are listed in [What's New in Tableau for Users](#) and [Web Authoring and Tableau](#)

Desktop Feature Comparison. For topics related to web authoring and interacting with views, see [Using Tableau on the Web](#).

Server or site administrator

- New Admin Views
- Changes to Create and Modify Schedule
- Data Acceleration
- Extract Query Load Balancing
- Dynamic Configuration

Add-ons

- Tableau Catalog updates - part of the Data Management Add-on

Combined What's New and What's Changed in Tableau Server

Beginning with version 2020.2, the "What's New" and "What's Changed" Help topics are combined in a single "What's New and Changed in Tableau Server" topic (this topic). The topic has been relocated in the Table of Content, and the old "What's Changed" topic now has a link to the combined file.

Applies to label for Data Management and Server Management Add-ons

Topics about add-on features list the add-on name in the Applies to section of the topic. For example:

The screenshot shows a help article page for 'About Tableau Resource Monitoring Tool'. The page includes a search bar at the top right, a breadcrumb trail: 'Tableau Help > Tableau Server on Windows Help > ... > Server Management Add-on > Resource Monitoring Tool', and a 'CONTENTS' sidebar with links to 'Get Started with Tableau Server' and 'What's New and Changed in Tableau Server'. The main title is 'About Tableau Resource Monitoring Tool'. Below the title, it says 'Version: 2020.2' and 'Applies to: Tableau Server Management Add-on', which is highlighted with a red box. On the right, under 'IN THIS ARTICLE', there are links for 'What is Resource Monitoring Tool?' and 'Other articles in this section'.

Use Relationships to Combine Data for Multi-table Analysis

Relationships are a new, flexible way to combine data for multi-table analysis in Tableau. Data sources now support multi-table data models in a new logical layer. Tableau recognizes the separate tables in your data source and performs aggregations at the native level of detail for each table.

Creators can combine tables using relationships. Using relationships between logical tables makes it easier to explore and analyze multi-table data at the right level of detail, without having to write specialized calculations to control aggregations like averages and totals.

You do not need to specify join types for relationships; during analysis Tableau automatically selects the appropriate join types based on the fields and context of analysis in the worksheet.

Note: Your workbook must use an embedded data source for you to be able to edit relationships in the Data Source page in web authoring in Tableau Server or Tableau Online. You cannot edit the data model of a published data source.

In support of multi-table analysis, several parts of the Tableau interface have changed. The Data Source page (canvas, data grid), View Data window, and the Data pane in the worksheet have all been updated to support a multi-table analysis experience. For more information, see [Changes to different parts of the interface](#).

Note: View Data column order has changed in Tableau 2020.2 and later. This change is required to support relationships and logical tables. For details, see [View Data column order changes in Tableau 2020.2 and later](#).

For more information about relationships and changes to data sources, see these topics:

- [What's Changed with Data Sources and Analysis in 2020.2](#)
- [Questions about Relationships, the Data Model, and Data Sources in 2020.2](#)

- [The Tableau Data Model](#)
- [Relate Your Data and Optimize Relationships Using Performance Options](#)
- [How Analysis Works for Multi-table Data Sources that Use Relationships](#)
- [Troubleshoot Multi-table Analysis](#)
- [Walkthrough: Don't Be Scared of Relationships](#)

Also see this Tableau blog post: [Relationships, part 1: Introducing new data modeling in Tableau](#)

Create Metrics to Track Data

Metrics help users track changes to their data. Users can create metrics from a view and use them to monitor key numbers like daily sales. For information on how users create metrics, see [Create and Troubleshoot Metrics](#).

As a Tableau Server administrator, you have the ability to control how often metrics refresh and how failing refreshes are handled. You also can ensure that users are able to create metrics or disable metrics for particular sites. For more information, see [Set Up for Metrics](#).

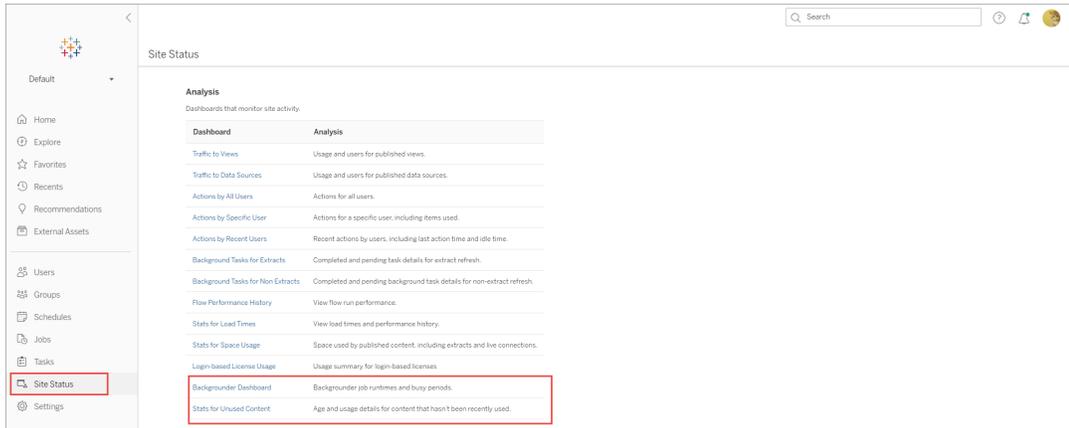
New Admin Views

Two new pre-built administrative views are available in this release:

1. **Backgrounder Dashboard:** Use this view to understand the background job details such as, the number of jobs that have completed successfully, failed, or canceled. You can also identify when your Backgrounder is busy or overloaded. For more information, see [Backgrounder Dashboard](#).
2. **Stale Content aka Stats for Unused Content:** Identify content that hasn't been accessed or opened in a while and how much disk space is being used by stale content. For more information, see [Stale Content](#).

In previous versions of Tableau Server, admin views were all displayed in the same workbook, in separate tabs. However, the two new admin views are displayed as separate workbooks and not part of the existing admin view workbook. You can navigate

to the new admin views from the Server or Site Status page:



Changes to Create and Modify Schedule

The web interface used to create and modify schedule has changed. You can now select more than one day when you create or modify monthly schedules. For more information and updated screen shots of the new web interface, see [Create or Modify a Schedule](#).

Data Acceleration

Administrators can now enable data acceleration for specific workbooks. The workbook loads faster because Tableau Server precomputes the workbook's data in a background process. For more information, see [Data Acceleration](#).

Extract Query Load Balancing

Improvements to load balancing for extract-based queries are now available in Tableau Server. For more information, see [Extract Query Load Balancing](#).

Dynamic Configuration

Version 2020.2 introduces dynamic configuration for certain configuration options or keys. When you make configuration changes with these keys, or configuration changes with these keys in combination with Tableau Server Dynamic Topology Changes, you do not have to restart Tableau Server. This step toward more flexible server management includes

dynamic configuration keys for changing logging levels for some Tableau Server processes, and changing some TSM base file paths.

Dynamically configurable keys include:

- `tsm.log.level`
- `tsm.controlapp.log.level`
- `tabadmincontroller.auth.expiration.minutes`
- `basefilepath.site_import.exports`
- `basefilepath.site_export.exports`
- `basefilepath.backuprestore`

For more information see [Change Logging Levels](#) and [tsm File Paths](#).

Tableau Catalog updates - part of the Data Management Add-on

Tableau Catalog is included in the separately licensed Data Management Add-on for Tableau Server and Tableau Online. For more information about Tableau Catalog, see [About Tableau Catalog](#).

The following features are included in this release:

Raise the visibility of important warnings - You can set a data quality warning to be highly visible, so that users of a visualization that's affected by the warning see a notification, and the warning displays in a more prominent yellow color elsewhere in Tableau. For more information, see [Set a Data Quality Warning](#) and "Use Data Details to see visualization information" in [Explore and Analyze Data in a View](#) in the Tableau Desktop and Web Authoring Help..

Dashboards appear in the Lineage pane - When you select the Dashboards item in the lineage, you can see a list of downstream dashboards. For more information, see [Use Lineage for Impact Analysis](#).

Filter external assets on Certified and data quality warnings - In addition to using search to narrow down your choices on External Assets, you can now filter databases, files, and tables by Certified and by data quality warnings.

More filters when connecting - When you connect to data, you can filter databases and table on connection type.

Changed in 2020.2

Version 2020.2 includes the following changes you should know about before upgrading.

Analytics extensions (external service) connections must be reconfigured

Tableau supports a set of functions that your users can use to pass expressions to analytics extensions for integration with R and Python. Previously, this feature was referred to "external services."

In previous versions of Tableau Server, the analytics extensions configuration applied to the Tableau Server global configuration and was administered with TSM.

Beginning with version 2020.2, you can configure a different analytics extensions connection for each site. Administration of the analytics extensions connection has moved out of TSM and to Tableau Server administrator. Additionally, configuring an analytics extensions connection no longer requires a restart of Tableau Server.

However, if you have configured Tableau Server with an analytics extensions connection, then you must reconfigure that connection after you upgrade Tableau Server to version 2020.2.

Before upgrading, document the existing analytics extensions configuration. On previous versions (2019.1-2020.1) you can run the `tsm security vizql-extsvc-ssl list` command to list the existing connection details. To retrieve the password that is stored for the analytics extensions connection (if any), run the following tsm command:

```
tsm configuration get -k vizqlserver.rserve.password
```

After you have upgraded, configure the analytics extension connection. See [Configure Connections with Analytics Extensions](#).

Oracle data source connections require updated driver

Beginning with version 2020.2, Oracle data sources use a new JDBC driver. You must download and install this driver on Tableau Server to enable existing content relying on Oracle connections to properly display. Until this driver is downloaded and installed on Tableau Server, existing content will not display or refresh. Follow the instructions on the [Driver Download](#) page to get and install the new driver. For details on the changes and how they impact Tableau authors, see [Oracle](#) in the Tableau Desktop and Web Authoring Help.

Version 2020.1

Added in 2020.1

User

New features and enhancements related to web authoring and interacting with views on the web are listed in [What's New in Tableau for Users](#) and [Web Authoring and Tableau Desktop Feature Comparison](#). For topics related to web authoring and interacting with views, see [Using Tableau on the Web](#).

Server or site administrator

- Run Now Server and Site Setting
- New configuration options for Hyper
- Support for Login-based License Management
- Support for Kerberos delegation with Vertica data source
- Support for managing site role limits
- Updates for Permission Dialog
- Lock nested projects

Add-ons

- Tableau Catalog updates - part of the Data Management Add-on
- Tableau Server External File Store (Server Management Add-on)

Run Now Server and Site Setting

This new setting allows Server Administrators to configure whether to allow or block users from manually running extract refreshes, flows runs, and subscriptions. This can be set at the server level to apply to all the sites on Tableau Server or at a site level to apply to only certain sites. For more information, see [Server Settings \(General and Customization\)](#).

New configuration options for Hyper

You can use these new tsm configuration set options to configure Hyper to control resource consumption and improve stability:

- `hyper.memory_limit`
- `hyper.network_threads`
- `hyper.objectstore_validate_checksums`
- `hyper.query_total_time_limit`
- `hyper.session_memory_limit`
- `hyper.temp_disk_space_limit`

For more information, see [tsm configuration set Options](#).

Support for Login-based License Management

Login-based license management, helps you manage licensing for users with Creator roles on Tableau Server and Tableau Online. Users with Explorer or Viewer roles cannot use this feature. If you're using Role Based Subscriptions with Tableau Server or Tableau Online, you can simplify your license management using login-based license management to eliminate separate Tableau Desktop and Tableau Prep Builder product keys. You only need to manage one or more product keys for on-premises Tableau Server, or in the case of Tableau Online, you don't need to manage any product keys at all.

Support for Kerberos delegation with Vertica data source

You can now authenticate to Vertica data sources with Kerberos delegation. To configure Kerberos delegation on Tableau Server, see [Enable Kerberos Delegation](#). To enable

Kerberos for Vertica in Active Directory, see [Enabling Kerberos Delegation for Vertica](#).

Support for managing site role limits

Server administrators can now set a maximum number of Creator, Explorer, and Viewer licenses available for a site. Site administrators can use those quotas to allocate licenses for users on their site. For a site that has user-based licensing with no licensing limits set by the server administrator, maximums for each role are determined by the number of available licenses. To learn more, see [Manage Site Role Limits](#). Server admins can also configure site role quotas via the [REST API](#).

Updates for Permission Dialog

Set permissions faster with the new permissions dialog. New permission [templates](#) and capability groupings match common user scenarios. Editing permission rules is easier than ever with direct editing—double click to enter edit mode. Search for groups and users simultaneously without preselecting your choice. [Copy](#) a custom permission rule and paste it for another group or user. And hand off project management to [Project Leaders](#) more intuitively with the new setting. For more information on permissions, see [Permissions](#).

Lock nested projects

Regulating permissions is now more flexible. Project permissions can be locked independently for nested projects. Top-level projects can act as organizational containers for projects with different permission models. Non-admins can create new locked projects. For more information, see [Lock content permissions](#).

Tableau Server External File Store (Server Management Add-on)

You can now configure Tableau Server to use a SAN or NAS to store File Store data. Using a network attached storage for your File Store data provides some benefits like reduced backup times, by using the snapshot backup technologies. For more information see, [Tableau Server External File Store](#)

Tableau Catalog updates - part of the Data Management Add-on

Tableau Catalog is included in the separately licensed Data Management Add-on for Tableau Server and Tableau Online. For more information about Tableau Catalog, see [About Tableau Catalog](#).

The following features are included in this release:

Added connectors - Connect to more data from external assets. We've added support for the following connectors: Google Big Query, Google Drive, Google Sheets, Box, Dropbox, and OneDrive.

Create workbooks - Create a workbook from the External Assets page or from a database or table page for Catalog-supported connections. For more information, see "Tableau Catalog users create workbooks from external assets" in [Creators: Get Started with Web Authoring](#) in the Tableau Desktop and Web Authoring Help.

Filter on Certified and data quality warnings - Filter data sources, databases, files, and tables by Certified, filter data sources by connections, and filter databases and tables by data quality warnings. For more information, see "Open the Connect to Data page" in [Creators: Connect to Data on the Web](#), or "Connect from Tableau Desktop" in [Connect to Data by Using Tableau Server or Tableau Online](#) in the Tableau Desktop and Web Authoring Help.

Enhanced Data Details pane -

- See a sparkline with the number of times a sheet or dashboard was viewed over the last 12 weeks.
- Easily see that there are Warnings on a view.
- Expand a field to see its metadata, such as a description or calculation.
- When a field in a workbook inherits its description, see where the description comes from.

For more information, see "Use Data Details to see visualization information" in [Explore and Analyze Data in a View](#) in the Tableau Desktop and Web Authoring Help.

Changed in 2020.1

Version 2020.1 includes the following changes you should know about before upgrading.

- Features Introduced in Previous Versions of Tableau Server
- `tableau-server-obliterate` script now preserves backup and log files
- Improvements to the `tsm topology deploy-coordination-service` command
- Large statistical file extracts consume system resources

`tableau-server-obliterate` script now preserves backup and log files

In previous versions of Tableau Server, running the `tableau-server-obliterate` script deleted all content from the Data directory. Beginning with version 2020.1, the default behavior of the `tableau-server-obliterate` script has changed: the script copies and saves Tableau Server backup and log files to the `logs-temp` directory. The default location for the `logs-temp` directory is at `/var/opt/tableau/logs-temp`. You can set options on the script to disable this new functionality.

See [Remove Tableau Server from Your Computer](#).

Improvements to the `tsm topology deploy-coordination-service` command

Beginning with version 2020.1, the command to deploy a new coordination service ensemble waits for the ensemble to be properly deployed and removes the old ensemble before returning control to the command prompt. This simplifies the process for deploying a new coordination service ensemble and eliminates the chances of continuing on with other commands or actions before Tableau Server is properly configured for the new ensemble. You no longer need to run the `tsm topology cleanup-coordination-service` command after deploying a new ensemble in version 2020.1.0 and later.

Note: With this change, all coordination service ensemble commands require input for a "y/n" prompt confirming that a server restart will take place. To run these commands without input, include the `--ignore-prompt` option.

For more information about deploying a coordination service ensemble, see [Deploy a Coordination Service Ensemble](#).

Large statistical file extracts consume system resources

Beginning with Tableau Server 2020.1, a change to the Statistical File (SAS) library may cause large SAS-based extract refreshes to fail. The following connection error will be visible in the backgrounder logs, along with spikes in system memory consumption.

"exceeded per-process memory threshold"

To work around the issue, run the following TSM commands after upgrading Tableau Server:

```
tsm configuration set -k features.ReadStat -v false
tsm pending-changes apply
```

For more information and steps to work around the issue in Tableau Desktop, see the Knowledge Base article, [Large Sas File Extract Refresh Failed And Huge Memory Consumption Observed After Upgraded Tableau Server To 2020.3.1](#).

Version 2019.4

Added in 2019.4

User

- Request access

New features and enhancements related to web authoring and interacting with views on the web are listed in [What's New in Tableau for Users](#) and [Web Authoring and Tableau Desktop Feature Comparison](#). For topics related to web authoring and interacting with views, see [Using Tableau on the Web](#).

Server or site administrator

- Configure recommendations for views
- Enable or disable Ask Data for a site
- Personal access tokens

- Secure SMTP
- Test Email Configuration
- Create Extracts on the Web
- New Installers for Tableau Server and tabcmd
- Custom welcome banner
- Limit visibility of user information
- Manage Sandboxed Extensions

Request access

Users can now request access to see a project, workbook, or view when they don't have viewing permission. When someone requests access to content, Tableau routes an email to the person who controls permissions for that content with information about the request and instructions on making access changes. For information on managing permissions, see [Permissions](#).

Configure recommendations for views

Tableau Server now recommends views to help users discover relevant content. Recommendations for views are an expansion of the system that provides recommendations for data sources and tables.

- To control whether the recommendations feature is enabled for your server, use the `tsm configuration set options recommendations.enabled` and `recommendations.vizrecs.enabled`. For more information, see [tsm configuration set Options](#).
- To schedule when the recommendations trainer runs on your server, adjust the server settings under **Recommendations Training Schedule**. For more information, see [Server Settings \(General\)](#).
- To hide all recommendations on a particular site, or to hide user names from appearing on the recommendation tooltips for a particular site, adjust the site settings under **Recommendations for Views**.

Enable or disable Ask Data for a site

As a site administrator, you can control the availability of Ask Data in the **General** area of site settings. For more information, see [Disable or Enable Ask Data for a Site](#).

Personal access tokens

Personal access tokens provide Tableau Server users the ability to create long-lived authentication tokens. The tokens allow users to run automation with Tableau REST APIs without requiring hard-coded credentials or interactive login. More information about using personal access tokens with Tableau REST APIs is at [Signing In and Out \(Authentication\)](#).

As an administrator, you can revoke and track usage of the tokens. See [Personal Access Tokens](#).

Secure SMTP

You can now configure Tableau Server to connect with your email server over a secure connection. You can encrypt and optionally authenticate with a certificate that you upload and configure on Tableau Server. See [Secure SMTP](#).

Test Email Configuration

After you configure an SMTP connection, use the new command, `tсм email`, to validate it.

Create Extracts on the Web

Now you can create extracts in the browser, without using Tableau Desktop. For more information, see [Create Extracts on the Web](#). You can also use the new `createextracts` and `deleteextracts` commands in `tabcmd` to create or delete extracts for a published workbook or data source. For more information, see [tabcmd Commands](#).

New Installers for Tableau Server and tabcmd

Tableau Server 2019.4 introduces completely rewritten Setup programs for both Tableau Server and `tabcmd`. For more information, see [What's Changed - Things to Know Before](#)

You Upgrade.

Custom welcome banner

You can add a custom message to the welcome banner on the home page for all server users to see. Add up to 240 characters of text and hyperlinks to provide important support information or resource to help server users get started. You can also turn off the default Tableau home banner for all server users. For more information, see [Customize Your Server](#).

Limit visibility of user information

The User Visibility site setting lets administrators manage if users with Viewer and Explorer site roles see other users and groups on the site, which can be important for sites that are used by multiple clients. When User Visibility is set to **Limited**, Explorers and Viewers cannot see information about other users, including aliases, project ownership, comments, and more. Creators and administrators (including Site Administrator Explorers) can still view user information. By default, User Visibility is set to Full. For more information, see [Manage Site User Visibility](#).

Manage Sandboxed Extensions

Tableau now supports Sandboxed extensions, which are a new type of dashboard extension that run in a protected environment hosted by Tableau. Like Network-enabled extensions, Sandboxed extensions are web applications that run in custom dashboard zones and can interact with the rest of the dashboard using the [Tableau Extensions API](#). While Network-enabled extensions have full access to the web, Sandboxed extensions can't share data or make network calls outside of the hosting Tableau Server. Sandboxed extensions provide a new level of security for Tableau users and administrators.

To learn more about controlling extensions, see [Manage Dashboard Extensions in Tableau Server](#). To learn more about extension security and strategies for deployment, see [Extension Security - Best Practices for Deployment](#).

Changed in 2019.4

Version 2019.4 includes the following changes you should know about before upgrading.

- Features Introduced in Previous Versions of Tableau Server
- New tabcmd installer for Windows
- Updates to Tableau Server Job management
- Updates to tsm maintenance cleanup command
- Changed behavior for Copy Link when sharing a view

New tabcmd installer for Windows

The 2019.4 release of Tableau Server features a newly rewritten Setup program for installing tabcmd on Windows.

This update changes some of the command line options. If you have automated the installation of tabcmd, be sure to review the current command line options for the tabcmd Setup program. For more information, see [tabcmd and Install Switches and Properties for tabcmd \(Windows\)](#)

Updates to Tableau Server Job management

The Job management page has several updates and new options including the following:

- The **Jobs** page now includes **Average Run** and **Average Queue** time information for each job.
- Improved job cancellation experience. Server and Site admins can now enter customized notes as well as pick recipients of the e-mail when cancelling a job.
- More information is provided in the job detail dialog box, such as: the last time the job was run successfully, and the name of the job creator.
- Canceling flow run jobs is now supported. Previously, in Tableau Server version 2019.3, only extract refreshes and subscriptions were supported.
- New **Task Type** has been added: Encryption jobs.
- Failed jobs now includes jobs that have been suspended. Suspended jobs are a sub-

status of the failed jobs and have their own icon to distinguish them from other failed jobs.

For more information, see [Managing Background Jobs in Tableau Server](#).

Updates to `tsm maintenance cleanup` command

The `tsm maintenance cleanup` command now includes an option to clear the image cache.

In addition, the `--verbose` option has been removed from the command. Running the command with the `--verbose` option will result in an error. If your organization has automated task using the `tsm maintenance cleanup` command with the `--verbose` option, update your automation scripts.

Changed behavior for Copy Link when sharing a view

Beginning with version 2019.4, the behavior of the **Copy Link** option of the **Share** button in a view no longer includes the "embed=y" parameter by default. To include this parameter you need to edit the copied URL before adding it to your web page code. For more information sharing views by embedding them in web pages, see [Embed Views into Webpages](#) in the Tableau Desktop and Web Authoring Help. For details about the parameter, see [URL parameters for iframe tags](#) in the Tableau Desktop and Web Authoring Help.

Version 2019.3

Added in 2019.3

Version 2019.3 includes the following new features.

User

- Add PDF Attachments to Subscription Emails
- Tableau Catalog features—part of the Data Management

New features and enhancements related to web authoring and interacting with views on the web are listed in [What's New in Tableau for Users](#) and [Web Authoring and Tableau Desktop Feature Comparison](#). For topics related to web authoring and interacting with views, see [Using Tableau on the Web](#).

Server or site administrator

- Support for Embedded Credentials in Bootstrap Files
- Tableau Catalog on the server—part of the Data Management
- Tableau Metadata API
- Advanced Management
- Support for Debian 9 Linux Distributions
- Extract Encryption at Rest
- Date-range Support for Ziplogs
- Tableau Server Usage Data
- Add a Note to Tableau Server Sign In Page
- Support added for Italian
- Job Management

Add PDF Attachments to Subscription Emails

You can give your users the option to attach PDF renderings for views and workbooks subscription emails. You must enable attachment functionality on the server. See [Configure Server Event Notification](#).

Tableau Catalog features—part of the Data Management

Tableau Catalog is included in the separately licensed Data Management Add-on for Tableau Server and Tableau Online, making a variety of features available to you in the data management space. You can use Tableau Catalog to discover data, curate data assets, communicate data quality, perform impact analysis, and trace the lineage of data used in Tableau content. For more information, see [About Tableau Catalog](#).

Tableau Catalog on the server—part of the Data Management

Tableau Catalog is included in the separately licensed Data Management. For more information, see [License the Data Management Add-on](#).

You can give your users access to metadata and features like expanded search, data details, and lineage when you Enable Tableau Catalog. To learn more about how Tableau Catalog and metadata permissions work, see [Manage Permissions for External Assets](#).

Tableau Metadata API

If you're a developer and interested in accessing metadata indexed by or surfaced through Tableau Catalog, see the [Tableau Metadata API Help](#).

Job Management

You can now view, monitor, and manage backgrounder jobs on Tableau Server. For more information, see [Managing Background Jobs in Tableau Server](#).

Support for Embedded Credentials in Bootstrap Files

You can now include embedded credentials when generating bootstrap file. This allows you to add a new node to your Tableau Server installation from the command line without needing to provide TSM administrative credentials when installing the node, and can be especially useful if you automate installation of Tableau. This feature is enabled by default and the embedded credentials are valid for a predetermined amount of time (see `tabadmincontroller.auth.expiration.minutes`) and serve as a session cookie.

Embedded credentials are designed to provide you with flexibility, so you can choose how you use this feature. You can choose to not include embedded credentials any time you generate a bootstrap file. You can also disable the feature entirely using the `features.PasswordlessBootstrapInit` option with the `tsm configuration set` command. When credentials are not included, you are prompted, when adding a node, for the name and password of a user with administrative permissions on the computer where Tableau Services Manager is installed.

Important: The bootstrap file contains a copy of the master keystore file used for encrypting the configuration secrets. The file can also be embedded with credentials which are valid for a pre-determined amount of time (see `tabadmincontroller.auth.expiration.minutes`) and serve as a session cookie. We strongly recommend that you take additional measures to secure the bootstrap file.

The following command set provides an example method to encrypt the bootstrap file output. This method is similar to the encryption process described in more detail at [Securing secrets for import and export operations](#).

Note however, the method here must be passed as separate arguments with trailing `&& \` operators as follows:

```
mkfifo -m 600 /tmp/secure1 && \  
  
tsm topology nodes get-bootstrap-file --file /tmp/secure1 && \  
  
gpg --symmetric --batch --yes --passphrase-file ~/.secret-  
s/pgppassphrase.txt --cipher-algo AES256 --output encrypted.enc  
< /tmp/secure1 && \  
  
rm /tmp/secure1
```

For more information, see `tsm topology nodes get-bootstrap-file`.

Advanced Management

Advanced Management is a separately licensed Add-on which will provide enhanced security, manageability, and scalability capabilities for Tableau Server.

- **Improved scalability:**
 - For Tableau Server deployments hosted on AWS, you can now choose to host your metadata repository externally with Amazon RDS Postgres. Amazon RDS offers scalability, reliability, high availability and security built-in for PostgreSQL. By integrating with AWS, you will be able to take advantage of these additional benefits of deploying the cloud. For more information, see [Tableau Server](#)

External Repository.

- You can also tune your Tableau Server processes by customizing which nodes process data engine queries and background jobs. For more information, see [Workload Management through Node Roles](#).

- **Increased operational efficiency:**

- A new Content Migration Tool which was acquired from InterWorks is included as part of Advanced Management.

The Content Migration Tool helps streamline your workbook and data source content work flows. You can manage the movement of content between development and production or between other environments. For more information, see [About Tableau Content Migration Tool](#)

- **Enhanced security:**

If your organization will be enabling extract encryption at rest, you can also integrate Tableau Server with AWS Key Management Service to provide an enhanced level of integration and security. For more information, see [Tableau Server Key Management System](#).

Support for Debian 9 Linux Distributions

Beginning with Tableau Server 2019.3, Debian 9 (and later) Linux distributions are supported. See [Before you install...](#)

Extract Encryption at Rest

Extract encryption at rest is a data security feature that allows you to encrypt .hyper extracts while they are stored on Tableau Server.

Tableau Server administrators can enforce encryption of all extracts on their site or allow users to specify to encrypt all extracts associated with particular published workbooks or data sources. See [Extract Encryption at Rest](#).

Date-range Support for Ziplogs

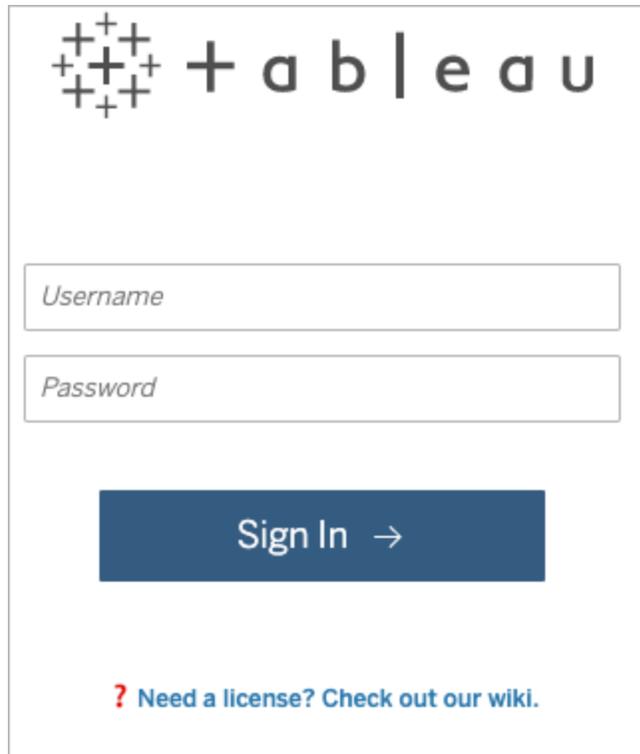
Two new date options for the `tsm maintenance ziplogs` command, `--startdate` and `--enddate`, give you the ability to specify a date range for the logs being gathered. These options must be used together, and cannot be used with the `--minimumdate` option. For more information, see `tsm maintenance ziplogs`.

Tableau Server Usage Data

Beginning with Tableau Server 2019.3, server usage data is sent to Tableau by default. This is the case for new installs and upgrades and can be disabled. You can opt out during installation, upgrade, or at any time after Tableau Server has been installed. For more information about usage data, see [What's Changed - Things to Know Before You Upgrade](#) in the [What's Change](#) article, and the general overview, [Server Usage Data](#).

Add a Note to Tableau Server Sign In Page

You can add an optional note and URL to your server's sign in page to help new users gain server access or for server-wide announcements. The setting also supports emojis. To set a custom note, sign in to a site on Tableau Server. On the left-side navigation pane, select **Manage all sites** from the drop-down site list. Select **Settings** to and add a message to **Sign In Customization**. For more information about customizing Tableau Server, see [Customize Your Server](#).

The image shows a login interface for Tableau. At the top left is the Tableau logo, which consists of a cluster of plus signs followed by the word "tableau" in a lowercase, sans-serif font. Below the logo are two input fields: the first is labeled "Username" and the second is labeled "Password". Below these fields is a dark blue button with the text "Sign In" and a right-pointing arrow. At the bottom of the interface, there is a red question mark icon followed by the text "Need a license? Check out our wiki." in blue.

Support added for Italian

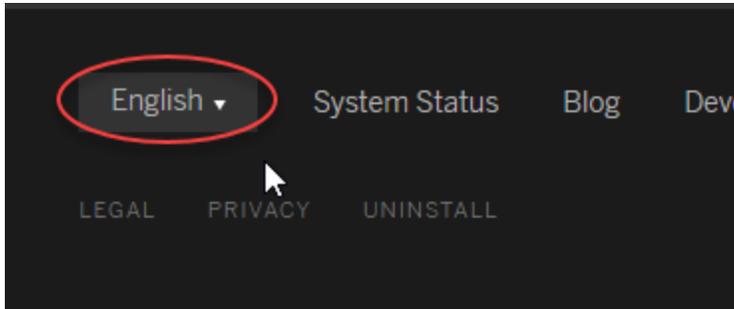
Tableau has added support for Italian. You can now set Tableau to display the user interface in one of eleven languages:

- Chinese (Simplified)
- Chinese (Traditional)
- English (United Kingdom)
- English (United States)
- French
- German
- Italian
- Japanese
- Korean
- Portuguese
- Spanish

For more information, see [Language and Locale for Tableau Server](#).

Tableau Server on Linux Administrator Guide

In addition to supporting Italian in all Tableau products, the help is also available in these languages. To read the help in your preferred language, navigate to the bottom of the main [Tableau Help](#) page, and select the language from the footer.



Changed in 2019.3

Version 2019.3 includes the following changes you should know about before upgrading.

- Default cipher suite no longer includes default support for old ciphers
- Tableau Server Service Manager has been removed
- Tableau Server Usage Data
- Upgrade without needing to provide a password
- Some tsm command options are no longer configurable

Default cipher suite no longer includes default support for old ciphers

The 2019.3 version of Tableau Server no longer includes default support for Triple-DES, IDEA, and CAMELIA ciphers for SSL connections. These ciphers are no longer considered adequately secure. In the previous versions of the Security Hardening Checklist, we included a recommendation to disable the Triple-DES cipher, which was enabled by default.

Depending on your environment, you may need to make additional configurations before our after upgrading:

- If you disabled the Triple-DES cipher as documented in the Checklist, then we recommend running the following commands before you upgrade to 2019.3:

```
tsm configuration set -k ssl.ciphersuite -d
```

```
tsm pending-changes apply
```

This command reverts Tableau Server to the 2019.2 (and older) default cipher suite configuration. After you have upgraded to 2019.3, Triple-DES, IDEA, and CAMELLIA will be removed from the default supported ciphers.

Do not run the command if you made other changes to the `ssl_ciphersuite` key.

- If you did not change the `ssl_ciphersuite` key, then Setup will update the key value to the new default as part of the upgrade process. The new default value for the `ssl_ciphersuite` key is

```
HIGH:MEDIUM:!aNULL:!MD5:!RC4:!3DES:!CAMELLIA:!IDEA:!SEED.
```

- If your users still require support for Triple-DES, then we recommend upgrading to more recent browsers that support more secure ciphers. If that is not an option, then you can run the following commands after you have upgraded. The first command sets the `ssl_ciphersuite` key to the pre-2019.3 configuration, which supports Triple-DES:

```
tsm configuration set -k ssl.ciphersuite -v
```

```
"HIGH:MEDIUM:!aNULL:!MD5:!RC4"
```

```
tsm pending-changes apply
```

Tableau Server Service Manager has been removed

The Tableau Server Service Manager (`tabsvc`) has been removed. Instead, `systemd` manages the Tableau business processes that were previously managed by `tabsvc`:

- Gateway
- Repository & Cluster Controller
- File Store

Tableau Server on Linux Administrator Guide

- Application Service
- SAML Service
- Backgrounder
- Data Server
- VizQL Server
- Data Engine
- Cache Server
- Search & Browse

You can use the following scripts to start and stop the business services that tabsvc managed in previous versions:

```
start-administrative-services
```

```
stop-administrative-services
```

The scripts are located at `/opt/tableau/tableau_server-
/packages/scripts.<version>/.`

Usage data is sent to Tableau

Beginning with Tableau Server 2019.3, server usage data is sent to Tableau by default. This is the case for new installs and upgrades. You have the ability to opt out during installation, upgrade, or at any time after Tableau Server has been installed. For more information about usage data, see [Server Usage Data](#).

To disable the sending of server usage data:

- If you are installing Tableau Server for the first time, you can disable this feature during the configuration of initial node settings. For details, see [Configure Initial Node Settings](#).
- If you are upgrading Tableau Server, the `upgrade-tsm` script will let you know about the option and how to disable it.
- If you want to have installed or upgraded Tableau Server and want to disable the

sending of server usage data, you can do this from the TSM Maintenance page, or from the command line. For details, see [Server Usage Data](#).

Upgrade without needing to provide a password

Beginning with Tableau Server 2019.3, when you upgrade from version 2019.2 or later, the `upgrade-tsm` script no longer requires you to specify a user and password. Version 2019.3 and later take advantage of the What's New and Changed in Tableau Server feature and uses the account you have logged in as to authenticate the upgrade. You can override this behavior with the `--user` option, to specify an administrative user. When you do this, you can also provide a password using the `--password` option, or enter a password when prompted. Upgrades from versions earlier than 2019.2.0 will still prompt you for a password.

Some tsm command options are no longer configurable

Beginning with Tableau Server 2019.3, the following options are enabled on install and are no longer configurable:

- `backgrounder.enable_sort_jobs_by_job_rank`
- `backgrounder.enable_task_run_time_and_job_rank`

Version 2019.2

Added in 2019.2

Version 2019.2 includes the following new features.

Help Design

A blue **Send feedback** icon is displayed in the bottom right corner of every Help page to make it easier for you to send feedback about the content in a topic. Click the icon to let us know if a topic is helpful to you, and also to comment on what needs improvement.

🔗 IN THIS ARTICLE

What is a site

Version: 2019.2

You might be used to using the term *site* to mean “a collection of connected computers,” or perhaps as the short form of “website.” In Tableau-speak, we use *site* to mean a collection of users, groups, and content (workbooks, data sources) that’s walled off from any other groups and content on the same instance of Tableau Server. Another way to say this is that Tableau Server supports multi-tenancy by allowing server administrators to create sites on the server for multiple sets of users and content.

User

- New features and enhancements related to web authoring and interacting with views on the web are listed in [What's New in Tableau for Users](#) and [Web Authoring and Tableau Desktop Feature Comparison](#). For topics related to web authoring and interacting with views, see [Using Tableau on the Web](#).

Server or site administrator

- Direct access with TSM CLI
- Local authentication password policy improvements
- Select from ten different languages for Tableau
- Set a site-level start page
- Elastic Server on multiple nodes
- SSL connection to TabPy external service
- Configuring the threshold for suspending data-driven alerts
- Configuring logging levels for TSM processes

Direct access with TSM CLI

Beginning in the 2019.2 release of Tableau Server, running tsm commands will not require you to enter a password if the following are true:

- The account you are running commands with is a member of the TSM-authorized group, by default, the `tsmadmin` group. The Tableau unprivileged user (by default, the `tableau` user) and root account may also run TSM commands.
- You are running commands locally on the Tableau Server that is running the Tableau Server Administration Controller service. By default, the Tableau Server Administration Controller service is installed and configured on the initial node in a distributed deployment.

See [Authenticating with tsm CLI](#).

Local authentication password policy improvements

Beginning in the 2019.2 release of Tableau Server, you can now set password policies for Tableau Server that is configured for local authentication. For example, you can specify password length, expiry, and complexity. Additionally, you can set accounts to be locked out after too many failed sign-in attempts. You can also enable users to reset their own passwords. See [Local Authentication](#).

Select from ten different languages for Tableau

Tableau has added support for two new languages, Chinese (Traditional) and English (United Kingdom). You can now set Tableau to display the user interface in one of ten languages:

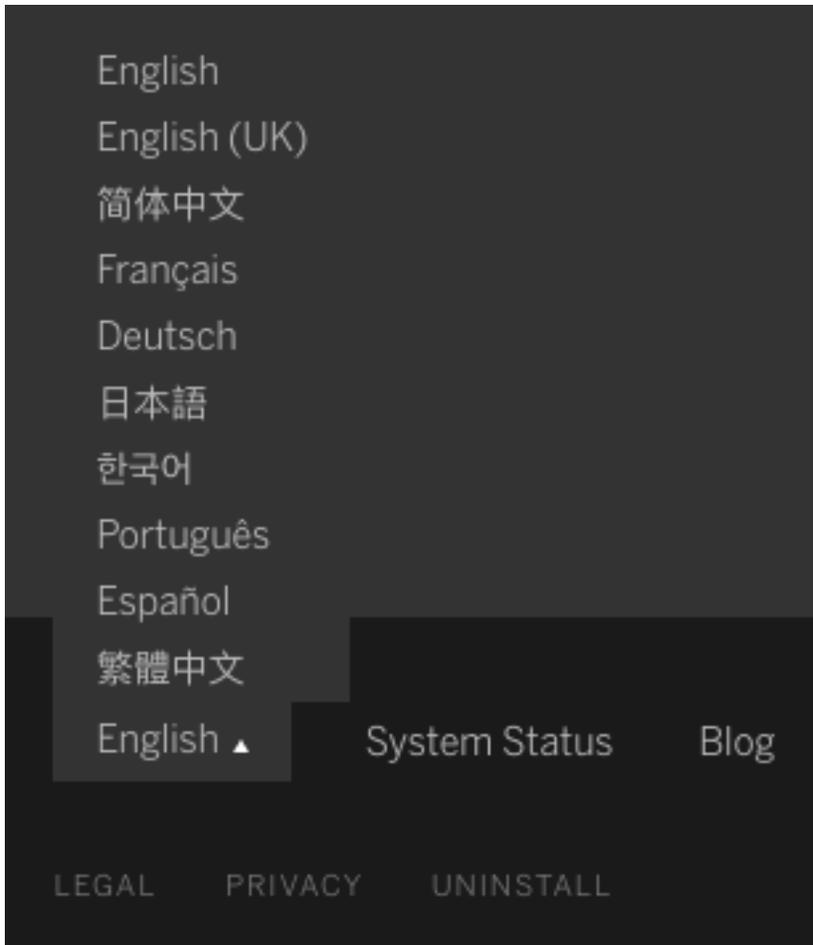
- Chinese (Simplified)
- Chinese (Traditional)
- English (United Kingdom)
- English (United States)
- French
- German
- Japanese
- Korean

Tableau Server on Linux Administrator Guide

- Portuguese
- Spanish

For more information, see [Language and Locale for Tableau Server](#).

In addition to supporting two new languages in all Tableau products, the help is also available in these languages. To read the help in your preferred language, navigate to the bottom of the main [Tableau Help](#) page, and select the language from the footer.



Set a site-level start page

Beginning in the 2019.2 release of Tableau Server, you can set a start page for all site users, so that they land on a particular page when they sign into Tableau Server. Previously, server

administrators could only set a start page at the server-level. See [Set the Default Start Page for All Users](#).

Elastic Server on multiple nodes

Elastic Server is used by Ask Data to index data. You can now run Elastic Server processes on more than one node in a cluster. It is recommended to have an odd number of Elastic Server processes running. For more information, see [Tableau Server Processes](#).

SSL connection to TabPy external service

In previous versions, SSL connections to TabPy were not supported. In Tableau Server 2019.2, you can configure SSL connections. Configuring external services is done with `tsm` commands at `tsm security`.

Configuring the threshold for suspending data-driven alerts

By default, an alert is suspended after 350 consecutive alert failures. Server administrators can configure the threshold number of alert failures before an alert is suspended. For more information, see [Set Up for Data-Driven Alerts](#).

Configuring logging levels for TSM processes

You can now easily change the logging levels of Tableau Services Manager (TSM) processes. This allows you to increase logging levels to troubleshoot issues (if you are working with Tableau Support for example). For more information, see [Change Logging Levels](#).

Changed in 2019.2

Version 2019.2 includes the following changes you should know about before upgrading.

- Minimum hardware requirements have changed
- Upgrade requires at least 15% free disk space on each node
- "Referrer-Policy" HTTP header added
- Map-related internet addresses have changed

Minimum hardware requirements have changed

Beginning with version 2019.2, Tableau Server requires a minimum of 4 cores and 16 GB of RAM in order to install. The installation program will check that the computer you are installing on satisfies these minimums before you can install. If the computer does not meet these hardware minimums, Tableau cannot be installed. For details, see [Minimum Hardware Requirements and Recommendations for Tableau Server](#). For more information about this change, see this Tableau blog post: [Updated machine requirements for Tableau Server trials](#).

Upgrade requires at least 15% free disk space on each node

Upgrading to version 2019.2 includes a version update to the PostGRES database that Tableau uses internally. This update means that a backup of the database is done before upgrading, and that requires a minimum of 15% free space on each node.

"Referrer-Policy" HTTP header added

Beginning in 2019.2, Tableau Server includes the ability to configure Referrer-Policy HTTP header behavior. This policy is enabled with a default behavior that will include the origin URL for all "secure as" connections (policy no-referrer-when-downgrade). In previous versions, the Referrer-Policy header was not included in responses sent by Tableau Server. This change is not likely to impact user scenarios for most browsers. However, we recommend testing browser-based user scenarios after you upgrade to 2019.2. In the event a user scenario is effected, you can disable or change this header behavior. See [HTTP Response Headers](#).

Map-related internet addresses have changed

Beginning in 2019.2, Tableau Server needs access using port 443 to two new map locations:

- mapsconfig.tableau.com
- api.mapbox.com

These replace the address used by previous versions: maps.tableau.com. For details on configuring Tableau Server for proper internet access, see [Configuring Proxies for Tableau Server](#).

Version 2019.1

Added in 2019.1

Version 2019.1 includes the following new features.

Changes to Tableau Help

Updates have been made to help you navigate our content, including a new right-hand menu to quickly scan and select different topics on a page. We've also created content within tabs, to easily show how to complete tasks for web or command line interface.

User

- New features and enhancements related to web authoring and interacting with views on the web are listed in [What's New in Tableau for Users](#) and [Web Authoring and Tableau Desktop Feature Comparison](#). For topics related to web authoring and interacting with views, see [Using Tableau on the Web](#).
- Web authoring auto recovery

Server or site administrator

- Option to install to non-default location on RHEL-distros
- Microservice Container Processes
- Pluggable Authentication Modules (PAM)
- TSM client authentication
- Ask Data and Elastic Server Processes
- Tableau Prep Conductor Server Process
- Node roles for Backgrounder Process
- Tableau Data Management Add-On
- SSL connection to Rserve external service

Web authoring auto recovery

With Tableau Server 2019.1, editing and authoring content on the server will auto-recover in the event of a process failure. As you make changes to content, Tableau Server will intermittently (and silently) save changes. In previous versions, a process failure results in lost changes if you did not save them prior to the failure. In Tableau Server 2019.1, changes you make prior to a failure will be persisted when you reconnect to the content.

By default, Tableau Server will save content after you make five changes to it. Your server admin can adjust this interval with the `vizqlserver.NumberOfWorkbookChangesBetweenAutoSaves` option using `tsm configuration set Options`. File size limit (`vizqlserver.WorkbookTooLargeToCheckpointSizeKiB`) and recovery attempts per session (`vizqlserver.RecoveryAttemptLimitPerSession`) can also be configured.

Option to install to non-default location on RHEL-distros

Starting with version 2019.1, if you are installing Tableau Server on a RHEL-like distribution of Linux, you can choose to install to a non-default location. The default location is `/opt`. If you want or need to install Tableau to another location, you can now do so, even when upgrading. Installing to a non-default location is not supported on Ubuntu because of limitations of the deb package. For more information about installing Tableau Server, see [Install and Initialize TSM](#). For more information about upgrading, see [Upgrading from 2018.1 and Later \(Linux\)](#).

Microservice Container Processes

The 2019.1 release adds some new processes to Tableau Server. Two container processes, Interactive Microservice Container and Non-Interactive Microservice Container now appear on the TSM Status page and in the output of the TSM command `tsm status -v`. These processes contain microservices and cannot be configured independently by administrators. For more information, see [Tableau Server Microservice Containers](#).

Pluggable Authentication Modules (PAM)

In the 2019.1 release, TSM authentication process uses PAM directly and then falls back to an authentication scheme using *substitute user* (`su`) if PAM fails or is not configured with a directory service. This improvement provides a more integration with LDAP directory services. See TSM Authentication.

TSM client authentication

In the 2019.1 release, certificate trust and management for TSM clients (Web UI and CLI) have been updated to make encrypted connections to Tableau Server easier to manage. See Connecting TSM clients.

Ask Data and Elastic Server Processes

With Tableau Server 2019.1, Ask Data and Elastic Server processes have been added to Tableau Server to support the new Ask Data feature. For more information, see Tableau Server Processes.

The Ask Data process is installed and configured automatically. It cannot be configured manually. Logs are stored in the `nlp` directory. For more information, see Tableau Server Logs and Log File Locations.

The Elastic Server process is installed and configured automatically. The Elastic Server heap size can be configured by using the `elasticserver.vmopts` TSM configuration option. For more information, see `tsm configuration set Options`. Only one Elastic Server process can be running and it can optionally be moved to any node in the cluster.

Tableau Prep Conductor Server Process

Tableau Prep Conductor is a new server process that has been added to Tableau Server to support automation of running and managing flows. Tableau Prep Conductor leverages the scheduling and tracking functionality of Tableau Server so you can automate running flows to update the flow output instead of logging into Tableau Prep to manually run individual flows as your data changes. For more details, see Tableau Prep Conductor.

Tableau Prep Conductor is licensed through Tableau Data Management Add-on. For more information, see [Licensing Tableau Prep Conductor](#).

Tableau Data Management Add-On

The Data Management Add-on is a new license that includes Tableau Prep Conductor. For more information, see [Licensing Tableau Prep Conductor](#).

Node roles for Backgrounder Process

A new `tsm topology` command that gives you the ability to specify the type of tasks that backgrounder can perform has been added. In 2019.1, using the new command, you can isolate certain nodes to run only flow tasks or run all tasks other than flows. This is useful if you are planning to use Tableau Prep Conductor to publish, run and schedule flows. For more information, see [Workload Management through Node Roles and Tableau Prep Conductor](#).

SSL connection to Rserve external service

In previous versions, SSL connections to Rserve and TabPy were not supported. In Tableau Server 2018.3, you can configure SSL connections. In addition, connecting to external services is now enabling, disabling, and configuring is now done with native `tsm` commands under `tsm security`.

Changed in 2019.1

Version 2019.1 includes the following changes you should know about before upgrading.

- Install to non-default locations (RHEL-like distributions only)
- High-resolution thumbnails
- Project card images

Project card images

In version 2019.1 and later, project card images will not appear on the project card unless they are added as the last item in the project description and encapsulated with ! (exclamation marks). For more information, see [Add a Project Image](#).

Version 2018.3

Added in 2018.3

Version 2018.3 includes the following new features.

Changes to Tableau Help

Search results within the product help have been expanded to include results for all Tableau products and help articles. Help layout and navigation, such as table of contents placement and breadcrumbs, have been updated to create a more consistent cross-product help experience.

User

- New features and enhancements related to web authoring and interacting with views on the web are listed in [What's New in Tableau for Users](#) and [Web Authoring and Tableau Desktop Feature Comparison](#). For topics related to web authoring and interacting with views, see [Using Tableau on the Web](#).
- Secure extract data at the row level

Server or site administrator

- Materialize calculations in extracts already published to Tableau Server
- Improved content browsing (in-release beta)
- Support for Content Security Policy
- Improved support for LDAP
- SSL connection to Rserve external service
- SAML now supports PKCS#8 key files

- Improved key and secrets refresh process
- Backup without compression (added in version 2018.3.2)

Materialize calculations in extracts already published to Tableau Server

If you need to retroactively materialize calculations in extracts that have already been published to Tableau Server, you can use `tabcmd` to refresh the published extract with the `--addcalculations` option. For more information about materializing calculations in extracts, see [Materialize Calculations in Your Extracts](#) in the Tableau User Help. For more information about the `tabcmd` command option, see `refreshextracts workbook-name or datasource-name`.

Secure extract data at the row level

Secure your extract data at the row level using existing approaches for implementing row-level security in Tableau. For more information, see [Restrict Access at the Data Row Level](#) in the Tableau User Help.

Improved content browsing (in-release beta)

With Tableau Online and Tableau Server 2018.3, you and your web users can get a sneak preview of our reorganized content browsing experience. The new experience lets you see multiple content types—views, workbooks, data sources, flows, projects—in one grid or list. For example, you can now see all content inside a particular project and its sub-projects.

This new experience intends to make content discovery more intuitive, and it's the first in a set of planned changes that are aimed at improving self-service and governance of your Tableau content.

To try the new browsing experience, enable the site-level setting we've provided for it. For more information, see [Preview the Updated Content Browsing Experience](#).

After you enable the new experience, encourage your users to sign in to our [Early Feedback](#) site, share their thoughts with us, and see our regular development updates.

Support for Content Security Policy

Tableau Server supports the Content Security Policy (CSP) standard. CSP is intended to be an additional layer of security against cross-site scripting and other malicious web-based attacks. CSP is implemented as a HTTP response header that allows you to specify where external resources, such as scripts and images, can be safely loaded from.

See [Content Security Policy](#).

Improved support for LDAP

Tableau Server 2018.3 includes improved support for LDAP, such as connection pooling, more flexible root configuration, and commas in object class names. See the new topic, [External Identity Store Configuration Reference](#).

Connections to external services support SSL

In previous versions, SSL connections to Rserve and TabPy were not supported. In Tableau Server 2018.3, you can configure SSL connections. In addition, connecting to external services is now enabling, disabling, and configuring is now done with native tsm commands under tsm security.

SAML now supports PKCS#8 key files

You can now use password-protected key files with SAML. See [Configure Server-Wide SAML](#). The passphrase is encrypted at rest along with other server secrets. See [Manage Server Secrets](#).

Improved key and secrets refresh process

In previous versions of Tableau Server, refreshing server secrets, keys, and tokens required that admins manually copied the keystore to each node in distributed deployments.

In Tableau Server 2018.3, the keystore is distributed to all nodes when you run the tsm security regenerate-internal-tokens command.

Backup without compression (added in version 2018.3.2)

Tableau Server version 2018.3.2 introduced a new option for the `tsm maintenance backup` command that allows you to disable compression when doing the backup. This can be beneficial for customers who have backups that take a long time. Using the option speeds up the time for the backup, but results in a larger backup file. If using this command option in a multi-node installation, we strongly recommend you have a File Store instance configured on your initial node to reduce the need to backup across nodes. To learn more about TSM backup, see `tsm maintenance backup`.

Changed in 2018.3

Version 2018.3 includes some changes you should know about before upgrading.

- Connecting to secondary trusted Active Directory domains may result in error
- Floating worksheet captions become transparent
- Changes to some Data Engine `tsm` configuration set options
- SSL Off-loading and SAML

Connecting to secondary trusted Active Directory domains may result in error

A secondary domain is one that Tableau Server connects to for user synchronization, but is a domain where Tableau Server is not installed. Tableau Server will attempt to connect to secondary domains for user and group synchronization. But in some cases, Tableau Server will be unable to connect to the secondary domain, which will result in the error, "Domain not in whitelist (errorCode=101015)."

Setting the `wgserver.domain.whitelist` option in TSM is required by a fix for the security vulnerability, [\[Important\] ADV-2020-003: Tableau Server Forced Authentication](#). As of February 2020, the fix for this vulnerability is included in all latest versions and maintenance releases of Tableau Server. See `wgserver.domain.whitelist`.

Floating worksheet captions become transparent

To make a floating caption visible again, content authors will need to change the caption background to a color.

Changes to some Data Engine tsm configuration set options

Two new configuration tsm set configuration options have been added: `hyper.hard_concurrent_query_thread_limit` and `hyper.soft_concurrent_query_thread_limit`

The new options replace the `hyper.num_job_worker_threads` and `hyper.num_task_worker_threads` options available in Tableau Server versions 2018.3 and earlier. The `hyper.num_job_worker_threads` and `hyper.num_task_worker_threads` have been deprecated and will no longer work in 2019.1 and later.

SSL Off-loading and SAML

Beginning with version 2018.3, Tableau Server validates SAML response message returned from the IdP. If your organization terminates SSL connections from the IdP at a proxy server before sending the authentication request to Tableau Server, then users may be unable to login with SAML after you upgrade from 2018.2.

In this scenario, SSL is "off-loaded" at the proxy server, which means the https request is terminated at the proxy server and then forwarded to Tableau Server over http. Since SSL is off-loaded at the proxy, Tableau Server will validate with the protocol that it receives (http), but the IdP response is formatted with https, so validation will fail unless your proxy server includes the X-Forwarded-Proto header set to `https`. See [Configuring Proxies for Tableau Server](#).

Version 2018.2

Added in 2018.2

Version 2018.2 includes the following new features.

User

- Include colleagues on data conversations with @mentions

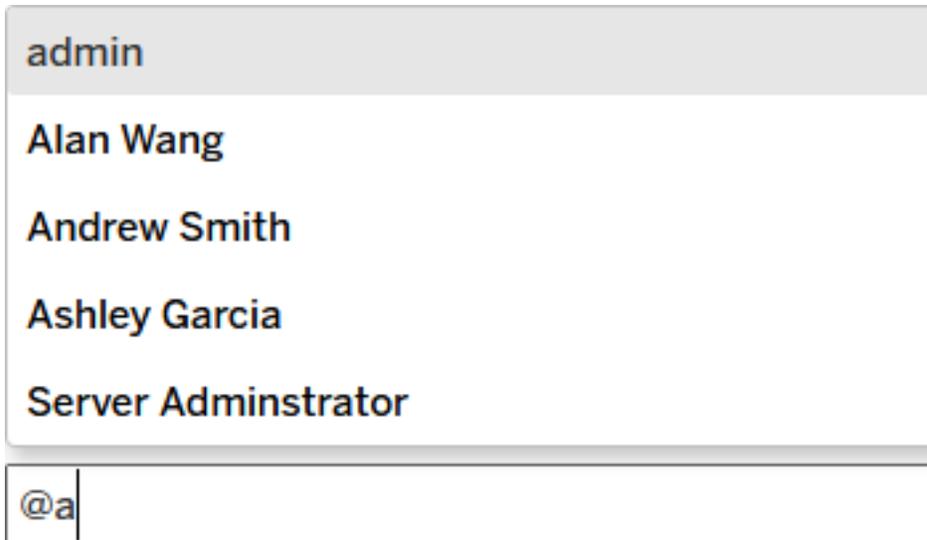
New features and enhancements related to web authoring and interacting with views on the web are listed in [What's New in Tableau for Users](#) and [Web Authoring and Tableau Desktop Feature Comparison](#). For topics related to web authoring and interacting with views, see [Using Tableau on the Web](#).

Server or site administrator

- Create a support case from the TSM UI or CLI
- Manage Dashboard Extensions in Tableau Server
- Automated daemon port configuration
- Amazon Linux 2 now supported

Include colleagues on data conversations with @mentions

To engage people in a conversation about compelling data you find in a view, @mention them in comments. As you begin to type a name, matching people with Tableau accounts at your organization appear in a list you can choose from. For more information, see [Comment on Views](#) in Tableau User Help.



Site administrators can disable @mentions in the General section of site settings.

Create a support case from the TSM UI or CLI

You can take a log file snapshot and send it to Tableau Support directly from the Tableau Services Manager UI or CLI. To learn how, see [Log File Snapshots \(Archive Logs\)](#)

Manage Dashboard Extensions in Tableau Server

Dashboard extensions are web applications that provide users new ways to interact with data from other applications directly in Tableau. You can control the dashboard extensions that you allow to run in Tableau. You can manage an extension's access to data and can block extensions to prevent them from running. To learn more about controlling extensions, see [Manage Dashboard Extensions in Tableau Server](#).

Automated daemon port configuration

In previous versions of Tableau Server on Linux, configuring the licensing daemon port was a required, manual step for multinode deployments. In the 2018.2 version, this process happens automatically. Tableau Server on Linux will automatically detect and configure the appropriate ports for the licensing daemon.

Amazon Linux 2 now supported

Beginning with version 2018.2, Amazon Linux 2 (AL2) is now a supported distro for Tableau Server on Linux. For details on all supported distros, see [Before you install...](#)

Changed in 2018.2

Version 2018.2 includes some significant changes you should know about before upgrading. The following sections summarize the significant changes to Tableau Server 2018.2 and provide links for additional information.

- [Updates to Tableau Server Apache Logs](#)
- [The Read Only site role is now deprecated](#)
- [The tabcmd initialuser command prompts for password](#)
- [New ports added](#)

Moving the repository in a two-node cluster not supported

Updates to Tableau Server Apache Logs

Apache log files stored in `/logs/httpd` now contain records with several formatting improvements, including the following:

- Server names are shown according to the `UseCanonicalName` setting, which makes it easier to distinguish between nodes in a multi-node server deployment.
- Date and time are now tracked in a single date-and-time string that can be readily recognized as a single value by a variety of data analysis tools, including Tableau Desktop.
- Time zones are now easier to parse
 - On Linux, the numerical offset from UTC is provided.
 - On Windows, the name of the timezone is enclosed in quotes

Tableau Server on Linux Administrator Guide

Date, time and time zone formatting changes are illustrated by an example log entry from Tableau Server version 2018.1 and earlier releases:

```
:::1 - - 2018-05-09 08:51:48.872 GMT Daylight Time 80 "POST
/vizql/w/Regional/v/Obesity/sessions/A971A92AF77A4F4A9813065BDCEB24D6-0:0/-
commands/tabsrv/render-tooltip-server HTTP/1.1" "-" 200 445
"593" 135021 WvKolPH4UcM5KrzUbl8EwAAA@E
```

For purposes of comparison, here is a log entry from Tableau Server version 2018.2:

```
10.210.24.3 127.0.0.1 - 2018-05-09T16:07:58.120 "GMT Daylight
Time" 80 "POST /vizql/w/Supper-
store/v/Overview/bootstrapSession/sessions/185CCDC854A44765BB0298E93B403879-
0:3 HTTP/1.1" "-" 200 136026 "784" 2370951
WvMOzgKIhfzh9kFWO@ow2gAAA1Y
```

The Read Only site role is now deprecated

The **Read Only** site role that was introduced in Tableau Server version 2018.1 is now deprecated. Any users assigned to the **Read Only** site role in version 2018.1 are reassigned to the **Viewer** site role after an upgrade to version 2018.2. To learn more about site roles, see [Set Users' Site Roles](#).

The `tabcmd initialuser` command prompts for password

In previous versions, the `tabcmd initialuser` command required a `--password` parameter to set the initial administrator password. In the `tabcmd` included with this version of Tableau Server, you can leave the `--password` parameter out of the command to be prompted in the shell instead. See `initialuser`.

New ports added

We've added new service ports in 2018.2.

Automatic port assignment

For installations where automatic port assignment is enabled (this is recommended and the default) there should be no impact to upgrading. The new ports will be automatically selected on upgrade.

Manual port assignment

If you have disabled automatic port assignment, *you will need to manually specify the new ports in your configuration before upgrading.* For more information about manually assigning ports, see Manual port assignment.

Ports added in 2018.2

Port names	Description
You must define each of these for every node if you disable dynamic port mapping.	
hyper.port	Data engine primary port.
hyper.connection.port	Data engine connection port.
licenseservice.vendor_daemon.port	License service vendor daemon port.
samlservice.port	SAML service port.
tabadminagent.columbo.port	Administration Agent service discovery port
tabadminagent.filetransfer.port	TSM Agent file transfer port.
vizportal_0.microservice.extensions.port	Application server extensions port.
vizportal_0.monolith_grpc.port	Application server GRPC port.

Version 2018.1

Added in 2018.1

Version 2018.1 includes the following new features.

User

- Resize headers on the web
- Streamline discussion threads by deleting comments

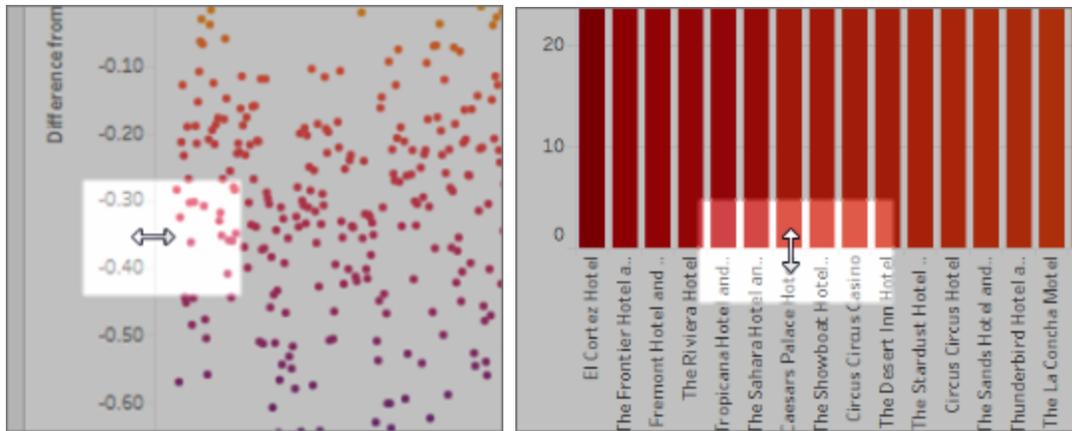
New features and enhancements related to web authoring and interacting with views on the web are listed in [What's New in Tableau for Users](#) and [Web Authoring and Tableau Desktop Feature Comparison](#). For topics related to web authoring and interacting with views, see [Using Tableau on the Web](#).

Server or site administrator

- New user-based term licenses are now available
- New and renamed site roles are now available
- Improved support for managing web data connectors

Resize headers on the web

When editing a view on the web, you can resize the width of row headers and the height of column headers.



Streamline discussion threads by deleting comments

If a comment on a view is unnecessary or inaccurate, users can quickly delete it by clicking the **X** in the upper-right corner. For more information, see [Comment on Views](#) in Tableau User Help.

To disable the ability to delete comments, change the [tsm configuration set](#) option, `viz-portal.commenting.delete_enabled`, to `false`.

New user-based term licenses are now available

Tableau offers multiple types of user-based term licenses that grant a range of web authoring and other capabilities at various price points, providing the flexibility for organizations to pay for the data analysis and data visualization capabilities that each type of user in their organization needs. To learn more, see [User-based licenses](#).

New and renamed site roles are now available

Starting in the version 2018.1 release, new and updated site roles are now available. To learn about these site roles, see [Set Users' Site Roles](#). To learn about changes in the REST API to accommodate new site roles, see [What's New in the REST API](#).

Improved support for managing web data connectors

Tableau Server now provides improved support for web data connectors (WDCs):

- A new set of tsm data-access commands are now available to help you manage WDCs.
- You can also manage WDCs using the new web-data-connector-settings Entity.

Changed in 2018.1

Version 2018.1 includes some changes you should know about before upgrading. The following sections summarize the significant changes to Tableau Server 2018.1 and provide links for additional information.

- Removed `tabadmin` command options
- Updated tsm commands
- Updated `upgrade-tsm` script

Removed `tabadmin` command options

The previously deprecated `--license` and `--administrator` options are no longer available when using the `tabcmd createsiteusers`, `tabcmd createusers` and `tabcmd syncgroup` commands. These are breaking changes for any scripts that still use the `--license` or `--administrator` options. To update your scripts, use the `--role` option with these commands to specify the license levels or site roles. For details, see `tabcmd` Commands.

Updated tsm commands

As of version 2018.1 we have added or expanded the following tsm commands:

- `tsm maintenace cleanup`—The `tsm maintenance cleanup` command was added in version 10.5.1. For version 2018.1 it has been enhanced with additional options to give you more flexibility. For details, see `tsm maintenance cleanup`.
- `tsm reset` —Added in version 2018.1, this command allows you to reset the initial administrative user. For details, see `tsm reset`.

Updated upgrade-tsm script

As of version 2018.1 the upgrade-tsm script you run when you upgrade from a previous version now requires you to include the name of a TSM administrator when you run it. For example:

```
sudo /opt/tableau/tableau_server/packages/scripts.<version_code>/upgrade-tsm -u <tsm_system_admin> -p <tsm_system_admin_password> --accepteula
```

where `tsm_system_admin` is a user with administrative permissions on the initial node computer.

Version 10.5

Added in 10.5

Version 10.5 includes the following new features.

User

- Extracts use the new .hyper format
- Changes to the way values are computed
- Filtered search hidden by default

New features and enhancements related to web authoring and interacting with views on the web are listed in [What's New in Tableau for Users](#) and [Web Authoring and Tableau Desktop Feature Comparison](#). For topics related to web authoring and interacting with views, see [Using Tableau on the Web](#).

Server or site administrator

Tableau Server on Linux Administrator Guide

- Optimized server performance for data alerts
- Nested projects
- Tableau Data Engine with Hyper
- Tableau Services Manager
- Hot topology support
- Lightweight Directory Access Protocol (LDAP) support

Optimized server performance for data alerts

To improve overall performance, Tableau Server now rechecks failing data alerts only when it next checks data conditions for all alerts. You can revert to the previous setting, which checked failing alerts every 5 minutes, by changing the [tsm configuration set](#) option, `dataAlerts.retryFailedAlertsAfterCheckInterval`, to `false`.

Nested projects

You can now create project hierarchies when you want to create content segments within a larger area. For example, create a project for each department, say Operations. In each of those top-level projects, separate content based on how the audience uses it, say connecting to certified data versus in-progress collaboration. For more information, see the following articles:

- Use Projects to Manage Content Access
- **Navigate projects** in the article [Navigate Tableau on the Web](#)

Filtered search hidden by default

Filtered search options are now located on the right side of pages, and are hidden by default.

The toggle is a dark icon () when the Filters pane is open and a light icon () when the Filters pane is closed. For details on searching, see [Search Content](#).

To use filtered search, click .

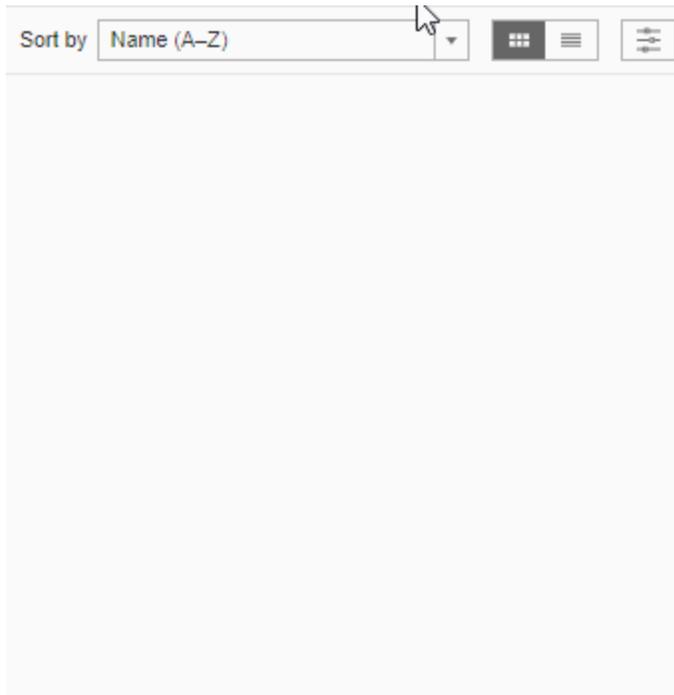


Tableau Data Engine with Hyper

Hyper is Tableau's new in-memory Data Engine technology optimized to bring faster extract and query performance, and increased scalability. You can now analyze large data sets faster and faster extract creation. For more information, see [Tableau Server Data Engine](#).

Extracts use the new .hyper format

Beginning with version 10.5, new extracts use the new .hyper format. Extracts in the new format take advantage of the improved data engine, which supports the same analytical and query performance as the data engine before it, but for even larger extracts. Similarly, when an extract task is performed on a .tde extract using version 10.5, the extract is upgraded to a .hyper extract. For more information, see [Extract Upgrade to .hyper Format](#).

Changes to the way values are computed

To improve data source efficiency and scalability, and produce results that are consistent with commercial databases, values in your data source can be computed differently in version 10.5. In some cases, these changes can cause differences with the values and marks in your view between version 10.4 (and earlier) and version 10.5 (and later). This applies to extract data sources, and can apply to multi-connection data sources, data sources that use live connections to file-based data, data sources that connect to Google Sheets data, cloud-based data sources, extract-only data sources, and WDC data sources. For more information, see [Changes to values and marks in the view](#) in the Tableau User Help.

Tableau Services Manager

Tableau Server version 10.5 features our new web-based server configuration and management application, Tableau Services Manager (TSM). To learn more about TSM, see [Tableau Services Manager Overview](#).

Hot topology support

Tableau Server includes support for hot topology changes. This allows you to change the number of Backgrounder and VizQL processes running on nodes that already have these processes - without the need to restart Tableau Server. To learn more about these and other Tableau Server processes, see [Tableau Server Processes](#).

Lightweight Directory Access Protocol (LDAP) support

Tableau Server now supports Lightweight Directory Access Protocol (LDAP). To learn more about LDAP support in Tableau Server, see [Identity Store](#)

Changed in 10.5

Version 10.5 includes some changes you should know about before upgrading. The following sections summarize the significant changes to Tableau Server 10.5 and provide links

for additional information.

- Tableau Server configuration changes
- Changes to extracts
- Version compatibility
- Viewer (can publish) and Unlicensed (can publish) site roles removed
- Nested projects
- Nested projects and tabcmd
- Filtered search hidden by default

Tableau Server configuration changes

Hyper powers the new Data Engine in Tableau Server 10.5 and introduces the changes described below. For more information, see [Tableau Server Data Engine](#).

Beginning with version 10.5, Tableau Installer **automatically** installs Data Engine when you install file store, Vizportal, VizQLServer, data server, or backgrounder and cannot be installed separately. Every node that has an instance of one of these processes will also have a single instance of the data engine process. You can no longer configure more than one instance of data engine on a node. To learn more about Tableau Server Processes see [Tableau Server Processes](#).

Previously, in versions 10.4 and earlier, for distributed installations, the backgrounder process launched temporary instances of data engine (tdeserver64.exe) process during extract creation or refreshes. In Tableau Server 10.5, a single instance of data engine is installed automatically when you install backgrounder. The backgrounder process uses this single instance of data engine that is installed on the same node. For more information on configuration recommendations, see [Distributed Installation Recommendations](#).

Upgrade may affect your current configuration for a multi-node cluster, especially if you have certain processes isolated on a separate node. For more information, see [Tableau Server Data Engine](#).

Changes to extracts

Beginning with version 10.5, new extracts use the new `.hyper` format. The improved data engine can read both the `.tde` and `.hyper` file formats, so the latest version of Tableau Server can continue to view and read workbooks that are using `.tde` extracts.

Existing extracts will **not** automatically be upgraded to the new `.hyper` format during the Tableau Server or Tableau Desktop upgrade process. Only when certain extract tasks are performed on the `.tde` extract, the extract is upgraded to the `.hyper` format. For example, an automatic or scheduled refresh will upgrade a `.tde` extract to a `.hyper` extract.

- For more information on extract upgrades, see [Extract Upgrade to .hyper Format](#).
- For recommendation on how to handle automated or schedule refresh tasks before you upgrade, see [Server Upgrade - Disable Scheduled Tasks](#).

NOTE: Use the Extract API 2.0 to create (`.hyper`) extracts, which requires Tableau 10.5 and later. If you are using Tableau 10.4 and earlier, continue to use the Tableau SDK to create `.tde` extracts. The Tableau SDK is no longer being maintained. For more information, see [Extract API 2.0](#).

Version compatibility

Upgraded extracts are not backward compatible. After extracts are upgraded from `.tde` to `.hyper` format, they cannot be opened or refreshed in Tableau Desktop 10.4 and earlier.

Tableau recommends that Tableau Server and all Tableau Desktops in your organization be on the same version and be upgraded to 10.5 on the same schedule. For more information on specific extract compatibility scenarios, see [Impact of extract upgrade](#)

Viewer (can publish) and Unlicensed (can publish) site roles removed

The site roles **Viewer (can publish)** and **Unlicensed (can publish)** have been deprecated and are no longer available. Users who have either of these site roles on Tableau Server 10.4 or earlier will be reassigned in Tableau Server 10.5.

Deprecated site role	In 10.5 is reassigned to
Viewer (can publish)	Viewer
Unlicensed (can publish)	Unlicensed

Nested projects and content navigation

You can now create project hierarchies when you want to segment content into sub-sections within a larger area. For example, you might create a project for each department. In each of those top-level projects, you can separate content based on how the audience uses it, such as connecting to certified data versus in-progress collaboration. For more information, see the following articles:

- Use Projects to Manage Content Access
- **Navigate projects** in the article [Navigate Tableau on the Web](#)

Nested projects and tabcmd

Using `tabcmd`, you can specify only a top-level project in a project hierarchy. To automate tasks you want to perform on a sub-project or content in a sub-project, you must use the equivalent Tableau [REST API](#) call.

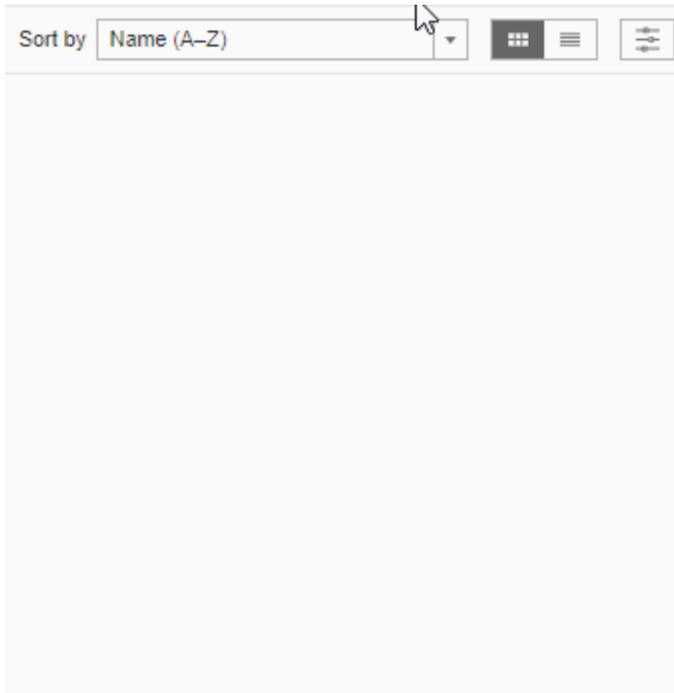
Not all actions are available through the REST API. If you use Server on Windows and want to run an unscheduled extract refresh or append a file to an extract, a possible alternative is to use the [Tableau Data Extract Command Line Utility](#).

Filtered search hidden by default

Filtered search options are now located on the right side of pages, and are hidden by default.

The toggle is a dark icon () when the Filters pane is open and a light icon () when the Filters pane is closed. For details on searching, see [Search Content](#).

To use filtered search, click .



Plan Your Deployment

The articles in this section provide information on planning a Tableau Server deployment.

In addition to consulting the articles in this section, we recommend that you review [Tableau Blueprint](#) as a first step in your planning workflow. Tableau Blueprint is a step-by-step guide to becoming a data-driven organization.

Server Administrator Overview	91
Tableau Services Manager Overview	99
Infrastructure Planning	103

Server Administrator Overview

Tableau Server on Linux integrates with a number of components in your IT infrastructure to provide a unique self-service data analytics culture for your users. It's important that you, as a server administrator, understand how Tableau Server fits into your IT infrastructure.

The topics in this section provide information on planning, deploying, tuning, and managing Tableau Server.

If you are new to Tableau Server, and you want to deploy it in your organization, we encourage you to deploy Tableau Server as a single server in a test environment first. The easiest way to do a single-server installation and to understand the essential requirements is to follow the steps in [Everybody's Install Guide](#).

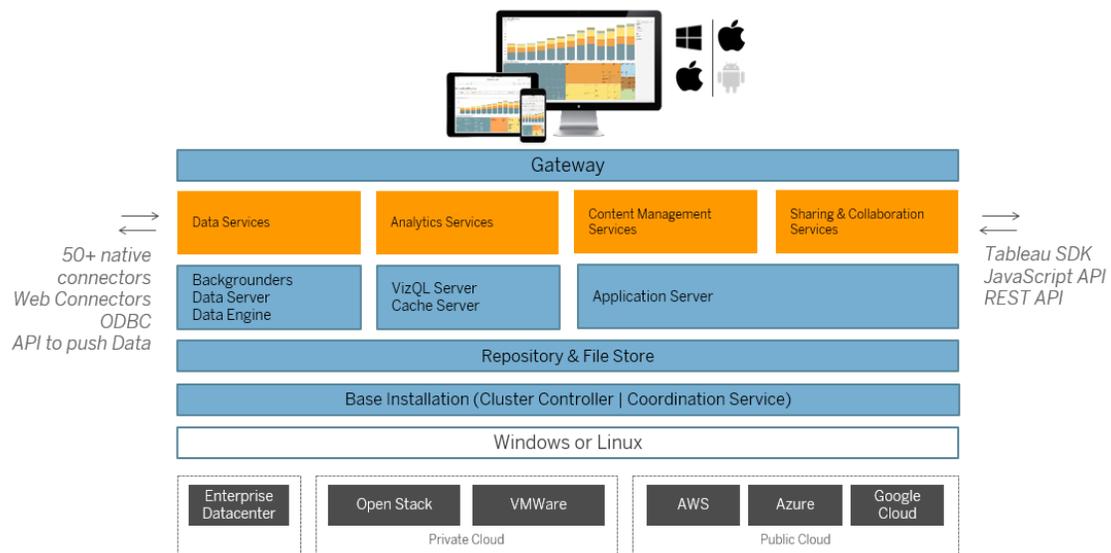
If you are deploying Tableau Server as part of a broader effort to transform your organization into a data-driven culture, see [Tableau Blueprint](#). Tableau Blueprint is a step-by-step guide to becoming a data-driven organization, whether your organization is new to modern, self-service analytics or you've already deployed and need to broaden, deepen, and scale the use of data.

This topic provides a brief overview of how to think about Tableau Server and how it interacts with your existing IT infrastructure.

Looking for Tableau Server on Windows? See [Server Administrator Overview](#).

Architectural overview

Tableau Server is a collection of processes that work together to provide a full self-service analytic platform for your users. The following diagram shows a high-level architectural view of Tableau Server.



Multiple server processes (shown in blue above) work together to provide services at various tiers. The Gateway process is the component that redirects traffic from all Tableau clients to the available server nodes in a cluster.

Data Services is a logical grouping of services that provide data freshness, shared meta data management, governed data sources, and in-memory data. The underlying processes that power Data Services are the Backgrounder, Data Server and Data Engine processes.

Analytics Services, composed of the VizQL and Cache Server processes, provide user-facing visualization and analytics services and caching services.

Sharing and Collaboration, and Content Management Service are powered by the Application Server process. Core Tableau Server functionality such as user login, content management (projects, sites, permissioning, etc.) and administration activities are provided by the Application Server process.

All of the above services use and rely on the Repository process, which contains structured relational data like metadata, permissions, workbooks, data extracts, user info, and other data. The File Store process enables data extract file redundancy across the cluster and ensures extracts are locally available on all cluster nodes. Under heavier loads, extract files are available locally across the cluster for faster processing and rendering.

Tableau's architecture is flexible, allowing you to run the platform just about anywhere. You can install Tableau Server on-premises, in your private cloud or data center, on Amazon EC2, on Google Cloud Platform, or on MS Azure. Tableau analytics platform can also run atop virtualization platforms. We recommend you follow the best practices for each virtualization platform to ensure the best performance from Tableau Server.

Tableau and your data

When you install Tableau Server into your organization, it becomes a core component of the analytics pipeline to the data your users need. It's important to understand how Tableau Server interacts with your business data. Specifically, Tableau Server can store extracts of data in your organization. It can also connect to live data sources. How you choose to provide the data to your Tableau users is informed by a number of variables: data source type, user scenario, performance and access requirements, and infrastructure conditions.

Tableau Server has not been architected as a data warehouse server where static, native data files are housed. In fact, using Tableau Server as a traditional data warehouse is a poor use of your investment. Rather, when it comes to data storage, we recommend hosting optimized data extracts on Tableau Server. While a data extract is often a subset of a larger

data source in your organization, you can also create extracts for data sources that are over-taxed during work hours by [scheduling the extract refresh](#) for off-hours.

Extracts are also useful for modeling data or to enable highly-performant visualization authoring. For example, to improve visualization authoring and interaction performance you may optimize extracts by filtering the source data to the essential fields for a given department or project.

Tableau Server also provides direct, authorized access to live data sources, allowing users to build and run complex filtered queries against a variety of connected data sources. For this scenario, Tableau requires highly performant network access to the data sources in your organization and to those in the cloud. Tableau Server and the target data sources also need to be properly sized to handle the processing load required by high-volume, complex data operations.

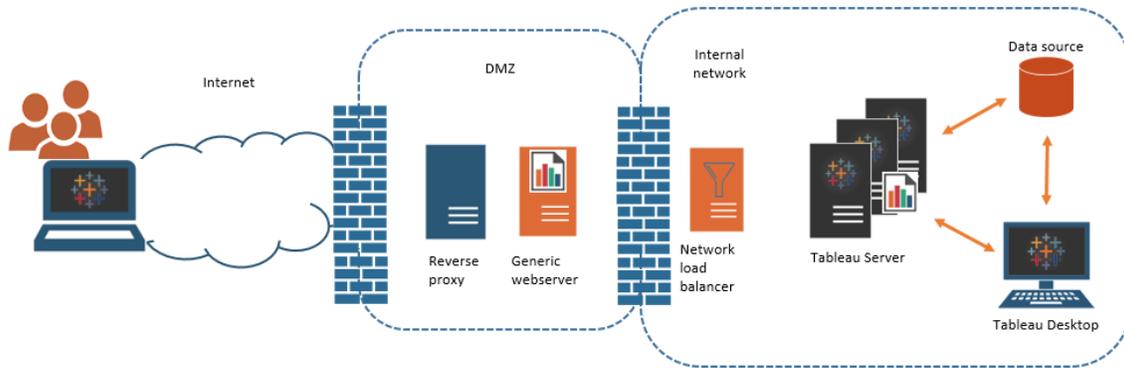
User access

Tableau Server is also a web-based collaboration platform, where users connect to share, view, and interact with data visualizations and data sources from a variety of devices. This means that Tableau Server must be accessible to Tableau users within your local protected network. You can also extend access to data visualizations to desktop, mobile, and authenticated web users outside your organization.

Tableau Server integrates with the following user [authentication](#) solutions: Active Directory, SAML, OpenId, and Kerberos.

Where should I install Tableau Server in my network?

Because of the highly-sensitive nature of most data that organizations manage with Tableau Server, and because Tableau Server requires access to internal data stores, Tableau Server must be run inside a protected network. Authenticated access from the internet is configured to connect to Tableau Server through a [reverse proxy](#) or a VPN solution.



Some organizations embed Tableau views in public webpages, or, for internal users, on generic web servers on the internal network.

Tableau Server can be configured to support such scenarios with either authenticated or anonymous access. For authorized access, where users can only view underlying data to which they have permission, you can configure trusted tickets with a generic web server. In this scenario, Tableau Server authorizes access to the underlying data in an embedded view. This scheme enables you to host interactive data visualizations on a web server in a DMZ or outside the protected network.

Anonymous access to embedded Tableau views requires that you enable "guest user" for Tableau Server. Guest user also requires that you license Tableau Server according to the number of cores you are running, rather than a named-user (interactor) model.

Sizing and scalability

Depending on the size and data usage in your organization, you can scale Tableau Server up or out. As you scale your server, you can also selectively allocate resources to meet your data needs and user needs.

When you scale up Tableau Server, you add hardware resources to a single server. For example, you might increase the memory and processing power of the computer running Tableau Server.

When you scale out Tableau Server, you add computers (or nodes). To create a highly available deployment with failover, you need at least three nodes. For example, you might run most CPU-intensive server processes on two nodes and use the third node for the gateway and coordination controller services.

Whether you scale up or scale out, you can selectively allocate resources by configuring the number and type of server processes that run. If your organization has a lot of data and creates a lot of data extracts, you can increase the number of processes that are dedicated to refreshing and storing extracts. Alternatively, if your organization wants to optimize for heavy user loads, you can increase the number of processes dedicated to responding to user requests. Additionally, you can integrate Tableau Server into industry-standard network load balancers to further optimize your server for user requests.

Tableau Server management model

Tableau Server has been designed to support a management scheme with two high-level administrators: server administrator and site administrator. In small organizations, these roles may be assumed by the same person or team, but in larger organizations, the roles often diverge.

In this model, server administrators are IT professionals who maintain and deploy heterogeneous server solutions. Essential areas for server administrators may include networking, hardware tuning and maintenance, security and access, and managing users and directory services. The tools and documentation that we deliver with Tableau Server for the server administrator support these core server IT areas.

Site administrator, on the other hand, is an administrative role specific to Tableau Server or Tableau Online deployments. The Tableau site administrator is fundamentally concerned with data content. The site administrator manages users and their access to projects, workbooks, and data sources. To learn about sites and how to plan your deployment for them, see [What is a site](#)

Administrative roles

In some small organizations a single administrator may manage the entirety of Tableau Server. But for bigger enterprise organizations, Tableau Server usually requires at least three administrative roles for management at scale.

Tableau Server administrator

The Tableau Server administrator has access to administrative pages for creating and editing sites, adding users and setting roles, and many content-related tasks after the Tableau Server installation is complete. The Tableau Server administrator also creates and manages other server and site administrators, who in turn may manage sites, user groups, and projects.

For information about signing into Tableau Server as a Tableau Server administrator, see [Sign in to the Tableau Server Admin Area](#).

TSM administrator

Tableau Services Manager (TSM) is a tool that gives server administrators command-line and web-based options for installing, upgrading, configuring, and maintaining Tableau Server. The TSM administrator installs the server and performs server-related administrative task like backing up server data, restoring backups, creating log archives, and managing multi-node clusters.

The TSM administrator must be an administrator on the local computer. See [Sign in to Tableau Services Manager Web UI](#).

Common tasks performed by the TSM administrator include:

- Initial configuration of Tableau Server after installation
- Ongoing configuration management, including editing settings and changing the server topology
- Running administrative tasks such as backup, restore, and ziplogs

To learn more about TSM, see [Tableau Services Manager Overview](#).

Tableau portal administrator

An important administrative role in a Tableau Server deployment is the Tableau customer portal administrator. The portal administrator manages licensing and the associated keys for the Tableau deployment. As the portal administrator, your first step is to purchase licenses on the [Tableau Customer Portal](#). When you purchase licenses, the portal will return corresponding product keys. To renew your license, visit the [Tableau renewal](#) web page.

Tableau has a number of products (e.g. Desktop, Server, Prep Builder, Add-ons, etc). Each of the Tableau products require that you activate licenses by updating the Tableau software with the product keys that are purchased and stored on the Tableau Customer Portal. As the administrator who is tasked with activating Tableau licenses, it important that you understand the relationship between licenses and keys. See [Understanding License Models and Product Keys](#).

Management tools

Tableau Server includes a number of toolsets for managing the system:

- **Tableau Server administrator page:** This is the web-based administrative site that is installed on each Tableau Server instance. Tasks performed on the administrator page are day-to-day tasks for both server and site administrators. Server-related tasks include creating sites and site administrator accounts, optionally importing users, setting up synchronization with directory services, setting up extract refresh schedules, monitoring server performance and usage, and other global settings.

Site-related tasks include managing content and assign permissions, running extract refreshes, create groups and projects, monitoring site activity, optionally adding users, and other content-related tasks.

See [Navigate the Admin Areas of the Tableau Web Environment](#).

Permissions required for the Tableau Server administrator page are based on site roles. The site roles are generated and managed by Tableau Server.

- **tsm Command Line Reference** - This is the primary interface for server-wide configurations. Many configurations made with TSM CLI are rarely revisited after initial configuration. For example: SSL, subscriptions, data caching, service account, SMTP alerting, user authentication, and single-sign on configuration are all performed with TSM CLI.
- You can also Sign in to Tableau Services Manager Web UI.
- **tabcmd**: You can use the tabcmd command-line utility on a Windows or Linux computer to create scripts to automate administrative tasks on your Tableau Server sites. For example, use tabcmd for creating or deleting users, projects, and groups.
- **REST API**: With the Tableau Server REST API you can manage and change Tableau Server resources programmatically, via HTTP. The API gives you simple access to the functionality behind the data sources, projects, workbooks, site users, and sites on a Tableau server. You can use this access to create your own custom applications or to script interactions with Tableau Server resources.

Security

As an application server connecting to data that may be highly-sensitive, Tableau Server supports and implements a number of industry security standards. Our server admin documentation includes best practices and implementation for user authentication, authorization, data security, and network security. While our default installation is secure by design, we also recommend following the [security hardening checklist](#) to further lock down your deployment.

For more information about security audit compliance, vulnerability reporting, and other security resources, visit <http://www.tableau.com/security>.

Tableau Services Manager Overview

This article provides an overview of Tableau Services Manager (TSM), which you can use to configure and administer Tableau Server. The TSM CLI was introduced with Tableau Server on Linux, version 10.5. Beginning with version 2018.2, the TSM Web UI is available.

- [Functionality](#)
- [Components](#)

- [Authentication](#)
- [Connecting](#)

Functionality

TSM gives server administrators command-line and web-based options for configuring and maintaining Tableau Server, including performing administrative tasks like backing up server data, restoring backups, creating log archives, and managing multi-node clusters. For example, you use TSM to perform the following tasks:

- Initial configuration of Tableau Server after installation
- Ongoing configuration management, including editing settings and changing the server topology
- Running administrative tasks such as backup, restore, and ziplogs

For administrators familiar with earlier versions of Tableau Server, TSM replaces the following tools from previous versions of Tableau Server:

- Tableau Server Configuration utility
- `tabadmin` command line utility
- Tableau Server Monitor

Components

TSM consists of *services* (called *processes* in this documentation) and *clients*. TSM processes are administrative services which manage Tableau Server processes. TSM processes run continuously after TSM is initialized, even when the rest of Tableau Server is offline.

TSM processes that run, even when Tableau Server is stopped include:

- Administration Agent
- Administration Controller
- Client File Service
- Coordination Service (based on Apache Zookeeper™)
- Service Manager
- Licensing Service

For more information about TSM processes and Tableau Server processes, see [Tableau Server Processes](#).

TSM Authentication

Whether you use the TSM Web UI, the command line interface, or the TSM API, you need to authenticate to Tableau Server before you can perform administrative tasks. This user account is distinct from Tableau Server user accounts, including Tableau Server administrators and site administrators.

TSM delegates authentication of users to the underlying operating system. On Linux, this means that authentication is handled using Pluggable Authentication Modules (PAM). PAM is the standard on all Linux distributions on which Tableau Server is supported. If your organization has configured PAM to authenticate with your directory service (Active Directory, LDAP), then you can authorize any user from that directory service to access TSM. In this scenario, any authenticated PAM user that is a member of the `tsmadmin` group is authorized to access TSM.

In the 2019.1 release, TSM authentication process uses PAM directly and then falls back to an authentication scheme using *substitute user* (`su`) if PAM fails or is not configured with a directory service. If PAM is not configured with a directory service then local accounts must be managed on the Linux computer. In these cases, TSM will use the `su` method of authentication: passing the user-provided credentials to run the `true` command in the `/bin` directory. If that command succeeds, then authentication is verified. Therefore, if the user is a member of the `tsmadmin` group, then the authenticated user is granted access to TSM.

Custom PAM service definition

TSM uses the standard PAM *login* service to authenticate. You can further customize TSM authentication behavior by creating a `tableau` PAM service file in `/etc/pam.d`. If this file exists, then it will be consulted instead of the PAM *login* service.

TSM authorization group

You authenticate to TSM with a user that exists on the Tableau Server computer. The TSM user account must use password-based authentication. By default, the TSM user account must be a member of the `tsmadmin` group on the computer where Tableau Server is running. Alternatively, you can specify a different authorization group for TSM administration. To specify a different default group during the install process, see Help Output for `initialize-tsm` Script. To specify a different authorization group after you have installed Tableau Server, see [Configure a Custom TSM Administration Group](#).

Connecting TSM clients

As a security measure, you can only connect to TSM with clients (CLI, Web UI, Rest API) over HTTPS. This is because TSM allows you to perform administrative tasks and to connect to TSM from other computers.

When you are connecting with a TSM client, you must connect to the Tableau Server instance running the TSM Administration Controller service.

Note: The TSM CLI tool does not require admin credentials in some scenarios. Specifically, if the account you are logged in as is a member of the TSM-authorized group, you do not need to provide credentials to run commands when running `tsm` CLI locally. For more information, see [Authenticating with tsm CLI](#).

TSM HTTPS connections rely on a self-signed certificate generated by the Tableau Server installer. This certificate is the Tableau installation CA certificate that signs the SSL certificates Tableau creates for encrypting traffic over HTTP. The Tableau installation CA certificate must be trusted by the systems connecting to TSM Administration Controller.

The TSM CLI client validates certificate trust from a different store than the TSM Web UI uses. The TSM CLI client refers to the trusted store in the local Java keystore to validate trust for CA certificates. Since the TSM Web UI must establish connection with a web browser, trust is validated with the operating system's trusted keystore. The difference in

how CA certificates are stored determines different trust configuration scenarios as outlined here:

- For TSM CLI communications on Tableau Server, the certificate trust is configured by default as part of the installation, node bootstrap, and upgrade processes. The Tableau installation CA certificate is added to the trusted store in the Java keystore. This allows you to access TSM using the CLI from any computer in the cluster without additional configuration. However, when accessing TSM Web UI, the browser will prompt you to trust the host running TSM Administration Controller service.
- For TSM CLI connections from remote computers, you will be prompted to trust the Tableau installation CA certificate the first time you connect to the Tableau Server running TSM Administration Controller. You can choose to trust the CA certificate, in which case you will not be prompted again on that computer until the certificate expires (default is 3 years). Or you can connect with a one-time trust by running your TSM command with the `--trust-admin-controller-cert` flag.
- For TSM Web UI connections from remote computers, the browser will prompt you to trust the host running TSM Administration Controller service.

Infrastructure Planning

Before you install Tableau Server, you should review the disk requirements, recommended configurations, user accounts, security, and networking requirements.

Before you install...

Note: You can find additional information about technical specifications for Tableau Server on the Tableau web site, [here](#).

This topic includes requirements and recommendations that you must consider before you install Tableau Server into a production environment.

If you want to install a single server, or if you want to do a minimal installation for test purposes, refer to our single-server installation guide, [Everybody's Install Guide](#).

If you are deploying Tableau Server in a distributed cluster, review Distributed Requirements in addition to the requirements and recommendations described in this topic.

If you are migrating from Tableau Server on Windows to Tableau Server on Linux, see [Migrate Tableau Server from Windows to Linux](#).

Hardware recommendations

The following list describes the minimum hardware recommendations for a single-node installation of Tableau Server:

<i>Install Type</i>	<i>Processor</i>	<i>CPU</i>	<i>RAM</i>	<i>Free Disk Space</i>
Single node	<ul style="list-style-type: none"> 64-bit Must support SSE4.2 and POPCNT instruction sets ARM-based processors are not supported 	8-core, 2.0 GHz or higher	32 GB	50 GB

If you are adding Tableau Prep Conductor to your Tableau Server installation, we recommend you add a second node and dedicate this to running Tableau Server Prep Conductor. This node should have a minimum of 4 cores (8 vCPUs), and 16 GB of RAM.

Multi-node and enterprise deployments	<p>Contact Tableau for technical guidance.</p> <p>Nodes must meet or exceed the minimum hardware recommendations, except:</p>
---------------------------------------	---

<i>Install Type</i>	<i>Processor</i>	<i>CPU</i>	<i>RAM</i>	<i>Free Disk Space</i>
---------------------	------------------	------------	------------	------------------------

- Nodes running backgrounder, where 4 cores may be acceptable.
- Dedicated node for Tableau Prep Conductor: Minimum of 4 cores (8 vCPUs), and 16 GB of RAM.

Important: The disk space requirement cannot be checked until you initialize TSM. If you don't have enough space, you won't be told this until after you install the Tableau Server package.

50 GB disk space available, with a minimum of 15 GB allocated to the `/opt` directory, and the remainder allocated to the `/var` directory for data storage.

- Free disk space is calculated after the Tableau Server Setup program is unzipped. The Setup program uses about 1 GB of space. You may need to allocate additional disk space depending on various factors like whether you will be using extracts.

The core Tableau Server bits must be installed in a directory with at least 15 GB of free disk space. If you attempt to install Tableau Server on a computer that does not have enough space, the Tableau Server package will install, but you will be unable to continue with setup. By default the install location is the `/opt` directory. You can change the installation path for Tableau Server on RHEL distros.

If you plan to make heavy use of extracts then you may need to allocate additional disk space. You can specify a different directory for data (extract) storage during installation.

- **Network attached storage space requirements for External File Store:** If you are planning to configure [Tableau Server with External File Store](#), you will need to estimate the amount of storage space to dedicate on your network attached storage.

Estimating the storage size: You must take into account the amount of storage needed for publishing and refreshing extracts. In addition, you must also take into account the

repository backup size unless you specifically choose the option to do your repository backup separately as described in the Option 2: Backup Repository Separately topic.

- Extracts:
 - Consider the number of extracts that will be published to Tableau Server and the size of each extract. Test your needs by publishing several extracts to Tableau Server, and then checking the disk space used. You can use this amount of disk space to help you figure out how many extracts will be published to Tableau Server over time as well as how each existing extract will increase in size.
 - Consider the space needed by the temp directory during an extract refresh. The temp directory, which is where an extract is stored to during a refresh, may require up to three times the final file size of the extract.
- Repository Backup:
 - To obtain an estimate of the repository data, check the size of `<data directory>/pgsql/data/base directory`.
 - To obtain the exact size of the repository data, open the backup file and use the size of the `workgroup.pg_dump` file.
- Core count is based on "physical" cores. Physical cores can represent actual server hardware or cores on a virtual machine (VM). Hyper-threading is ignored for the purposes of counting cores.
- RAM shown is the minimum recommended for a single-node installation. Your installation may function better with more RAM, depending on activity, number of users, and background jobs, for example.

To see the full list of recommendations and to see the minimum requirements, see Minimum Hardware Requirements and Recommendations for Tableau Server.

For public cloud deployments on Amazon Web Services and Google Cloud Platform, their "vCPU" is actually a CPU hyper-thread, and not a full CPU core. When sizing cloud instances, you will need twice as many vCPU as the Tableau Server CPU core require-

ments given (8 vCPU required for a minimum trial installation, 16 vCPU recommended for a single-node installation).

Operating system requirements

The following distributions of Linux are supported:

- Red Hat Enterprise Linux (RHEL) 7.3 and higher (not 8.x), and Amazon Linux 2
- CentOS 7.3 and higher (not 8.x)
- Oracle Linux 7.3 and higher (not 8.x)
- The latest versions of Ubuntu 16.04 LTS and 18.04 LTS (not 17.04)

Note: As of July 2022, Debian distributions are no longer supported. For more information, see [this Tableau Community post](#).

Additional notes on Linux distributions:

- Red Hat Enterprise Linux (RHEL), CentOS, Oracle Linux, and Amazon Linux distributions are collectively referred to in this documentation as RHEL-like.
- RHEL 8 is not supported.
- As of July 2022, Debian distributions are no longer supported. For more information, see [this Tableau Community post](#).
- Non-LTS releases of Ubuntu are not supported.
- Ubuntu version 17.04 is not supported.
- Previous versions of CentOS and Ubuntu are not supported because Tableau Server requires `systemd` for process management.
- The version of the installer with the file suffix, `.deb`, installs on both Ubuntu and

Debian distributions.

- Custom kernels are not supported.

In a multi-node installation of Tableau Server, all of the computer nodes where you are installing Tableau must run Linux and the same distribution of Linux.

Installation directory

The core Tableau Server bits are installed in the `/opt` directory by default.

- The directory where you install Tableau Server must have at least 15 GB of free disk space allocated to it. If you attempt to install Tableau Server on a computer that does not have enough space, the Tableau Server package will install, but you will be unable to continue with setup.
- You can specify a non-default install location on RHEL-like systems, but cannot change the location on Ubuntu.
- Do not specify a symbolic link or a directory location on a Network File System (NFS) volume when specifying a non-default install location on RHEL-like systems.

Data directory

By default, Tableau Server will create a data directory for all content and extracts that are managed by Tableau. The directory is created at `/var/opt/tableau/tableau_server`.

You can specify a different directory for data (extract) storage during installation. If you plan to use a different directory, do not create the directory. Instead, let Tableau Server setup create the directory. The data directory requires specific permissions that are set during the installation process.

To change the data directory, you must pass a flag along with the data directory path when you run the `initialize-tsm` script. See Help Output for `initialize-tsm` Script.

If you are changing the default data directory:

- Do not specify a symbolic link or a data directory location on a Network File System (NFS) volume.

Tableau Server on Linux Administrator Guide

- Do not specify a data directory location with a path that includes a period or space. If there is a period or space in the path, initialization will fail.
- The data directory must be installed into a different directory than the installation directory.

Important: You cannot change the data directory location after you've run `initialize-tsm`. The data directory location will persist for the life of the deployment, including subsequent upgrades.

Tableau Prep Conductor

Tableau Prep Conductor is one of the process on Tableau Server. It runs a flow, checks connection credentials, and sends alerts if a flow fails. Tableau Prep Conductor leverages the scheduling and tracking functionality of Tableau Server so you can automate running flows to update the flow output instead of logging into Tableau Prep Builder to manually run individual flows as your data changes.

Tableau Prep Conductor is licensed separately and is available through the Data Management Add-on license. For more information on Tableau Prep Conductor licensing, see [License the Data Management Add-on](#).

We recommend you enable Tableau Prep Conductor on a dedicated node. For more information:

- If you are installing a new Tableau Server, see [Step 1 \(New Install\): Install Tableau Server with Tableau Prep Conductor](#).
- If you are adding Tableau Prep Conductor to an existing installation of Tableau Server, see [Step 1 \(Existing Install\): Enable Tableau Prep Conductor](#).

Additional requirements

Make sure that your environment also meets the following additional requirements:

Hostname

- Tableau Server must be able to resolve the hostname to an IP address either using the domain name server (DNS) or with a local host file on the computer running Tableau Server. By default, host files are stored at `/etc/hosts`.
- The hostname of the server must not change after you start Tableau Services Manager during the setup process. For example, this might happen if you use the cloud-init package to initialize a virtual machine, and you install Tableau Server on that virtual machine.
- Hostnames that include underscores (`_`) are not supported by Tableau Server.

Static IP address

The computer where you install Tableau Server must have a static IPv4 or IPv6 address.

Database drivers

To connect to specific data sources, the computer where you install Tableau Server must have the correct database drivers installed. For more information, see [Database Drivers](#).

Available ports

TSM and Tableau Server each require an available TCP port in order for you to access them. TSM defaults to port 8850, and the Tableau Server Gateway service defaults to port 80. We strongly recommend that you ensure that both port 8850 and 80 are not in use on your system before installing Tableau Server. If those ports are not available, the TSM and gateway ports may be dynamically remapped to different port numbers, and there is currently no interface for displaying which port they have been remapped to.

See [Tableau Services Manager Ports](#).

Local firewall configuration

If you are running a firewall on the computer where you will be installing Tableau Server, then you will need to open the following default ports for Tableau Server traffic. All port numbers, except 443 can be changed.

Port	TCP/UDP	Used by ...	TYPE OF INSTALLATION	
			All	Distributed / High Availability
80	TCP	Gateway	X	
443	TCP	SSL. When Tableau Server is configured for SSL, the application server redirects requests to this port. Do not change this port.	X	
8850	TCP	Tableau Services Manager.	X	
8060	TCP	PostgreSQL database.	X	
8061	TCP	PostgreSQL backup verification port	X	
8000-9000	TCP	Range of ports reserved by default for dynamic mapping of Tableau processes		X
27000-27009	TCP	Range of ports used by Tableau Server for License service. This range must be open on the node running the License service and accessible from other nodes. By default, the initial node runs the License service.	X	

See Tableau Services Manager Ports and Configure Local Firewall.

System user and groups

Tableau Server on Linux uses one unprivileged user, and two groups for proper operation. Tableau will create the default account and groups during setup. Alternatively, you can specify existing accounts. See System user and groups and TSM authorization group.

Sudo and root access

All installation tasks and administrative tasks for Tableau Server must be run as root. Often this is accomplished using the `sudo` command, but running the commands directly as the root user is also possible.

To install Tableau Server with the root account, you must specify a user account during installation. The account will be used for managing TSM. Specify the account by running the `initialize-tsm` script with the `-a` option. See Help Output for `initialize-tsm` Script.

Account password

The user account that you use to install and administer Tableau Server must be able to authenticate with a password. That is, the user must not use another means of authenticating (for example public key authentication).

If the account you are using to install and initialize Tableau Server does not have a password, you can set one using the `passwd` command:

```
sudo passwd $USER
```

Port access requirements

If you want to install Tableau Server remotely, for example by means of SSH, ensure that the following ports are open:

- 8850. The port used for the Tableau Services Manager (TSM) web interface. You can use this interface to configure Tableau Server.
- 80. The port used for the main Tableau Server web interface.

The Tableau Server installer attempts to open these ports during the installation process, but it can only open these ports for the `firewalld` firewall. If your computer runs another firewall, you must open the ports before you install.

Virtual Container environments

Tableau Server on Linux Administrator Guide

Beginning with version 2021.2, certain configurations of Tableau Server on Linux can be run in a container. For details on supported configurations, see [Tableau Server in a Container](#).

Previous versions of Tableau Server on Linux and unsupported configurations have not been tested and are not supported in virtual container environments such as Docker. In these cases, Tableau Server on Linux will not function as expected if installed in these environments.

Package requirements

Systemd

Tableau Server requires `systemd` to manage services. This package is installed by default on CentOS 7 and Ubuntu 16. If you decide to test Tableau Server on a modified version of these distributions, you can run the following command to confirm that `systemd` is installed:

```
whereis systemd
```

If `systemd` is installed, the installation location is displayed. For example, you might see the following output:

```
systemd: /usr/lib/systemd /etc/systemd /usr/share/systemd /usr/share/man/man1/systemd.1.gz
```

If you have `systemd` installed but the Tableau installer is failing requirements checks for `systemd`, it's likely that `systemd` is not running. To verify that `systemd` is running, run the following command:

```
ls /run/systemd
```

The output will be a list of files and directories. If `systemd` is running, the output will include `system`. If `system` is not in the output, then `systemd` is not running.

Continue to the next step: [Install and Configure Tableau Server](#).

Disk Space Requirements

In general, when estimating the amount of additional disk space to allocate for Tableau Server for day-to-day usage, you must consider whether or not extracts will be published to Tableau Server, and consider the number of workbooks that you expect to publish to Tableau Server. If you anticipate using extracts, Tableau recommends that you begin with a few hundred gigabytes (GB). If you do not anticipate using extracts, you may only need around 50 GB to fulfill your usage needs. To setup drive space alerts, see [Configure Server Event Notification](#).

Looking for Tableau Server on Windows? See [Disk Space Requirements](#).

It is critical for Tableau Server to have adequate disk space. If you run out of disk space on any node in a Tableau Server installation, you can experience erratic performance, including not being able to access Tableau Server or the TSM Web UI. For troubleshooting steps, see the Tableau Knowledgebase.

Here are the factors that affect disk space requirements and where you might choose to install Tableau Server:

Publishing extracts to Tableau Server

Consider the number of extracts that will be published to Tableau Server and the size of each extract. Test your needs by publishing several extracts to Tableau Server, and then checking the disk space used. You can use this amount of disk space to help you figure out how many extracts will be published to Tableau Server over time as well as how each existing extract will increase in size.

Refreshing extracts

Consider the space needed by the temp directory during an extract refresh. The temp directory, which is where an extract is stored to during a refresh, may require up to three times the final file size of the extract.

While there may be some cases where .hyper extracts become bigger after an extract upgrade, there are other cases when .hyper extracts become smaller after an extract upgrade. In general, the overall disk size requirements should be the same for .hyper extracts as it was for .tde extracts.

Creating many workbooks

If using workbooks, consider the number of workbooks that will be published to Tableau Server. Individual workbooks tend to take up a small amount of disk space. However, if you anticipate thousands of workbooks being published, you may want to allocate additional disk space to accommodate those workbooks.

Logging

To assist with daily management and troubleshooting, Tableau Server creates log files as a part of its normal operations. Depending on the level at which the logging is configured, it can significantly impact the amount of disk space necessary on the Tableau Server computer.

Backup and restore processes

The free disk space required to create a backup varies depending on the amount of data in the Tableau Server repository and file store services, and their collocation with the tabadmincontroller service. During backups, the background tasks for cleaning up old extracts are temporarily paused. This means that, for the duration of the backup, extract refreshes will leave extra files in place, adding to disk space usage. If your backup takes a long time, or if your organization uses many extracts that are regularly updated, this can result in a significant amount of temporary disk space usage. These temporary files will be removed after the backup is complete.

The following table lists the disk space requirements for backup based on whether the node hosts the repository, file store, controller, or some combination of them.

Repository	File Store	Controller	Disk Space Required
------------	------------	------------	---------------------

✓			<p>3x repository data + 250 MB</p> <p>To obtain an estimate of the repository data, check the size of <code><data directory>/pgsql/data/base</code> directory.</p> <p>To obtain the exact size of the repository data, open the backup file and use the size of the <code>workgroup.pg_dump</code> file.</p>
	✓		<p>1.5x file store data</p> <p>To obtain an estimate of file store data (extracts, flows, etc.), check the size of <code><data directory>/dataengine</code> directory.</p>
		✓	<p>3x repository data + 250 MB + 2.5x file store data</p>
✓	✓		<p>3x repository data + 250 MB + 1.5x file store data</p>
	✓	✓	<p>3x repository data + 250 MB + 1.5x file store data</p>
✓		✓	<p>3x repository data + 250 MB + 2.5x file store data</p>
✓	✓	✓	<p>3x repository data + 250 MB + 1.5x file store data</p>

Restore disk space requirements

You must have adequate disk space for the database restore process to run successfully.

To restore Tableau Server:

- On controller nodes, you need free space equal to at least the size of the backup archive.
- On repository nodes, you need free space equal to at least three times the size of the repository data in the backup archive, plus 250 MB, plus the size of the pgsql data directory.
- On file store nodes, you need free space equal to at least twice the size of the dataengine folder in the backup archive.

Recommended Baseline Configurations

Determining the topology (number of nodes, number of Tableau Server processes) of your Tableau Server deployment requires you to consider these variables: your environment, sources of data and management to provide self-service data access, workload, and usage. However you may not have enough information about these variables when you deploy Tableau Server for the first time. This topic describes three baseline architectures that can be used as starting points for your Tableau Server installations.

Hardware Recommendations

The hardware recommendations seen below are based on the hardware that the Tableau team uses to test Tableau Server scalability. We suggest that you use these recommendations as starting points for your production deployments. For proof-of-concept deployments, see Minimum Hardware Requirements and Recommendations for Tableau Server.

<i>Install Type</i>	<i>Processor</i>	<i>CPU</i>	<i>RAM</i>	<i>Free Disk Space</i>
Single Server	64-bit (x64) ARM-based processors are not supported.	8 physical cores (16vCPUs), 2.0 GHz or higher	64 GB (8GB/physical Core)	500 GB - 1 TB

<i>Install Type</i>	<i>Processor</i>	<i>CPU</i>	<i>RAM</i>	<i>Free Disk Space</i>
----------------------------	-------------------------	-------------------	-------------------	-------------------------------

If you are adding Tableau Prep Conductor to your Tableau Server installation, it is recommended to add a second node and dedicate this to running Tableau Server Prep Conductor. This node should have a minimum of 4 physical cores (8 vCPUs), and 16 GB of RAM.

Multi-node
and enter-
prise deploy-
ments

Nodes must meet or exceed the minimum hardware recommendations, except in the following scenarios where a node can be configured with 4 physical cores (8 vCPUs):

- Dedicated node for Backgrounder.
- Dedicated node for Tableau Prep Conductor.
- Move all the licensed process from the initial node to additional nodes.

Note: For deployments using virtual machines, Tableau recommends dedicated CPU affinity. If you are running Tableau Server in a virtual environment, use your VM host's best practices for vCPU allocation in relation to the number of physical CPU cores on the VM host. Typically 2 vCPUs = 1 physical core for Tableau Server. For example, for AWS installations, the 4 core minimum recommendation is equivalent of 8 AWS vCPUs. Similarly, follow the best practices provided by your virtual infrastructure provider to make sure Tableau Server has access to the appropriate compute, memory, and data resources. If you are installing Tableau Server in a virtual environment or a cloud-based deployment, see *Virtual Machines and Public Cloud Deployments* section later in this topic.

Estimating Disk Space

There are several factors that affect disk space requirements, including whether or not you will be publishing extracts, flows, and the number of workbooks to Tableau Server. For more information see *Disk Space Requirements*.

Baseline Configurations

- Single Server Installations
- Two Node Installation - Specialized for extract heavy environments
- Two Node Installation - Specialized for flow environments
- High Availability Installations (HA)

Single Server Installations

Recommendations

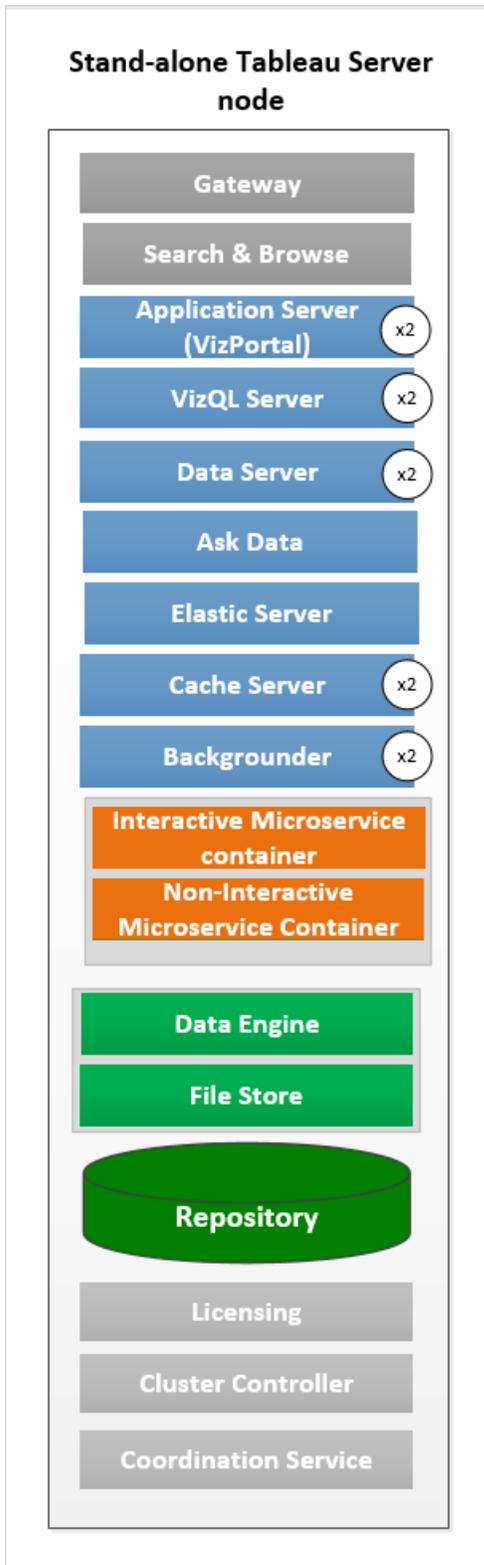
We recommend using a single machine to install your Tableau Server for initial deployments with limited usage and are not mission critical. Single Server installations can also be expanded to multi-node installation as your workloads grow.

Here are some instances when a single server installation may not be right for you:

- If your system is considered mission critical and needs to be highly available. High availability is about minimizing the system downtime. It is achieved by eliminating single points of failure, and having a reliable failover mechanism. Tableau Server requires a minimum of a three-node configuration to provide redundancy and eliminate the single points of failure. This is one of the primary reasons to move to a multi-node configuration.
- If you have a lot of active users and a lot of extract refreshes, the two types of loads may be competing for the same resources on the machine. In such a scenario, a single server configuration may not be the right option as you may need additional specialized nodes to isolate the difference workloads.

Note: Active users represent the interactive, concurrent requests made to Tableau Server, including consuming dashboards on a laptop or mobile device, web authoring, and connecting to and querying Published Data Sources.

Server Configuration



- Stand-alone single server node with all the processes installed on one machine.
- Below are the number of processes for an 8 core machine:
 - VizQL Server: Set to 2 instances (Number of physical cores divided by 4, up to a maximum of 4).
 - Backgrounder, Cache Server, and Data Server: Set to 2 instances.
 - All other processes, only one instance of the process is installed, regardless of hardware.

Note: One instance of Tableau Prep Conductor is automatically configured with Backgrounder, when you have the Data Management Product Key activated on your server. However, it is recommended that you have a dedicated node for Tableau Prep Conductor. If you plan to have flows on your Tableau Server, it is recommended that you use two or more nodes and dedicated one of these nodes to run only flows. The example configuration described above does not include Tableau Prep Conductor since it is a single node server.

Multi-Node Installations

Running Tableau Server on more than one machine is called a multi-node installation, or a cluster. There are various reasons why you might want to have a multi-node installation. For example, you may have heavy extract environments which can mean dedicating some hardware resources to Backgrounder process. For systems that have high availability requirements, you need a multi-node environment that has at least three nodes.

Two Node Installation - Specialized for extract heavy environments

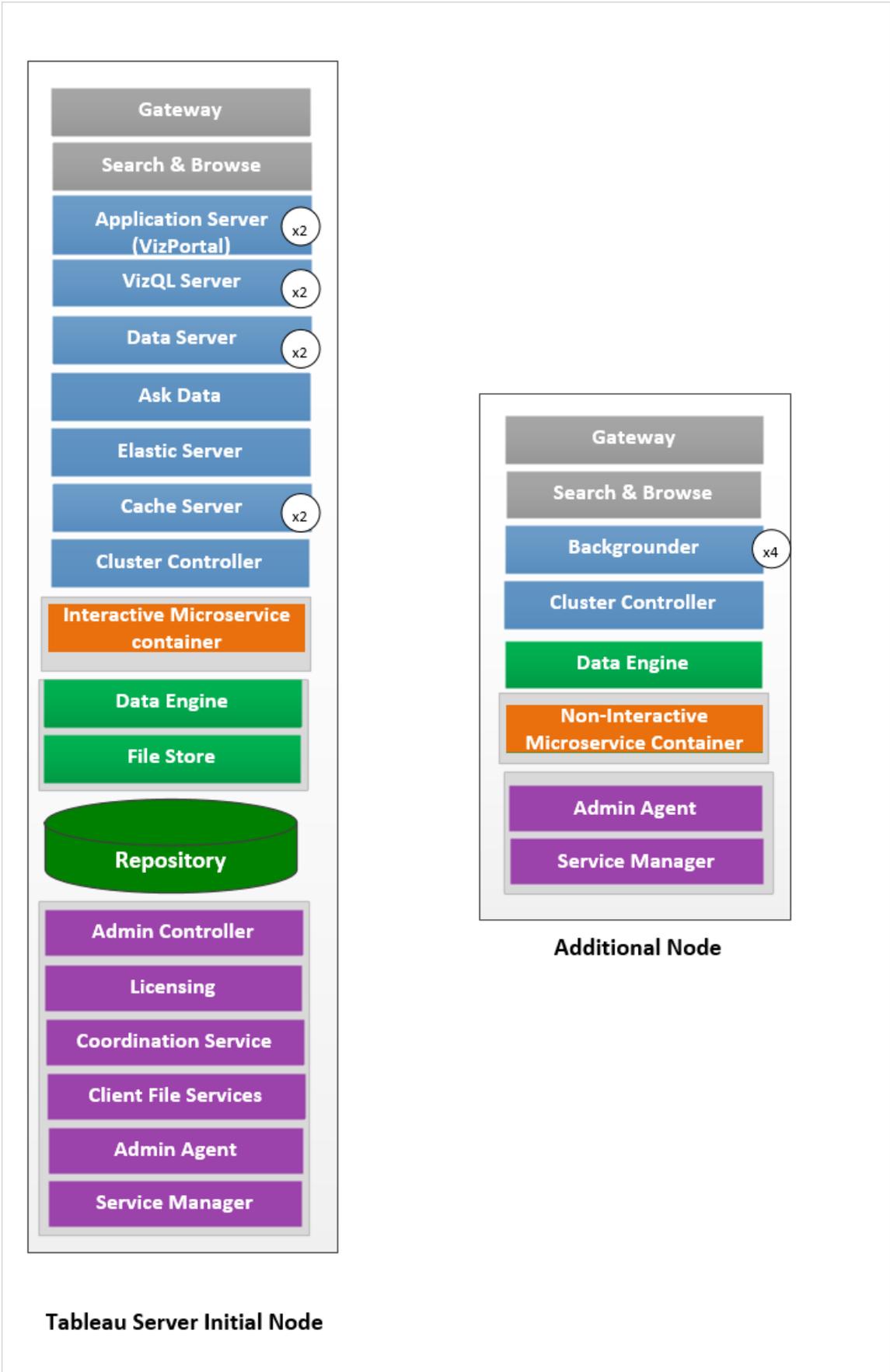
Recommendations

Start with a two node configuration when the following conditions apply to you:

- **Extract heavy environment:** Majority of your data sources are extracts. Having just a few, extremely large extracts could put your deployment in this category, as would having very many small extracts.
- **Frequent extract refreshes:** Refreshing an extract is a CPU-intensive task. Deployments where extracts are frequently refreshed (for example, several times a day during business hours) are often helped by more emphasis on the background process, which handles refresh tasks.

Important: Two-node configurations do not meet the minimum requirements for high availability. If you need a system that is highly available, see High Availability Installations (HA).

Server Configuration



- On the initial node, install all the processes except for the backgrounder. Below is the number of instances of the processes for an 8 core machine:
 - VizQL Server: Set to 2 instances. (default calculation: Number of physical cores divided by 4, up to a maximum of 4).
 - Cache Server, and Data Server: Set to 2 instances. One instance of Ask Data is automatically configured on the node that has Data Server.
 - Elastic Server: Elastic Server memory is configured to 1GB by default and can be configured to improve performance by using the `elasticserver.vmopts` TSM configuration option. For more information, see `tsm` configuration set Options.
 - All other processes, only one instance of the process is installed, regardless of hardware. One instance of Interactive Microservice Container is installed on a node that has Application Server enabled, and one instance of Non-Interactive Microservice Container is installed on a node that has Backgrounder enabled.
- Isolate backgrounder on the additional node. To calculate the minimum number of backgrounder processes to run on this node, divide the computer's total number of physical cores by 4. To calculate the maximum number, divide the computer's total physical cores by 2. In the example shown above, both the nodes are on machines with 8 physical cores. When you install the backgrounder, Tableau Server automatically installs one instance of the Data Engine.

Note: This configuration assumes that you do not have Tableau Prep Conductor enabled on your Tableau Server. If are using Tableau Prep Conductor to schedule and manage flows, and have an extract heavy environment, we recommend that you have at least 3 nodes and use the 3 node configuration described later in this topic.

As you monitor and gather data about the performance and usage, you can fine tune and configure the number of instances for these processes. For example, on the node where it

Tableau Server on Linux Administrator Guide

is dedicated to running backgrounder, initially, you can set the number of backgrounders to minimum (total number of cores divided by four), and increase the number of backgrounder processes later if you find that:

- Extract refreshes are taking a long time to complete
- Subscriptions and alerts are not completing on time

For more information on performance tuning, see [Performance Tuning](#) topic.

Two Node Installation - Specialized for flow environments

Start with a two node configuration if you are planning to publish, schedule, and manage flows on your Tableau Server.

Important: Two-node configurations do not meet the minimum requirements for high availability. If you need a system that is highly available, see [High Availability Installations \(HA\)](#).

Server Configuration

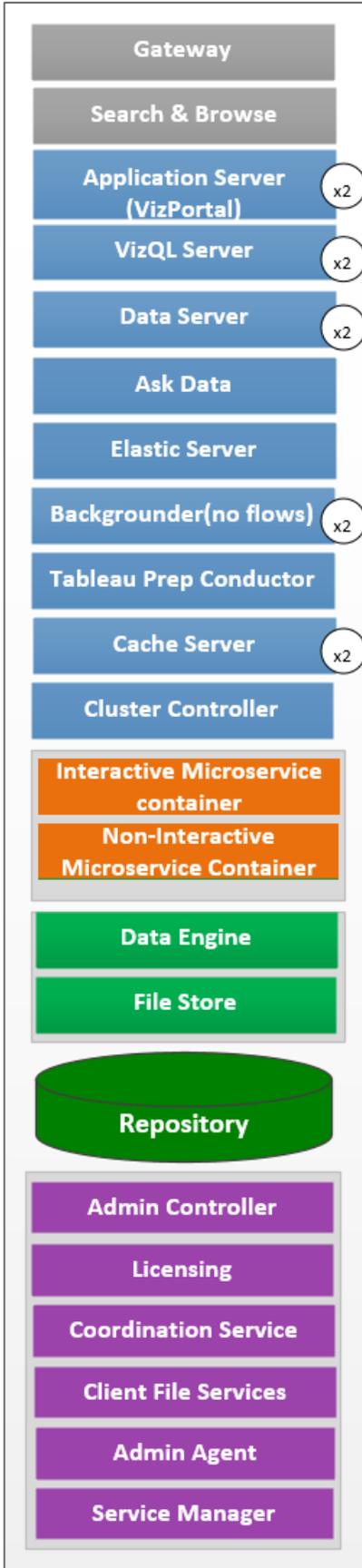
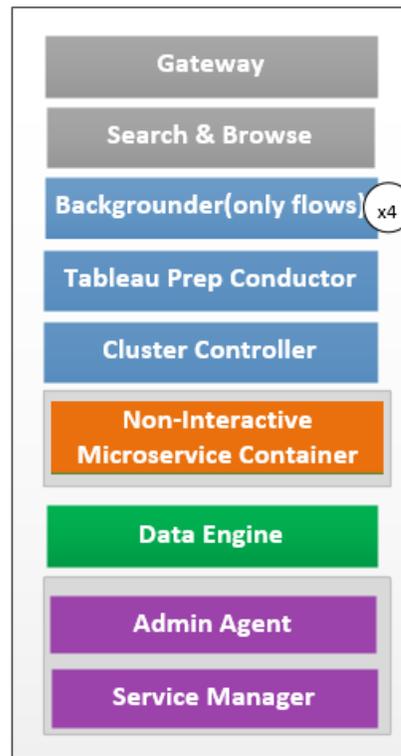


Tableau Server Initial Node



Additional Node

- On the initial node, install all the processes. Below is the number of instances of the processes for an 8 core machine:
 - VizQL Server: Set to 2 instances. (default calculation: Number of physical cores divided by 4, up to a maximum of 4).
 - Cache Server, and Data Server: Set to 2 instances. One instance of Ask Data is automatically configured on the node that has Data Server.
 - Backgrounder: Minimum 2, maximum 4. The diagram above shows the maximum for an 8 core node. Tableau Prep Conductor is automatically configured on the node where you have backgrounder installed. On the initial node, set the Backgrounder node role to run all job types including flows using the `tsm topology set-node-role` tsm configuration. For more information, see `tsm topology set-node-role`
 - Elastic Server: Elastic Server memory is configured to 1GB by default and can be configured to improve performance by using the `elasticserver.vmopts` TSM configuration option. For more information, see `tsm configuration set Options`.
 - All other processes, only one instance of the process is installed, regardless of hardware. One instance of Interactive Microservice Container is installed on a node that has Application Server enabled, and one instance of Non-Interactive Microservice Container is installed on a node that has Backgrounder enabled.
- Isolated the backgrounder on the additional node to run only flows. Use the `tsm topology set-node-role` tsm configuration to configure this setting. For more information, see `tsm topology set-node-role`

Note: If you have both a heavy extract environment, and schedule and manage flows on your server, we recommend that you use the 3 node configuration described below.

Tableau Server on Linux Administrator Guide

High Availability Installations (HA)

Recommendations

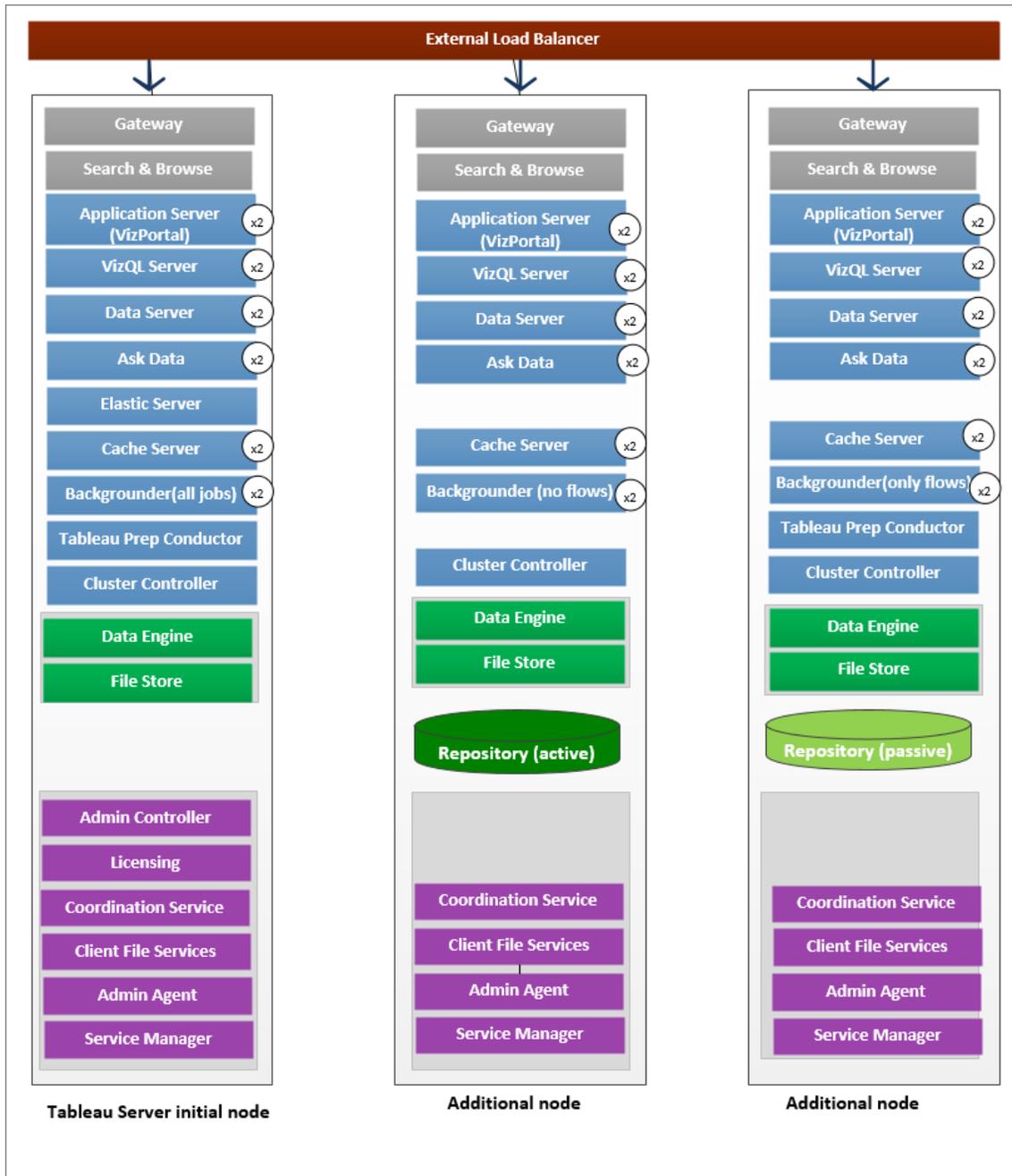
A highly available installation of Tableau Server is a distributed installation that is designed to maximize the availability of Tableau Server. High availability basically means that the system is available with minimal amount of downtime. To build in redundancy for HA related items such as repository, file redundancy, and failover, you need a **minimum of three nodes**. The tolerance for downtime will vary for each organization and depends on the SLAs you have established in your organization.

High availability is achieved by eliminating single points of failure and detecting failures and setting up a reliable failover system. HA in Tableau Server is mainly achieved by:

- File redundancy with multiple File Store/Data Engine instances.
- Active/Passive Repository across two nodes.
- Adding an external load balancer to make sure your installation is robust to Gateway failures and make sure that requests only get routed to functioning Gateway processes.

Server Configuration

Three-node configuration:



- To build in redundancy, you need to add additional nodes to host instances of the repository and File Store/Data Engine processes. You can add instances of other processes, including multiple instances of a process on a node.
- To build redundancy for the type of backgrounder jobs, have one of the nodes (initial

node in this example) run all type of jobs. Backgrounders run all types of jobs by default. On one of the additional nodes, set the backgrounder to run only flows, and the other additional node to run all jobs except for flows.

- The successful functioning of Tableau Server depends on a properly functioning Coordination Service. For server installations of three or more nodes, we recommend that you add additional instances of the Coordination Service by deploying a new Coordination Service ensemble. This provides redundancy and improved availability in the event that one instance of the Coordination Service has problems. For more information, see [Deploy a Coordination Service Ensemble](#) .
- Elastic Server memory is configured to 1GB by default and can be configured to improve performance by using the `elasticsearch.vmopts` TSM configuration option. For more information, see [tsm configuration set Options](#).
- To reduce the system's vulnerability, you can run multiple gateways and additional instances of some of the server processes. The fewest number of computers required to achieve this configuration is three.
- The repository has also been moved from the initial node to one of the additional nodes, and a second, passive instance has been added to the other new node.
- One instance of Interactive Microservice Container is installed on a node that has Application Server enabled, and one instance of Non-Interactive Microservice Container is installed on a node that has Backgrounder enabled.

NOTE: In certain circumstances you may want to limit the processes running on your initial node. Reasons for doing this include wanting to run as few processes as possible on the node to limit processing requests on the node. You might also remove licensed Tableau Server processes from the node if you have a core-based license and do not want the initial node cores to count against your core use. For more information on Tableau Server licensed processes, see [Tableau Server Processes from the node](#).

Virtual Machines and Public Cloud Deployments

In general, the considerations and recommendations described in this topic apply to virtual environment and cloud deployments.

If you are running Tableau Server in a virtual environment, use your VM host's best practices for vCPU allocation in relation to the number of physical CPU cores on the VM host. Typically 2 vCPUs = 1 physical core for Tableau Server. For example, for AWS installations, the 4 core minimum recommendation is equivalent of 8 AWS vCPUs.

For more information on cloud-based deployments, see:

- [Install Tableau Server in the Cloud.](#)

Beyond Baseline Configurations

If you are planning a system whose configuration is beyond the limits documented here, contact [Tableau Professional Services](#).

Disaster Recovery Considerations

While HA configurations reduce downtime, you may still encounter failures in case of a disaster or hardware failures. In addition to the above considerations, you should evaluate the importance of disaster recovery in your organization and plan for a deployment that helps you meet your disaster recovery goals and objectives.

When planning for disaster recovery (DR) in your Tableau environment, there are two main factors to consider:

- Recovery Time Objective (RTO), a measure of how much downtime your business can accept before a full recovery.
 - Influences how often you restore your backups to an alternative cluster and the amount of infrastructure investment.
- Recovery Point Objective (RPO), a measure of how much data loss your business can tolerate.

Tableau Server on Linux Administrator Guide

- Influences how often you will need to take backups of your system.
- For Tableau Server the RPO cannot be shorter than the time it takes to complete a full backup of your server.

The diagram below illustrates how to plan for a range of RTO requirements:

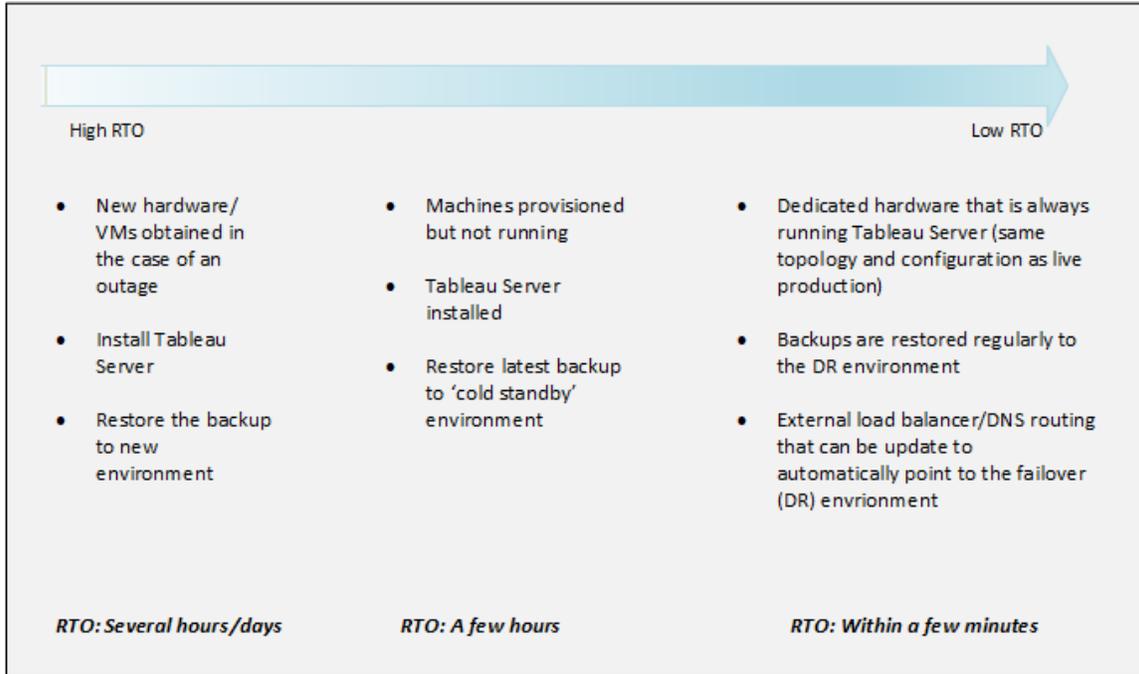
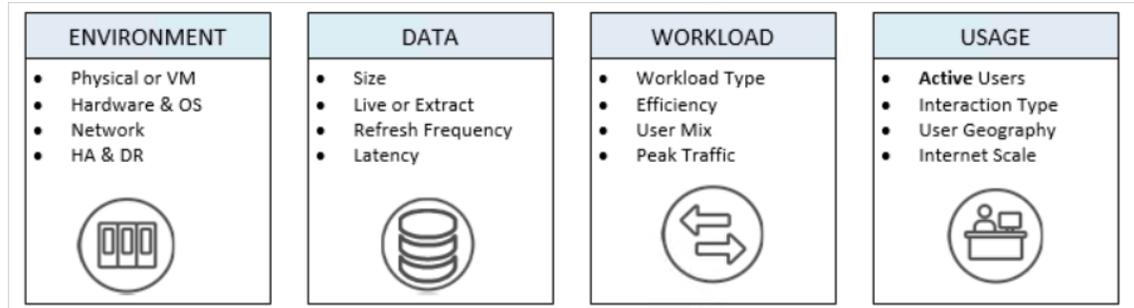


Tableau Server Scalability

These baseline configurations may not be enough as your needs change and grow, and you might need to scale your Tableau Server beyond these configurations. Like other enterprise platforms, Tableau Server scales up by adding processor, memory, and disk to existing node, and scales out by adding more nodes to a cluster. However, scalability and performance are heavily dependent on external systems and user activity. The configuration of your Tableau Server can be different depending on your requirements and variables:



For more information on Tableau Server scalability and the variables affecting scalability, see [Tableau Server Scalability whitepaper](#).

Identity Store

Tableau Server requires an identity store to manage user and group information. There are two kinds of identity stores: local and external. When you install Tableau Server you must configure either a local identity store or an external identity store.

For information about configuration options for the identity store, see [identityStore Entity and External Identity Store Configuration Reference](#).

Local identity store

When you configure Tableau Server with a local identity store, all user and group information is stored and managed in the Tableau Server repository. In the local identity store scenario, there is no external source for users and groups.

External identity store

When you configure Tableau Server with an external store, all user and group information is stored and managed by an external directory service. Tableau Server must synchronize with the external identity store so that local copies of the users and groups exist in the Tableau Server repository, but the external identity store is the authoritative source for all user and group data.

If you have configured the Tableau Server identity store to communicate with an external LDAP directory, then all users (including the initial admin account) that you add to Tableau Server must have an account in the directory.

When Tableau Server is configured to use an external LDAP directory, you must first import user identities from the external directory into the Tableau Server repository as system users. When users sign in to Tableau Server, their credentials are passed to the external directory, which is responsible for authenticating the user; Tableau Server does not perform this authentication. However, the Tableau user names stored in the identity store are associated with rights and permissions for Tableau Server. Therefore, after authentication is verified, Tableau Server manages user access (authorization) for Tableau resources.

Active Directory is an example of an external user store. Tableau Server is optimized to interface with Active Directory. For example, when you install Tableau Server on an Active Directory domain-joined computer using the Configure Initial Node Settings, Setup will detect and configure most Active Directory settings. If, on the other hand, you are using TSM CLI to install Tableau Server, you must specify all the Active Directory settings. In this case, be sure to use the LDAP - Active Directory template to configure identity store.

If you are installing into Active Directory, we recommend that you review User Management in Deployments with External Identity Stores before you deploy.

For all other external stores, Tableau Server supports LDAP as a generic way to communicate the identity store. For example, OpenLDAP is one of several LDAP server implementations with a flexible schema. Tableau Server can be configured to query the OpenLDAP server. To do so, the directory administrator must provide information about the schema. During setup, you must use Configure Initial Node Settings to configure a connection to other LDAP directories.

LDAP bind

Clients that wish to query a user store using LDAP must authenticate and establish a session. This is done by binding. There are multiple ways to bind. Simple binding is authenticating with a username and password. For organizations that connect to Tableau Server

with simple bind, we recommend configuring an SSL encrypted connection, otherwise the credentials are sent over the wire in plaintext. Another type of binding Tableau Server supports is GSSAPI binding. GSSAPI uses Kerberos to authenticate. In Tableau Server's case, Tableau Server is the client and the external user store is the LDAP server.

LDAP with GSSAPI (Kerberos) bind

We recommend binding to LDAP directory with GSSAPI using a keytab file to authenticate to the LDAP server. You will need a keytab file specifically for the Tableau Server service. We also recommend encrypting the channel with the LDAP server using SSL/TLS. See [Configure Encrypted Channel to LDAP External Identity Store](#).

If you are installing into Active Directory, and the computer where you are installing Tableau Server is already joined to the domain, then the computer may already have a configuration file and a keytab file. In this case, the Kerberos files are for the operating system functionality and authentication. Strictly speaking, you can use these files for GSSAPI bind, but we don't recommend using them. Instead, contact your Active Directory administrator and request a keytab specifically for the Tableau Server service. See [Understanding Keytab Requirements](#).

Assuming your operating system has a properly configured keytab for authentication to the domain, then the Kerberos keyfile for GSSAPI bind is all you need for the base installation of Tableau Server. If you plan to use Kerberos authentication for users, then [configure Kerberos for user authentication](#) and [Kerberos delegation to data sources](#) after installation is complete.

LDAP over SSL

By default, LDAP with simple bind to arbitrary LDAP servers is not encrypted. User credentials that are used to establish the bind session with the LDAP server are communicated in plaintext between Tableau Server and the LDAP server. We strongly recommend that you encrypt the channel between Tableau Server and the LDAP server.

Beginning with version 2021.2, Tableau Server on Linux requires an encrypted LDAP channel when you use Active Directory as an identity store. You must install a valid SSL/TSL certificate before installing or upgrading to 2021.2 or newer. Although not recommended, you can also disable the default encrypted LDAP channel. For more information about enabling or disabling encryption for Active Directory and other LDAP servers, see [Configure Encrypted Channel to LDAP External Identity Store](#).

System user and groups

Tableau Server on Linux uses one user, and two groups for proper operation. The user and groups can be local or from an LDAP directory service.

User

Tableau Server requires a service account. This account is an unprivileged user with normal login privileges. By default, Tableau Server installation will create a local user, `tableau`, for the service account.

If you want to use an existing user account for the Tableau Server service account then you must disable account creation during installation.

Specifically, you will need to set the `--disable-account-creation` option when you run the `initialize-tsm` script. You will also need to specify the account name with the `--unprivileged-user` option. If the account that you specify does not exist, then the `initialize-tsm` script will create it. See [Help Output for initialize-tsm Script](#) for more details.

If you want to specify an existing account with the `--unprivileged-user` option, verify that the user account is an unprivileged user with normal login privileges. Configure the account with the following characteristics:

- Shell set to `/bin/bash`.
- For convenience, consider setting the home directory to the data directory path. The account must have ownership and write privileges to the home directory.

If you specify a different unprivileged account during setup, you must manually add that same user to the `systemd-journal` group. The unprivileged user must be a member of the `systemd-journal` group so that Tableau Server can collect logs from some services (such as Ask Data) when running the `tsm maintenance ziplogs` command. If the unprivileged user is not a member of the group, ziplogs will not contain logs from the affected services

Groups

Tableau Server requires two groups for operation.

In a default installation, the local `tableau` service account belongs to a primary group named `tableau`. However, if you specify an alternate unprivileged user during installation, then the primary group for that alternate account will be used. As a convenience, any account can be added to this group to be able to read the Tableau Server log files (without becoming root).

The second group is used to authorize which users are authorized to authenticate to Tableau Services Manager (TSM). Any user in this group will be able to send commands to TSM, so it should be restricted to Tableau Server administrators. By default, this group is named `tsmadmin`.

If you are not going to use the default name, you will need to specify the group name with the `--tsm-authorized-group` option when you run `initialize-tsm`. See Help Output for `initialize-tsm` Script for more details.

Authenticating clients

Basic user authentication in Tableau Server is by username and password sign-in for both local and external user stores. In the local case, user passwords are stored as a hashed password in the repository. In the external case, Tableau Server passes the credentials to the external user store and awaits a response as to whether the credentials are valid.

External user stores can also handle other kinds of authentication like Kerberos or SSPI (Active Directory only), but the concept is still the same, Tableau Server delegates the credentials or user to the external store and awaits a response.

You can configure Tableau Server such that username-password sign-in is disabled. In these scenarios other authentication methods, such as trusted authentication, OpenID, or SAML can be used. See [Authentication](#).

In some cases, you may need to update LDAP external directories to allow bind operations with username + DN format from Tableau Server. See [User binding behavior on sign in](#).

User Management in Deployments with External Identity Stores

This topic describes important technical details that you should be familiar with if you use an external identity store to manage users for Tableau Server. Tableau Server supports connecting to an external directory using LDAP. In this scenario, Tableau Server imports users from the external LDAP directory into the Tableau Server repository as system users.

Arbitrary LDAP directories

The system username in Tableau is whatever attribute you set as part of LDAP configuration, for example "cn". This is true for both individual user import and group sync functionality. See [External Identity Store Configuration Reference](#).

User binding behavior on sign in

You may need to update your LDAP configuration to allow binding with usernames appended with the DN. Specifically, you will need to update your LDAP configuration when Tableau Server is configured with an arbitrary LDAP directory. (e.g., OpenLDAP) that uses UPN or Email addresses as usernames.

Tableau Server will search for a given user based on the username that is supplied during sign in. Tableau Server will then attempt to bind with the username appended with the DN. In the case where Tableau Server has been configured with GSSAPI, then the username@REALM (domain name) will be used.

Active Directory

This content in rest of this topic assumes that you are familiar with Active Directory user management and basic Active Directory schema and domain concepts.

Note: In the context of user and group synchronization, Tableau Server configured with LDAP identity store is equivalent to Active Directory. Active Directory synchronization features in Tableau Server function seamlessly with properly configured LDAP directory solutions.

Active Directory user authentication and Tableau Server

Tableau Server stores all user names in the Tableau Server identity store, which is managed by the repository. If Tableau Server is configured to use Active Directory for authentication, you must first import user identities from Active Directory to the identity store. When users sign in to Tableau Server, their credentials are passed to Active Directory, which is responsible for authenticating the user; Tableau Server does not perform this authentication. (By default, NTLM is used for authentication, but you can enable Kerberos or SAML for single sign-on functionality—however, in all these cases, authentication is left to Active Directory.) However, the Tableau user names stored in the identity store are associated with rights and permissions for Tableau Server. Therefore, after authentication is verified, Tableau Server manages user access (authorization) for Tableau resources.

Active Directory user name attributes and Tableau Server

Active Directory uniquely identifies user objects using several attributes. (For details, see [User Naming Attributes](#) on the MSDN website.) Tableau Server relies on two Active Directory user naming attributes:

- `sAMAccountName`. This attribute specifies the logon name that was originally designed for use with older versions of Windows. In many organizations, this name is combined with the NetBIOS name for authentication, using a format like

Tableau Server on Linux Administrator Guide

`example\jsmith`, where `example` is the NetBIOS name and `jsmith` is the `sAMAccountName` value. Due to the original design in Windows, the `sAMAccountName` value must be less than 20 characters.

In the Windows **Active Directory Users and Computers** administrative console, this value is in the field labeled **User logon name (pre-Windows 2000)** on the **Account** tab of the user object.

- `userPrincipalName` (UPN). This attribute specifies a user name in the format `jsmith@example.com`, where `jsmith` is the UPN prefix and `@example.com` is the UPN suffix.

In the Windows **Active Directory Users and Computers** administrative console, the UPN is a concatenation of two fields on the **Account** tab of the user object: the **User logon name** field, and the domain drop-down list next to it.

Adding users from Active Directory

You can **add users individually** from Active Directory, either by typing them in the server environment or by creating a CSV file and importing the users. You can also add Active Directory users by **creating a group via Active Directory** and importing all of the group's users. The result can be different depending on which approach you're using.

Adding user groups

If you import an Active Directory user group, Tableau will import all users from the group using the `sAMAccountName`.

Sync behavior when removing users from Active Directory

Users cannot be automatically removed from Tableau Server through an Active Directory sync operation. Users that are disabled, deleted, or removed from groups in Active Directory remain on Tableau Server so that you can audit and reassign the user's content before removing the user's account completely.

However, Tableau Server will act upon user objects differently based how the status of that user object changes in Active Directory. There are two scenarios: deleting/disabling users in Active Directory or removing users from synchronized groups in Active Directory.

When you delete or disable a user in Active Directory and then synchronize that user's group on Tableau Server, the following occurs:

- The user is removed from the Tableau Server group you synchronized.
- The user's role is set to "unlicensed."
- The user will still belong to the All Users group.
- The user is unable to sign in to Tableau Server.

When you remove a user from a group in Active Directory and then synchronize that group on Tableau Server, the following occurs:

- The user is removed from the Tableau Server group you synchronized.
- The user's role is retained: it is not set to "unlicensed."
- The user will still belong to the All Users group.
- The user will still have permission to the Tableau Server with access to everything that the All Users group is granted permission to use.

In both instances, to remove a user from Tableau Server, the server administrator must delete the user from the Server Users page in Tableau Server.

Domain nicknames

In Tableau Server, domain nickname is equivalent to the Windows NetBIOS domain name. In a Windows Active Directory forest, a fully qualified domain name (FQDN) can have an arbitrary NetBIOS name. The NetBIOS name is used as the domain identifier when a user logs in to Active Directory.

For example, the FQDN `west.na.corp.lan` might be configured with a NetBIOS name (nickname) of `SEATTLE`. The user `jsmith` in that domain could log on to Windows using either of the following user names:

- `west.na.corp.example.com\jsmith`
- `SEATTLE\jsmith`

Tableau Server on Linux Administrator Guide

If you want your users to sign in to Tableau Server with a NetBIOS name instead of the FQDN, then you'll need to verify that the nickname value for each domain where users log in is set. See `editdomain` for information on how to view and set the nickname value for each domain.

Support for multiple domains

You can add users from a domain that's different from the domain of the Tableau Server computer in these cases:

- Two-way trust has been established between the server's domain and the users' domain.
- The server's domain trusts the users' domain (one-way trust). See [Domain Trust Requirements for Active Directory Deployments](#).

The first time you add a user from the non-server domain, use the fully qualified domain name with the user name by setting the `authentication saml map-assertions` option with TSM. Any additional users you add from that domain can be added using the domain's nickname, provided the nickname matches the NetBIOS name. If Tableau Server connects to multiple domains, you must also specify the other domains that Tableau Server connects to by setting the `wgserver.domain.whitelist` (version 2020.3 and earlier) or `wgserver.domain.accept_list` (version 2020.4 and later) option with TSM. For more information, see `wgserver.domain.whitelist` or `wgserver.domain.accept_list`.

Duplicate display names

If user display names are not unique across multiple domains, then managing users with the same display name in Tableau can be confusing. Tableau Server will display the same name for two users. For example, consider an organization with two domains, `example1.lan` and `example2.lan`. If user John Smith exists in both domains, then adding that user to groups and other administrative tasks will be confusing in Tableau Server. In this scenario, consider updating the display name in Active Directory for one of the users to differentiate the accounts.

Sign in to Tableau Server with NetBIOS name

Users can sign in to Tableau Server using the domain nickname (NetBIOS name), for example, `SEATTLE\jsmith`.

Tableau Server cannot query for NetBIOS name for a given FQDN. As a result, Tableau sets the nickname of a given FQDN according to the first entry in the namespace. For example, given the FQDN `west.na.corp.lan`, Tableau sets the nickname to `west`.

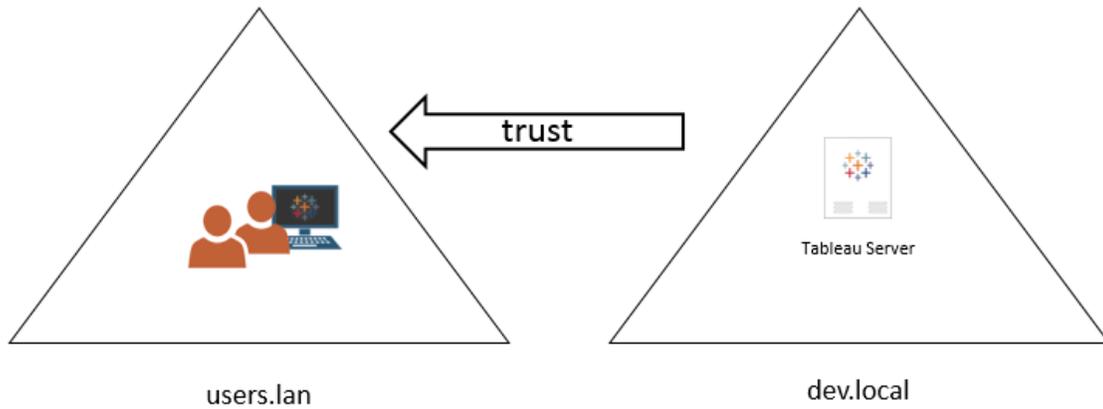
Therefore, you might need to update the domain nickname on Tableau Server before users can sign in using the nickname. If you do not update the nickname, users will have to sign in using a fully qualified domain name. For more information, see [Users From New Domain Unable to Log In and Do Not Appear in User List](#) in the Tableau Knowledge Base.

Domain Trust Requirements for Active Directory Deployments

When you run Tableau Server in an Active Directory environment across multiple domains (either in the same Active Directory forest or in different forests), some Tableau functionality is dependent on the trust relationship between the domains. For example, some administrators manage users in domains that are separate from where they deploy server applications, such as Tableau Server. In other organizations, a Tableau Server deployment might be shared with external partners or with different partners in the organization. Finally, Windows-authenticated data sources, such as SQL Server, MSAS, or Oracle, that Tableau Server connects to may also be in other domains.

If it's feasible, we recommend configuring two-way trust between all domains that interact with Tableau Server. If this is not possible, Tableau Server can be configured to support user authentication where a one-way trust has been configured. In this case, a one-way trust between domains is supported when the domain in which Tableau Server is installed is configured to trust the domain where user accounts reside.

The following illustration shows one-way trust between the domain where Tableau Server is installed and the domain where user accounts reside:



In this scenario, Tableau Server is in the dev.local domain, and users from the users.ian Active Directory domain are imported into Tableau Server. A one-way trust is required for this scenario; specifically, the dev.local domain is configured to trust the users.ian domain. Users in the users.ian domain can access Tableau Server in the dev.local with their normal Active Directory credentials. However, you may need to update the domain nickname on Tableau Server before users log on with the nickname. Refer to the [Tableau Knowledge Base](#) for more information.

When you configure Tableau Server for this scenario, specify the primary user domain during installation. See [Configure Initial Node Settings](#). To ensure that Tableau Server can connect to other Active Directory domains, you must also specify other domains that Tableau Server connects to by setting the `wgserver.domain.accept_list` option with TSM. For more information, see `wgserver.domain.accept_list`.

Duplicate bind accounts for domain trust

Tableau Server on Linux relies on JDK's LDAP implementation that uses a simple bind to authenticate with Active Directory. Simple bind is not domain-aware and as a result, does not support a cross-domain bind. When you set up the initial identity store, you must supply the bind account that you will use to authenticate to Active Directory.

To enable cross-domain trust and directory look-ups, you must duplicate this bind account in every target domain. Each bind account in each domain must use the same username (`sAMAccountName` or `dn`) and password.

Kerberos single sign-on is supported in this one-way trust scenario.

Review [User Management in Deployments with External Identity Stores](#) to understand how multiple domains, domain naming, NetBIOS, and Active Directory user name format influence Tableau user management.

Connecting to live data in one-way trust scenarios

In the one-way trust scenario, users connecting to Tableau Server can connect to live data that's hosted in the cloud or on any other data source on premises that does not rely on Windows authentication.

Data sources that require Windows-authentication might have additional authentication requirements that complicate the scenario, or that can even prevent Tableau Server users from connecting. This is because Tableau Server uses a service account for authentication with such data sources. If you are running Tableau Server in a different domain than data sources that use Windows authentication, verify that the service account that is used for Tableau Server can access the data source.

Communicating with the Internet

In most enterprises, Tableau Server needs to communicate with the internet. Tableau Server was designed to operate inside a protected internal network. Do not set up Tableau Server directly on the internet or in a DMZ. Instead, communications between your network and the internet should be mediated using proxy servers. If the computer running Tableau Server cannot access the internet directly, then you may need to deploy forward proxy servers to mediate traffic from inside the network to targets on the internet. Tableau Server doesn't support pass-through or manual proxy authentication.

For inbound traffic, we recommend running Tableau Server behind reverse proxy servers.

How Tableau communicates with the internet

Tableau Server requires outbound access to the internet for these scenarios:

- Working with maps. Tableau uses map data that is hosted externally.

Tableau Server needs to connect to the following internet locations with port 443 to use maps:

- `mapsconfig.tableau.com`
- `api.mapbox.com`

If Tableau cannot make these connections, maps may fail to load.

You can test connectivity by accessing each of those addresses in a browser:

`https://mapsconfig.tableau.com/v1/config.json` and `https://api.mapbox.com/` will prompt you to download a json file.

If you use a proxy to connect to the internet and are unable to connect to `api.mapbox.com`, see [Working with firewalls](#) on the Mapbox website.

For Tableau Server version 2019.1 and earlier, see the documentation for your version: [Tableau Help](#)

- Connecting to the Tableau send-logs server.

You can upload log files to Tableau when working with Support. See `tsm maintenance send-logs`. To successfully upload files to Tableau, your Tableau Server must be able to communicate with the send-logs server on port 443:

`report-issue.tableau.com:443`

- Sending Basic Product Data.

The domain, `prod.telemetry.tableausoftware.com`, is used by Tableau to receive the Basic Product Data about process launch and shutdown. It is also used for the more general Product Usage Data.

Traffic to this domain will occur on port 80 (for initial registration of our Product Data clients) and on port 443 (for all subsequent traffic).

prod.telemetry.tableausoftware.com:80

prod.telemetry.tableausoftware.com:443

- Licensing. Tableau products connect to the internet to activate product keys. Unless you activate Tableau software with the [Offline Activation Tool](#), all Tableau products must have access to the internet to validate licenses. Specifically Tableau requires internet access during the following licensing operations: activation, deactivation, and on the refresh maintenance date. For more information about these operations, see [Manage Licenses](#).

Tableau Server needs to connect to the following internet locations for licensing purposes:

- licensing.tableau.com:443
- atr.licensing.tableau.com:443
- s.ss2.us
- ocsp.rootg2.amazontrust.com
- ocsp.rootca1.amazontrust.com
- ocsp.sca1b.amazontrust.com
- crt.sca1b.amazontrust.com
- crt.rootca1.amazontrust.com
- ocsp.sca0a.amazontrust.com
- crt.sca0a.amazontrust.com

Tableau Server on Linux Administrator Guide

- `ocsp.sca1a.amazontrust.com`
- `crt.sca1a.amazontrust.com`
- `ocsp.sca2a.amazontrust.com`
- `crt.sca2a.amazontrust.com`
- `ocsp.sca3a.amazontrust.com`
- `crt.sca3a.amazontrust.com`
- `ocsp.sca4a.amazontrust.com`
- `crt.sca4a.amazontrust.com`
- `crl.rootca1.amazontrust.com`
- `crl.rootg2.amazontrust.com`
- `crl.sca1b.amazontrust.com`

Requests to the above domains may be on port 80 or 443. Port 80 is used for certificate validation (revocation, certificate chain, etc). Port 443 is used for SSL connections.

If Tableau Server cannot make a connection while attempting to activate its license, you will be prompted to do an offline activation.

- Working with external or cloud-based data.

Tableau Server needs to connect to the following internet location for Box, Dropbox, OneDrive, Google Drive, Google Sheets, and Anaplan services:

```
galop.connectors.tableau.com:443
```

Tableau Server can run without internet access. For more information about deploying Tableau Server in organizations without access to the internet, see [Install Tableau Server in a Disconnected \(Air-Gapped\) Environment](#).

In many enterprises, users also need to access Tableau Server from outside the network (that is, from the internet). For example, in many enterprises, users want to be able to reach Tableau Server from their mobile devices in order to interact with views that are stored on the server. To configure access to Tableau Server from the internet or from mobile devices, you should use a reverse proxy. See [Configuring Proxies for Tableau Server](#).

Configuring Proxies for Tableau Server

In most enterprises, Tableau Server needs to communicate with the internet. Tableau Server was designed to operate inside a protected internal network. Do not set up Tableau Server directly on the internet or in a DMZ. Instead, communications between your network and the internet should be mediated using proxy servers. Forward proxy servers mediate traffic from inside the network to targets on the internet. Reverse proxy servers mediate traffic from the internet to targets inside the network.

Who should read this article?

This article is for IT professionals who are experienced with general networking and gateway proxy solutions. The article describes how and when Tableau requires internet access, and describes how to configure your network and Tableau to use forward and reverse proxy servers for access to and from the internet. There are many third-party proxy solutions available, so some of the content in the article is necessarily generic.

Before you configure a proxy server, see [Communicating with the Internet](#).

Configure a forward proxy server

To enable communication from Tableau Server to the internet, deploy Tableau Server behind a forward proxy server. When Tableau Server needs access to the internet, it doesn't send the request directly to the internet. Instead, it sends the request to the forward proxy, which in turn forwards the request. Forward proxies help administrators manage traffic out to the internet for tasks such as load balancing, blocking access to sites, etc.

Tableau Server on Linux Administrator Guide

If you use a forward proxy, you must configure the computers that run Tableau Server inside the network to send traffic to the forward proxy. Tableau Server doesn't support pass-through or manual proxy authentication.

Configuring Tableau Server on Linux to work with a forward proxy

We recommend configuring Tableau Server to work with your forward proxy solution as part of the installation process. Specifically, configure Tableau Server when you run `./initialize-tsm` as described in *Install and Initialize TSM*, or as part of *Automated Installation of Tableau Server*.

The procedure below describes how to create a forward proxy configuration file for Tableau Server on Linux.

The configuration file is stored in the following directory:

```
~<unprivileged_user>/.config/systemd/tableau_server.conf.d
```

By default, Tableau Server creates the unprivileged user, `tableau`. Therefore, the default path to the configuration directory is:

```
~tableau/.config/systemd/tableau_server.conf.d
```

The proxy configuration file name in this topic and in the configuration file below is referred to as `20-proxy.conf`. You can name this file according to your own convention, but it must use the `.conf` extension. `systemd` will process files stored in the `tableau_server.conf.d` directory in lexical order according to file name.

1. Run the `tsm stop` command.
2. Start a session as the unprivileged user. By default, `tableau`, is the unprivileged user created by Tableau Server during installation. Run the following command:

```
sudo su -l tableau
```

3. Create or open the `20-proxy.conf` file in the `tableau_server.conf.d`

directory. If you configured forward proxy during setup, then the `20-proxy.conf` file has already been created.

- Create the file. Run the following command:

```
touch ~tableau/.config/systemd/tableau_serv-  
er.conf.d/20-proxy.conf
```

- Open the `20-proxy.conf` file in a text editor.

4. Copy the *Proxy configuration file contents* into the file. If you are editing an existing file, take care not to delete the configuration. The *Proxy configuration file contents* include instructions on how to set forward proxy configurations. After you have edited and saved the file go to Step 5.

Proxy configuration file contents

```
# Always edit this file on Tableau Server as the unpriv-  
ileged user. By default, tableau, is the unprivileged user  
created by Tableau Server during installation.  
# Set environment variables http_proxy and https_proxy to  
point to your proxy host.  
# For example, to set the proxy to example-host for ports  
80 and 443, run the following commands:  
#  
http_proxy=http://example-host:80/  
https_proxy=http://example-host:443/  
#  
# Take care to use 'http' when you specify the URL for the  
https_proxy environmental variable.  
# Do not specify the 'https' protocol for the value of  
the https_proxy environmental variable.  
#
```

Tableau Server on Linux Administrator Guide

```
# To bypass the proxy server, specify exceptions in the no_
proxy variable. Use this variable if your proxy server does
not route internal addresses.
# You must also add exceptions to this proxy configuration
to guarantee that all communications within a local Tableau
Server cluster (if you have one now or will have one later)
do not route to the proxy server.
# Enter both the host name and the IP address for each com-
puter. Additionally, include the canonical host name (loc-
alhost) and IP address (127.0.0.1) for the local computer.
# For example, to specify exceptions for a three-node
cluster:
#
no_proxy-
="lo-
cal-
host,127.0.0.1,hostname1,hostname2,hostname3,IP1,IP2,IP3"
#
# Where "hostname1" is the actual hostname of node 1, and
"IP1" is the actual IP address of node 1, etc.
```

5. Exit the Tableau shell. Run the following command:

```
exit
```

6. Restart the TSM business services. Run the following script:

```
sudo /opt/tableau/tableau_server-
/packages/scripts.<version>/start-administrative-services
```

7. Restart TSM.

```
tsm restart
```

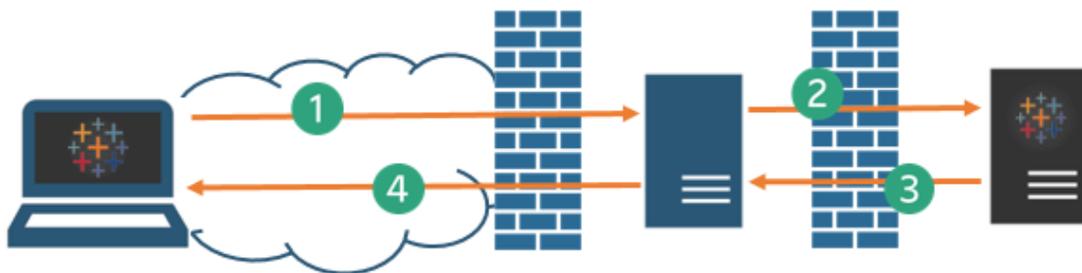
Server crash reporter

If your organization uses a proxy server to connect to the internet then you must configure Tableau Server's crash reporter to use the proxy. Even if you have already configured Tableau Server to use a proxy, you must also configure server crash reporter separately. To configure proxy for server crash reporter, see [Configure Server Crash Reporter](#).

How a reverse proxy works with Tableau Server

A reverse proxy is a server that receives requests from external (internet) clients and forwards them to Tableau Server. Why use a reverse proxy? The basic answer is security. A reverse proxy makes Tableau Server available to the internet without having to expose the individual IP address of that particular Tableau Server to the internet. A reverse proxy also acts as an authentication and pass-through device, so that no data is stored where people outside the company can get to it. This requirement can be important for organizations that are subject to various privacy regulations such as PCI, HIPAA, or SOX.

The following diagram illustrates the communication path when a client makes a request to Tableau Server that is configured to work with a reverse proxy server.



1. An external client initiates a connection to Tableau Server. The client uses the public URL that's been configured for the reverse proxy server, such as `https://tableau.example.com`. (The client doesn't know that it's accessing a reverse proxy.)
2. The reverse proxy maps that request in turn to a request to Tableau Server. The reverse proxy can be configured to authenticate the client (using SSL/TLS) as a

precondition to passing the request to Tableau Server.

3. Tableau Server gets the request and sends its response to the reverse proxy.
4. The reverse proxy sends the content back to the client. As far as the client is concerned, it just had an interaction with Tableau Server, and has no way to know that the communication was mediated by the reverse proxy.

Proxy servers and SSL

For better security, you should configure reverse proxy servers to use SSL for any traffic that's external to your network. This helps to ensure privacy, content integrity, and authentication. Unless you've deployed other security measures to protect traffic between your internet gateway and Tableau Server, we also recommend configuring SSL between the gateway proxy and Tableau Server. You can use internal or self-signed certificates to encrypt traffic between Tableau Servers and other internal computers.

Mobile access

Tableau Server adds an X-header to all HTTP responses for Tableau Mobile sessions. By default, most proxy solutions will preserve X-headers. If your proxy solution does not preserve X-headers, then you will need to configure your proxy server to preserve the following header to all HTTP responses for Mobile client sessions: `X-Tableau: Tableau Server`.

If you have configured authentication at the proxy server gateway, then your proxy server must respond to Tableau Mobile HTTP requests with a HTTP 302 response. The 302 must include a redirect to the identity provider login page. To view a diagram that describes the 302 authentication sequence, see [Tableau Mobile Authentication Sequence](#) in the Tableau Community.

Reverse proxy and user authentication

Tableau Server will always authenticate users. This means that even if you are authenticating inbound connections at the gateway for your organization, Tableau Server will still

authenticate the user.

However, not all clients will support user authentication with a reverse proxy:

- For supported web browsers, you can use SAML, OpenID Connect, Kerberos, Trusted Tickets or manual authentication with a reverse proxy. However, we recommend a transparent scenario where user requests are not prompted for authentication at the gateway. This recommendation doesn't prohibit using SSL for client/server system-level authentication at the gateway proxy, in fact, we strongly recommend SSL system-level authentication.
- Tableau Mobile supports SAML or manual authentication with a reverse proxy. The iOS version of Tableau Mobile additionally supports Kerberos with a reverse proxy. The same recommendation above applies.
- Tableau Desktop does not support authentication with a reverse proxy. To support remote access with Tableau Desktop, use a VPN solution or configure your reverse proxy to route traffic from Tableau Desktop directly to Tableau Server for authentication.

If your organization is authenticating with Active Directory:

- Active Directory with Enable automatic logon (SSPI) is not supported with a reverse proxy.
- Tableau Server must be configured for reverse proxy before configuring Tableau Server for Kerberos. For more information, see [Configure Kerberos](#).

Configure Tableau Server to work with a reverse proxy server

Before you configure Tableau Server, you'll need to collect the following information about the proxy server configuration. To configure Tableau Server, you use the `tsm configuration set` command. The information you need to collect corresponds to options you'll need when you run `tsm`.

Most of the following `tsm` options are also used to configure Tableau Server deployments that operate behind a load balancer. For more information, see [Add a Load Balancer](#).

Item	Description	Corresponding <code>tsm configuration set</code> option
IP address or CNAME	<p>You can either enter an IP address or a CNAME for this option.</p> <p>The public IP address or addresses of the proxy server. The IP address must be in IPv4 format, such as <code>203.0.113.0</code>, and it must be a static IP.</p> <p>If you are unable to provide a static IP, or if you are using cloud proxies or external load balancers, you can specify the CNAME (Canonical Name) DNS value that clients will use to connect to Tableau Server. This CNAME value must be configured on your reverse proxy solution to communicate with Tableau Server.</p>	<code>gateway.trusted</code>
FQDN	<p>The fully qualified domain name that people use to reach Tableau Server, such as <code>tableau.example.com</code>. Tableau Server doesn't support context switching for this option. For example, the following URL is not supported: <code>example.com/tableau</code>.</p>	<code>gateway.public.host</code>
Non-FQDN	<p>Any subdomain names for the proxy server. In the example of <code>tableau.example.com</code>, the subdomain name is <code>tableau</code>.</p>	<code>gateway.trusted_hosts</code>

Item	Description	Corresponding <code>tsm configuration set</code> option
Aliases	Any public alternative names for the proxy server. In most cases, aliases are designated using CNAME values. An example would be a proxy server <code>bigbox.example.com</code> and CNAME entries of <code>ftp.example.com</code> and <code>www.example.com</code> .	<code>gateway.trusted_hosts</code>
Ports	Port numbers for traffic from the client to the reverse proxy server.	<code>gateway.public.port</code>

If you are using a distributed installation of Tableau Server, then run the following `tsm` commands on the initial node in your cluster.

1. Enter the following command to set the FQDN that clients will use to reach Tableau Server through the proxy server, where *name* is the FQDN:

```
tsm configuration set -k gateway.public.host -v 'name'
```

For example, if Tableau Server is reached by entering `https://tableau.example.com` in the browser, enter this command:

```
tsm configuration set -k gateway.public.host -v 'tableau-example.com'
```

2. Enter the following command to set the address or the CNAME of the proxy server, where *server_address* is the IPv4 address or CNAME value:

```
tsm configuration set -k gateway.trusted -v 'server_ip_address'
```

If your organization uses multiple proxy servers, enter multiple IPv4 addresses, separating them with commas. IP ranges are not supported. To improve start up and initialization of Tableau Server, minimize the number of entries for `gateway.trusted`.

3. Enter the following command to specify alternate names for the proxy server, such as its fully qualified domain name, any not fully qualified domain names, and any aliases. If there's more than one name, separate the names with a comma.

```
tsm configuration set -k gateway.trusted_hosts -v 'name1,
name2, name3'
```

For example:

```
tsm configuration set -k gateway.trusted_hosts -v
'proxy1.example.com, proxy1, ftp.example.com, www.ex-
ample.com'
```

4. If the proxy server is using SSL to communicate with the internet, run the following command, which tells Tableau that the reverse proxy server is using port 443 instead of port 80:

```
tsm configuration set -k gateway.public.port -v 443
```

Note: If the proxy server is using SSL to communicate with Tableau Server, SSL must be configured and enabled on Tableau Server.

5. Enter the following command to commit the configuration change:

```
tsm pending-changes apply
```

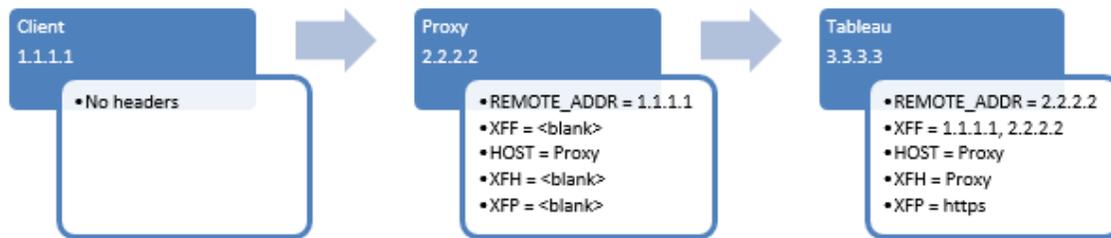
If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the

restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

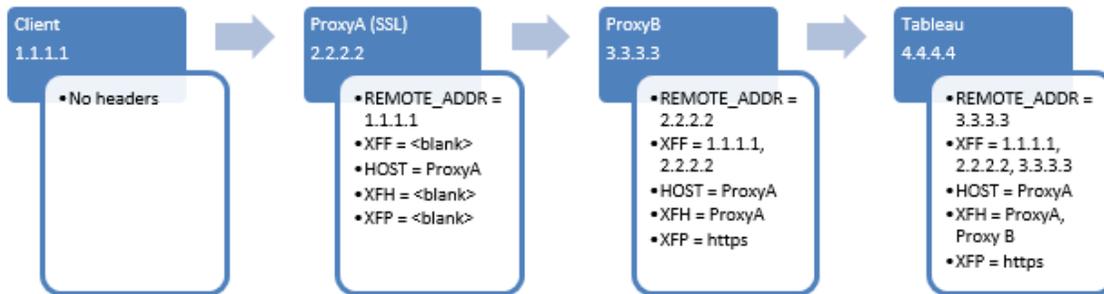
Configure the reverse proxy server to work with Tableau Server

When a client accesses Tableau Server through a reverse proxy, specific message headers have to be preserved (or added). Specifically, all proxy servers in the message chain must be represented in the `gateway.trusted` and `gateway.trusted_hosts` settings.

The following graphic shows example headers for a single-hop message chain, where the proxy server is communicating directly with Tableau Server:



The following graphic shows example headers for a multiple-hop message chain, where the message traverses two proxy servers before connecting to Tableau Server:



The following table describes what these headers are and how they relate to the configuration settings on Tableau Server:

Headers	Description	Related Tableau Server settings
---------	-------------	---------------------------------

<p>REMOTE_ADDR and X-FORWARDED-FOR (XFF)</p>	<p>Tableau Server needs these headers to determine the IP address of origin for requests. X-FORWARDED-FOR header must present IP address chain to Tableau Server in the order the connections have occurred.</p>	<p>The IP address that you set in <code>gateway.trusted</code> must match the IP presented in REMOTE_ADDR. if you sent multiple addresses in <code>gateway.trusted</code>, one of them must match the IP presented in REMOTE_ADDR.</p>
<p>HOST and X-FORWARDED_HOST (XFH)</p>	<p>These headers are used to generate absolute links to Tableau Server when it replies to the client. X-FORWARDED-HOST header must present host names to Tableau Server in the order the connections have occurred.</p>	<p>The host names that are presented in X-FORWARDED-HOST header must be included in the host names that you specify in <code>gateway.trusted_hosts</code>.</p>
<p>X-FORWARDED-PROTO (XFP)</p>	<p>This header is required if SSL is enabled for traffic from the client to the proxy, but not for traffic from the proxy to Tableau Server.</p> <p>The X-FORWARDED-PROTO headers are important for scenarios where HTTP or HTTPS is not maintained along each hop of the message route. For example, if the reverse proxy requires SSL for outside requests, but traffic between the reverse proxy and Tableau Server is not configured to use SSL, X-FORWARDED-PROTO head-</p>	<p>Port configuration on reverse proxy (inbound connections from client and outbound connections to Tableau Server) must be specified in the corresponding parameter: <code>gateway.public.port</code>, which is the port clients use to connect to the proxy.</p> <p>If the proxy server is using SSL to communicate with Tableau Server, SSL must be configured and enabled</p>

	<p>ers are required. Some proxy solutions add the <code>X-FORWARDED-PROTO</code> headers automatically, while others do not. Finally, depending on your proxy solution, you might have to configure port forwarding to translate the request from port 443 to port 80.</p> <p>Related KB article: "Unable to Sign In" and "Invalid username or password" Error With SAML After Upgrading.</p>	<p>on Tableau Server.</p>
--	---	---------------------------

Validate reverse proxy setup

To validate your reverse proxy setup, publish workbooks and datasources using Tableau Server web authoring or Tableau Desktop. If you are connecting with a web browser to Tableau Server from the internet, verify that you are using a [recommended browser](#). Publish and view workbooks that use existing datasources as well as a datasource that you've published. Use the links below to familiarize yourself with connecting with Tableau Server as an end-user.

Task	Documentation
Overview of web authoring.	Use Tableau on the Web
Log in to Tableau Server from Tableau Desktop or a web browser.	Sign in to Tableau Server or Online
Publish a workbook to Tableau Server.	Publish a Workbook
Publish a data source.	Publish a Data Source
Open workbook from Tableau Server.	Opening Workbooks from

	the Server
Log out Server (with Desktop).	Sign in to Tableau Server or Online
Download workbook from a web browser.	Download Workbooks
Check to make sure tabcmd (from a non-server client) works.	tabcmd

Related topics

- [Tableau Desktop Internet Access Requirements](#)
- [Add a Load Balancer](#)

Deploy

The topics in this section provide information on installing, configuring, and upgrading Tableau Server on Linux.

Looking for Tableau Server on Windows? See [Install and Configure Tableau Server](#)

- [Install and Configure Tableau Server](#) 167
- [Automated Installation of Tableau Server](#) 241
- [Install Tableau Server in a Disconnected \(Air-Gapped\) Environment](#) 248
- [Tableau Server in a Container](#) 256
- [Install Tableau Server in the Cloud](#) 318
- [Post Installation Tasks](#) 319
- [Distributed and High Availability Tableau Server Installations](#) 359
- [Migrate](#) 475
- [Upgrade Tableau Server Overview](#) 501
- [Uninstall Tableau Server](#) 581

Install and Configure Tableau Server

The topics referenced at the bottom of this page describe the steps to install and configure Tableau Server. If you are installing a distributed deployment (cluster), use the steps in this topic to install the initial node, then, to install additional nodes, see Distributed and High Availability Tableau Server Installations.

After you run the installation, you must then continue setup by activating a license, registering Tableau Server, and configuring various settings including authentication.

Other installation methods

There are a few alternative methods that you can use to install Tableau Server.

- If you want a quick start procedure to install Tableau Server in a non-production environment, see [Jump-start Installation](#).
- [Automated Installation of Tableau Server](#).
- If you are installing Tableau Server in an environment without a connection to the internet, see [Install Tableau Server in a Disconnected \(Air-Gapped\) Environment](#).
- You can also install Tableau Server onto various cloud platforms. See [Install Tableau Server in the Cloud](#).
- For single-server installations in departments or small businesses, see [Everybody's Install Guide](#).

Before you begin

To install Tableau Server you must have a computer that satisfies the hardware requirements. You will get an informational message if your computer meets the minimum requirements but does not satisfy the recommended minimum requirements. In this case, your computer hardware can handle a trial installation of Tableau but is not adequate for a production environment. For more information, see [Before you install...](#)

Installation steps

The following steps describe how to install Tableau Server on a single computer. Use the steps to install Tableau Server in a single server deployment. Use the steps to install the initial node in a multi-node Tableau Server deployment. Run the steps sequentially.

Tableau Server on Linux Administrator Guide

1. Install and Initialize TSM
2. Activate and Register Tableau Server
3. Configure Initial Node Settings
4. Add an Administrator Account
5. Validate Installation

Before you install...

Note: You can find additional information about technical specifications for Tableau Server on the Tableau web site, [here](#).

This topic includes requirements and recommendations that you must consider before you install Tableau Server into a production environment.

If you want to install a single server, or if you want to do a minimal installation for test purposes, refer to our single-server installation guide, [Everybody's Install Guide](#).

If you are deploying Tableau Server in a distributed cluster, review Distributed Requirements in addition to the requirements and recommendations described in this topic.

If you are migrating from Tableau Server on Windows to Tableau Server on Linux, see [Migrate Tableau Server from Windows to Linux](#).

Hardware recommendations

The following list describes the minimum hardware recommendations for a single- node installation of Tableau Server:

<i>Install Type</i>	<i>Processor</i>	<i>CPU</i>	<i>RAM</i>	<i>Free Disk Space</i>
Single node	<ul style="list-style-type: none"> • 64-bit • Must support SSE4.2 and POPCNT instruction sets • ARM-based processors are not supported 	8-core, 2.0 GHz or higher	32 GB	50 GB

If you are adding Tableau Prep Conductor to your Tableau Server installation, we recommend you add a second node and dedicate this to running Tableau Server Prep Conductor. This node should have a minimum of 4 cores (8 vCPUs), and 16 GB of RAM.

Multi-node and enterprise deployments	<p>Contact Tableau for technical guidance.</p> <p>Nodes must meet or exceed the minimum hardware recommendations, except:</p> <ul style="list-style-type: none"> • Nodes running backgrounder, where 4 cores may be acceptable. • Dedicated node for Tableau Prep Conductor: Minimum of 4 cores (8 vCPUs), and 16 GB of RAM.
---------------------------------------	--

Important: The disk space requirement cannot be checked until you initialize TSM. If you don't have enough space, you won't be told this until after you install the Tableau Server package.

50 GB disk space available, with a minimum of 15 GB allocated to the `/opt` directory, and the remainder allocated to the `/var` directory for data storage.

- Free disk space is calculated after the Tableau Server Setup program is unzipped. The Setup program uses about 1 GB of space. You may need to allocate additional disk space depending on various factors like whether you will be using extracts.

The core Tableau Server bits must be installed in a directory with at least 15 GB of free disk space. If you attempt to install Tableau Server on a computer that does not have enough space, the Tableau Server package will install, but you will be unable to continue with setup. By default the install location is the `/opt` directory. You can change the installation path for Tableau Server on RHEL distros.

If you plan to make heavy use of extracts then you may need to allocate additional disk space. You can specify a different directory for data (extract) storage during installation.

- **Network attached storage space requirements for External File Store:** If you are planning to configure [Tableau Server with External File Store](#), you will need to estimate the amount of storage space to dedicate on your network attached storage.

Estimating the storage size: You must take into account the amount of storage needed for publishing and refreshing extracts. In addition, you must also take into account the repository backup size unless you specifically choose the option to do your repository backup separately as described in the [Option 2: Backup Repository Separately](#) topic.

- Extracts:
 - Consider the number of extracts that will be published to Tableau Server and the size of each extract. Test your needs by publishing several extracts to Tableau Server, and then checking the disk space used. You can use this amount of disk space to help you figure out how many extracts will be published to Tableau Server over time as well as how each existing extract will increase in size.
 - Consider the space needed by the temp directory during an extract

refresh. The temp directory, which is where an extract is stored to during a refresh, may require up to three times the final file size of the extract.

- Repository Backup:
 - To obtain an estimate of the repository data, check the size of `<data directory>/pgsql/data/base` directory.
 - To obtain the exact size of the repository data, open the backup file and use the size of the `workgroup.pg_dump` file.
- Core count is based on "physical" cores. Physical cores can represent actual server hardware or cores on a virtual machine (VM). Hyper-threading is ignored for the purposes of counting cores.
- RAM shown is the minimum recommended for a single-node installation. Your installation may function better with more RAM, depending on activity, number of users, and background jobs, for example.

To see the full list of recommendations and to see the minimum requirements, see [Minimum Hardware Requirements and Recommendations for Tableau Server](#).

For public cloud deployments on Amazon Web Services and Google Cloud Platform, their "vCPU" is actually a CPU hyper-thread, and not a full CPU core. When sizing cloud instances, you will need twice as many vCPU as the Tableau Server CPU core requirements given (8 vCPU required for a minimum trial installation, 16 vCPU recommended for a single-node installation).

Operating system requirements

The following distributions of Linux are supported:

- Red Hat Enterprise Linux (RHEL) 7.3 and higher (not 8.x), and Amazon Linux 2
- CentOS 7.3 and higher (not 8.x)
- Oracle Linux 7.3 and higher (not 8.x)

Tableau Server on Linux Administrator Guide

- The latest versions of Ubuntu 16.04 LTS and 18.04 LTS (not 17.04)

Note: As of July 2022, Debian distributions are no longer supported. For more information, see [this Tableau Community post](#).

Additional notes on Linux distributions:

- Red Hat Enterprise Linux (RHEL), CentOS, Oracle Linux, and Amazon Linux distributions are collectively referred to in this documentation as RHEL-like.
- RHEL 8 is not supported.
- As of July 2022, Debian distributions are no longer supported. For more information, see [this Tableau Community post](#).
- Non-LTS releases of Ubuntu are not supported.
- Ubuntu version 17.04 is not supported.
- Previous versions of CentOS and Ubuntu are not supported because Tableau Server requires `systemd` for process management.
- The version of the installer with the file suffix, `.deb`, installs on both Ubuntu and Debian distributions.
- Custom kernels are not supported.

In a multi-node installation of Tableau Server, all of the computer nodes where you are installing Tableau must run Linux and the same distribution of Linux.

Installation directory

The core Tableau Server bits are installed in the `/opt` directory by default.

- The directory where you install Tableau Server must have at least 15 GB of free disk space allocated to it. If you attempt to install Tableau Server on a computer that does not have enough space, the Tableau Server package will install, but you will be unable to continue with setup.

- You can specify a non-default install location on RHEL-like systems, but cannot change the location on Ubuntu.
- Do not specify a symbolic link or a directory location on a Network File System (NFS) volume when specifying a non-default install location on RHEL-like systems.

Data directory

By default, Tableau Server will create a data directory for all content and extracts that are managed by Tableau. The directory is created at `/var/opt/tableau/tableau_server`.

You can specify a different directory for data (extract) storage during installation. If you plan to use a different directory, do not create the directory. Instead, let Tableau Server setup create the directory. The data directory requires specific permissions that are set during the installation process.

To change the data directory, you must pass a flag along with the data directory path when you run the `initialize-tsm` script. See Help Output for `initialize-tsm` Script.

If you are changing the default data directory:

- Do not specify a symbolic link or a data directory location on a Network File System (NFS) volume.
- Do not specify a data directory location with a path that includes a period or space. If there is a period or space in the path, initialization will fail.
- The data directory must be installed into a different directory than the installation directory.

Important: You cannot change the data directory location after you've run `initialize-tsm`. The data directory location will persist for the life of the deployment, including subsequent upgrades.

Tableau Prep Conductor

Tableau Prep Conductor is one of the process on Tableau Server. It runs a flow, checks connection credentials, and sends alerts if a flow fails. Tableau Prep Conductor leverages the

scheduling and tracking functionality of Tableau Server so you can automate running flows to update the flow output instead of logging into Tableau Prep Builder to manually run individual flows as your data changes.

Tableau Prep Conductor is licensed separately and is available through the Data Management Add-on license. For more information on Tableau Prep Conductor licensing, see [License the Data Management Add-on](#).

We recommend you enable Tableau Prep Conductor on a dedicated node. For more information:

- If you are installing a new Tableau Server, see [Step 1 \(New Install\): Install Tableau Server with Tableau Prep Conductor](#).
- If you are adding Tableau Prep Conductor to an existing installation of Tableau Server, see [Step 1 \(Existing Install\): Enable Tableau Prep Conductor](#).

Additional requirements

Make sure that your environment also meets the following additional requirements:

Hostname

- Tableau Server must be able to resolve the hostname to an IP address either using the domain name server (DNS) or with a local host file on the computer running Tableau Server. By default, host files are stored at `/etc/hosts`.
- The hostname of the server must not change after you start Tableau Services Manager during the setup process. For example, this might happen if you use the cloud-init package to initialize a virtual machine, and you install Tableau Server on that virtual machine.
- Hostnames that include underscores (`_`) are not supported by Tableau Server.

Static IP address

The computer where you install Tableau Server must have a static IPv4 or IPv6 address.

Database drivers

To connect to specific data sources, the computer where you install Tableau Server must have the correct database drivers installed. For more information, see Database Drivers.

Available ports

TSM and Tableau Server each require an available TCP port in order for you to access them. TSM defaults to port 8850, and the Tableau Server Gateway service defaults to port 80. We strongly recommend that you ensure that both port 8850 and 80 are not in use on your system before installing Tableau Server. If those ports are not available, the TSM and gateway ports may be dynamically remapped to different port numbers, and there is currently no interface for displaying which port they have been remapped to.

See Tableau Services Manager Ports.

Local firewall configuration

If you are running a firewall on the computer where you will be installing Tableau Server, then you will need to open the following default ports for Tableau Server traffic. All port numbers, except 443 can be changed.

Port	TCP/UDP	Used by ...	TYPE OF INSTALLATION	
			All	Distributed / High Availability
80	TCP	Gateway	X	
443	TCP	SSL. When Tableau Server is configured for SSL, the application server redirects requests to this port. Do not change this port.	X	
8850	TCP	Tableau Services Manager.	X	
8060	TCP	PostgreSQL database.	X	

Port	TCP/UDP	Used by ...	TYPE OF INSTALLATION	
			All	Distributed / High Availability
8061	TCP	PostgreSQL backup verification port	X	
8000-9000	TCP	Range of ports reserved by default for dynamic mapping of Tableau processes		X
27000-27009	TCP	Range of ports used by Tableau Server for License service. This range must be open on the node running the License service and accessible from other nodes. By default, the initial node runs the License service.	X	

See Tableau Services Manager Ports and Configure Local Firewall.

System user and groups

Tableau Server on Linux uses one unprivileged user, and two groups for proper operation. Tableau will create the default account and groups during setup. Alternatively, you can specify existing accounts. See System user and groups and TSM authorization group.

Sudo and root access

All installation tasks and administrative tasks for Tableau Server must be run as root. Often this is accomplished using the sudo command, but running the commands directly as the root user is also possible.

To install Tableau Server with the root account, you must specify a user account during installation. The account will be used for managing TSM. Specify the account by running the initialize-tsm script with the -a option. See Help Output for initialize-tsm Script.

Account password

The user account that you use to install and administer Tableau Server must be able to authenticate with a password. That is, the user must not use another means of authenticating (for example public key authentication).

If the account you are using to install and initialize Tableau Server does not have a password, you can set one using the `passwd` command:

```
sudo passwd $USER
```

Port access requirements

If you want to install Tableau Server remotely, for example by means of SSH, ensure that the following ports are open:

- 8850. The port used for the Tableau Services Manager (TSM) web interface. You can use this interface to configure Tableau Server.
- 80. The port used for the main Tableau Server web interface.

The Tableau Server installer attempts to open these ports during the installation process, but it can only open these ports for the `firewalld` firewall. If your computer runs another firewall, you must open the ports before you install.

Virtual Container environments

Beginning with version 2021.2, certain configurations of Tableau Server on Linux can be run in a container. For details on supported configurations, see [Tableau Server in a Container](#).

Previous versions of Tableau Server on Linux and unsupported configurations have not been tested and are not supported in virtual container environments such as Docker. In these cases, Tableau Server on Linux will not function as expected if installed in these environments.

Package requirements

Systemd

Tableau Server requires `systemd` to manage services. This package is installed by default on CentOS 7 and Ubuntu 16. If you decide to test Tableau Server on a modified version of these distributions, you can run the following command to confirm that `systemd` is installed:

```
whereis systemd
```

If `systemd` is installed, the installation location is displayed. For example, you might see the following output:

```
systemd: /usr/lib/systemd /etc/systemd /usr-  
r/share/systemd /usr/share/man/man1/systemd.1.gz
```

If you have `systemd` installed but the Tableau installer is failing requirements checks for `systemd`, it's likely that `systemd` is not running. To verify that `systemd` is running, run the following command:

```
ls /run/systemd
```

The output will be a list of files and directories. If `systemd` is running, the output will include `system`. If `system` is not in the output, then `systemd` is not running.

Continue to the next step: [Install and Configure Tableau Server](#).

Minimum Hardware Requirements and Recommendations for Tableau Server

The following minimum hardware requirements and recommendations apply to all computers running Tableau Server, including physical hardware and virtual machines (VMs):

- **Minimum requirements for testing and prototyping** are the minimum hardware your computer must have in order to install Tableau Server. If your computer does not meet these requirements, the Setup program will not install Tableau Server. These requirements are appropriate for testing and prototyping, but are not appropriate for most production environments.
- **Minimum recommendations for production** go beyond minimum requirements, and represent the minimum hardware configuration you should use for a production installation of Tableau Server. If your computer meets the minimum requirements but does not meet these recommendations, the Setup program will warn you but you can continue the installation.

In addition, Tableau Server should not be installed on a physical computer or on a VM instance that is also running resource-intensive applications such as databases or application servers.

Note: If you install Tableau Server on a computer that meets the minimum requirements but does not have at least 8 cores and 16 GB of system memory, the default number of all processes installed is reduced to one of each process by design. For more information about processes, see [Server Process Limits](#)

Looking for Tableau Server on Windows? See [Minimum Hardware Requirements and Recommendations for Tableau Server](#).

Minimum Hardware Requirements for Testing and Prototyping

The computer on which you are installing or upgrading Tableau Server must meet the minimum hardware requirements. If the Setup program determines that your computer does not meet the following requirements, you will not be able to install Tableau Server.

Tableau Server on Linux Administrator Guide

These minimum requirements are appropriate for a computer that you use for prototyping and testing of Tableau Server. They apply to single-node installations and to each computer in a distributed installation.

Minimum Hardware Requirements	Processor	CPU	RAM	Free Disk Space
Note: These minimum requirements are not recommended for use in production environments. For production minimum recommendations, see Minimum Hardware Recommendations.	<ul style="list-style-type: none">• 64-bit• Must support SSE4.2 and POPCNT instruction sets• ARM-based processors are not supported	4- core	16 GB	15 GB

- Free disk space is calculated after the Tableau Server Setup program is unzipped. The Setup program uses about 1 GB of space. You may need to allocate additional disk space depending on various factors like whether you will be using extracts.

The core Tableau Server bits must be installed in a directory with at least 15 GB of free disk space. If you attempt to install Tableau Server on a computer that does not have enough space, the Tableau Server package will install, but you will be unable to continue with setup. By default the install location is the `/opt` directory. You can change the installation path for Tableau Server on RHEL distros.

If you plan to make heavy use of extracts then you may need to allocate additional disk space. You can specify a different directory for data (extract) storage during installation.

- Core count is based on "physical" cores. Physical cores can represent actual server

hardware or cores on a virtual machine (VM). Hyper-threading is ignored for the purposes of counting cores.

Minimum Hardware Recommendations

For production use, the computer on which you install or upgrade Tableau Server should meet or exceed the minimum hardware recommendations. These recommendations are general. Actual system needs for Tableau Server installations can vary based on many factors, including number of users and the number and size of extracts. If the Setup program determines that your computer does not meet the following recommendations, you will get a warning, but you can continue with the setup process. The minimum recommendations listed here are intended as general guidance. However the recommendations for your environment may vary. For more information, see the [Hardware recommendations section](#) of the [Recommended Baseline Configurations](#) topic.

<i>Install Type</i>	<i>Processor</i>	<i>CPU</i>	<i>RAM</i>	<i>Free Disk Space</i>
Single node	<ul style="list-style-type: none"> 64-bit Must support SSE4.2 and POPCNT instruction sets ARM-based processors are not supported 	8-core, 2.0 GHz or higher	32 GB	50 GB

If you are adding Tableau Prep Conductor to your Tableau Server installation, we recommend you add a second node and dedicate this to running Tableau Server Prep Conductor. This node should have a minimum of 4 cores (8 vCPUs), and 16 GB of RAM.

<i>Install Type</i>	<i>Processor</i>	<i>CPU</i>	<i>RAM</i>	<i>Free Disk Space</i>
Multi-node and enterprise deployments	Contact Tableau for technical guidance.			Nodes must meet or exceed the minimum hardware recommendations, except: <ul style="list-style-type: none"> • Nodes running backgrounder, where 4 cores may be acceptable. • Dedicated node for Tableau Prep Conductor: Minimum of 4 cores (8 vCPUs), and 16 GB of RAM.

Important: The disk space requirement cannot be checked until you initialize TSM. If you don't have enough space, you won't be told this until after you install the Tableau Server package.

50 GB disk space available, with a minimum of 15 GB allocated to the `/opt` directory, and the remainder allocated to the `/var` directory for data storage.

- Free disk space is calculated after the Tableau Server Setup program is unzipped. The Setup program uses about 1 GB of space. You may need to allocate additional disk space depending on various factors like whether you will be using extracts.

The core Tableau Server bits must be installed in a directory with at least 15 GB of free disk space. If you attempt to install Tableau Server on a computer that does not have enough space, the Tableau Server package will install, but you will be unable to continue with setup. By default the install location is the `/opt` directory. You can change the installation path for Tableau Server on RHEL distros.

If you plan to make heavy use of extracts then you may need to allocate additional disk space. You can specify a different directory for data (extract) storage during installation.

- **Network attached storage space requirements for External File Store:** If you

are planning to configure [Tableau Server with External File Store](#), you will need to estimate the amount of storage space to dedicate on your network attached storage.

Estimating the storage size: You must take into account the amount of storage needed for publishing and refreshing extracts. In addition, you must also take into account the repository backup size unless you specifically choose the option to do your repository backup separately as described in the [Option 2: Backup Repository Separately](#) topic.

- Extracts:
 - Consider the number of extracts that will be published to Tableau Server and the size of each extract. Test your needs by publishing several extracts to Tableau Server, and then checking the disk space used. You can use this amount of disk space to help you figure out how many extracts will be published to Tableau Server over time as well as how each existing extract will increase in size.
 - Consider the space needed by the temp directory during an extract refresh. The temp directory, which is where an extract is stored to during a refresh, may require up to three times the final file size of the extract.
- Repository Backup:
 - To obtain an estimate of the repository data, check the size of `<data directory>/pgsql/data/base` directory.
 - To obtain the exact size of the repository data, open the backup file and use the size of the `workgroup.pg_dump` file.
- Core count is based on "physical" cores. Physical cores can represent actual server hardware or cores on a virtual machine (VM). Hyper-threading is ignored for the purposes of counting cores.
- RAM shown is the minimum recommended for a single-node installation. Your installation may function better with more RAM, depending on activity, number of users, and background jobs, for example.

For hardware recommendations for Tableau Server in the cloud, see the following:

Tableau Server on Linux Administrator Guide

- Selecting an AWS Instance Type and Size in the [Tableau Server on Linux in the AWS Cloud Administrator Guide](#)
- Selecting a Google Compute Engine Virtual Machine Type and Size in the [Tableau Server on Linux in the Google Cloud Platform Administrator Guide](#)
- Selecting a Microsoft Azure Virtual Machine Type and Size in the [Tableau Server on Linux in Microsoft Azure Administrator Guide](#)

Install and Initialize TSM

This topic guides you through the process of installing Tableau Server and initializing Tableau Services Manager (TSM).

Important: Do not install a beta version of Tableau Server in your production environment. You should also never restore a production Tableau Server installation using a backup of a beta version.

Prerequisites

Before proceeding, review the topic, [Before you install...](#)

Review optional initialization parameters

Before you install and initialize TSM, it is critical that you review the parameters that you may optionally set as part of the initialization operation. You can only run `initialize-tsm` once, so be sure to run it with all of the options that your organization needs. Some options, such as non-default system user and group can only be configured as part of the initialization operation. Other configurations, such as forward proxy and dynamic port settings can be manually set after you run initialization, but doing so is a much more labor-intensive process.

For a complete list of optional parameters, see [Help Output for initialize-tsm Script](#).

Some common scenarios where optional initialization parameters are used :

- Configuring Tableau Server to work with a forward proxy server. See [Optional: common initialize-tsm parameters](#), later in this topic, to configure Tableau Server during installation. You can also configure Tableau Server after you install, see [Configuring Tableau Server on Linux to work with a forward proxy](#)
- Specifying dynamic port mapping. By default, most ports needed by Tableau Server are assigned (mapped) dynamically from a predefined range of ports. The port assignments are made for each service or process during installation. If you want to modify port mapping, we recommend configuring this during installation, see [Controlling port remapping with initialize-tsm](#).
- Specifying non-default system user or group This configuration change can only be performed during initialization. See [System user and groups](#).
- Specifying a non-default data directory. This configuration change can only be performed during initialization. See [Data directory](#).

Install Tableau Server

Install Tableau Server with your distribution's package manager, then run a script to initialize TSM. The script is included with the installed package.

Important: The hostname of the server must not change after you start TSM. For example, this might happen if you use the cloud-init package to initialize a virtual machine, and you install Tableau Server on that virtual machine.

Install the Tableau Server package

By default, Tableau Server is installed in the `/opt` directory. On RHEL-like distributions you can specify a different install location.

1. Log on as a user with `sudo` access to the computer where you want to install Tableau Server.

Note: To avoid possible complications, we recommend a user account that does not include any special characters (for example, non-ASCII, "+", "-"). These may cause problems, including a failure to fully install Tableau Server, depending on how your environment is configured.

2. Download the `.rpm` or `.deb` installer package from the [Tableau Server Downloads and Release Notes](#) page.
3. Navigate to the directory where you copied the `.rpm` or `.deb` package.
4. Use the package manager to install the Tableau Server package.

Do not install to a location using a symbolic link or to a directory on a Network File System (NFS) volume.

- On RHEL-like distributions, including CentOS, you have the option to install Tableau to a non-default location.
 - **Default location**—To install to the default location (`/opt/tableau/tableau_server`), run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: 2019-2-5):

```
sudo yum update

sudo yum install tableau-server-<version>.x86_64.rpm
```
 - **Non-default location**—To install to a non-default location, you must use `rpm -i`. You will also need to install all dependent packages. See the note below.

Run the following command:

```
sudo rpm -i --prefix /preferred/install/path
tableau-server.rpm
```

Note: When you use `yum` to install Tableau Server, all dependent packages are automatically downloaded and installed. This is the preferred method for installing Tableau. If you want to install to a non-default location, or your organization does not allow you to use `yum` and you must install using `rpm -i`, you must also install all dependent packages separately. For information about installing dependent packages, see [Installing Tableau Server on an Air-Gapped Computer Running Linux](#).

- On Ubuntu, run the following commands, where `<version>` is formatted as `major-minor-maintenance` (ex: `2019-2-5`):

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get -y install gdebi-core
sudo gdebi -n tableau-server-<version>_amd64.deb
```

Initialize TSM

You can specify a separate location for data (extracts and extract metadata) storage, whether or not you install Tableau to the default location.

1. Navigate to the `scripts` directory:

```
cd /opt/tableau/tableau_server/packages/scripts.<version_
code>/
```

2. Run the following script to start TSM:

```
sudo ./initialize-tsm --accepteula --<optional_parameters>
```

The only required parameter for the `initialize-tsm` script is `--accepteula`. You must include this parameter to accept the Tableau Server End User License Agreement (EULA). The EULA is available in the following location:

```
/opt/tableau/tableau_server/packages/docs.<version_
code>/Commercial_EULA.txt
```

However, review the following common parameters before running the script.

Optional: common initialize-tsm parameters

There are a number of parameters (also referred to as, *flags*) that you can set when you run `initialize-tsm`. Common flags are listed below. For a complete list, run the command, `sudo ./initialize-tsm -h`, or see Help Output for `initialize-tsm` Script.

- The `-a` flag to specify a user to be added to the `tsmadmin` and `tableau` groups instead of the user running this script. If you are installing with the root account, the you must specify the `-a` flag,
- The `--unprivileged-user` flag to set a different service account. By default a new user called `tableau` will be created. This account is an unprivileged service account under which most Tableau process run. We recommend creating a different user only in the case where a `tableau` user account already exists on the computer.
- The `-d` flag to specify a non-default location of the "data directory" where Tableau Server stores extracts, information about extracts, and more.

By default, Tableau Server uses the following location for the directory:

```
/var/opt/tableau/tableau_server
```

When you set this flag, the `initialize-tsm` script will create and apply permissions to the directory that you specify. There are important restrictions that apply to changing the default directory path. See [Data directory](#).

- The `--debug` flag for troubleshooting. This option displays each command in the script as it is run and can make it easier to troubleshoot issues. Use of this option results in extensive output to the screen.
- We recommend configuring Tableau Server for a forward proxy solution during installation.

To do so, include the `--http_proxy` and/or `--https_proxy` flags to specify the forward proxy server. Specify the URL with the port, for example:

```
--http_proxy=http://proxy.example.lan:80/
--https_proxy=http://1.2.3.4:443/
```

Take care to use `http` when you specify the URL for the `https_proxy` environmental variable. Do not specify the `https` protocol for the value of the `https_proxy` environmental variable.

To configure Tableau Server to bypass the forward proxy, include the `--no_proxy` flag. You should also add exceptions to this proxy configuration to guarantee that all communications within a local Tableau Server cluster (if you have one now or will have one later) do not route to the proxy server. For example:

```
--no_proxy-
y=localhost,127.0.0.1,localaddress,.localdomain.com.
```

- If you want to manually manage port assignment for TSM and Tableau Server processes, you may need to use one or more port-related switches with `initialize-tsm`. For more information, see [Controlling port remapping with initialize-tsm](#).
- If you want to use the Tableau authorization-to-run (ATR) service to activate

Tableau Server, use `--activation-service`. This option is for new installs only, not upgrades. This option is ideal for cloud-based or virtual environments, but is available to anyone who can activate their copy of Tableau Server online. Selecting ATR for product activation is a permanent choice that cannot be undone later. For more information, see [Activate Tableau Server using the authorization-to-run \(ATR\) service](#).

3. Log off and log on again to the terminal before you configure Tableau Server.

When you log on again, you create a new session in which group membership changes have taken effect. The new session also has access to the environment variables added by the `initialize-tsm` script.

Next step

- [Activate and Register Tableau Server](#)

Activate and Register Tableau Server

This topic describes how to activate and register Tableau Server.

Tableau Server requires at least one product key that both activates the server and specifies the number of license levels you can assign to users. You can access your product keys from the [Customer Portal](#).

If you need to activate Tableau Server on a computer that is offline, see [Activate Tableau Server Offline - Adding a License](#). If you need to activate additional product keys to add capacity to an existing Tableau Server installation, see [Add Capacity to Tableau Server](#).

Prerequisite

Before proceeding with the procedures in this topic, complete the following prerequisites as outlined in [Install and Configure Tableau Server](#):

- [Install and Initialize TSM](#)
- TSM uses port 8850. If you are running a local firewall, open port 8850. See [Configure](#)

Local Firewall.

Use the TSM web interface

1. Sign in to Tableau Services Manager Web UI.

The account you use must have administrative privileges on the computer where TSM is installed.

2. On the **Activate** page, Enter or paste your product key and click **Activate Product Key**.

After your product key is activated, it appears under **Activated Product Keys**.

Activate Register Setup Initialize

Enter your license product key to get started with Tableau Server.

Product Key
The key has 20 characters
0000-0000-0000-0000-000
[I can't find my product key.](#)

Tableau Server requires at least one product key that activates the server and specifies the number of license levels you can assign to users. You can access your product keys from the [Tableau Customer Portal](#).

Activated Product Keys
No product key currently activated

Activate Product Key Next

Try it free for 14 days
[Start Tableau Server Trial](#)

3. To activate another product key, enter the new product key, and then click **Activate Product Key**. After your product key is activated, it appears under **Activated Product Keys**.
4. When you're finished activating product keys, click **Next**.

Note: If the product keys you have activated don't have the necessary capacity, such as not enough cores or only a Viewer role product key, Tableau Services

Manager displays the **Insufficient licenses applied** dialog box. Click **Activate Another Product Key** to add another product key and increase capacity on your Tableau Server installation.

Insufficient licenses applied

You have not activated enough product keys for your Tableau Server deployment.

If you try to activate Tableau Server using these licenses, it may not run properly. We recommend that you activate additional licenses to support this Tableau Server deployment

[Tableau Customer Portal Troubleshooting](#)

Activate Another Product Key

5. On the **Register** page, enter your information into the fields and click **Register**.

● ——— ○ ——— ○ ——— ○
Activate Register Setup Initialize

Register with Tableau. All fields are required.

Contact Information

First Name _____ Last Name _____

Phone Number _____ Email _____

Company Information

Organization _____ Industry _____ ▼

Department _____ ▼ Job Role _____ ▼

Region Information

City _____ Postal Code _____

Country/Region _____ ▼ State/Province _____ ▼

Register

Use the TSM CLI

Activate Tableau Server

To activate Tableau Server for production use, you must have a product key.

To activate a product key, run the following command:

```
tsm licenses activate -k <product key>
```

To activate a two-week trial, run the following command:

```
tsm licenses activate -t
```

If you are unable to activate Tableau, for example, if you get an error like this:

```
License Server not available
```

see [Activate Tableau Server Offline - Adding a License](#).

What if I get an access denied error when I attempt to run TSM commands?

The account that you use to configure the rest of the installation must be a member of the `tsmadmin` group that was created during initialization. To view the user accounts in the `tsmadmin` group, run the following command:

```
grep tsmadmin /etc/group
```

If the user account is not in the group, run the following command to add the user to the `tsmadmin` group:

```
sudo usermod -G tsmadmin -a <username>
```

Register Tableau Server

After activation, you need to register Tableau Server. To do this, create a registration file and then pass it as an option with the `tsm register` command.

1. Generate a template that you can edit by running the following command:

```
tsm register --template > /path/to/<registration_
file>.json
```

2. Edit the template file to create your completed registration file.

Here is an example of a registration file in the required format:

```
{
  "zip" : "97403",
  "country" : "USA",
  "city" : "Springfield",
  "last_name" : "Simpson",
  "industry" : "Energy",
  "eula" : "yes",
  "title" : "Safety Inspection Engineer",
  "phone" : "5558675309",
  "company" : "Example",
  "state" : "OR",
  "department" : "Engineering",
  "first_name" : "Homer",
  "email" : "homer@example.com"
}
```

3. After saving changes to the file, pass it with the `--file` option to register Tableau Server:

```
tsm register --file /path/to/<registration_file>.json
```

For example:

```
tsm register --file /usr/share/tableau-reg-file.json
```

If you have a product key for Add-on license, you must activate that product key to be able to use the features included in each Add-on. Product keys for add-on licenses should only be activated after at least one capacity product key has been activated on the server.

- For license information on Tableau Data Management see, [License the Data Management Add-on](#).
- For license information on Tableau Advanced Management, see [About Tableau Advanced Management](#).

Next step

- [Configure Initial Node Settings](#)

Activate Tableau Server Using the Authorization-To-Run (ATR) Service

You can use the Server authorization-to-run (ATR) service to activate Tableau Server deployed in cloud or virtual environments without running out of license activations. The ATR service achieves this by providing short-term leases of configurable duration (ATR duration) that ties the license to the device until product key expiration date is met. ATR abstracts Tableau licensing from underlying hardware changes, which is common on virtual environments. This option is ideal for cloud-based or virtual environments, but is available to anyone who can activate their copy of Tableau Server online because Tableau Server needs to connect to various internet locations for licensing purposes. For more information, see [Communicating with the Internet](#).

Select this option if:

- You plan to activate Tableau Server online
- You plan to install Tableau Server in the cloud
- You plan to install Tableau Server on a virtual machine or other non-persistent environment

Do not use this option if:

- You plan to activate Tableau Server offline
- You plan to install Tableau Server in an air-gapped environment

Selecting ATR for product activation is a permanent choice that cannot be easily undone. To change this later, you must backup, remove, and then reinstall Tableau Server.

Note: When using Server ATR with subscription licenses, you cannot refresh product keys in Tableau Server. To renew a subscription or make licensing changes, you must identify the new product key in the Tableau Customer Portal and activate it.

Select the Server ATR option

You can select the Server authorization-to-run (ATR) option when installing Tableau Services Manager (TSM) using the TSM web interface or the TSM CLI. For more information, see [Install TSM](#).

Set or change the Server ATR duration

When using Server authorization-to-run (ATR) to activate Tableau Server, you can change the ATR duration and use a setting different than the default of 432000 seconds (5 days). The ATR duration is the length of time that Tableau Server is authorized to run and will refresh while Tableau Services Manager (TSM) processes are up, requiring no new activation until the key(s) have reached their expiration date. If you're creating a test server or virtual machine (VM) that will have a short lifespan, you might want to shorten the ATR duration. Similarly, if you have a server that will remain for a long time, you might want to lengthen the ATR duration. The license refresh duration varies from hourly to daily, depending on the ATR duration. You should keep your Tableau Server running for at least 50 percent of the ATR duration. Otherwise if the Tableau Server is left shut down too long, you may need to reapply your license.

To change the ATR duration, you use the `tsm licenses atr-configuration set -duration <value_in_seconds>` command. For more information, see [tsm licenses atr-configuration set](#).

ATR duration in seconds	Minimum	Maximum	Default
Tableau Server (container)	3600 (1 hour)	2593000 (30 days)	14400 (4 hours)
Tableau Server (non-con-	14400 (4	7776000 (90	432000 (5

tainer)	hours)	days)	days)
---------	--------	-------	-------

To set ATR duration and manually start the Administration Controller Manager, run the following commands:

1. On the initial node, open a terminal session, and they use the following commands:
2. `tsm licenses atr-configuration set --duration <value_in_seconds>`
3. `tsm pending-changes apply`
4. `sudo su -l tableau`
5. `systemctl --user stop tabadmincontroller_0`
6. `systemctl --user start tabadmincontroller_0`
7. Verify that the controller is running by using the `tsm status -v` command. Tableau Server Administration Controller should be listed as "is running". If the Administration Controller Manager is not started, then run:

```
systemctl --user restart tabadmincontroller_0
```

Move a Server ATR license to another Tableau Server

If you want to stop using your license on an existing Tableau Server and use it on a new installation of Tableau Server, you can move your Server ATR license. You might want to move licenses when:

- Switching between development or pre-production environments.
- Moving Tableau Server to upgraded hardware.
- Moving Tableau Server to cloud infrastructure.
- Using an ephemeral Tableau Server, such as a virtual machine (VM) or container.

How do ATR leases work?

When you sign into Tableau Server and activate it using your product key, your ATR lease remains active as long as the Administration Controller, also called the TSM Controller, and the Activation Service are running. If you stop using Tableau Server, such as when you shut down the computer where Tableau Server is installed, you are automatically logged out and the ATR lease expires. Tableau reclaims the license and frees the capacity for new use after the ATR lease duration has elapsed.

Activate the same license on another Tableau Server

You can have one production and two non-production installations per license. If you have unused activations on a license, you can activate the same license on another Tableau Server. If you're out of activations, you can still activate the license after its lease expires.

1. You have an existing Tableau Server that is licensed using Server ATR.
2. Stop the existing Tableau Server from refreshing its leased activation. You can do this using any of the following methods:
 - Shut down your existing Tableau Server.
 - Uninstall the existing Tableau Server.
 - Disconnect the existing Tableau Server from the Internet by unplugging the ethernet cable or disconnecting wifi.
3. After the ATR lease expires, you can reuse the license on another Tableau Server.
4. On the new computer, install Tableau Server.

When prompted, activate Tableau Server using the same license.

Remove product keys that were activated using Server ATR

You can remove product key(s) that were activated using Server ATR when you want to reclaim license capacity from one Tableau Server installation and use it on another. This procedure removes all product keys activated using Server ATR on the specified Tableau

Tableau Server on Linux Administrator Guide

Server. Before you begin, view license details for Server ATR product keys by running the `atrdiag -product "Tableau Server"` command.

Note: If your Tableau Server is in a container, use the `-container` option with each `atrdiag` command. For example, `atrdiag -product "Tableau Server" -container`. For more information about Tableau Server in a container, see [Tableau Server in a Container](#).

1. On the initial node, open a terminal session, and they use the following commands:
2. Login as user "Tableau":

```
sudo su -l tableau
```

3. Stop Administration Controller Manager (`tabadmincontroller`):

```
systemctl --user stop tabadmincontroller_0
```

4. Use the following `atrdiag` command to view activated product keys and locate the **UUID**.

```
atrdiag -product "Tableau Server"
```

5. Copy the **UUID** field from the top of the ATRDIAG output to delete the ATR lease containing all activated product keys.

Note: You cannot remove an individual product key from an ATR lease.

```
atrdiag -deleteATR <UUID> -product "Tableau Server"
```

6. Start Administration Controller Manager (`tabadmincontroller`) if it isn't already running:

```
systemctl --user start tabadmincontroller_0
```

7. **Activate** a new set of product keys.

8. Restart Tableau Server.

Stop using Server ATR for activation

If you no longer want to use the Server ATR activation method for a specific Tableau Server, you must perform a backup, delete the existing Tableau Server, install a new Tableau Server, and then restore from backup.

1. Perform a full backup of your Tableau Server. For more information, see [Perform a Full Backup and Restore of Tableau Server](#).
2. Completely remove Tableau Server using the `tableau-server-obliterate` script with the `-l` option. For more information, see [Remove Tableau Server from Your Computer](#).
3. Install and configure Tableau Server. When installing TSM, on the **Product Activation** page, clear the **Use ATR for product activation** check box. For more information, see [Install and Configure Tableau Server](#).
4. Restore from backup. For more information, see [Perform a Full Backup and Restore of Tableau Server](#).

Offline Activation When Installing Tableau Server

When you install Tableau Server, you have to activate at least one product key. Doing this activates the server, and specifies the number of license levels you can assign to users. There are also times you may need to activate licenses after Tableau Server is installed, for example, if you add capacity to your server, or get a new product key. If you don't have your product key, you can get it from the [Tableau Customer Account Center](#).

In most cases, you can activate your key directly from Tableau, either during installation, or later, using the Tableau Services Manager (TSM) Licenses page, but there are some circumstances that don't allow you to do this. If your computer is not connected to the internet for example, or has a firewall that restricts access outside your intranet. In these cases you need to do an offline activation.

Tableau Server on Linux Administrator Guide

- Offline activation when adding a license—To complete an offline activation when you are adding a license to Tableau Server, see [Activate Tableau Server Offline - Adding a License](#).
- Offline activation of licenses during install—To complete an offline activation when you are installing Tableau Server, follow the steps below.

Offline activation overview and prerequisites

Offline activation of Tableau Server involves several steps. If you have never activated a product key before on your computer, you need to run through the steps twice, once to initialize licensing on the computer, and a second time to activate the key. If you have had a license on the computer in the past, you run through the steps only once.

Offline activation steps:

1. Generate an offline activation request file (`offline.tlq`).
2. Upload the offline activation request to Tableau and download the resulting activation file (`activation.tlf`).
3. Upload the activation file to Tableau Server.

If this is the first time a Tableau product key has been activated on the computer, you repeat these three steps a second time.

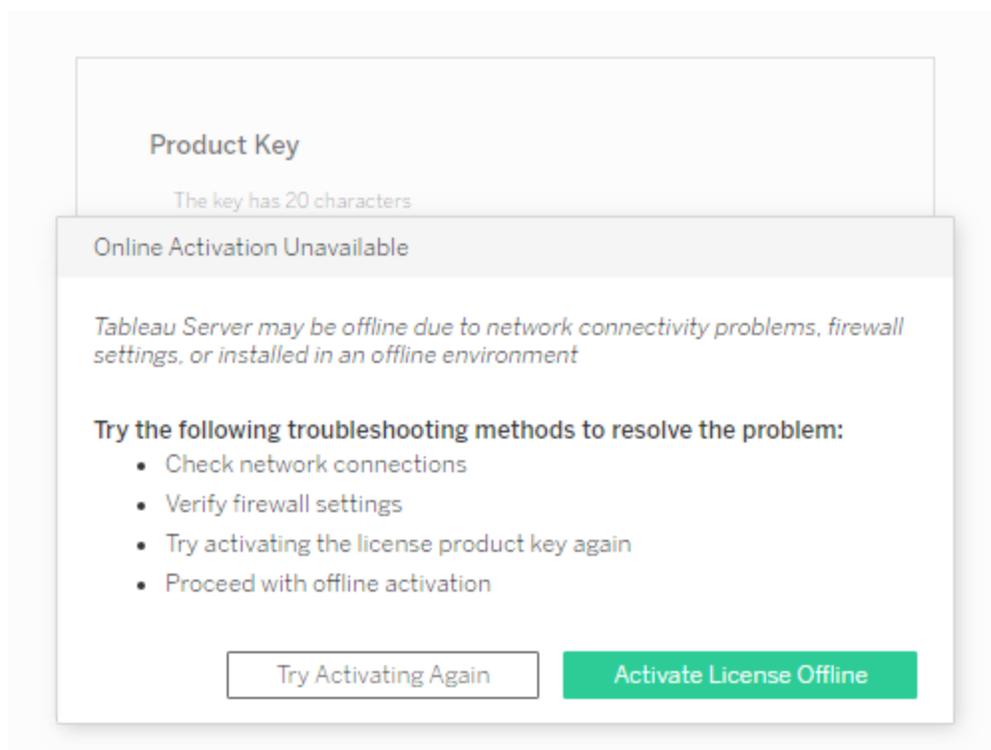
In this article

Use the TSM web interface

Use the TSM CLI

Use the TSM web interface

If you are unable to do an activation of your product key from within the TSM Setup UI, you can do an offline activation. This might be necessary if you are installing on a computer that is isolated without internet access. If you attempt to activate your product key and see a dialog that says online activation is unavailable, you can activate the key offline. Click **Activate License Offline**.



Instructions for a five-step process display.

1. Create an offline file for each product key you want to activate.

Create an offline file you will upload to Tableau. If your product key is not pre-filled in the form, enter your key and click **Create Offline File** to generate an offline file (`offline.tlq`) on the local computer.

Step 1: Create Offline File

Enter a product key then create, save, and move the `offline.tlq` file to an online computer. You will then need to upload and submit the offline file to Tableau (Step 2).

The key has 20 characters

Create Offline File

I can't find my product key.

Step 2: Upload and Submit Offline File

Leave the Tableau Server install program open and copy the `offline.tlq` file to a computer with internet access. You need to upload this file to Tableau to generate an activation file.

2. Upload and submit the offline file.

On the computer where you copied the `offline.tlq` file, open a browser and go to <http://www.tableau.com/support/activation> to open the Tableau Support Activation page. You will upload and submit the `offline.tlq` file to Tableau. This automatically generates an activation file (`activation.tlf`) that you can download and copy back to the Tableau Server computer.

- a. On the Offline Activation page, click **Choose File** to select the `offline.tlq` file.
- b. Click **Upload Activation File** to submit the file to Tableau.
- c. Click the [here](#) link to download the `activation.tlf` file to your computer.

Offline Activation

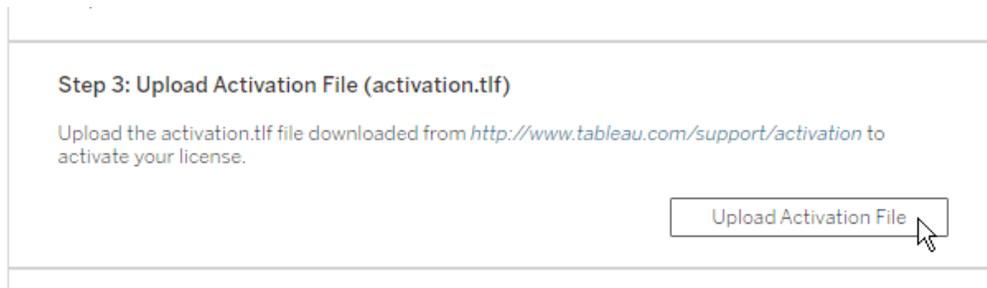
The activation was successful. Please click [here](#) to download your activation file.

For help creating the offline activation file, see [Activate Tableau Desktop Offline](#) or [Activate Tableau Server Offline. \(Linux\)](#)

- d. Copy the downloaded `activation.tlf` file to the computer where Tableau Server is installed.

3. Upload the activation file.

On the Tableau Server computer, click **Upload Activation File** to upload the file to Tableau Server.



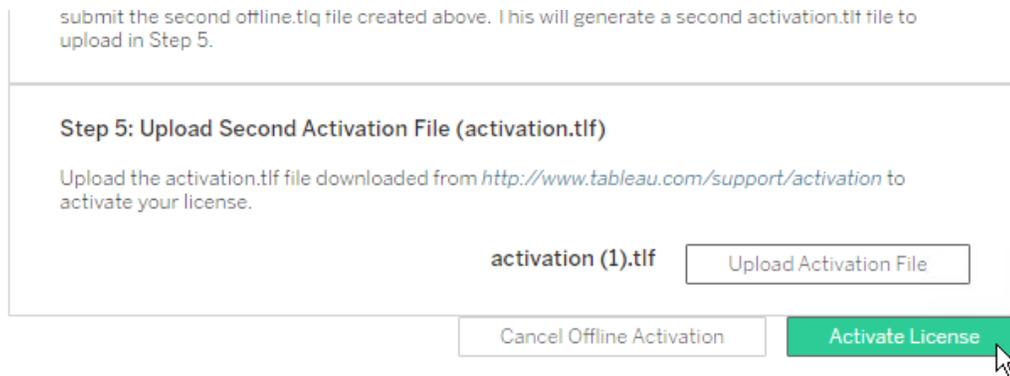
4. Create, upload, and submit a second offline file.

If this is the first Tableau license activated on the computer, then you will need to activate again.

Perform Steps 1 and 2 again: create a new `offline.tlq` file, submit it to Tableau, and download another `activation.tlf` file.

5. Upload the second activation file.

On the Tableau Server computer, upload the second `activation.tlf` file. When you do this successfully, the **Activate License** option is enabled.



6. Click **Activate License** to complete the offline activation and continue with the installation process.

Use the TSM CLI

Log in to Tableau Services Manager

Tableau Server on Linux Administrator Guide

Before you can proceed you must log in to Tableau Services Manager (TSM).

To log in to TSM, run the following command:

```
tsm login -u <username>
```

What if I can't log in?

The account that you use to configure the rest of the installation must be a member of the `tsmadmin` group that was created during initialization. To view the user accounts in the `tsmadmin` group, run the following command:

```
grep tsmadmin /etc/group
```

If the user account is not in the group, run the following command to add the user to the `tsmadmin` group:

```
sudo usermod -G tsmadmin -a <username>
```

After you have added the user to the `tsmadmin` group, run the `tsm login` command.

Step 1 - Generate an offline activation request file

1. On the initial node, open a terminal session.
2. Type this command to get your offline activation file:

```
tsm licenses get-offline-activation-file -k <product-key> -  
o <target-directory>
```

You can get your product key from the [Tableau Customer Portal](#). The target directory must already exist.

3. Copy the offline activation file (`offline.tlq`) from the target directory to a computer that has internet access.

Step 2 - Upload the offline activation request to Tableau

1. On the computer that has internet access, go to the Tableau [Product Activations](#) page.
2. Complete the instructions to upload your `offline.tlq` file.

This creates an activation file, `activation.tlf`.

3. Download the resulting activation file from Tableau.

Step 3 - Initialize or activate your license

1. Copy the activation file (`activation.tlf`) to a location accessible from your Tableau Server computer.
2. Run the following command:

```
tsm licenses activate -f <path-and-activation-file>
```

3. (Skip this step if you are installing Tableau Server for the first time.)

Restart the server for licensing changes to take effect:

```
tsm restart
```

First time license activation

If this is the first Tableau license activated on the computer, you will see this message:

```
Your license has been initialized. To complete the activation,
we need one more exchange. Generate and send to Tableau a
second activation request file.
```

Repeat steps 1 through 3 above to activate your license. You need to generate a second `offline.tlq` file, upload it to Tableau, and download the resulting `activation.tlf` file, then use that file to activate your license.

If you are not sure whether you need to repeat the steps, you can run this command:

```
tsm licenses list
```

Tableau Server on Linux Administrator Guide

If you see a message like this one you need to repeat the steps above:

```
No licenses are currently activated.
```

Subsequent license activation

If this is not the first Tableau license activated on the computer, or if you have completed the steps above two times, you should see a success message:

```
Activation successful.
```

Tableau Server is activated. If you need additional assistance, contact [Tableau Technical Support](#).

Deactivate Tableau Server Offline

If Tableau Server does not have Internet access, you can use the instructions in this topic to deactivate Tableau Server. You will need to use a second computer that is able to access the Internet to complete this activation process.

Note: These instructions require you to save a file that you will use for deactivation. You must complete these steps using a browser other than Microsoft Edge, such as Google Chrome. If you use Edge, the product key return file used for deactivation is not created. To prevent existing users from becoming unexpectedly unlicensed, do not restart Tableau Server until you have activated a new license or transferred the site role for the users.

1. On the Tableau Server initial node, create a directory to store the offline deactivation file that is created in the next step.
2. Log in to TSM using a Tableau Administrator account, and then run the following command:

```
tsm licenses get-offline-deactivation-file -k <productkey>
-o <deactivation-file-directory>
```

3. Move the `deactivate.tlq` file from the deactivation file directory that you specified in the previous command to a trusted computer that has Internet access.
4. On the trusted computer that has Internet access, open a web browser and visit the Tableau [Product Activations](#) page. Follow the instructions on that page to submit your `deactivate.tlq` file.
5. When prompted, save the product key return file (`return.tlr`) from the **Product Activations** page.
6. Move the product key return file (`return.tlr`) from the trusted computer that has Internet access to the Tableau Server initial node that runs the Licensing Server service.
7. Log in to TSM using a Tableau Administrator account, and then run the following command:

```
tsm licenses deactivate -f <path-to-license-key-return-
file>
```

Configure Initial Node Settings

This topic describes how to configure essential server settings as part of the initial Tableau Server installation process.

Prerequisite

Before proceeding with the procedures in this topic, complete the following prerequisites as outlined in [Install and Configure Tableau Server](#):

- Install and Initialize TSM
- Activate and Register Tableau Server

You may also need to configure your local firewall for Tableau Server traffic. See [Configure Local Firewall](#).

Use the TSM web interface

After you have activated and registered Tableau Server, the installation program will display the Setup page.

Note: If you need to configure Tableau Server to connect to an LDAP directory that is not Active Directory, then you must use the TSM CLI.

Identity Store

You cannot change the identity store after initializing.

Local
 Active Directory

Gateway Port

Port Number: (Default)

Product Usage Data

Disable sending usage data to Tableau

Include samples

Include sample workbooks

Initialize

Identity store settings

You must configure the identity store settings for the Tableau Server computer. The identity store manages Tableau Server accounts. You can configure the identity store to synchronize with an external directory (for example, OpenLDAP or Active Directory) or you can configure the identity store to manage and store accounts on Tableau Server. If you will be

using a single sign on solution (OpenID, SAML, Kerberos, etc) then review the following topics before configuring the identity store:

- Identity Store
- Authentication

Important: After you have configured and applied settings for the identity store, it cannot be changed.

If you select **Active Directory**, Tableau Server will populate the **Domain** and **NetBIOS** fields from the computer on which you are running Setup. In some cases, Setup may not display these attributes. For more information about how Tableau Server connects and communicates with Active Directory, see [User Management in Deployments with External Identity Stores](#).

Tableau Server requires read access to Active Directory.

You can use simple bind or GSSAPI bind to authenticate Tableau Server with Active Directory. If Tableau Server requires access to domains outside of the domain where you are installing, you will need to create duplicate bind accounts. See [Duplicate bind accounts for domain trust](#).

We recommend configuring an encrypted channel for LDAP. See [Configure Encrypted Channel to LDAP External Identity Store](#).

LDAP simple bind

Identity Store

You cannot change the identity store after initializing.

Local
 Active Directory

Domain	NetBIOS (Nickname)
<input type="text" value="example.lan"/>	<input type="text" value="example"/>

Hostname	Port
<input type="text" value="Hostname"/>	<input type="text" value="Port"/>

Specify and configure the encryption method Tableau Server will use to communicate with Active Directory. Encrypted communication (TLS/SSL) requires a valid certificate in the Tableau certificate store.

To use LDAPS, you must specify a hostname and port.

LDAP over StartTLS (encrypted channel)
 LDAPS (encrypted channel)
 LDAP (unencrypted channel)

Tableau Server requires read access to Active Directory. Specify how Tableau Server will authenticate with Active Directory.

LDAP simple bind
 LDAP GSSAPI bind

Username	Password
<input type="text" value="Username"/>	<input type="text" value="Password"/>

If you are using simple bind to authenticate with Active Directory, enter a domain account and password.

LDAP GSSAPI bind

Identity Store

You cannot change the identity store after initializing.

- Local
- Active Directory

Domain	NetBIOS (Nickname)
<input type="text" value="example.lan"/>	<input type="text" value="example"/>

Hostname	Port
<input type="text" value="main-dir"/>	<input type="text" value="636"/>

Specify and configure the encryption method Tableau Server will use to communicate with Active Directory. Encrypted communication (TLS/SSL) requires a valid certificate in the Tableau certificate store.

To use LDAPS, you must specify a hostname and port.

- LDAP over StartTLS (encrypted channel)
- LDAPS (encrypted channel)
- LDAP (unencrypted channel)

Tableau Server requires read access to Active Directory. Specify how Tableau Server will authenticate with Active Directory.

- LDAP simple bind
- LDAP GSSAPI bind

Specify a user principal name (UPN) and upload the Kerberos configuration file Tableau Server will use to authenticate to the Identity Store.

UPN

Configuration file

Specify and configure the method Tableau Server will use to authenticate to Active Directory.

- Keytab file
- Local authentication

Keytab file

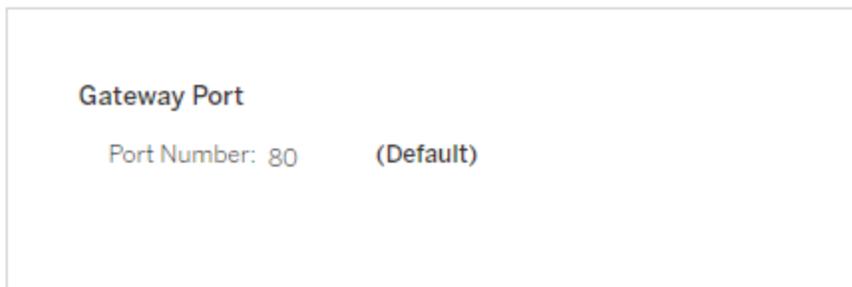
Username	Password
<input type="text" value="Username"/>	<input type="text" value="Password"/>

To bind with GSSAPI you can bind with credentials or with a keytab file. If you are using a keytab file, we recommend creating a keytab specifically for the Tableau Server service. See [Understanding Keytab Requirements](#).

Gateway port

The default port for web access to Tableau Server (via HTTP) is port 80. If the installation program determines that port 80 is in use when you first install Tableau Server, an alternate port (for example 8000) is used and shown in the Port number box.

You may need to change the port for other networking needs, for example, if you have a hardware firewall or proxy in front of the Tableau Server host, this might make running a back-end system on port 80 undesirable.

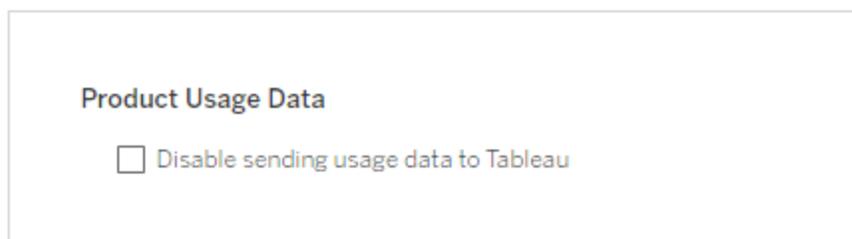


The screenshot shows a configuration box titled "Gateway Port". Below the title, it displays "Port Number: 80" followed by "(Default)" in parentheses.

Product usage data

By default, Tableau Server shares usage data with Tableau that helps us better understand how you use our products, improve your overall experience, and build highly intelligent features that make Tableau even more powerful.

Clear this option if you do not want usage data sent to Tableau.



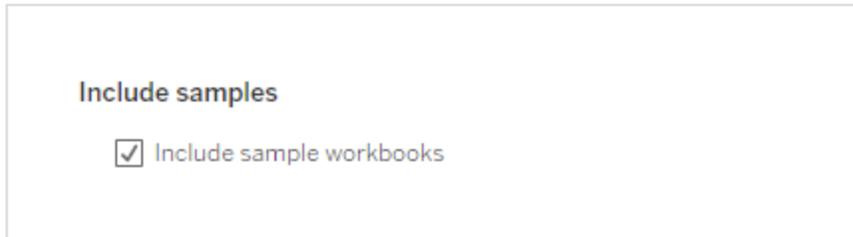
The screenshot shows a configuration box titled "Product Usage Data". Below the title, there is a checkbox followed by the text "Disable sending usage data to Tableau".

Tableau Server on Linux Administrator Guide

You can also change this setting after installation, on the TSM Maintenance tab in the TSM Web UI, or using the TSM CLI. For more information, see [Server Usage Data](#).

Sample workbook installation

By default, Tableau Server will install sample workbooks in the Default site when you initialize the server.



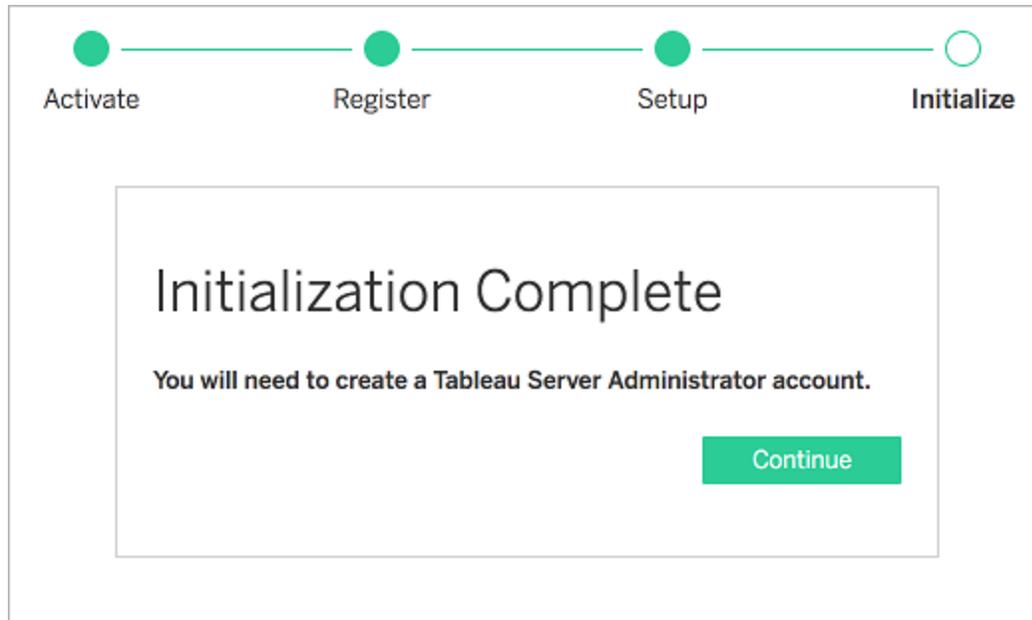
The image shows a rectangular box with a light gray border. Inside the box, the text "Include samples" is displayed in a bold, dark font. Below this text, there is a checked checkbox followed by the text "Include sample workbooks".

Alternatively, you can publish samples after installation by using the `publishsamples tabcmd` command.

Initializing install

After you have configured the options on this page, click **Initialize**.

The initialization process can take a while. When initialization is complete the following page is displayed:



Use the TSM CLI

First, configure identity store, gateway settings, and sample workbook installation. Then apply the changes, optionally verify your LDAP connection, and then initialize Tableau Server

Configure identity store settings

You must configure the identity store settings for the Tableau Server computer.

Important: After you have configured and applied settings for the identity store, it cannot be changed.

Use the json template in identityStore Entity to create a json file. After you have filled in the options with the appropriate values, you can then pass the json file with this command:

```
tsm settings import -f path-to-file.json.
```

Tableau Server on Linux Administrator Guide

Configure gateway settings (optional)

Depending on your network requirements, you may need to configure the gateway settings for the Tableau Server computer. For example, if you are enabling SSL or configuring access to Tableau Server with a reverse proxy, you may need to configure gateway settings. See `gatewaySettings` Entity for more information.

Use the json template in `gatewaySettings` Entity to create a json file. After you have filled in the options with the appropriate values, you can then pass the json file with this command:

```
tsm settings import -f path-to-json-file.json.
```

Configure product usage data (optional)

By default, Tableau Server shares usage data with Tableau to help us better understand how you use our products. This allows us to improve your overall experience and build highly intelligent features that make Tableau even more powerful.

Tableau collects only behavioral and usage data, never any of your confidential database values, and your usage data will never be shared or sold; its sole purpose is to improve your Tableau experience.

If you do not want to share product usage data, use the json template in `shareProductUsageDataSettings` Entity to create a json file, and specify a value of `false`. Then pass the json file with this command:

```
tsm settings import -f path-to-json-file.json.
```

You can also change this setting after installation, on the TSM Maintenance tab or using the TSM CLI. For more information, see [Server Usage Data](#).

Configure sample workbook installation (optional)

By default, Tableau Server will install sample workbooks in the Default site when you initialize the server.

If you do not want to install sample workbooks during installation, run the following command:

```
tsm configuration set -k install.component.samples -v false
```

You can publish samples after installation by using the `publishsamples tabcmd` command.

Apply pending configuration changes

Now that you've created and set initial configuration, you must apply them. When you apply configuration changes, `tsm` will verify the settings you've set before committing them.

To apply configuration changes to Tableau Server, run this command:

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

Once this command has completed, TSM processes are running, and Tableau Server is configured but is not running.

Verify LDAP configuration (Optional)

If your identity store uses LDAP, then we recommend verifying LDAP connectivity before proceeding.

To do so, run the following commands before you initialize the server:

```
tsm user-identity-store verify-user-mappings -v <user name>
```

```
tsm user-identity-store verify-group-mappings -v <group name>
```

Tableau Server on Linux Administrator Guide

User and group names must be valid names that exist in the LDAP server that you are connecting to. If your LDAP connection is set up correctly, then the user or group attributes will be returned to the shell. If your connection is not set up correctly, then an error will be returned.

Initialize and start Tableau Server

- To initialize and start Tableau Server, use the `--start-server` option:

```
tsm initialize --start-server --request-timeout 1800
```

This saves time by starting the server running after initialization.

- If you are going to reconfigure Tableau Server after initialization, leave the `--start-server` option off:

```
tsm initialize --request-timeout 1800
```

This stops the server after initialization.

Start Tableau Server. If you did not use the `--start-server` option during initialization and are finished configuring Tableau Server, use this command to start the server:

```
tsm start --request-timeout 900
```

Note: If you experience timeouts when installing or configuring Tableau Server, you may need to specify a longer timeout. For more information, see [Install fails due to timeouts](#).

Next Step

After initialization is complete, create the Tableau Server administrator user account. See [Add an Administrator Account](#).

Configuration File Example

This article provides an example of a complete JSON configuration file, with gatewaySettings and identityStore entities specified. In addition, a configuration key sets the gateway timeout to 900 seconds.

Your configuration file will look different depending on the options you need to set.

You might set multiple .json configuration files during installation. To set the values for each file in Tableau Server, you run the following command, once for each configuration file:

```
tsm settings import -f path-to-file.json
```

After you set the configuration files, run `tsm pending-changes apply` to apply the changes from all of the .json files you've set.

```
{
  "configEntities": {
    "gatewaySettings": {
      "_type": "gatewaySettingsType",
      "port": 80,
      "publicHost": "localhost",
      "publicPort": 80
    },
    "identityStore": {
      "_type": "identityStoreType",
      "type": "local",
      "domain": "example.lan",
      "nickname": "EXAMPLE"
    }
  },
  "configKeys": {
    "gateway.timeout": "900"
  }
}
```

Entities vs keys

As shown in the example above, there are two classes of configuration parameters: `configEntities` and `configKeys`.

configEntities

Certain types of configuration are done through entity sets that map to specific scenarios, such as the identity store and gateway configurations. When you pass a set of `configEntities` with the `tsm settings import -f path-to-file.json` command, TSM validates the configuration. If values passed are invalid, TSM will provide an error. This enables you to make changes during the configuration process, rather than experience a configuration failure at initialization or run time.

Entities can be set only by including a `configEntities` block in a `.json` file.

Important: All files that are referenced in `configEntities` must be located on the local computer. Do not specify UNC paths.

configKeys

Entities cover only a small portion of the configuration values that can be set. Hundreds of keys correspond to parameters stored in `.yaml` files. Tableau Server uses these parameters to store all of the configuration information for all services.

You can set individual keys with the `tsm configuration` command. But during deployment, setting them along with other configuration scenarios in JSON files, as shown above, is more convenient.

Unlike `configEntities`, `configKeys` are not validated.

Note: We do not recommend setting parameters that are not documented in `tsm configuration set Options`.

Server Usage Data

The Tableau Server administrator can control whether or not usage data from Tableau Server is sent to Tableau. By default this option is enabled, and can be disabled at initial install, or after installing Tableau Server, using the TSM Web UI or command line. For details about this usage data, see [Tableau Product Usage Data](#).

In addition to product usage data, Tableau products send Basic Product Data to Tableau. This data is sent whether or not you have disabled the sending of product usage data. You can disable the sending of Basic Product Data separately. For details, see [Basic Product Data](#).

Disable Usage Data Sharing

You can disable the sharing of usage data when you install Tableau Server, or at any time after installation.

Disabling the sharing of usage data at install

To disable the sharing of usage data with Tableau when you are installing Tableau Server, clear the option during the initial configuration of server. For details, see [Product usage data](#).

Disabling the sharing of usage data after install

Use the TSM web interface

1. Open TSM in a browser:

`https://<tsm-computer-name>:8850`. For more information, see [Sign in to Tableau Services Manager Web UI](#).

2. Click the **Maintenance** tab.
3. Under Other Maintenance Tasks, in Server Usage Data, clear **Send usage data to improve Tableau features**:

Product Usage Data

Help us improve your Tableau experience by sharing how you use the product. Tableau collects information about your feature usage. All usage data will be handled according to our Privacy Policy -

<http://www.tableau.com/privacy>

[Learn more](#)

Disable sending usage data to Tableau

4. When you are finished, click **Pending Changes**, and then click **Apply Changes and Restart**.

Use the TSM CLI

If you do not want to share product usage data, disable the option using this tsm configuration command:

```
tsm configuration set -k shareproductusagedata.enabled -v false  
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

Basic Product Data

By default Tableau products send usage data to Tableau so that we can understand how customers use our software, and gain insights into where they are successful and where they might run into problems that we can address. For example, this data can help us learn where

upgrades are commonly failing and allow us to make product changes to address those issues, or identify what portion of our user base needs informed about a security issue that applies to a specific version of Tableau Server. You can disable the sending of this data at installation time, or later. For details on how to do this, see the instructions for [Tableau Desktop](#) or [Tableau Server](#).

Even when you disable the sending of product usage data, certain basic product data is sent to Tableau. This Basic Product Data includes information about products and their processes, including which product or process is running, when they start up, what operating system they are running on, licensing information, which machine or cluster of machines has sent the data (using unique pseudonymized identifiers), and whether the product is configured to send product usage data.

You can disable the sending of Basic Product Data at the machine level, or at the enterprise level, by blocking traffic sent to prod.telemetry.tableausoftware.com.

Disabling sharing of Basic Product Data on individual computers

Important: This procedure involves modifying your local `hosts` file. If you do not know what this is, you should not change it. You should only make this change if you understand the implications of making changes to the file, know how to change the file, and have made a backup of the file for safety.

Modifying `hosts` files changes network behavior for computers. Detailed instructions for modifying `hosts` files are provided by operating system providers such as Microsoft, Apple or Linux Distributions.

1. Make a copy of your existing `hosts` file and save it to a computer that is not your Tableau computer. This is your backup, in case you need to reverse your changes. Do not start modifying the file until you have made a backup copy of it.
2. Modify your computer's `hosts` file to include these lines:

Tableau Server on Linux Administrator Guide

```
# Stops sending Product Usage to Tableau (prod.telemetry.tableausoftware.com) .  
# Learn more here: http:\\tableau.com\\derived-data  
127.0.0.1    prod.telemetry.tableausoftware.com
```

The first and second lines are comments, explaining the third line.

The third line prevents all traffic to `prod.telemetry.tableausoftware.com` (`http://prod.telemetry.tableausoftware.com/`) from leaving your local machine by sending it to the Internal host loopback address. The data does not get sent outside the computer.

Disabling the sharing of Basic Product Data at the enterprise level

To disable sending of Basic Product Data on an enterprise level, modify your Network Firewall to prevent outbound traffic to `prod.telemetry.tableausoftware.com`.

This domain is used by Tableau to receive the Basic Product Data about process launch and shutdown. It is also used for the more general Product Usage Data. Blocking traffic to this domain it you will prevent both kinds of data from being sent.

Traffic to this domain will occur on Ports 80 (for initial registration of our Product Data clients) and on Port 443 (for all subsequent traffic). To completely prevent product data from being sent, block all traffic to this domain.

For details on how to configure your network firewall, refer to your vendor or your internal IT department. Tableau cannot provide these instructions.

Add an Administrator Account

The final step in activating Tableau Server is to add the initial administrator account. The administrator will have all access to the server including the ability to manage users, groups, and projects.

The server must be running when you create the initial admin user.

If you have configured the Tableau Server identity store to use LDAP or Active Directory, then the initial administrative user that you specify must be an account in the directory. The initial administrative user is generally distinct from the user account on the Tableau Server computer that you use to run `t-sm`.

However, these accounts can be the same if you have configured the Tableau Server identity store to use LDAP or Active Directory and the initial admin user is a member of the `t-smadmin` group on the Tableau Server computer.

Prerequisites

Before proceeding with the procedures in this topic, complete the following prerequisites as outlined in *Install and Configure Tableau Server*:

- Install and Initialize TSM
- Activate and Register Tableau Server
- Configure Initial Node Settings

Use web UI

After Tableau Server is finished initializing, the installation program will display a page to create the Tableau Server administrator.

- If you configured a local identity store during setup, then specify a name and password that you want to use.
- If you configured a LDAP or Active Directory identity store during setup, then you must specify a user account that is a member of the directory.

If you are installing remotely, then you must sign in to TSM on the physical computer where Tableau Server is installing, or you can access the computer remotely and run the `tabcmd initialuser` command from a shell.

Use tabcmd CLI

You must create the initial administrative account for Tableau Server.

- If you configured a local identity store during setup, then specify a name and password that you want to use.
- If you configured a LDAP or Active Directory identity store during setup, then you must specify a user account that is a member of the directory.

To create the initial user, run the following `tabcmd` command:

```
tabcmd initialuser --server http://localhost --username '<new-admin-username>'
```

For example:

```
tabcmd initialuser --server http://localhost --username 'tableau-admin'
```

If you are running the HTTP protocol on a port other than 80, specify the port after the host name, for example: `--server http://localhost:8080`.

After you run the command, the shell will prompt for an administrative password.

Next steps

After you have created the Tableau Server administrator account, continue your deployment by working through the configuration topics at Post Installation Tasks.

Important: You must install the PostgreSQL driver if you want to use the built-in administrative views. You can find driver links and installation instructions for all the supported connectors on the [Driver Download page](#).

Validate Installation

To validate that Tableau Server is installed and running properly and to review the built-in administrative views, you must install the PostgreSQL driver.

Prerequisites

Before proceeding with the procedure in this topic, complete the following prerequisites as outlined in [Install and Configure Tableau Server](#):

- Install and Initialize TSM
- Activate and Register Tableau Server
- Configure Initial Node Settings
- Add an Administrator Account

Install PostgreSQL driver and validate installation

To validate installation:

1. Download PostgreSQL drivers from the [Driver Download page](#).
2. Copy the .jar file to this folder (you may have to create it manually): `/opt/tableau/tableau_driver/jdbc`.
3. Restart TSM:

```
tsm restart
```
4. To validate that the drivers installed, navigate to the Administrative Views in Tableau Server.

Initial Node Installation Defaults

By default, the Tableau Server installer configures the number of process instances that Tableau Server runs based on the hardware detected by the installer. The default

Tableau Server on Linux Administrator Guide

configuration applies to single-server installations and to the initial server of a multi-node installation.

You can calculate the default configuration based on the following rules for each process, where the "number of cores" refers to the number of logical processors:

Process Name	Number of Processes
VizQL Server	Equal to the number of cores divided by four, up to a maximum of four process instances.
Backgrounder	Set to two unless the number of cores is fewer than eight.
Cache Server	Set to two unless the number of cores is fewer than eight.
Data Server	Set to two unless the number of cores is fewer than eight.

For all other process types, the number of process instances is set to one, regardless of the hardware.

Here's an example default configuration for a computer with 16 cores:

Process Name	Number of Processes
VizQL Server	4
Application Server	1
Backgrounder	2
Cache Server	2
Data Server	2
File Store	1
Data Engine	1

Jump-start Installation

This topic provides all of the steps required to perform a basic, quick-start installation of Tableau Server using the command line. The purpose of the configuration described here is to provide the quickest, simplest path to a Tableau Server installation on a computer running the CentOS 7.3 (and higher) or Ubuntu distribution of Linux. Use this procedure as practice, to try out the Tableau Server installation, management, and user experience before your actual server deployment.

Important: Do not use this procedure as a stand-alone resource for deploying Tableau Server into a production environment. To deploy Tableau Server into a production environment, refer to the content at [Install and Configure Tableau Server](#).

The procedures in this topic will install an instance of Tableau Server for Linux with the following characteristics:

- Operating system: Supported RHEL-like Linux distribution or Ubuntu. (As of July 2022, Tableau no longer supports Debian distributions. For more information, see [this Tableau Community post](#).)
- Identity store: local authentication
- Gateway port: 80
- Tableau Server administrator account: admin

Before you begin

Review the topic, [Before you install...](#) The procedure below assumes that you have installed Linux on conforming hardware and according to the environmental requirements specified in that topic.

Note that the computer you install on must meet the minimum hardware requirements specified in [Minimum Hardware Requirements and Recommendations for Tableau Server](#). The setup program will not install Tableau Server onto systems that do not meet the minimum hardware requirements.

Step 1: Install Tableau Server package and start Tableau Services Manager

Install Tableau Server with your distribution's package manager, then run a script to initialize Tableau Services Manager (TSM). Tableau Services Manager is the management toolset used to install, configure, and manage Tableau services.

The initialize script is included with the installed package. For more details, see [Install and Initialize TSM](#).

1. Log on as a user with `sudo` access to the computer where you want to install Tableau Server.
2. Navigate to the directory where you copied the Tableau Server installation package.
3. Use the package manager to install the Tableau Server package.

Do not install to a location using a symbolic link or to a directory on a Network File System (NFS) volume.

- On RHEL-like distributions, including CentOS, run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: 2019-2-5):

```
sudo yum update
```

```
sudo yum install tableau-server-<version>.x86_64.rpm
```

Note: When you use `yum` to install Tableau Server, all dependent packages are automatically downloaded and installed. This is the preferred method for installing Tableau. If your organization does not allow you to use `yum` and you must install using `rpm -i`, you must also install all dependent packages separately.

- On Ubuntu, run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: 2019-2-5):

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get -y install gdebi-core
sudo gdebi -n tableau-server-<version>_amd64.deb
```

4. Navigate to the `scripts` directory:

```
cd /opt/tableau/tableau_server/packages/scripts.<version_
code>/
```

5. Run the following script to start TSM:

```
sudo ./initialize-tsm --accepteula
```

6. After initialization is complete, close the terminal session:

```
exit or logout
```

Step 2: Activate and register Tableau Server

Before you can configure Tableau Server you must activate a license and register. For more details, see [Activate and Register Tableau Server](#).

1. Open a Bash session and sign in with the same account that you previously used to run `initialize-tsm`.
2. Activate the Tableau Server license. Run the following command:

```
tsm licenses activate -k <license_key>
```

3. Register Tableau Server. Generate a template that you can edit by running the following command:

```
tsm register --template > /path/to/registration_file.json
```

4. Open a text editing program, fill in the registration file, save it, then pass it with the following command:

```
tsm register --file /path/to/registration_file.json
```

Step 3: Configure local identity store

You must configure the identity store settings. This procedure simplifies installation setting identity store to local authentication. Workbook examples are installed by default. For more details on customizing these defaults, see [Configure Initial Node Settings](#).

- Pass the configuration file with the following command:

```
tsm settings import -f /opt/tableau/tableau_server-  
/packages/scripts.<version_code>/config.json
```

Step 4: Finalize installation

The final steps of installation are to apply changes, initialize and start TSM, and then to create the administration account. More details about these steps are at [Configure Initial Node Settings](#).

1. Apply the configurations you made in the previous steps. Run the following command:

```
tsm pending-changes apply
```

2. Initialize and start Tableau Server. Run the following command:

```
tsm initialize --start-server --request-timeout 1800
```

3. Create the Tableau Server administrator account. Run the following command:

```
tabcmd initialuser --server 'localhost:80' --username  
'admin' --password '<password>'
```

Where '[<password>](#)' is a strong password. Enclose the password and other arguments in single quotes.

Use the `admin` account that you created to access the Tableau Server admin web pages. See Sign in to the Tableau Server Admin Area.

Step 5: Install PostgreSQL drivers

To validate that Tableau Server is installed and running properly and to review the built-in administrative views, you must install the PostgreSQL driver.

1. Download PostgreSQL drivers from the [Driver Download page](#).
2. Copy the `.jar` file to this folder (you may have to create it manually): `/opt/tableau/tableau_driver/jdbc`.
3. Restart TSM:

```
tsm restart
```
4. To validate that the drivers installed, navigate to the Administrative Views in Tableau Server.

Configure Local Firewall

This topic explains how to configure the firewall on the computer running Tableau Server.

A local firewall should be enabled on the operating system to protect Tableau Server in single and multi-node deployments. In a distributed (multi-node) installation of Tableau Server, communication between nodes does not use secure communication. Therefore, you should enable firewalls on the computers that host Tableau Server.

We recommend that you configure the firewall so that only two ports are accessible to external traffic: the `gateway` port and the `tabadmincontroller` port. By default, these are ports 80 and 8850 respectively. Additionally, if you are running in a distributed deployment, then you will need to open the port range, 27000-27009, so licensing can communicate across nodes.

The `gateway` port is used for HTTP connection to Tableau Server. We recommend that you use SSL for the `gateway` port. If you will be using SSL, then the port must be 443 because Tableau Server does not support other ports for SSL. The procedures below describe how to configure the firewall for the `gateway` port. Configure the Tableau Server gateway (Configure Initial Node Settings) to match the port you set here.

The examples below describe how to configure the firewall on single and multi-node deployments of Tableau Server running on RHEL/CentOS distributions. The examples use **Firewalld**, which is the default firewall on CentOS.

Single-node configuration

1. Open a bash shell and run the following TSM command to retrieve the port number for the `tabadmincontroller` port:

```
tsm topology list-ports
```

Make a note of the `tabadmincontroller` port. By default, this port is 8850.

2. Start `firewalld`:

```
sudo systemctl start firewalld
```

3. Verify that the default zone is a high-security zone, such as `public`. If it is not, we recommend changing it to a high-security zone.

```
sudo firewall-cmd --get-default-zone
```

```
sudo firewall-cmd --set-default-zone=public
```

4. Add ports for the `gateway` port and the `tabadmincontroller` port. In the example below we use the default ports (80 and 8850).

```
sudo firewall-cmd --permanent --add-port=80/tcp
```

```
sudo firewall-cmd --permanent --add-port=8850/tcp
```

5. Reload the firewall and verify the settings.

```
sudo firewall-cmd --reload  
  
sudo firewall-cmd --list-all
```

Multi-node cluster configuration

In addition to enabling ports, configuring the firewall on a multi-node cluster requires additional steps to ensure that nodes can communicate with each other.

Before you begin

You will need the IP address for each node in the cluster. The example here uses `<node1IP>` as a placeholder for the initial node IP address, and `<node2IP>` and `<node3IP>` as placeholders for the IP addresses of two additional nodes.

Step 1: Configure initial node.

1. Open a bash shell and run the following TSM command to retrieve the port number for the `tabadmincontroller` port:

```
tsm topology list-ports
```

Make a note of the `tabadmincontroller` port. By default, this port is 8850.

2. Run the following commands to determine the range of port numbers that TSM may dynamically select. You will specify this range later in this procedure. Make a note of the port range.

```
tsm configuration get -k ports.range.min
```

```
tsm configuration get -k ports.range.max
```

A typical range is 8000 to 9000.

3. Start `firewalld`:

```
sudo systemctl start firewalld
```

4. Verify that the default zone is a high-security zone, such as `public`. If it is not, we

Tableau Server on Linux Administrator Guide

recommend changing it to a high-security zone.

```
firewall-cmd --get-default-zone
```

```
sudo firewall-cmd --set-default-zone=public
```

5. Add ports for the `gateway` port and the `tabadmincontroller` port. In the example below we use the default ports (80 and 8850). You must also add a port range (27000-27010) to enable licensing communication between nodes.

```
sudo firewall-cmd --permanent --add-port=80/tcp
```

```
sudo firewall-cmd --permanent --add-port=8850/tcp
```

```
sudo firewall-cmd --permanent --add-port=27000-27010/tcp
```

6. Configure the firewall to allow all traffic from the other nodes in the cluster. For the `ports` option, specify the range you noted in Step 2. Run the command for each of the additional nodes in your cluster. For example:

```
sudo firewall-cmd --permanent --add-rich-rule='rule family=ipv4 source address=<node2IP>/32 port port=8000-9000 protocol=tcp accept'
```

```
sudo firewall-cmd --permanent --add-rich-rule='rule family=ipv4 source address=<node3IP>/32 port port=8000-9000 protocol=tcp accept'
```

7. Reload the firewall and verify the settings.

```
sudo firewall-cmd --reload
```

```
firewall-cmd --list-all
```

Step 2: Configure additional nodes

Each node in the cluster must be able to communicate with the initial node and with the other nodes.

Run this procedure on each additional node in the cluster. In this example, the node at IP address, <node2IP>, communicates with the initial node at <node1IP> and a third node at <node3IP>.

1. Start firewalld:

```
sudo systemctl start firewalld
```

2. Verify that the default zone is a high-security zone, such as `public`. If it is not, we recommend changing it to a high-security zone.

```
firewall-cmd --get-default-zone
```

```
sudo firewall-cmd --set-default-zone=public
```

3. Configure the firewall to allow `gateway` and `tabadmincontroller` access from the other nodes in the cluster. For example:

```
sudo firewall-cmd --permanent --add-rich-rule='rule family=ipv4 source address=<node1IP>/32 port port=80 protocol=tcp accept'
```

```
sudo firewall-cmd --permanent --add-rich-rule='rule family=ipv4 source address=<node1IP>/32 port port=8000-9000 protocol=tcp accept'
```

```
sudo firewall-cmd --permanent --add-rich-rule='rule family=ipv4 source address=<node3IP>/32 port port=80 protocol=tcp accept'
```

```
sudo firewall-cmd --permanent --add-rich-rule='rule family=ipv4 source address=<node3IP>/32 port port=8000-9000 protocol=tcp accept'
```

In this example, since the `tabadmincontroller` port (8850) is included in the port range, it is not explicitly specified in a command.

4. Reload the firewall and verify the settings.

```
sudo firewall-cmd --reload
```

```
firewall-cmd --list-all
```

Automated Installation of Tableau Server

Tableau provides an `automated-installer` script to automate an install of Tableau Server. The script is **community supported**. You can download the script and use it as written, or modify it for your specific needs.

Benefits of using the automated installer

- With a single command, you can install, configure and get to a working instance of Tableau Server.
- The command can be run without user input making it suitable for automation.
- The configuration can be set once and used for all your installations, making this a repeatable process.

When not to use the automated installer:

- If you are installing for the first time, we recommend that you manually test the installation before automating the process. Any issues that block installation are easier to resolve interactively, and after you have resolved these issues, you can use the automated installer.
- If you are testing or trying new configuration parameters such as authentication methods, we recommend that you manually run the installation first. TSM validates configuration entities and rejects configuration parameters that are not valid. Once you have the correct parameters identified, you can use the automated installer.

- If you are unable to or do not want to enter passwords into the secrets file, using the automated installer might not be an option for you.

Before you begin

Review the Before you install... topic to make sure you have installed Linux on a computer that meets the operating system requirements and the minimum hardware requirements for Tableau Server.

Note: If you are installing Tableau Server in a production environment, review the minimum hardware **recommendations**. The recommendations represent the minimum hardware configuration you should use for a production installation of Tableau Server.

To perform an automated installation, you have to use the automated installer package, which uses the Tableau Server install package as an input. We recommend that you download **both of these packages** before you begin as follows:

1. Download both the automated installer package and the Tableau Server installer package:
 - a. Download the **automated installer package** from [GitHub](#) for the distribution you are using. The automated installer packages can be found in the [packages](#) sub directory.

Note: The version of automated installer package you use must match the version of the Tableau Server installer package. For example, use 10.5.0 version of the automated installer package with 10.5.0 version of the Tableau Server installer package.

- b. Select and download the appropriate **Tableau Server installer package** from the [Tableau Server Product Downloads and Release Notes](#) page. The one you choose depends on which Linux distribution you are using. For example, for RHEL like systems, tableau-server-<version>.x86_64.rpm.
- c. Download the config.json, reg_tmpl.json, and the secrets templates.

2. Copy the packages and templates to a location on or accessible from the computer where you are going to install Tableau Server.

How to use the automated installer

The automated installer installs the Tableau Server installer package, creates the directories, sets the permissions required to run Tableau Server, and starts the Tableau Services Manager (TSM) setup. After the TSM setup is completed, the automated installer runs `tsm` commands to install, configure, and start Tableau Server. By default, during installation, the automated installer activates a trial license. If you have an actual product key, you can provide the product key at the command line or activate the product key after you run the script. Most of the command line options in the automated installer are the same as the options used by the `tsm initialize` command.

To run the automated installer without user input, you must provide the following required command line options:

Option	Description
<code>-s <secrets-file></code>	<p>The name of the secrets file. The secrets file should have the user names and passwords for TSM administrator and the Tableau Server administrator accounts.</p> <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px; margin: 10px 0;"> <p>Note: Providing the password in the secrets file is optional. However, if passwords are not found in the secrets file, you will be prompted to provide them during installation.</p> </div> <p>The automated installer package includes the template for the secrets file.</p>
<code>-f <config-file></code>	The name of the configuration JSON file. The automated installer package includes the template for the configuration file.
<code>-r <regis-</code>	The name of the registration file. The automated installer package

tration-file>	includes the template for the registration file.
--accepteula	Indicates that you have accepted the End User License Agreement.
<package-file>	The rpm or deb Tableau Server installer.

Use the -h option to see a full list of all the required command line options.

Configure Tableau Server for a forward proxy

If your organization uses a forward proxy solution to access the internet, then configure Tableau Server to use the proxy server. Tableau Server must access the internet for map data and for default licensing functionality.

We recommend configuring Tableau Server for a forward proxy solution during installation.

To configure proxy server during unattended installation, include the `--http_proxy` and/or `--https_proxy` flags to specify the forward proxy server.

Specify the URL with the port, for example:

```
--http_proxy=http://proxy.example.lan:80/ --https_proxy-
y=http://1.2.3.4:443/
```

Take care to use `http` when you specify the URL for the `https_proxy` variable. Do not specify the `https` protocol for the value of the `https_proxy` variable.

To configure Tableau Server to bypass the forward proxy, include the `--no_proxy` flag. You should also add exceptions to this proxy configuration to guarantee that all communications within a local Tableau Server cluster (if you have one now or will have one later) do not route to the proxy server. For example:

```
--no_proxy=localhost,127.0.0.1,localaddress,.localdomain.com.
```

If you do not configure the forward proxy during installation, then refer to [Configuring Tableau Server on Linux to work with a forward proxy](#), after you have installed.

Perform an unattended installation

Step 1: Install the automated installer

1. Log onto the computer as a user with sudo access.
2. Use the package manager to install the script package:
 - ON RHEL-like distributions, including CentOS, run the following command:

```
sudo yum install /path/to/tableau-server-automated-  
installer-<version>.noarch.rpm
```

- On Ubuntu, run the following commands:

```
sudo apt-get update
```

```
sudo apt-get -y install gdebi-core
```

```
sudo gdebi -n /path/to/tableau-server-automated-  
installer-<version>.deb
```

The automated installer package download includes templates for the configuration file (config.json), registration file (reg_tmpl.json) and the secrets (secrets) file that you can use to modify for your requirements as described in the next step. The installer script, and the templates for the initial node configuration, Tableau Server registration, and secrets file are installed to:

```
/opt/tableau/tableau_server_automated_installer/automated-  
installer.<version>
```

Step 2: Create files to provide additional configuration information required to run the automated install

Since the automated installer is meant to run without user interaction, you must provide the following additional information:

1. Run the following command to copy the templates, `config.json`, `reg_tmpl.json`, and `secrets`, to another directory like your home directory. We don't recommend that you edit the template files directly:

```
cp /opt/tableau/tableau_server_automated_installer-  
/automated-installer.<version>/{config.json,reg_tem-  
pl.json,secrets} ~
```

2. Edit the configuration template, **config.json**, to provide the initial node configuration settings. You must provide identity store settings for the Tableau Server computer. Depending on your network requirements, you may need to also provide the gateway settings. The caching option is set to cache and reuse data for as long as possible. Sample workbooks are installed by default. The template includes the minimum required information, so the template is a starting point. For more information on configuration settings, see [Configure Initial Node Settings](#).
3. Edit the registration file **reg_tmpl.json** to provide your unique identifying information needed to register Tableau Server in accordance with the End User License Agreement (EULA). For more information, see [End User License Agreement](#) and [Activate and Register Tableau Server](#).
4. Edit the secrets file using the **secrets** template with the user name and password for the TSM administrator and Tableau Server administrator accounts.
 - The TSM administrator account should be the same user as the sudo admin running the script. If you do not want to specify the password in the secrets file, you can leave it blank, and you will be prompted to provide the password during installation.
 - The Tableau Server administrator account is the initial account that is created by the installer and is used to administer Tableau Server.

Step 3: Run the automated install

1. Log onto the computer as a user with sudo access.

ON RHEL-like distributions, including CentOS, run the following command:

```
sudo /opt/tableau/tableau_server_automated_installer-  
/automated-installer.<version>/automated-installer -s  
/path/to/secrets -f /path/to/config.json -r /path/to/reg_  
templ.json --accepteula /path/to/tableau-server-<ver-  
sion>.x86_64.rpm
```

On Ubuntu, run the following command:

```
sudo /opt/tableau/tableau_server_automated_installer-  
/automated-installer.<version>/automated-installer -s  
/path/to/secrets -f /path/to/config.json -r /path/to/reg_  
templ.json --accepteula /path/to/tableau-server-<version>_  
amd64.deb
```

Important: You must specify `-accepteula` key to acknowledge and accept the end user license agreement (EULA) in the command that you use to run the script. The EULA is available in the following location: `/opt/tableau/tableau_server/packages/docs.<version_code>/EULA.rtf`.

Note: If you are adding this machine as an additional node to an existing cluster, you must specify the `-b bootstrap` flag and the node configuration file from the initial server. For more information on how to generate the node configuration file, see [Install and Configure Additional Nodes](#)

Install Tableau Server in a Disconnected (Air-Gapped) Environment

You can install Tableau Server in a disconnected environment that has no outside network access of any kind. Such disconnected environments, commonly referred to as air-gapped, are used when high security is needed to prevent data breaches or to guard against hacking. Air-gapped environments have no internet access, no outside network access, no outside wireless access, etc. The only means of getting software and data into or out of an air-gapped environment is by using removable media such as USB sticks or writeable optical CDs or DVDs.

Installing Tableau Server in an air-gapped environment is an advanced task for IT administrators who are familiar with the security considerations, best practices, and pitfalls of installing software in air-gapped environments.

The following Tableau Server features will be unavailable or will have reduced functionality in an air-gapped environment:

- **Maps** – Tableau Server uses externally hosted map data by default. Beginning with version 2020.4.0, you can configure Tableau Server to use offline maps. With earlier versions of Tableau, maps are unavailable in an air-gapped environment unless you also install a map server in your air-gapped environment. For more details, see [Displaying Maps in an Air-Gapped Environment](#).
- **Licensing** – Tableau Server needs to connect to the internet in order to activate product keys. However, you can **manually activate** the product keys.
- **External data** – Any data located outside your air-gapped environment is unavailable.

Prerequisites

In order to install Tableau Server in an air-gapped environment, you'll need the following:

- Trusted computer with limited access to the internet that you can use to download the installation packages and resources required by Tableau Server. A trusted computer has been scanned and cleared of any viruses and malware.
- Trusted removable media that you can use to transfer the downloaded software to your air-gapped environment. Trusted removable media is removable media that is new and previously unused and comes from a reputable or known source. Trusted removable media has been scanned and verified that it does not contain any viruses or malware.
- Air-gapped environment with computers and storage that meet the [requirements](#) for installing Tableau Server.

Installing Tableau Server on an Air-Gapped Computer Running Linux

The easiest way to install Tableau Server on a computer in an air-gapped environment is to do so before the computer is placed into the air-gapped environment. If that's not possible you'll need to download the required packages to a trusted computer outside the air gap:

1. On a trusted computer outside the air gap with internet access, download the Tableau Server installation package.
2. Extract the list of dependent packages:

On Ubuntu:

```
dpkg --field <debfile> Depends (where <debfile> is the name of the .deb package you downloaded from Tableau).
```

Example command:

```
dpkg --field tableau-server-linux-1.deb Depends
```

Example output:

```
ca-certificates, fontconfig, net-tools, bash-completion,
ca-certificates-java, freeglut3, libegl1-mesa, lib-
freetype6, libgssapi-krb5-2, libxcompositel, libxrender1,
libxslt1.1, lsb-core
```

On RHEL and RHEL-like Linux distributions:

`yum -q deplist <RPM file>` (where <RPM file> is the .rpm package you downloaded from Tableau).

Example command:

```
yum -q deplist tableau-server-linux_1.rpm
```

Example output:

```
package: tableau-server-10400.17.0703.1600.x86_64 10400-
17.0703.1600
dependency: /bin/sh
provider: bash.x86_64 4.2.46-21.e17_3
dependency: bash-completion
provider: bash-completion.noarch 1:2.1-6.e17
dependency: ca-certificates
provider: ca-certificates.noarch 2017.2.14-70.1.e17_3
dependency: fontconfig
provider: fontconfig.x86_64 2.10.95-10.e17
provider: fontconfig.i686 2.10.95-10.e17
dependency: freeglut
provider: freeglut.x86_64 2.8.1-3.e17
provider: freeglut.i686 2.8.1-3.e17
dependency: freetype
provider: freetype.x86_64 2.4.11-12.e17
provider: freetype.i686 2.4.11-12.e17
```

Tableau Server on Linux Administrator Guide

```
dependency: krb5-libs
provider: krb5-libs.x86_64 1.14.1-27.el7_3
provider: krb5-libs.i686 1.14.1-27.el7_3
dependency: libXcomposite
provider: libXcomposite.x86_64 0.4.4-4.1.el7
provider: libXcomposite.i686 0.4.4-4.1.el7
dependency: libXrender
provider: libXrender.x86_64 0.9.8-2.1.el7
provider: libXrender.i686 0.9.8-2.1.el7
dependency: libxslt
provider: libxslt.x86_64 1.1.28-5.el7
provider: libxslt.i686 1.1.28-5.el7
dependency: mesa-libEGL
provider: mesa-libEGL.x86_64 11.2.2-2.20160614.el7
provider: mesa-libEGL.i686 11.2.2-2.20160614.el7
dependency: net-tools
provider: net-tools.x86_64 2.0-0.17.20131004git.el7
dependency: redhat-lsb-core
provider: redhat-lsb-core.x86_64 4.1-27.el7.centos.1
provider: redhat-lsb-core.i686 4.1-27.el7.centos.1
```

3. Download each of the dependent packages:

On Ubuntu:

```
apt-get download <package1> <package2>...
```

On RHEL and RHEL-like Linux distributions:

```
yumdownloader <package1> <package2>...
```

4. Transfer the packages to your removable media.
5. On your air-gapped computer, insert the removable media containing the Tableau

Server installation package and dependent packages, and then **run the installer**.

6. After installation is complete, you can activate the Tableau Server product keys. For more information, see [Activating Tableau Server in an Air-Gapped Environment](#).

Activating Tableau Server in an Air-Gapped Environment

Because an air-gapped computer is not connected to the internet, you'll need to perform the Tableau Server activation process manually.

Step 1. Transcribe data from your air-gapped system into an activation request template.

1. On your Tableau Server in the air-gapped environment, use TSM to obtain the offline activation file. At a command prompt:

```
tsm licenses get-offline-activation-file -k <product-key>
-o <target-directory>
```

The `<target-directory>` must exist. You can obtain your product key in the [Tableau Customer Portal](#).

2. On a trusted computer that is connected to the internet and is capable of sending files via email, right-click and choose **Save link as** to download either the `server_firstpass_linux.tlq` or the `server_secondpass_linux.tlq` file. The firstpass template is used when the computer has never had Tableau Server installed on it before.
3. Open both the `offline.tlq` file from the disconnected computer, and the appropriate `server_firstpass_linux.tlq` or `server_secondpass_linux.tlq` file in an XML text editor such as Notepad++ on the trusted computer that has Internet access. You'll need to write down the values from the disconnected computer in order to copy them to the computer that has Internet access.
4. Update the following XML elements in the appropriate .tlq file (`server_`

Tableau Server on Linux Administrator Guide

`firstpass_linux.tlq` or `server_secondpass_linux.tlq`) with the values for the same elements in `offline.tlq` on the disconnected computer. The template file has X's where you need to change the values. Do not change anything other than the values below.

All the Machine / Hash values in the .tlq files are Hex values. The only valid characters are 0 - 9 and A - F. Use all caps for any letters.

Line 2 - `<EntitlementId>`

Line 5 - `<ClientVersion>`

Line 5 - `<RevisionType>` (This value is only present in the `server_secondpass_linux.tlq` file.)

Line 5 - `<MachineIdentifier>` (This value is only present in the `server_secondpass_linux.tlq` file.)

Line 11 - `<Value>` (If the value is not present, remove the "X" place holder, leaving `<Value></Value>`.)

Line 12 - `<Value>` (If the value is not present, remove the "X" place holder, leaving `<Value></Value>`.)

Line 13 - `<Value>` (If the value is not present, remove the "X" place holder, leaving `<Value></Value>`.)

Line 15 - `<SequenceNumber>`

Line 61 - `<Hash>`

5. Upload the edited template (`offline.tlq`) to the [Tableau Offline Activation](#) website.
6. The website should say The activation was successful. Please [click here](#) to download your activation file.

Download the `activation.tlf` file and transfer it to your Tableau Server.

Step 2. Initialize or activate your license

1. Move the activation file (`activation.tlf`) to your disconnected computer using trusted removable media.
2. Run the following command:

```
tsm licenses activate -f <path-and-activation-file>
```

First time license activation

If this is the first Tableau license activated on the computer, you will see this message:

```
Your license has been initialized. To complete the activation,  
we need one more exchange. Generate and send to Tableau a  
second activation request file.
```

You need to follow the steps below to activate your license. If you are not sure whether you need to repeat the steps, you can run this command:

```
tsm licenses list
```

If you see a message like this one you need to repeat the steps above:

```
No licenses are currently activated.
```

Subsequent license activation

If this is not the first Tableau license activated on the computer, or if you have completed the steps above two times, you should see the message `"Activation successful."`, which indicates that Tableau Server is activated.

Note: You may have to perform offline activation twice to completely activate your air-gapped Tableau Server. The first offline activation sets up Trusted Storage on the computer if Tableau has never been licensed on it before. The second offline activation propagates the licensing information into Trusted Storage. If you're upgrading an existing install of Tableau Server, right-click and choose **Save link as** to download the [server_secondpass_linux.tlq](#) file. It will match all the places to put the values from the generated .tlq file.

Displaying Maps in an Air-Gapped Environment

In an air-gapped environment, maps in Tableau Server will be unavailable by default due to the lack of internet access. If you have Tableau Server 2020.4.0 or later, you can configure Tableau to use local maps. If you have an earlier version, you can use one of several workarounds to display maps in an offline environment.

Configuring Tableau Server to use offline maps (version 2020.4.0 and later):

1. Open a command prompt as administrator.
2. Configure Tableau to use locally available offline maps:

```
tsm configuration set -k vizqlserver.force_maps_to_offline  
-v true  
  
tsm pending-changes apply
```

Workarounds to display maps in Tableau Server in an offline environment (versions earlier than 2020.4.0):

- **Option 1** – Use the offline background maps

To use the offline background map select **Map > Background maps > Offline**. For more information, see [Select Background Maps](#).

Note: Due to size limitations with the offline map, zooming in too deep may result in errors as the images for the more close-up maps are not stored offline.

- **Option 2** – Use a map as a background image

Locate the desired map online and save it as an image file, or copy a map from another location. Set this map image as a static background image in Tableau Desktop. For more information, see [Use Background Images in Your Views](#).

Note: Zooming in and out will not work with background images.

- **Option 3** – Use a local GeoServer

For more information, see [Use Web Map Service \(WMS\) Servers](#).

Tableau Server in a Container

Introduction

Tableau Server in a Container is Tableau's first container-based server offering. Tableau Server in a Container is an all-in-one Tableau Server instance running inside of a Linux Docker container. In other words, a Tableau Server in a Container image is a docker image that runs an entire self-contained Tableau Server application. Tableau Server in a Container is our first of many steps to support Tableau Server running in container-based environments. The easiest way to understand the concept of Tableau Server in a Container is to think of it like a VM with Tableau Server pre-installed. The image is based on a CentOS 7.x image and runs `supervisord` (instead of `systemd`) inside the container. When the container starts `supervisord`, it will immediately attempt to initialize and start Tableau Server. Much of the documentation here aims to describe how to provide configuration and leverage automation so you can run Tableau Server in Docker environments.

The Tableau Server in a Container Image Setup Tool helps you create and customize container images to include custom packages and artifacts. One of the primary functions of the tool is to build the container image and install custom data connectors.

To test out the Tableau Server in a Container Image quickly, refer to the **Quick Start Guide** below.

Limitations

- Tableau Server in a Container only supports license activation using Server ATR, which requires the container to have internet access. Therefore, offline activation in an air-gapped environment is not possible.
- Tableau Server in a Container does not currently support the Resource Monitoring Tool (RMT) agent.
- Kerberos is not supported in Tableau Server in a Container

Tableau Server in a Container - Quick Start Guide

This topic provides all of the steps required to perform a basic, quick start deployment of a Tableau Server in a Container Image using the command line. The purpose of the configuration described here is to provide the quickest, simplest path to a Tableau Server on Linux installation running inside a container. For a more in-depth look at what Tableau Server in a Container has to offer skip down to the [Tableau Server in a Container Setup Tool](#) or [Tableau Server in a Container Image](#) sections.

Important: Do not use the following Quick Start procedure as a stand-alone resource for deploying Tableau Server in a production environment.

Before you begin

There are two basic steps required to use Tableau Server in a Container:

1. Building the Docker image

Building the Tableau Server in a Container Docker image is only supported on a RHEL-based Linux system (RHEL, CentOS, or Amazon Linux 2). Building on any other Linux distributions may be possible but is currently untested and unsupported. Building images on macOS is not supported.

You must use a distro that includes Docker version of 17 or later, with the latest version of Docker being preferred. Docker versions earlier than version 17 do not include features that are required for Tableau Server in a Container.

2. Running the Docker image

Production use of Tableau Server in a Container is only supported on Linux. For exploratory prototyping and testing work, any system that can run Linux-based Docker images should be able to run Tableau Server in a Container images (assuming it satisfies the hardware and operating systems requirements outlined in *Before you install...*).

Tableau Server in a Container images require the same hardware resources as Tableau Server itself in order to run. For production deployments, follow the recommendations shown in *Minimum Hardware Recommendations*. For exploratory work, follow the recommendations shown for *Minimum Hardware Requirements for Testing and Prototyping*. If you are deploying in a public cloud environment, see the links at the bottom of the *Minimum Hardware Requirements and Recommendations for Tableau Server* page for detailed recommendations on instance sizing. If running the image in Docker Desktop on Windows or macOS (for exploratory work), be aware that the default resource limits for containers are set far below the minimum requirements and must be increased to run the container successfully.

Note: Tableau Support cannot help with issues on unsupported platforms or configurations.

The steps below assume that you have a Linux system with Docker installed and you have downloaded the Tableau Server in a Container Setup Tool and a version compatible Tableau Server RPM installer.

1. Untar the Tableau Server in a Container Setup Tool

The Tableau Server in a Container Setup Tool will be available as a tarball. To unpack the tar archive, simply use the following command:

```
tar -xzf tableau-server-container-setup-tool-  
<VERSION>.tar.gz
```

2. Edit the registration file provide your unique identifying information needed to register Tableau Server in accordance with the End User License Agreement. The file, `reg-info.json`, serves as a template for your registration information and is located in the top directory of the Tableau Server in a Container Setup Tool. This file is used to register the Tableau Server instance running in the image. Providing accurate information will ensure the registration process completes properly.

In the file, the `eula` field value is pre-filled with "accept" to indicate you are accepting our End User License Agreement (EULA). You can view the EULA in the EULA directory of the build tool. When you are done the other fields should have values that reflect your unique information.

Note: You must accept the EULA to use Tableau Server. If you do not accept the EULA, you cannot run Tableau Server.

The registration file template `reg-info.json` before editing:

```
{  
  "zip" : "<value>",  
  "country" : "<value>",  
  "city" : "<value>",  
  "last_name" : "<value>",  
  "industry" : "<value>",  
  "eula" : "accept",  
  "title" : "<value>",
```

```
"phone" : "<value>",
"company" : "<value>",
"state" : "<value>",
"department" : "<value>",
"first_name" : "<value>",
"email" : "<value>"
}
```

3. Run the `build-image` script in the Tableau Server in a Container setup tool to build the Tableau Server in a Container image:

```
./build-image --accepteula -i <Tableau Server Installer-
>.rpm
```

4. Run the Tableau Server in a Container Image:

- a. Execute the Docker run command with all arguments filled out. You will need to provide the following information in the command line:

- License key or license key file
- Username and password for a new initial admin account (you will use this to sign into Tableau Server after it starts)
- A static (internal) host name for computer
- Name of the Tableau Server in a Container image

- b. When you have collected this information run this docker command on a minimum-spec host:

```
docker run \
-e LICENSE_KEY=<key> \
-e TABLEAU_USERNAME=<username> \
-e TABLEAU_PASSWORD=<password> \
--hostname=<static (internal) name of host machine> \
-p 8080:8080 -d <Tableau Server in a Container image
ID or tag>
```

- c. After 10-20 minutes Tableau Server will be initialized.

5. Access Tableau Server

When Tableau Server is completely initialized, check that the server is running properly by opening a web browser and going to `http://<hostname>:8080` where `<hostname>` is the host name of the machine running the image.

You should have a running instance of Tableau Server in a Docker container at this point. For more advanced and customized deployments use the documentation below to guide you.

Tableau Server in a Container Setup Tool

The Tableau Server in a Container Setup Tool `build-image` builds a custom Tableau Server in a Container image from a Tableau `.rpm` Installer and the provided configuration files.

The Setup tool takes a Tableau Server installer and your drivers and other artifacts as input and creates a Docker image. When the `build-image` tool is used properly the newly generated image will have installed the desired artifacts.

You must use a Linux distro that includes Docker version of 17 or later, with the latest version of Docker being preferred. Docker versions before version 17 do not include features that are required for Tableau Server in a Container. For details on supported distros, see [Before you begin](#) above.

Download the necessary files

To use the Setup Tool, you need to download both the tool and a compatible Server Installer `.rpm` file. The Installer file must be version 2021.2.0 or later. Both files can be downloaded from the [Tableau Server page](#).

1. Download the Server installer file, `tableau-server-<version>.rpm` version 2021.2.0 or later.
2. Download the Server in a Container Setup tool, `tableau-server-container-setup-tool-<version>.tar.gz`.

The Tableau Server in a Container Setup Tool is provided as a tarball. You will need to extract the contents of the compressed file. Here is an example which assumes the Tableau Server in a Container setup tool archive is in your current directory:

```
tar -xzf tableau-server-container-setup-tool-<VERSION>.tar.gz
```

This creates a new directory, `tableau-server-container-setup-tool-<VERSION>` with the `build-image` script you use to run the tool.

Installation

The Tableau Server in a Container Setup Tool is provided as a tarball. You will need to extract the contents of the compressed file. Here is an example which assumes the Tableau Server in a Container setup tool archive is in your current directory:

```
tar -xzf tableau-server-container-setup-tool-<VERSION>.tar.gz
```

This creates a new directory, `tableau-server-container-setup-tool-<VERSION>` with the `build-image` script you use to run the tool.

Complete the registration form

Edit the registration file to add your information to it. The file, `reg-info.json`, serves as a template for your registration information and is located in the top directory of the Tableau Server in a Container Setup Tool. This file is used to register the Tableau Server instance running in the image. Providing accurate information will ensure the registration process completes properly.

In the file, the `eula` field value is pre-filled with "accept" to indicate you are accepting our End User License Agreement (EULA). You can view the EULA in the EULA directory of the build tool. When you are done the other fields should have values that reflect your unique information.

Note: You must accept the EULA to use Tableau Server. If you do not accept the EULA, you cannot run Tableau Server.

The registration file template `reg-info.json` before editing:

```
{
"zip" : "<value>",
"country" : "<value>",
"city" : "<value>",
"last_name" : "<value>",
"industry" : "<value>",
"eula" : "accept",
"title" : "<value>",
"phone" : "<value>",
"company" : "<value>",
"state" : "<value>",
"department" : "<value>",
"first_name" : "<value>",
"email" : "<value>"
}
```

How to use the tool

There is an executable script in the setup tool called `build-image`. Running this with the `-h` option displays the help:

```
./src/build-image -h
Usage: build-image --accepteula -i [INPUT_IMAGE_NAME] -o
[OUTPUT_IMAGE_NAME] [optional arguments]
Creates new Tableau Server image with pre-installed database
drivers, configuration, and instructions.
REQUIRED
```

`--accepteula` Indicate that you have accepted the

End User License Agreement (EULA).

The EULA is available in the EULA directory of this tool.

`-i installer` Path to the Tableau Server installer.

OPTIONAL

`-o output name` Tag of target generated Tableau Server image.

`-e environment file` Optional environment file to configuration overrides to docker image at build time.

`-v setup tool version` Prints the Setup tool version.

`-b base image version` Prints the default base image URL.

Basic example usage:

```
./build-image --accepteula -i tableau-server-image.rpm
```

Running the command without providing artifacts or a setup script will work, but doesn't really do anything because it would not be copying or installing any additional resources into the original Tableau Server in a Container image.

Orchestration

Tableau only provides documentation and support for Tableau Server containers running on Linux. Tableau does not provide documentation or support for container orchestration systems like Kubernetes or Docker Swarm. Kubernetes can, however, be used to deploy Tableau Server in a Container. For resources and guidance on deploying Tableau containers in Kubernetes, refer to our community-supported GitHub project: <https://github.com/tableau/tableau-server-in-kubernetes>.

Customizing the image

Setting Environment Variables at Build Time

A subset of environment variables that can customize your image can only be set when the image is built. This includes user, group, and other properties that require root privilege inside the container. Root actions are not available by default at runtime. Additionally it can be useful to bake certain environment variables into the image at build time so they do not have to be set every time the image is run. All of these environment variables can be set by passing in an environment file to the build-image script.

Environment File

The environment file can be passed to the build-image script using the `-e` argument. The file must conform to this format:

```
<KEY>=<VALUE>  
<KEY>=<VALUE>
```

Example Usage:

Create an environment file with the correct format:

```
UNPRIVILEGED_TABLEAU_UID=1012  
UNPRIVILEGED_TABLEAU_GID=1020  
TABLEAU_USERNAME=myuser  
TABLEAU_PASSWORD=pw
```

Pass the file to the image builder

```
./build-image --accepteula -i tableau-server-2020-3.x86_64.rpm -  
e <path-to-env-file>
```

Environment Variables

Any environment variable can be set in the environment file. Take a look at the Environment Variables section to see the complete list.

Build time environment variables can only be set when this script is run to build the image:

Environment name	Default	Description
BASE_IMAGE_URL	Use the build tool command: <code>build-image -b</code>	The default image specified in the build-image tool and Dockerfile is the only officially supported base image. This parameter can be used to either pull a copy of this specific base image from a custom docker image repository or define a custom base image. If you choose to use a custom-defined base image (see Defining a Custom Base Image for more details), it is your responsibility to ensure it is based on CentOS or RHEL 7 and contains the necessary resources to run Tableau Server properly.
PRIVILEGED_TABLEAU_GID	997	The GID of the privileged tableau group.
UNPRIVILEGED_TABLEAU_GID	998	The GID of the unprivileged tableau group.
UNPRIVILEGED_TABLEAU_UID	999	The UID of the user that runs tableau processes (single user deployment).
UNPRIVILEGED_USERNAME	tableau	The string name of the unprivileged user.
UNPRIVILEGED_GROUP_NAME	tableau	The string name of the unprivileged group.
PRIVILEGED_GROUP_NAME	t-smadmin	The string name of the privileged group.
LANG	en_US.UTF-8	Locale setting

Tableau Server on Linux Administrator Guide

Drivers, Certificates, and other files

The Tableau Server image does not come with pre-installed data connectors or drivers. You will need to create a bash setup-script that will instruct the `build-image` script to install the data connectors Tableau Server will need. These are the steps you would take:

1. Make sure the Tableau Server in a Container setup tool is installed properly
2. Download the driver from the Tableau driver page: <https://www.tableau.com/en-us/support/drivers>
3. Copy the downloaded driver file into the `customer-files` directory in the Tableau Server in a Container setup tool.

The `customer-files` directory will be copied into the docker image. It will be located in the following path inside the container: `/docker/customer-files`

4. Edit the file in the Tableau Server in a Container setup tool `customer-files/-setup-script` to tell Docker how to install the driver.

The script is just an executable bash file that is executed when the Docker image is built. (Technically it can be used to perform any arbitrary actions in the image including environment setup, configuration, etc.)

The script will be run inside the container so be mindful that the commands must work when executed inside the Tableau Server in a Container image.

For example, if you want to install a driver named `mydriver.rpm` you would write this in `setup-script`:

```
yum install -y /docker/customer-files/mydriver.rpm
```

Tableau Administrative Views

Tableau Administrative views require the PostgreSQL driver to be installed. If you plan on using these views you will need to follow the steps above and install the PostgreSQL driver. For more information about Administrative views, see [Administrative Views](#)

Defining a Custom Base Image

The default base image used to build the Tableau Server container is a CentOS 7.x image sourced from Docker Hub. In some cases you may want to configure the `build-image` tool to pull the image from a different docker image repository. For example, your company may manage an internal docker repository and not want the `build-image` pulling from a public repository. Use the following steps to customizing the base image path:

1. Use the following command to view the current `build-image` tool's default base image name and version:

```
./build-image -b
```

2. Pull the exactly matching base image name and version from Docker Hub and store/cache it in your preferred image repository (per your company's image policies)
3. Come back to the `build-image` tool. Create or modify an existing environment file to include the `BASE_IMAGE_URL` environment key with the value set to a new docker image registry path:

```
BASE_IMAGE_URL=<custom-image-registry-path>
```

4. Build the image with the environment file:

```
./build-image --accepteula -i <rpm> -e <path-to-env-file>
```

These steps enable you to specify a completely different base image. This capability is supported only for RHEL and CentOS 7.x based images and is provided to help customers create more secure images.

Using a base image other than the default specified by Tableau carries the risk of producing an image that does not start or function properly. If you choose to use a custom base image, you are responsible for ensuring that the base image enables Tableau Server to run properly. We recommend using the default base image unless it is unacceptable to your organization for some reason, such as security concerns.

The custom base image must be based on RHEL 7 or CentOS 7. Using any other distro will result in an unsupported image.

Base Images and Security

Many customers will run container scanning tools (such as AquaScan or TwistLock) against the generated Tableau Server Docker image. These security tools will generate a report of potential security vulnerabilities (or Common Vulnerabilities and Exposures or CVE). There are two types of vulnerabilities associated with the Tableau Server in a Container image:

- CVEs associated with Tableau Server or with a library we have a dependency on.
- CVEs associated with the underlying Linux distro.

Tableau is responsible for CVEs associated directly with Tableau Server. The Security Team analyzes these reports to determine impact and severity to help prioritize the issues for resolution. The baseline remediation priority and timelines will be determined by the original CVSS severity scoring. Third party component security updates will not usually be back-ported into older releases unless there is an executable code path that exposes the vulnerability.

With the containerized distribution model, customers are faced with a different set of challenges around OS Layer vulnerabilities. Traditionally, with Tableau Server, Tableau would focus its efforts on delivering a secure application and the customer is responsible for managing the Operating Systems (Linux or Windows). However, with containerization, the OS (CentOS 7.x or RHEL 7.x) and the application are packaged together in one container. Tableau takes responsibility for the CVE associated with Tableau Server and for determining if a CVE from a third party library would impact customers. However customers must be responsible for the OS layer issues. In order for customers to address the security issues in the OS layer customers may replace the base OS layer with their own version (as long as it is RHEL or CentOS 7.x based). Note, in doing this you must validate that Tableau Server runs correctly. Removing a library Tableau Server is dependent on because of security issues may result in a non-functioning instance of Tableau Server. If you change the base OS layer customers are responsible for validating that it works.

Hardening Tableau Server in a Container

You can apply the standard hardening steps to Tableau Server in a Container. For more information on hardening Tableau Server, see Security Hardening Checklist.

Tableau Server in a Container Image

The Tableau Server in a Container Image is a Docker image that contains all of Tableau Server. The image is built using the Tableau Server in a Container Setup Tool. The image comes with Tableau Server installed and not yet initialized. The default user in a Tableau Server in a Container image is a non-root unprivileged user.

Prerequisites

Run the `configure-container-host` script

When Tableau Server is installed outside a container on a machine certain resource limits and coredump properties are changed as part of the installation process. This is to help optimize the performance of Tableau Server. A Tableau Server in a Container image does not have the ability to make these changes on the host machine, so we recommend running the `configure-container-host` script that is provided in the Tableau Server in a Container Setup Tool on any machine that will be running Tableau Server in a Container images. This will help ensure the performance of the Tableau Server in a Container image is on-par with its bare metal counterpart.

To run the `configure-container-host` script:

1. Locate the script (`configure-container-host`) in the top-level directory of the Tableau Server in a Container Setup Tool.
2. Copy it to the environments you plan on running Tableau Server.
3. Determine the unprivileged user account/uid that will run as the default user of the Tableau Server in a Container image. This user should exist on the host machine and

should match the UID set in the Tableau Server container UNPRIVILEGED_TABLEAU_UID environment variable. If you did not set this when creating your Docker image, the default unprivileged user id inside the container is 999. If you are using Docker user-mapping, this uid should correspond with the user that exists on the host machine.

4. Execute the script as root:

```
sudo ./configure-container-host -u <uid>
```

Running the Image

To run a Tableau Server in a Container docker image, the simplest command to get a Tableau Server in a Container Image running is the following:

```
docker run \  
-e LICENSE_KEY=<key>  
-p 8080:8080  
--hostname=<static (internal) name of host machine> \  
-d <Tableau Server in a Container image ID or tag>
```

This will result in a fully installed Tableau Server instance. At this point the initial admin account will need to be set. Note: this step can be automated: take a look at the Automate Initial Admin. Tableau Server can take 10-20 minutes to reach a fully operational state (this time varies depending on the hardware specs running the image).

Basic Run Arguments Summary

All the options used in the Docker run command are necessary, often times more options will be provided to leverage different functionality in the image. For now, let's take a closer look at just the arguments used in the simplest Docker run command for Tableau Server in a Container:

Argument	Description
-e LICENSE_KEY=Y=<key>	Tableau Server must be licensed. This environment variable will store the key that will be used to license the server. This is a

Argument	Description
	required component of the initialization process. You can provide multiple licenses separated with comma.
-p 8080:8080	This tells docker to expose port 8080 inside the container and bind it to port 8080 on the host machine. The first 8080 value is configurable, changing this will modify the port mapped on the host. Tableau Server by default expects to receive user traffic on port 8080 inside the container, you can choose whether to expose this port on a different host port or not at all.

Automate Initial Admin

When Tableau Server starts up for the first time an initial admin user must be created before remote network connections to Tableau Server are permitted. This can be done by running the `tabcmd initialuser -s localhost:8080 -u <username> -p <password>` command inside the container. You may also set admin credentials via environment variables. `TABLEAU_USERNAME` and `TABLEAU_PASSWORD` or `TABLEAU_PASSWORD_FILE` (preferred) are the environment variables that can be set to pass in initial admin credentials. For more information on password management take a look at the Password section of this documentation.

For more information on the `tabcmd initialuser` command see `initialuser`.

Example

```
docker run \
-e LICENSE_KEY=<key> \
-e TABLEAU_USERNAME=<myadmin> \
-e TABLEAU_PASSWORD_FILE=/etc/tableau-admin-secret \
-v <full-path-to-pw-file>:/etc/tableau-admin-secret \
--hostname=<static (internal) name of host machine> \
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

Licensing

Licensing in Containers

Licensing of Tableau Server in a Container uses the Server Authorization-To-Run (ATR) service to activate Tableau Server deployed in the cloud, containers, or virtual environments without running out of license activations. The ATR service achieves this by providing short-term leases of configurable duration (ATR duration) until the product key expiration date is met. ATR abstracts Tableau licensing from underlying hardware changes, which is a fundamental aspect of container deployments. Since Server ATR requires the container to be able to reach the ATR service hosted by Tableau, the containers require internet access. Tableau Server in a Container does not support offline or manual activation. See [Activate Tableau Server Using the Authorization-To-Run \(ATR\) Service](#) for more details.

Important: You must provide either the `LICENSE_KEY` or `LICENSE_KEY_FILE` environment variables (only set one).

License Environment Variable

Tableau Server in a Container supports setting license keys using an environment variable: the `LICENSE_KEY` can contain one or more keys (`-e LICENSE_KEY="<key1> , <key2>"`) via a comma separated list.

Example

```
docker run \  
-e LICENSE_KEY="<key1>, <key2>" \  
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

License File

Tableau Server in a Container also supports setting license keys using a file. Mount a file to the default license key file location in the container (`/docker/config/license_file`) or as otherwise specified by the environment variable `LICENSE_KEY_FILE`.

Example

```
docker run \
-v <full-path-to-license-file>:/docker/config/license_file \
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

Requested license lease time

You can specify the ATR license lease time in a Tableau Server container by setting the environment variable `REQUESTED_LEASE_TIME`. You must provide the lease time in seconds, with the minimum duration being 14,400 seconds (or 4 hours). It is recommended that you lower the lease time when experimenting and testing Tableau Server to reduce the likelihood of reaching the max activated lease limit. For production deployments, it is strongly recommend to not set the `REQUESTED_LEASE_TIME` parameter (thus using the default value), so Tableau can determine the ideal lease time.

Example

```
docker run \
...
-e REQUESTED_LEASE_TIME=<time-in-seconds> \
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

Running an Uninitialized Image

Tableau Server has two phases of installation, first the Tableau Service Manager (TSM) services are installed. In a typical On-Premise installation this step is a time for server admins to register their server, activate their licenses, and configure the server to behave the way they want it to. The second phase of installation is setting up and starting the Tableau Server processes which will handle end-user traffic and related business logic.

The default behavior of Tableau Server in a Container images is to automate all installation steps so the `docker run` command eventually results in a fully functional server. However, if you want to start a Tableau Server in a Container image and have it only running the TSM services (what a server administrator would expect if they just ran `initialize-tsm`), you can do this by passing the `TSM_ONLY` flag as an environment variable.

For example:

```
docker run \  
-e TSM_ONLY=1 \  
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

Interacting with the Image

When you have a Tableau Server in a Container image running, you can call TSM and `tabcmd` commands directly. Those tools are added directly to the environment path of the `pid 1` user (which is root at this time). This means you can call TSM and `tabcmd` commands as follows:

```
docker exec -it <container> tsm status -v  
  
docker exec -it <container> tabcmd initialuser -s localhost -u  
<admin> -p <secret>
```

It is also possible to open a shell in the container to perform more general operations. This is generally not recommended except for debugging purposes:

```
docker exec -it <container> bash
```

TSM Web UI and Remote CLI

The TSM Web UI and remote CLI is not accessible by default. This is because it requires a username and password to authenticate and by default the user running tableau server processes in the container is not provided a password. This is done for security reasons (we do not recommend shipping images with a default password inside them as that would allow remote access). In some cases the TSM Web UI and making remote access calls using the TSM CLI can be useful. If you want to use these features you will need to follow the steps outlined below to create a remote access user account.

More detailed information about the TSM Web UI and CLI see [Sign in to Tableau Services Manager Web UI](#).

Create a TSM Remote User

Specify the `TSM_REMOTE_UID` and `TSM_REMOTE_USERNAME` environment variables when you build the Tableau Server in a Container image using the Setup Tool. Creation of a TSM enabled account in the image requires privileged access in the image that isn't available at runtime, and therefore it can only be done when the Docker image is being built by the Tableau Server in a Container Setup Tool (`build-image`).

Example Tableau Server in a Container Setup Tool `environment` file:

```
TSM_REMOTE_UID=1010
TSM_REMOTE_USERNAME=myuser
```

Set the password for the TSM Remote User

The Tableau Server in a Container image requires a password for the account when the image is run. There are two ways you can set the password for this account.

Secrets File (Recommended)

Create a file named `remote-user-secret`, write the password into the file and mount it into the container at runtime. `TSM_REMOTE_PASSWORD_FILE` determines the expected location (default location is `/docker/config/remote-user-secret`) of the secrets file in the container.

Example `remote-user-secret` file:

```
mypassword
```

Example `docker run` command:

```
docker run \
-e LICENSE_KEY=<key>
-v {absolute-path}/remote-user-secret:/docker/config/remote-
user-secret
-p 8080:8080 \
-p 8850:8850 \
```

Tableau Server on Linux Administrator Guide

```
--hostname=<static (internal) name of host machine> \  
-d <Tableau Server in a Container image ID or tag>
```

Environment Variable

You can also simply set the `TSM_REMOTE_PASSWORD` environment variable when starting the docker image.

Example docker run command:

```
docker run \  
-e LICENSE_KEY=<key>  
-e TSM_REMOTE_PASSWORD=<password>  
-p 8080:8080 \  
-p 8850:8850 \  
--hostname=<static (internal) name of host machine> \  
-d <Tableau Server in a Container image ID or tag>
```

Security Notes

- Port 8850 must be exposed to receive TSM request traffic.
- If the password is not set properly in the image at runtime the container will exit immediately.
- TSM relies on the image's Linux user account system. In this case the account is a restricted inside the image. This means the account will have a restricted shell and is limited to executing two commands: `/bin/true` and `passwd`.

How to Rotate the TSM Remote User's Password

If you want to rotate the account password of the TSM remote user, you can do this using either of these options:

Start a new Tableau Server in a Container

The account's password is set every time the container starts up. If you are persisting Tableau data outside the container then starting a fresh image with a new password will effectively rotate the password.

1. Shut down and remove the running image
2. Set a new password value in either `TSM_REMOTE_PASSWORD` or `TSM_REMOTE_PASSWORD_FILE` environment variables (see above) in your image configuration.
3. Start the image up again.

Rotate the password manually inside a running container

If you don't want to shut down the image, you can still rotate the password manually.

1. Open a shell in the running container
2. Log in as the remote user account using the `su` command
3. Run the `passwd` command to change the password.

Warning: These manual rotations only persist as long as the container instance's write layer remains. If you delete the container, the manual changes will not be applied when a new container is started.

Initial Configuration Options

Configuring Tableau Server in a Container is essential for getting the Tableau Server behavior you want. Tableau Server in a Container is a clean installation of Tableau Server, so you need to provide the same information to the container as when configuring Tableau Server outside a container..

Runtime Environment Variables

The runtime environment variables below tell the Tableau Server in a Container image how to deploy Tableau Server. A subset of these will be described in greater detail.

All of these values are designed to be overridden to enable more flexibility for configuration.

Environment name	Default	Description
ACCEPTULA	0	Automatically set to 1 when an image is built using the Tableau Server in a Container Setup tool.
LICENSE_KEY		Set to the license key that will be used to license the server. Accepts multiple licenses separated with commas.
LICENSE_KEY_FILE	/docker/config/license_file	File path to license file. The format of the license file should be one license key per line.
REGISTRATION_FILE	/docker/config/tableau_reg.json	File path to registration file inside the image. By default this contains the registration information that was provided when the Tableau Server in a Container image was built. This can be overwritten at run time. For more information, see Activate and Register Tableau Server .
REGISTRATION_DATA		An alternative way to overwrite registration information at runtime. This environment variable must be set to a serialized JSON string containing the same registration information that would be found in a Tableau Server registration file. For more information, see Activate and Register Tableau Server .
TABLEAU_USERNAME		This refers to the initial administrator account on Tableau Server. This is recommended but optional. If this user is not set the initial admin account for Tableau Server will need to be set using <code>tabcmd</code> . If this variable is set to a value

Environment name	Default	Description
		then a password is also required. This is only used when Tableau Server is initialized for the first time. Setting this value will tell Tableau Server in a Container to automatically attempt to initialize the user. For more information, see Add an Administrator Account .
TABLEAU_PASSWORD		A plain text password for the tableau user. This refers to the initial administrator account on Tableau Server. This is required if <code>TABLEAU_USERNAME</code> is specified. For more information, see Add an Administrator Account .
TABLEAU_PASSWORD_FILE		A file path to a file containing only the password text for the tableau user. This refers to the initial administrator account on Tableau Server. This is required if <code>TABLEAU_USERNAME</code> is specified. For more information, see Add an Administrator Account .
CONFIG_FILE	/docker/- config/config.json	File path in the container to default TSM config file. The file will be used to configure Tableau Server. You must arrange for the config file to be found in that location in the container (this can be done through container file mount). For more information, see Configuration File Example . Do not set <code>CONFIG_DATA</code> if <code>CONFIG_FILE</code> is used.
CONFIG_		This can be used as a substitute for <code>CONFIG_FILE</code> . If you want to provide configuration to the

Environment name	Default	Description
DATA		<p>server without mounting an external file, set this environment variable to the equivalent serialized contents of a TSM config file.</p> <p>Example <code>CONFIG_DATA="{\"configEntities\":{\"identityStore\":{\"_type\": \"identityStoreType\", \"type\": \"local\"}}}"</code> For more information, see Configuration File Example</p> <p>Do not set <code>CONFIG_FILE</code> if <code>CONFIG_DATA</code> is used</p>
IGNORE_TOPOLOGY_CONFIG	0	0 or 1. If set to 1, the container will ignore any topology related configuration present in the config file designated by <code>CONFIG_FILE</code> .
BACKUP_FILE	/docker/-config/backup/backup-file.tsbak	A file path to a Tableau Server backup file (.tsbak). If provided during initialization the server will attempt a restore.
INIT_CONTAINER	0	0 or 1. If set to 1 Tableau Server will only attempt to initialize TSM and initialize Tableau Server and the container will exit upon completion.
TSM_	0	0 or 1. Equivalent to installing the Tableau Server

Environment name	Default	Description
ONLY		rpm and running initialize-tsm. Only the TSM (Tableau Service Manager) services will start. ONLY works if the container is initializing for the first time (this will not work if a Tableau Server in a Container is being started up with a previously initialized server directory).
BOOTSTRAP_INSTALL	0	0 or 1. Indicates whether or not the server is an initial node or an additional node. If set to 1, the container will wait indefinitely until a bootstrap file exists at the location specified by <code>\$BOOTSTRAP_FILE</code>
ALWAYS_WRITE_BOOTSTRAP_FILE	0	0 or 1. If set to 1, the container will write a bootstrap file to the location given in <code>BOOTSTRAP_FILE</code> .
WAIT_FOR_BOOTSTRAP_FILE	1	0 or 1. If set to 1 (default), if the container detected it is a worker installation (<code>BOOTSTRAP_INSTALL=1</code>). The container will wait indefinitely until a file is detected located at the path set in <code>BOOTSTRAP_FILE</code> . If set to 0 when the startup process is run this wait will be skipped. This can be useful in some debug cases.
BOOTSTRAP_FILE	<code>/docker/-con-</code>	File path to bootstrap file. Only applies to worker containers. This file should only point to a bootstrap file. The typical usage would be to mount the directory of the target file (default would be

Environment name	Default	Description
	fig/- boot- strap/- bootstrap.json	/docker/config/bootstrap) to the host. Do not set BOOTSTRAP_FILE if using BOOTSTRAP_DATA.
BOOTS- TRAP_ DATA		This can be used as a substitute for BOOTSTRAP_FILE. If you want to provide a bootstrap file without mounting an external file, set this environment variable to the equivalent serialized contents of a TSM bootstrap file. Do not set BOOTSTRAP_DATA if using BOOTSTRAP_FILE.
PORT_ RANG- E_MIN	8800	For performance reasons, we recommend exposing only 200 ports (8800-9000) instead of the Tableau Server On-Premise default 8000-9000 port range because exposing 1000 ports in docker can negatively impact the start up time of the docker image. See Exposing Licensing and TSM ports below for more information.
PORT_ RANG- E_MAX	9000	We recommend exposing only 200 ports (8800-9000) instead of the Tableau Server On-Premise default 8000-9000 port range because exposing 1000 ports in docker can negatively impact the start up time of the docker image. See Exposing Licensing and TSM ports below for more information.
HTTP_ PROXY		To forward http requests to your proxy server, set this environment variable to point your proxy host. For example, to set the proxy to example-

Environment name	Default	Description
		host for port 8080, <code>HTTP_PROXY=</code> <code>Y=http://example-host:8080/</code>
HTTP- S_ PROXY		To forward https requests to your proxy server, set this environment variable to point your proxy host. For example, to set the proxy to example-host for port 443, <code>HTTPS_PROXY=</code> <code>Y=http://example-host:443/</code> Be sure to use 'http' when you specify the URL for the <code>HTTPS_PROXY</code> environmental variable.
NO_ PROXY		To bypass the proxy server, specify exceptions in the <code>no_proxy</code> variable. Use this variable if your proxy server does not route internal addresses. You must also add exceptions to this proxy configuration to guarantee that all communications within a local Tableau Server cluster (if you have one now or will have one later) do not route to the proxy server. Enter both the host name and the IP address for each computer, and add host name of the container. Additionally, include the canonical host name (<code>localhost</code>) and IP address (<code>127.0.0.1</code>) for the local computer. For example, to specify exceptions for a three-node cluster: <code>NO_PROXY=</code> <code>="lo-</code> <code>cal-</code> <code>host,127.0.0.1,host-</code> <code>name1,host-</code> <code>name2,hostname3,IP1,IP2,IP3"</code>
COORD-	Any port between <code>PORT_</code>	Client port for the coordination service.

Environment name	Default	Description
DISTRIBUTION_SERVICE_CLIENT_PORT	RANGE_MIN and PORT_RANGE_MAX	
COORDINATION_SERVICE_PEER_PORT	Any port between PORT_RANGE_MIN and PORT_RANGE_MAX	Peer port for the coordination service.
COORDINATION_SERVICE_LEADER_PORT	Any port between PORT_RANGE_MIN and PORT_RANGE_MAX	Leader port for the coordination service.
LICENSE_SERVICE_VENDOR_DAEMON_PORT	Any port between PORT_RANGE_MIN and PORT_RANGE_MAX	Vendor daemon port for the licensing service.

Environment name	Default	Description
N_PORT		
AGENT_FILE_TRANSFER_PORT	Any port between PORT_RANGE_MIN and PORT_RANGE_MAX	File transfer port for the agent service.
CONTROLLER_PORT	Any port between PORT_RANGE_MIN and PORT_RANGE_MAX	https port for the controller service.
REQUESTED_LEASE_TIME	Default is currently set to 4 hours.	Set the requested lease time for Server ATR activations. You need to provide the time value in seconds and the minimum duration is 14400 seconds (or 4 hours). Changing this value is generally not recommended for production deployments. However when developing or prototyping with Tableau Server in a Container you may want to set this to the minimum value so as to minimize the loss of activations.

Readonly Environment Variables

These are environment properties that describe some of the basic properties of the Tableau Server in a Container image. Overriding these values is not recommended.

Environment name	Default	Description
PRE_INIT_COMMAND_SCRIPT	<code>\${DOCKER_CONFIG}/-customer-files/pre_init_command</code>	Path to a user custom bash/executable file to be run in Tableau Server prior to Tableau Server initialization. Note: Make sure the file has execute permission for all users, otherwise run <code>chmod +rx <path-to-pre-init-command-file></code>
POST_INIT_COMMAND_SCRIPT	<code>\${DOCKER_CONFIG}/-customer-files/post_init_command</code>	Path to a user custom bash/executable file to be run in Tableau Server after the server is fully functional and running. Note: Make sure the file has execute permission for all users, otherwise run <code>chmod +rx <path-to-post-init-command-file></code>
DATA_DIR	<code>/var/opt/tableau/tableau_server</code>	The data directory where Tableau Server bits should be written.
INSTALL_DIR	<code>/opt/tableau/tableau_server</code>	The installation directory where Tableau Server installation bits are written.
SERVICE_NAME	Tableau Server	Name of the application running in the container.
SERVICE_	N/A	Version of Tableau Server

Environment name	Default	Description
VERSION		installed in the container.
DOCKER_CONFIG	/docker	Directory that stores Tableau specific docker configuration.
ENV_FILE	<code>\${DOCKER_CONFIG}/-customer-files/environment</code>	File that contains all user environment overrides.

Build-Time Environment Variables

BASE_IMAGE_URL	Use the build tool command: <code>build-image -b</code>	The default image specified in the build-image tool and Dockerfile is the only officially supported base image. This parameter can be used to either pull a copy of this specific base image from a custom docker image repository or define a custom base image. If you choose to use a custom-defined base image, it is your responsibility to ensure it is based on CentOS or RHEL 7 and contains the necessary resources to run Tableau Server properly. For more information on custom base images, see Defining a Custom Base Image .
PRIVILEGED_TABLEAU_GID	997	The GID of the privileged Tableau group.

Tableau Server on Linux Administrator Guide

UNPRIVILEGED_ TABLEAU_GID	998	The GID of the unprivileged Tableau group.
UNPRIVILEGED_ TABLEAU_UID	999	The UID of the user that runs Tableau processes (single user deployment).
UNPRIVILEGED_ USERNAME	tableau	The string name of the unprivileged user.
UNPRIVILEGED_ GROUP_NAME	tableau	The string name of the unprivileged group.
PRIVILEGED_ GROUP_NAME	tsmadmin	The string name of the privileged group.
LANG	en_US.UTF-8	Locale setting

Tableau Server Configuration Overrides

These environment variables can be overwritten by Docker to point at any file in the container. So if you want to specify a different mount point you are welcome to do so.

Tableau Server needs a configuration file to start and run:

```
CONFIG_FILE=/docker/config/config.json
```

`CONFIG_FILE` refers to a TSM configuration file. The format and usage is identical to the configuration file described in Configuration File Example.

Pre-initialization and Post-initialization Commands

Tableau Server runs an automated installation script designed to take Server from a pre-initialized state to fully running. However, sometimes you may want to add in your own auto-

mation code in the initialization process. We offer two hooks to do this, the pre-initialization script and post-initialization script.

Pre-initialization script

This script will run immediately after the base TSM processes are initialized and before any other TSM setup steps are executed. This is useful for executing TSM configuration commands before Tableau Server runs. For configuration changes made at this point, you do not need to apply the changes because the normal Tableau Server automation does this after your script completes.

Post-initialization script

This script will run after all other Tableau Server initialization and start-up automation completes. Tableau Server will be fully functional and running when this script executes. Configuration changes made at this point must be applied.

Instructions

To add a custom script to one of these hooks in your image, follow these steps:

1. Write your custom script
2. Copy the custom script into the `customer-files` directory of the Tableau Server in Containers Build Image Tool.
3. Rename the script to be either `pre_init_command` or `post_init_command` depending on when you would like the script to run (you can use both hooks independently from each other).
4. Ensure the permissions of the script are either executable by other (`chmod +rx <command-file>`) or the ownership permissions match the unprivileged user in the container.

User Configuration

Tableau Server uses an unprivileged user to run server processes. This user created inside the container when Tableau Server in a Container is initializing. By default the user is named `tableau` with a UID of 999. If you are deploying a Tableau Server in a Container that uses mounts to externally store data on the host machine you may prefer to change the

UID to map a to a UID on the host machine. Using docker user namespaces is another way to achieve the same result.

Tableau Server in a Container Utilities and Tools

All Tableau Server in a Container utility and tool functions are placed under this directory:

```
/docker/
```

File Permission Management

When passing any configuration files to the container you will want to ensure the user running the Tableau Server process inside the container has permission to access the files. To avoid granting all users access to files being mounted to the container you can change the UID and/or the GID of the user running Tableau Server inside the container to match the user/group owner on the host. The container user will have a UID determined by the `UNPRIVILEGED_TABLEAU_UID` environment variable (default: 999) and GID determined by `UNPRIVILEGED_TABLEAU_GID` (default: 998). These values can be changed by overriding the environment variable or you can use a Docker user namespace mapping to associate the UID/GID in the container to a different UID/GID on the host.

Password Management

Certain features and options require user credentials to be provided as a configuration setting into the container. Tableau Initial Admin credentials is an example of optional credentials that enable additional features. In these cases we always provide two methods of setting the password. First, you can provide a file containing the password and supply a file path to an environment variable. Alternatively, you can set an environment variable to store the password directly.

The recommended and more secure option is to provide the password as a file path to the container. Providing a secret in a file is a well supported pattern in Docker, Docker Swarm, Kubernetes, and other container orchestration systems. Storing passwords directly in environment variables is a common pattern so we do support it, but it typically means the password is less secure.

Examples

Let's take a look at the `TABLEAU_USERNAME` credential. You can provide the password for the user as either `TABLEAU_PASSWORD` or `TABLEAU_PASSWORD_FILE`. When running a Tableau Server in a Container image you can provide either environment variable to supply the password.

The password file environment variable expects a file path inside the container to a valid secrets file. The secrets file should be a single line containing the secret.

Example of using a secrets file

```
docker run \
...
-e TABLEAU_USERNAME=admin \
-e TABLEAU_PASSWORD_FILE=/etc/admin-secret \
-v <full-path-to-pw-file>:/etc/admin-secret \
-d <Tableau Server in a Container image ID or tag>
```

Example contents of a secrets file

```
mypassword23879172
```

Alternatively one can directly store the password in plain text in the password environment variable. This approach is considered less secure but it is more convenient and a common pattern with containers.

Example

```
docker run \
...
-e TABLEAU_USERNAME=admin \
-e TABLEAU_PASSWORD=password \
-d <Tableau Server in a Container image ID or tag>
```

Configuring Tableau Server after it is running

Once Tableau Server has been initialized and is running, the best way to interact with the server is to use the TSM CLI tool. This is the classic Tableau Server tool for performing administrative tasks. In the future we will support Tableau Server reacting to changes in the static configuration provided in the `CONFIG_FILE` environment variable in between runs. But for now, after Tableau Server is initialized, you must interact with the server using the classic tools.

For more information on the TSM command-line see [tsm Command Line Reference](#).

Status

There are two basic status checks for Tableau Server provided in the image. These can be used to check the aliveness and readiness of the server.

Liveness Check

The liveness check indicates whether or not TSM services are running. This means it will indicate whether the orchestrated services of Tableau Server are operating and are functioning. This check is callable here:

```
/docker/alive-check
```

Another option is to expose port 8850 which the Tableau Controller service runs to provide administrative functions through a web browser. One could periodically check the health of the service by checking the health of the service through `tcp` health checks.

Readiness Check

The readiness check indicates whether Tableau Server is running and business services are ready to receive traffic. This can be determined using the following script:

```
/docker/server-ready-check
```

Another option is to use `tcp` health checks against port 8080 (or whatever port Tableau Server is bound to receive traffic). Sometimes this kind of `tcp` health check is more reliable

than the server-ready-check, as the server-ready-check is based on service status reported to TSM which can sometimes be delayed as the service state is updated.

Persisting Data

Often times with containers we want the capability to shut it down and then turn it back on without losing any important information. Tableau Server in a Container images support this in that you can mount certain directories outside of the container so you can completely destroy or remove container instances and still preserve your data. This data can be used to start another container instance and resume where the previous container left off.

The following sections cover the different kinds of managed state.

Tableau Server Data

Server Data is all stored in the data directory. The data directory is where all user-related data and service runtime metadata is stored. Externalizing this data means your user's data can be persisted even after the Tableau Server in a Container is completely removed.

This data is transferable and can be usable with cloud-managed block storage system, like AWS EBS volumes.

Important: Do not use a network file system (i.e. NFS) for either of these directories.

Static Hostnames

Tableau Server does not handle dynamic hostname changes well so it is important to specify the container's internal hostname so it is consistent between container runs. The hostname inside a container is arbitrary and can be set to any value. Using the `--hostname` option allows one to specify the internal hostname of the container. **Make sure subsequent containers using the same persistent data are run using the same hostname value.**

This is not to be confused with multi-node server installations. In these, additional nodes should each be assigned a different hostname. What matters is when any single container

Tableau Server on Linux Administrator Guide

is restarted the replacing container that will use the same persistent data for that instance must have a matching hostname.

Complete Example

Here is an example where the data directory is mounted outside the container.

```
docker run \  
-v <empty-data-dir>:/var/opt/tableau \  
-e LICENSE_KEY=<key> \  
--hostname=localhost \  
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

Backup and Restore

Tableau Server in a Container supports Tableau Server creating backups and restoring from a backup file (.tsbak). The first step is to get a Tableau Server in a Container image running, have the backup file (.tsbak) mounted in the image and set the environment variable `BACKUP_FILE` with the file path to the backup file. Additionally you must provide the backup json configuration file in the `CONFIG_FILE` environment variable. The Tableau Server container automates the restore process even for multi-node deployments. If at any point this automation fails to fully restore the system, you can always fallback on the classic Tableau Server tools and processes such as TSM commands to interact with Tableau Server in the same way you would with a non-container deployment.

For more information on how to perform a backup and restore of a standard Tableau Server instance see [Perform a Full Backup and Restore of Tableau Server](#).

Backup in Tableau Server Container

1. Open shell inside the Tableau Server in a Container version A. Create repository backup, and topology and configuration backup files.

```
docker exec -it my-server bash
```

```
# Just providing filename automatically produces the backup
```

```

file at /var/opt/tableau/tableau_server-
/data/tabsvc/files/backups/
tsm maintenance backup -f <repository-backup>.tsbak -d

# Any filepath where current user (UNPRIVILEGED USER) can
write.
tsm settings export -f /var/opt/tableau/tableau_server-
/data/tabsvc/files/backups/<topology-conf-backup>.json

```

2. Copy the files created in previous step into the host machine. Change the file permission to have read-all permission set for both files.

```

docker cp my-server:/var/opt/tableau/tableau_server-
/data/tabsvc/files/backups/<repository-backup>.tsbak ./<re-
pository-backup>.tsbak
docker cp my-server:/var/opt/tableau/tableau_server-
/data/tabsvc/files/backups/<topology-conf-backup>.json
./<topology-conf-backup>.json
chmod a+r ./<repository-backup>.tsbak ./<topology-conf-
backup>.json

```

3. Store backup artifacts in a secure location. Follow the restore steps below when needed.

Restore inside Tableau Server Container

Backups from any supported Tableau Server version (container and non-container) can be restored inside the Tableau Server container.

Prerequisites

- Tableau Server backup file.
- Configuration json file containing both configuration and topology information.
- **Note:** You will likely need to change the backup files to have read-all permission set. Backup files are typically locked to the user that created the file, and this one will likely be different than the Tableau user running in the container.

Tableau Server on Linux Administrator Guide

```
docker run \  
-v <full-path-to-backup-file>:/docker/config/backup/backup-  
file.tsbak \  
-v <full-path-to-config-only-file>:/docker/config/config.json:ro  
\  
-e LICENSE_KEY=<key> \  
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

Notes:

- If you are restoring a multi-node system you must also start the other nodes in order for the restore automation to work. See the Multi-node Tableau Server in a Container section of this document for more information. Only the initial node requires the backup file, backup configuration file, and license.
- The backup files only need to be provided in the first run of the container. Once the server is initialized you do not need to keep mounting in the backup files.

Migrating from Tableau Server to Tableau Server in a Container

In order to migrate from a standard Tableau Server installation to Tableau Server in a Container you must use the backup and restore technique. Backups from any supported Tableau Server version (container and non-container) can be restored inside the Tableau Server container. See the Restore inside Tableau Server Container section above for more information.

Upgrading Tableau Server Versions

There are two ways to upgrade Tableau Server. The Upgrade-Image method listed in this section is the recommended solution. However, as a fallback it is also possible to upgrade Tableau Server using Backup/Restore.

Upgrading through Upgrade-Image method

The Upgrade Image is a Docker image that can be built using the `build-upgrade-image` script from the Tableau Server in a Container setup tool. The purpose of the image is

solely to upgrade the currently running Tableau Server in a Container.

Follow the below steps to do the upgrade.

1. Create an upgrade-image using the `build-upgrade-image` script. The new version's tableau server rpm is needed to build this container.
2. Shutdown the container currently running the Tableau Server.
3. Start the upgrade-image, mounting the same data directory from the container shut-down in the previous step.
4. The upgrade process takes a while, but the tableau server will be upgraded, check docker logs for upgrade process update. The container will shut down after the upgrade process.
5. Start a new Tableau Server in a Container of newer version. Mount the same directory from the previous steps.

Example:

Let's say we have a Tableau Server in a Container running Tableau Server. Here are some assumptions made in this example:

- I have valuable data, and I don't want to lose any data during the upgrade process. The data directory needs to be persisted outside the container.
- The container is named `my-server`. The docker image is named as `tableau-server-versionA`.
- The server version `my-server` is currently using is version A.
- The server version I want to upgrade to is version B.

1. Get tableau server rpm for version B. Create an upgrade-image.

```
# For all the options available in the script
./build-upgrade-image -h

# Frequently used command to create a upgrade-image
./build-upgrade-image --installer=<path to the tableau
server version B> -i tableau-server:versionA -o tableau-
server-upgrade:versionAB
```

2. Stop the `my-server` container.

Tableau Server on Linux Administrator Guide

```
docker stop my-server -t 120
```

3. Start the newly created image `tableau-server-upgrade:versionAB`. Mount the same data directory from the previously stopped container. The container starts the upgrade process to version B.

```
docker run --name my-upgrade-server \  
-v <data-dir mount from previous step>:/var/opt/tableau \  
...  
tableau-server-upgrade:versionAB
```

4. The container will stop once the upgrade is complete. Check the docker logs for upgrade process logs and make sure the upgrade process is a success. You can also check the exit code of the docker container, to make sure the upgrade process is completed successfully.

```
# The log file /var/opt/tableau/tableau_server-  
/logs/upgrade-console.log is created after 3-4 mins into  
the start of upgrade container. When the upgrade completes  
successfully, "upgrade is complete" log will be # seen.  
docker logs my-upgrade-server  
...  
...  
Verifying licensing state.  
Tableau Server has been upgraded to version  
near.20.0801.1050.  
>> The upgraded Tableau binary directory will be added to  
PATH for new shells. To get the  
>> updated path, either start a new session, or for bash  
users run:  
>> source /etc/profile.d/tableau_server.sh  
Starting service...  
Starting service...  
Job id is '12', timeout is 30 minutes.
```

```
Service was started successfully.
Status: RUNNING
Tableau Server is Running
upgrade is complete
```

5. Stop the my-upgrade-server container. Start the new version B of Tableau Server in a Container image and mount the data directory from the stopped my-upgrade-server container

```
# Stop the server.
docker stop my-upgrade-server -t 120

# Run the new version Hu
docker run --name my-upgraded-server \
-v <data-dir mount from previous step>:/var/opt/tableau \
...
...
tableau-server:versionB
```

Upgrading though Backup-Restore method

Follow the steps in the Backup and Restore Section of this document. The only adjustment needed to change a backup-restore operation into an upgrade operation, is to restore the backup on a new version of Tableau Server.

Multi-node Tableau Server in a Container

Multi-node Tableau Server in a Container refers to a single deployment of Tableau Server distributed across multiple nodes. Multi-node in this context is the same as Tableau Server multi-node where certain processes can be run on other nodes to increase capacity, compute power, etc. This is distinct from starting up multiple individual Tableau Server in a Container where each container is an independent server with its own distinct data.

Multi-node Tableau Server in a Container works much like a non-container Tableau Server multi-node installation, and uses the same underlying mechanism. To get an overview of

Tableau Server on Linux Administrator Guide

setting up a non-container Tableau Server multi-node installation, see [Distributed and High Availability Tableau Server Installations](#).

Here is an example:

Multi-Node Basic Usage

Initial node

Option 1: Use this if the server configuration (`CONFIG_FILE`) specifies a multi-node topology:

```
docker run \  
-v <network-shared-directory>:/docker/config/bootstrap \  
-v <full-path-to-config-file>:/docker/config/config.json:ro \  
-e LICENSE_KEY=<key> \  
-p 8080:8080 -p 8800-9000:8800-9000 -p 27000-27010:27000-27010 \  
--hostname=<static (internal) name of host machine> \  
-d <Tableau Server in a Container image ID or tag>
```

Option 2: Use this if you want a multi-node deployment even if server configuration does not specify multi-node topology:

```
docker run \  
-v <network-shared-directory>:/docker/config/bootstrap \  
-e LICENSE_KEY=<key> -e ALWAYS_WRITE_BOOTSTRAP_FILE=1 \  
-p 8080:8080 -p 8800-9000:8800-9000 -p 27000-27010:27000-27010 \  
  
--hostname=<static (internal) name of host machine> \  
-d <Tableau Server in a Container image ID or tag>
```

Additional node

```
docker run \  
-v <network-shared-directory>:/docker/config/bootstrap \  
-e BOOTSTRAP_INSTALL=1 \  
-p 8080:8080 -p 8800-9000:8800-9000 \  

```

```
--hostname=<static (internal) name of host machine> \  
-d <Tableau Server in a Container image ID or tag>
```

Exposing Licensing and TSM ports

In order for worker nodes to communicate with the primary instance we need to open up additional ports. You will need to allow traffic from other nodes on your primary Tableau Server in a Container instance in the following port ranges:

```
Service Ports: 8800-9000  
Postgres Port: 8060  
Licensing Ports: 27000-27010
```

Be careful how many ports you open: We recommend exposing only 200 ports, 8800-9000, instead of the Tableau Server default port range of 8000-9000, because exposing 1000 ports in Docker can negatively impact the performance and start up time of the Docker image. You can use a smaller or larger port range depending on how complex your Tableau Server topology is. Generally we do not recommend exposing fewer than 100 ports, otherwise you run the risk of services in a cluster being unable to talk to certain services. If you specify your own port range make sure you expose port 8850 (this is implicitly included in the 8800-9000). The port range is specified by setting the `PORT_RANGE_MIN` and `PORT_RANGE_MAX` environment variables.

Additional nodes will also need to expose the Service Port range (8800-9000), but not the Licensing Port range. It is important to note that these port ranges are only to allow Tableau Server inter-process communication. These ports should not be exposed to users or any other machines other than computers that are running Tableau Server in a Container for the same multi-node cluster.

These port rules are consistent with Tableau Server firewall documentation. For more information, see [Configure Local Firewall](#).

Resolving Hostnames

The multiple nodes of Tableau Server in a Container need to be run with consistent hostnames because Tableau Server does not handle dynamic hostname changes. When

running Tableau Server multi-node, those nodes will want to communicate with each other. Tableau Server nodes will attempt to reach each other using the hostnames that multi nodes' Tableau Server in a Container are configured to use. For example, if you run your initial node with a hostname of "initial", additional nodes will attempt to send traffic to a host called "initial". There are multiple ways you can configure images to resolve hostnames to other images. `/etc/hosts` file in each container to map the arbitrary container hostname (i.e. "initial") to the IP address that is actually running the other container.

Bootstrapping additional nodes

The initial Tableau Server container that runs as part of a cluster generates a bootstrap file that subsequent additional nodes need to use to join the cluster. After additional nodes are registered to the cluster's topology you can start assigning Tableau Server processes to run on them. This process can be fully automated. If you have provided a Tableau Server configuration file (commonly supplied by mounting a configuration file to the file path specified by `CONFIG_FILE`, default path: `/docker/config/config.json`) that specifies a Multi-node topology, the initial node will automatically wait until all additional nodes have registered. Once registered the Multi-node topology will be applied across the cluster.

Once the initial node in Tableau Server in a Container is fully running Tableau Server you can have it generate a bootstrap file for additional nodes:

```
docker exec -it <container-name> tsm topology nodes get-bootstrap-file -f $BOOTSTRAP_FILE
```

This command is automatically called for you if you set the value of `ALWAYS_WRITE_BOOTSTRAP_FILE` to 1.

Security Considerations

The bootstrap file contains server secrets that allow it to establish a TSM session with the initial node. This means if a malicious user obtained the file they could send TSM commands to the server for a period of time. The file itself also contains data that would enable the decryption of server configuration secrets. This file should be treated as sensitive and should be

accessible only by services and systems directly pertaining to establishing a multi-node deployment.

Bootstrap Expiration

Bootstrap files carry a limited-time session that lasts for 2 hours. In that window the additional nodes will not need to supply credentials to the initial node in order to join as an additional node. It is possible to use a bootstrap file once the session has expired, however it would mean needing to supply credentials to the initial node.

Transferring the Bootstrap File

The bootstrap file needs to be made available to and consumed by worker nodes' Tableau Server in a Container. The bootstrap file will need to be shared with all other nodes' Tableau Server in a Container that you want to as worker nodes for this deployment. This can be done in many different ways.

Transfer the file over a secure network

Part of your automation on the initial node can involve sending the file directly to additional nodes. This should be done using some secure file transfer client/tool. This is probably more useful in scenarios where multiple bootstrap files may be generated throughout the life of the initial node (possibly to add more additional nodes at a later time).

Use a network file mount

A network file mount shared by all the containers in a given deployment is another option.

Other

The end goal is to securely transfer a file produced by one container and transfer it to a specific set of other containers. So any method that achieves this and is secure is sufficient.

Starting additional nodes

To start up a Tableau Server in a Container additional node, simply start the container with the `BOOTSTRAP_INSTALL` environment variable set to 1.

This will tell the Tableau Server in a Container instance to sleep until a bootstrap file exists at the path specified by the `BOOTSTRAP_FILE` environment variable (which is also configurable). Refer to the environment variable table to view the default file path. To clarify, if you run a Tableau Server in a Container image in "additional node mode" the container will not start `supervisord` or any other process other than a bash script running as pid 1 that checks every 5 seconds to see if the bootstrap file exists. Once the file is present the Tableau Server in a Container will proceed to initialize as an additional node.

Configuring additional nodes

Configuring additional nodes to run a specific topology works the same as it does in a normal Tableau Server deployment. It also comes with the same requirements, which means adding new processes on a node may require a cluster-wide restart. For more information, see [Configure Nodes](#).

Tableau Server Feature Considerations

Some Tableau Server features works differently in containers. This section covers specific features that have special or different considerations in a container environment.

Active Directory

Set AD Domain Controller

If you plan on using Active Directory as an Identity Store for Tableau Server web pages and sites, there is an additional consideration to account for. Tableau Servers running in Linux environments dynamically determine which AD Domain Controller to communicate with by examining their IP subnet. Containers can be assigned arbitrary IP addresses, and in this case Tableau Server will not necessarily be able to use its IP address to find an appropriate domain controller. For this reason it may be necessary to configure a specific domain controller / hostname for Tableau Server to communicate with. To do this follow these steps:

1. Determine which domain controller you want Tableau Server to use and get the hostname.
2. Set the configuration key `wgserver.domain.ldap.hostname` to the hostname

using the standard Tableau Server Admin configuration options:

- Set the value in the json configuration file `CONFIG_FILE`.
- Use the TSM configuration command

```
tsm configuration set -k wgserver.domain.ldap.hostname
-v <hostname>
```

Import AD certificate to Tableau Server Keystore

By default Tableau Server in a container communicates with AD via StartTLS whenever simple bind is used. So when the container is run in this configuration, it is necessary to import the AD server certificate to the Tableau Server Keystore, otherwise server initialization will fail. To do this follow these steps:

1. Create a `pre-init-command` script (check Pre-initialization script section). Add the following line to add the AD certificate to tableau server keystore.

```
${INSTALL_DIR}/packages/repository.${SERVICE_
VERSION}/jre/bin -importcert -noprompt -alias startTlsCert
-file <mounted-certificate-path> -storetype JKS -storepass
changeit -keystore ${DATA_DIR}/-
config/tableauservicesmanagerca.jks
```

2. Mount the AD server certificate at the filepath provided for `-file` parameter in the `pre-init-command` script.

Alternatively, the default setting to communicating with AD via StartTLS can be disabled. Set `wgserver.domain.ldap.starttls.enabled` to `false` to disable the StartTLS. But it is not recommended.

Troubleshooting

If you encounter issues when running Tableau Server, there are a number of avenues one can pursue to find a solution. This section covers general Tableau Server troubleshooting advice, such as where to find logs and what they mean. It also covers some specific known scenarios and mitigation paths.

Tableau Server on Linux Administrator Guide

If you are working with Tableau Support to debug an issue, it can be helpful if you provide the following:

- Tableau Server logs (gathering these logs is explained below).
- Docker container `stdout` logs.
- Dockerfile of Tableau Server (if any customizations have been made).
- Deployment configuration including:
 - `Kubeconfig` (or any equivalent deployment config).
 - Static configuration files that configure the Tableau Server Container.

Installation and initialization failures

If you are initializing Tableau Server for the first time, or you have done a fresh install inside a container, the server will not recover by simply restarting the container. Every installation attempt should use a clean data directory. This may mean deleting persistent volume data from previous container runs. If you do this make sure to save logs and information that could be useful for debugging.

Debugging failing installation

Tableau Server containers are designed to exit when an installation failure is encountered. This pattern makes it easy to automate and identify when an install failure has occurred. However, it can make debugging challenging because the container will exit and not leave any examinable runtime state. If you would like to have a debug session inside a running container that is failing during initialization, follow these steps:

1. Prepare a fresh Tableau Server in a Container deployment.
2. Configure the container to run with the `TSM_ONLY=1` environment variable. The `TSM_ONLY=1` environment variable tells Tableau Server to only initialize TSM. This is the equivalent of just running the `initialize-tsm` script in a standard, non-container installation.
3. Run the Tableau Server container.
4. Open a shell inside the container.
5. You can now run TSM commands, even though Tableau Server has not been initialized. To resume the automation that takes place normally during initialization,

execute the `tsm-commands` script: `"${DOCKER_CONFIG}"/config/tsm-commands`

Tableau Support and Kubernetes

Tableau Server in a Container can be run using Kubernetes, but it is not a requirement to do so. Our expectation is that most customers will use Kubernetes or one of its associated managed cloud environments (EKS, AKS or GKS) to run and manage Tableau Server in a Container.

Kubernetes can be a complex environment to run and debug and often includes dependencies on individual companies infrastructure and setup. Because of this, Tableau Support cannot help customers resolve Kubernetes (or infrastructure deployment) issues associated with running Tableau Server in a Container. However, Tableau does support running Tableau Server in a Docker container. Therefore, if you are having issues with running Tableau Server in a Container using Kubernetes, Tableau Support can only go as far as validating that the Docker container works properly by itself.

For more information on how to run Tableau Server in a Container using Kubernetes, see this Github site: <https://github.com/tableau/tableau-server-in-kubernetes>.

Logs

Logs are an essential resource for finding, understanding, and solving problems in Tableau Server. They are useful for helping our support teams find the root the cause of issues you encounter. Logs may also be useful for your own debugging and troubleshooting.

Extracting All Logs

If you need to extract all logs for further debugging or to send to our support teams, there are a couple methods for retrieving this information.

Ziplogs

TSM can create a compressed archive containing all relevant server logs. You can trigger this by running the `tsm maintenance ziplogs` command. When the command completes it will report the filepath of the log archive. You will need to copy the archive using

Tableau Server on Linux Administrator Guide

whatever file transfer method works best for your situation. For details on ziplogs, see tsm maintenance ziplogs.

Example command executed inside container:

```
tsm maintenance ziplogs
```

Manual Tar Command

If you cannot run the ziplogs command, for example, if the server does not manage to reach a consistent state, you can still retrieve the logs by executing a tar command inside the container. You will need to copy out the archive using whatever file transfer method works best for your situation.

Example command executed inside container (writes the tar to a temp directory in the container's data directory):

```
tar -zcvf /var/opt/tableau/tableau_server/temp/<archive_
name>.tar.gz \
/var/opt/tableau/tableau_server/data/tabsvc/logs/. \
/var/opt/tableau/tableau_server/supervisord/ \
/var/opt/tableau/tableau_server/data/tabsvc/config/ \
/docker/.metadata.conf \
--exclude='*keystores' --exclude='*.jks' --exclude='*.tkr' \
--exclude='*asset_keys.yml' --exclude='*.ks' --exclude='*.ts' \
--exclude='*.crt' --exclude='*cacerts' --exclude='*.key'
```

Navigating Logs and Debugging Tips

There are common steps to diagnosing most problems in Tableau Server. If you are thinking of taking a look at your server logs, it can be helpful to break down what information to look for depending on where in the server life cycle the error occurred.

Starting The Container (initial / install)

If the container is crashing immediately or failing to install or initialize, check the following resources:

Container stdout

Examine `stdout` for the docker container. This is most often accessible by looking at the container output collected by your container orchestration system (for example, Kubernetes). Because Tableau Server is a multi-process system running inside a container, `stdout` is often not useful and will not report the root cause of the problem, unless there are catastrophic failures on startup. It is recommended you check `stdout` of the failing container before digging further into Tableau Server logs.

Example:

```
docker logs <container-name>
```

Tableau Server Container Startup Log

The Tableau Server Container startup log captures output from automation that is initializing, configuring, and starting Tableau Server. If you find your container is encountering issues while booting or running for the first time, this is the first log to check:

```
/var/opt/tableau/tableau_server/supervisord/run-tableau-server.log
```

Check the bottom of the log and see if there is a reported failure. Sometimes the error will be reported and be immediately obvious from the log. If the error is not clear from the log, it's possible the root cause is only visible in a stage-specific or service-specific log file. The logs listed below cover these possibilities.

Tableau Server Install Log

If the startup log indicates there was an issue with the automation handling the initialize TSM stage, check this log:

```
/var/opt/tableau/tableau_server/logs/app-install.log
```

Tableau Server Controller Log

If the startup log indicates there was an issue with the initializing and starting Server stage (CLI only), check the `tabadmincontroller` service log:

Tableau Server on Linux Administrator Guide

```
/var/opt/tableau/tableau_server-  
/data/tabsvc/logs/tabadmincontroller/tabadmincontroller_node1-  
0.log
```

This log file is for a specific service called tabadmincontroller. Tabadmincontroller is responsible for orchestrating the initialization and startup functionality in Server. This log can be complex and verbose. Errors in this log file may still not point to the root cause. Sometimes the errors are caused by services that tabadmincontroller is relying on to complete a certain task. Check the Server runtime section below for more details.

Service Logs - Server Runtime

If Tableau Server encounters issues during normal runtime or issues with services failing to complete tasks or are down, you can check service logs for more information. Every service running as part of Tableau Server has a service log file. If you know which service you would like to examine, you can find that service's logs under this general directory:

```
/var/opt/tableau/tableau_server/data/tabsvc/logs/<service_name>
```

Provide the name of the service in the <service_name> arg of the file path. Any service can write multiple kinds of logs files. Also, if you have more than one of the same service running (more than one instance), all service logs will be written into the same service directory.

General Service-Specific Log File Classifications

This table covers the most common Service Log file names, types, and descriptions for Tableau Server services. The "Failure types" column indicates which log files are likely to be useful in a given failure scenario.

Name	Filename format	Description	Failure types	Example
Control-App	control_<service_name>_<node_id>-<instance_	Contains information of the control-app process which is responsible for installing and configuring a service.	Install, Configure, Status	control_back-grounder_node1-0.log

Name	Filename format	Description	Failure types	Example
	id>.log	This is often the first log written related to a service. For service install and configure failures, look here first.		
Service log	<service_name>_<node_id>-<instance_id>.log	Primary log for a running service. Most often this log contains output from the spring/-java application layer.	Start, Runtime, Status	backgrounder_node1-1.log
Stdout log	stdout_<service_name>_<instance_id>.log	Contains <code>stdout</code> output for the service. Most services do not output much content to <code>stdout</code> and instead write to the primary log. Sometimes this log can contain useful information when a service exits.	Start, Stop	stdout_back-grounder_0.log
NativeAPI log	nativeapi_<service_name>_<instance_id>.txt	Some services run a native code layer. This log captures that portion of the application's runtime.	Licensing, Start, Runtime, Status	nativeapi_backgrounder_1-1_2021_05_10_00_00_00.txt
Tomcat log	tomcat_<service_name>_<node_id>-	This is only for services that run inside a tomcat container and contains tomcat logs. It	Network, Start	tomcat_back-grounder_node1-0.2021-05-10.log

Name	Filename format	Description	Failure types	Example
	<instance_id>.log	rarely provides information regarding service failure. It can be useful to debug some network issues.		

Stopped Container

If the container is stopped or is otherwise difficult to execute commands in, you can still look at the logs if the data directory of the server is externalized to a mounted volume. Otherwise, only the `stdout` of the container will be examinable in the container's orchestration system, which is often not going to contain the root cause.

Failure to set authentication properties

There appears to be an issue with setting authentication properties in Tableau Server without the identity store getting set first. To work around this issue just set the identity store in the pre-initialization hook.

1. Create a file called `./customer-files/pre_init_command` in the Tableau Server Image Build Tool `customer-files` directory and edit it to contain:

```
#!/bin/bash
tsm configuration set -k wgserver.authenticate -v local --force-keys
```

2. Set the script to be executable.

```
chmod +x ./customer-files/pre_init_command
```

3. Build and run the image.

Failure during fresh startup (e.g. why isn't Tableau Server starting?)

- If you are encountering issues with Tableau Server initializing or starting up there are a number of troubleshooting options that may help uncover the issue.

- If the container cannot start at all, you'll want to check the stdout from the PID 1 process using `docker logs <container-name> command`.
- If the container is running but Tableau Server does not seem to be initializing or running properly, the second place to check for errors is this file:

```
`${DATA_DIR}/supervisord/run-tableau-server.log
```

Example:

```
docker exec -it <container-name> bash -c 'cat `${DATA_DIR}/supervisord/run-tableau-server.log'
```

This log file contains all events orchestrated by the tableau container initializing service that is handling the startup of tableau server as well as executing any setup scripts or custom configuration that you may have provided in the container. Most startup errors will report issues here. Sometimes if the error is related to a TSM or Tableau Server process it will suggest another log file to look at for more detailed information.

Failure during restart or starting a container with existing data

Server Won't Start PostGRES (or other processes)

When data is persisted outside the container and you are starting another Tableau Server in a Container image instance using that old data, it is important to note that the internal hostname of the new container must match the hostname of the container that initialized the persisted data. Tableau Server does not handle dynamic hostname changes well and starting up a new container with a different internal hostname is effectively causing that scenario.

The remedy is to simply make sure the container's hostname is set to the same value as the container that was previously running with that data. This is not to be confused with Multi-node, workers can (and probably should) have different hostnames from each other. What matters is when a given container is restarted or killed the subsequent container must have the same hostname as its predecessor.

Deployment Configuration Examples

Docker

Tableau Server in a Container Basic Usage

```
docker run \  
-e LICENSE_KEY=<key> \  
-p 8080:8080 \  
--hostname=<static (internal) name of host machine> \  
-d <Tableau Server in a Container image ID or tag>
```

Tableau Server in a Container Basic Usage with Automated Initial Admin User

```
docker run \  
-e LICENSE_KEY=<key> \  
-e TABLEAU_USERNAME=<myadmin> \  
-e TABLEAU_PASSWORD_FILE=/etc/tableau-admin-secret \  
-v <full-path-to-pw-file>:/etc/tableau-admin-secret \  
--hostname=<static (internal) name of host machine> \  
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

TSM only mode

```
docker run \  
-e TSM_ONLY=1 \  
--hostname=<static (internal) name of host machine> \  
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

Multi-Node Basic Usage

Initial Node

Option 1: Use this if the server configuration (`CONFIG_FILE`) specifies a multi-node topology:

```
docker run \  
-v <network-shared-directory>:/docker/config/bootstrap \  
-v <full-path-to-config-file>:/docker/config/config.json:ro \  
-d <Tableau Server in a Container image ID or tag>
```

```
-e LICENSE_KEY=<key> \
-p 8080:8080 -p 8800-9000:8800-9000 -p 27000-27010:27000-27010
\
--hostname=<static (internal) name of host machine> \
-d <Tableau Server in a Container image ID or tag>
```

Option 2: Use this if you want a multi-node deployment even if server configuration does not specify multi-node topology:

```
docker run \
-v <network-shared-directory>:/docker/config/bootstrap \
-e LICENSE_KEY=<key> -e ALWAYS_WRITE_BOOTSTRAP_FILE=1 \
-p 8080:8080 -p 8800-9000:8800-9000 -p 27000-27010:27000-27010
\
--hostname=<static (internal) name of host machine> \
-d <Tableau Server in a Container image ID or tag>
```

Additional node

```
docker run \
-v <network-shared-directory>:/docker/config/bootstrap \
-e BOOTSTRAP_INSTALL=1 \
-p 8080:8080 -p 8800-9000:8800-9000 \
--hostname=<static (internal) name of host machine> \
-d <Tableau Server in a Container image ID or tag>
```

Externalize Data Usage

```
docker run \
-v <empty-data-dir>:/var/opt/tableau \
-e LICENSE_KEY=<key> \
--hostname=<static (internal) name of host machine> \
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

Init-Container Basic Usage

Init Container

Tableau Server on Linux Administrator Guide

```
docker run \  
-v <empty-data-dir>:/var/opt/tableau \  
-e LICENSE_KEY=<key> \  
-e INIT_CONTAINER=1 \  
--hostname=<static (internal) name of host machine> \  
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

Run Container

```
docker run \  
-v <empty-data-dir>:/var/opt/tableau \  
--hostname=<static (internal) name of host machine> \  
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

Basic Restore from Backup Single-Node

```
docker run \  
-v <full-path-to-backup-file>:/docker/config/backup/backup-  
file.tsbak \  
-v <full-path-to-config-only-file>:/docker/config/config.json:ro  
\  
-e LICENSE_KEY=<key> \  
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

Docker-Compose

```
version: '3.2'  
services:  
  tableau-server:  
    hostname: localhost  
    volumes:  
      - <your-tsm-command-file>:/docker/config/tsm-com-  
mands:ro  
      - <your-config-file  
>:/docker/config/config.json:ro  
    ports:  
      - "8080:8080"
```

```
image: ${IMAGE_NAME}
environment:
  - LICENSE_KEY=<license-key>
```

Install Tableau Server in the Cloud

Even if you don't have your own infrastructure and server hardware, you can deploy an enterprise-level Tableau Server installation in the cloud. Building a cloud-based solution has many benefits over an on-premises installation. For example, the overall total cost of ownership for building a Tableau Server solution in the cloud is normally much less than a similar on-premises solution because you don't have to buy all of the expensive hardware. In addition, the cloud can provide better uptime, reliability, and fault-tolerance, especially if you deploy your solution across different regions and availability zones.

Looking for Tableau Server on Windows? See [Install Tableau Server in the Cloud](#).

You can build and scale your Tableau environment in the following cloud environments:

- **Tableau Online** - If you don't want to install software or purchase hardware, you can choose analytics as a service with Tableau Online, the Tableau-hosted environment in the cloud. For more information, see [Tableau Online](#).
- **Amazon Web Services** - You install and manage Tableau Server on Amazon Web Services (AWS). For more information, see [Install Tableau Server in the AWS Cloud](#).
- **Google Cloud Platform** - You install and manage Tableau Server on the Google Cloud Platform. For more information, see [Install Tableau Server on the Google Cloud Platform](#).

- **Microsoft Azure** - You install and manage Tableau Server on Microsoft Azure. For more information, see [Install Tableau Server on Microsoft Azure](#).
- **Alibaba Cloud** - You install and manage Tableau Server on Alibaba Cloud. For more information, see [Install Tableau Server in the Alibaba Cloud](#).

Post Installation Tasks

After you install Tableau Server, you can perform other configuration tasks, such as reviewing the security hardening checklist, configuring server event notifications, configuring the data cache for views, and configuring server crash reporting.

Security Hardening Checklist

The following list provides recommendations for improving the security ("hardening") of your Tableau Server installation.

Looking for Tableau Server on Windows? See [Security Hardening Checklist](#).

Installing security updates

Security updates are included in the latest versions and maintenance releases (MR) of Tableau Server. You cannot install security updates as patches. Rather, you must upgrade to a current version or MR to update Tableau Server with the latest security fixes.

Always reference the most current version of this topic after upgrading. The current version includes `/current/` in the topic URL.

For example, the US version URL is: https://help.tableau.com/current/server/en-us/security_harden.htm.

1. Update to the current version

We recommend that you always run the latest version of Tableau Server. Additionally, Tableau periodically publishes maintenance releases of Tableau Server that include fixes for known security vulnerabilities. (Information regarding known security vulnerabilities can be found on the Tableau [Security Bulletins](#) page and the [Salesforce Security Advisories](#) page.) We recommend that you review maintenance release notifications to determine whether you should install them.

To get the latest version or maintenance release of Tableau Server, visit the [Customer Portal](#) page.

2. Configure SSL/TLS with a valid, trusted certificate

Secure Sockets Layer (SSL/TLS) is essential for helping to protect the security of communications with Tableau Server. Configure Tableau Server with a valid, trusted certificate (not a self-signed certificate) so that Tableau Desktop, mobile devices, and web clients can connect to the server over a secured connection. For more information, see [SSL](#).

3. Disable older versions of TLS

Tableau Server uses TLS to authenticate and encrypt many connections between components and with external clients. External clients, such as browsers, Tableau Desktop, Tableau Mobile connect to Tableau using TLS over HTTPS. Transport layer security (TLS) is an improved version of SSL. In fact, older versions of SSL (SSL v2 and SSL v3) are no longer considered to be adequately secure communication standards. As a result, Tableau Server does not allow external clients to use SSL v2 or SSL v3 protocols to connect.

We recommend that you allow external clients to connect to Tableau Server with TLS v1.3 and TLS v1.2.

TLS v1.2 is still regarded as a secure protocol and many clients (including Tableau Desktop) do not yet support TLS v1.3.

TLS v1.3 capable clients will negotiate TLS v1.3 even if TLS v1.2 is supported by the server.

The following `tsm` command enables TLS v1.2 and v1.3 (using the "all" parameter) and disables SSL v2, SSL v3, TLS v1, and TLS v1.1 (by prepending the minus [-] character to a given protocol). TLS v1.3 is not yet supported by all components of Tableau Server.

```
tsm configuration set -k ssl.protocols -v 'all -SSLv2 -SSLv3 -
TLSv1 -TLSv1.1'
```

```
tsm pending-changes apply
```

You can also modify the default list of cipher suites that Tableau Server uses for SSL/TLS sessions. For more information see the *ssl.ciphersuite* section at `tsm configuration set Options`.

4. Configure SSL encryption for internal traffic

Configure Tableau Server to use SSL to encrypt all traffic between the Postgres repository and other server components. By default, SSL is disabled for communications between server components and the repository. We recommend enabling internal SSL for all instances of Tableau Server, even single-server installations. Enabling internal SSL is especially important for multi-node deployments. See [Configure SSL for Internal Postgres Communication](#).

5. Enable firewall protection

Tableau Server was designed to operate inside a protected internal network.

Important: Do not run Tableau Server, or any components of Tableau Server on the internet or in a DMZ. Tableau Server must be run within the corporate network protected by an internet firewall. We recommend configuring a reverse proxy solution for internet clients that need to connect to Tableau Server. See [Configuring Proxies for Tableau Server](#).

A local firewall should be enabled on the operating system to protect Tableau Server in single and multi-node deployments. In a distributed (multi-node) installation of Tableau Server, communication between nodes does not use secure communication. Therefore, you should enable firewalls on the computers that host Tableau Server. See [Configure Local Firewall](#).

To prevent a passive attacker from observing communications between nodes, configure a segregated virtual LAN or other network layer security solution.

See [Tableau Services Manager Ports](#) to understand which ports and services Tableau Server requires.

6. Restrict access to the server computer and to important directories

Tableau Server configuration files and log files can contain information that is valuable to an attacker. Therefore, restrict physical access to the machine that is running Tableau Server. In addition, make sure that only authorized and trusted users have access to the Tableau Server files in the `/var/opt/tableau/tableau_server/` directory.

7. Generate fresh secrets and tokens

Any Tableau Server service that communicates with repository or the cache server must first authenticate with a secret token. The secret token is generated during Tableau Server setup. The encryption key that internal SSL uses to encrypt traffic to Postgres repository is also generated at during setup.

We recommend that after you install Tableau Server, you generate new encryption keys for your deployment.

These security assets can be regenerated with the `tsm security regenerate-internal-tokens` command.

Run the following commands:

```
tsm security regenerate-internal-tokens
```

```
tsm pending-changes apply
```

8. Disable services that you're not using

To minimize the attack surface of the Tableau Server, disable any connection points that are not needed.

JMX Service

JMX is disabled by default. If it's enabled but you're not using it, you should disable it by using the following:

```
tsm configuration set -k service.jmx_enabled -v false
```

```
tsm pending-changes apply
```

9. Verify session lifetime configuration

By default, Tableau Server does not have an absolute session timeout. This means that browser-based client (Web authoring) sessions can remain open indefinitely if the Tableau Server inactivity timeout is not exceeded. The default inactivity timeout is 240 minutes.

If your security policy requires it, you can set an absolute session timeout. Be sure to set your absolute session timeout in a range that allows the longest-running extract uploads or workbook publishing operations in your organization. Setting the session timeout too low may result in extract and publishing failures for long-running operations.

To set the session timeout run the following commands:

```
tsm configuration set -k wgserver.session.apply_lifetime_limit -v true
```

```
tsm configuration set -k wgserver.session.lifetime_limit -v value, where value is the number of minutes. The default is 1440, which is 24 hours.
```

```
tsm configuration set -k wgserver.session.idle_limit -v value, where value is the number of minutes. The default is 240.
```

```
tsm pending-changes apply
```

Sessions for connected clients (Tableau Desktop, Tableau Mobile, Tableau Prep Builder, Bridge, and personal access tokens) use OAuth tokens to keep users logged in by re-establishing a session. You can disable this behavior if you want all Tableau client sessions to be solely governed by the browser-based session limits controlled by the commands above. See [Disable Automatic Client Authentication](#).

10. Configure a server allowlist for file-based data sources

By default, Tableau Server allows authorized Tableau Server users to build workbooks that use files on the server as file-based data sources (such as spreadsheets). In this scenario, files are accessed by the `tableau` system account.

To prevent unwanted access to files, we recommend that you configure allowlist functionality. This lets you limit `tableau` account access to just the directory paths where you host data files.

1. On the computer running Tableau Server, identify the directories where you will host data source files.

Important Make sure the file paths you specify in this procedure exist on the server. If the paths do not exist when the computer starts, Tableau Server will not start.

2. Run the following commands:

```
tsm configuration set -k native_api.allowed_paths -v  
"path" , where path is the directory to add to the allowlist. All subdirectories of the  
specified path will be added to the allowlist. If you want to specify multiple paths, sep-  
arate them with a semicolon, as in this example:
```

```
tsm configuration set -k native_api.allowed_paths -v "/data-  
sources;/HR/data"
```

```
tsm pending-changes apply
```

11. Enable HTTP Strict Transport Security for web browser clients

HTTP Strict Transport Security (HSTS) is a policy configured on web application services, such as Tableau Server. When a conforming browser encounters a web application running HSTS, then all communications with the service must be over a secured (HTTPS) connection. HSTS is supported by major browsers.

For more information about how HSTS works and the browsers that support it, see The Open Web Application Security Project web page, [HTTP Strict Transport Security Cheat Sheet](#).

To enable HSTS, run the following commands on Tableau Server:

```
tsm configuration set -k gateway.http.hsts -v true
```

By default, HSTS policy is set for one year (31536000 seconds). This time period specifies the amount of time in which the browser will access the server over HTTPS. You should consider setting a short max-age during initial roll-out of HSTS. To change this time period, run `tsm configuration set -k gateway.http.hsts_options -v max-age=
e=<seconds>`. For example, to set HSTS policy time period to 30 days, enter `tsm configuration set -k gateway.http.hsts_options -v max-age=2592000`.

```
tsm pending-changes apply
```

12. Disable Guest access

Core-based licenses of Tableau Server include a Guest user option, which allows any user in your organization to see and interact with Tableau views embedded in web pages.

Guest user access is enabled by default on Tableau Servers deployed with core-based licensing.

Guest access allows users to see embedded views. The Guest user cannot browse the Tableau Server interface or see server interface elements in the view, such as user name, account settings, comments, and so on.

If your organization has deployed Tableau Server with core licensing and Guest access is not required, then disable Guest access.

You can disable Guest access at the server or site level.

You must be a server administrator to disable the Guest account at either the server or the site level.

To disable Guest access at the server level:

1. In the site menu, click **Manage All Sites** and then click **Settings > General**.
2. For **Guest Access**, clear the **Enable Guest account** check box.
3. Click **Save**.

To disable Guest access for a site:

1. In the site menu, select a site.
2. Click **Settings**, and on the Settings page, clear the **Enable Guest account** check box.

For more information, see [Guest User](#).

13. Set referrer-policy HTTP header to 'same-origin'

Beginning in 2019.2, Tableau Server includes the ability to configure Referrer-Policy HTTP header behavior. This policy is enabled with a default behavior that will include the origin URL for all "secure as" connections (`no-referrer-when-downgrade`), which sends origin referrer information only to like connections (HTTP to HTTP) or those that are more secure (HTTP to HTTPS).

However, we recommend setting this value to `same-origin`, which only sends referrer information to same-site origins. Requests from outside the site will not receive referrer information.

To update the `referrer-policy` to `same-origin`, run the following commands:

```
tsm configuration set -k gateway.http.referrer_policy -v same-  
origin
```

```
tsm pending-changes apply
```

For more information about configuring additional headers to improve security, see [HTTP Response Headers](#).

14. Configure TLS for SMTP connection

Beginning in 2019.4, Tableau Server includes the ability to configure TLS for the SMTP connection. Tableau Server only supports STARTTLS (Opportunistic or Explicit TLS).

Tableau Server can be optionally configured to connect to a mail server. After configuring SMTP, Tableau Server can be configured to email server administrators about system failures, and email server users about subscribed views and data-driven alerts.

To configure TLS for SMTP:

1. Upload a compatible certificate to Tableau Server. See `tsm security custom-cert add`.
2. Configure TLS connection using TSM CLI.

Run the following TSM commands to enable and force TLS connections to the SMTP server and to enable certificate verification.

```
tsm configuration set -k svcmonitor.notification.smtp.ssl_  
enabled -v true
```

```
tsm configuration set -k svcmonitor.notification.smtp.ssl_  
required -v true
```

```
tsm configuration set -k svcmonitor.notification.smtp.ssl_
check_server_identity -v true
```

By default, Tableau Server will support TLS versions 1, 1.1, and 1.2, but we recommend that you specify the highest TLS version that the SMTP server supports.

Run the following command to set the version. Valid values are `SSLv2Hello`, `SSLv3`, `TLSv1`, `TLSv1.1`, and `TLSv1.2`. The following example sets the TLS version to version 1.2.:

```
tsm configuration set -k svcmonitor.notification.smtp.ssl_
versions -v "TLSv1.2"
```

For more information about other TLS configuration options, see [Configure SMTP Setup](#).

- Restart Tableau Server to apply changes. Run the following command:

```
tsm pending-changes apply
```

15. Configure SSL for LDAP

If your Tableau Server deployment is configured to use a generic LDAP external identity store, we recommend configuring SSL to protect authentication between Tableau Server and your LDAP server. See [Configure Encrypted Channel to LDAP External Identity Store](#).

If your Tableau Server deployment is configured to use Active Directory, we recommend enabling Kerberos to protect authentication traffic. See [Kerberos](#).

Change List

Date	Change
May 2018	Added clarification: Do not disable REST API in organizations that are running Tableau Prep.
May 2019	Added recommendation for referrer-policy HTTP header.

June 2019	Removed recommendation to disable Triple-DES. As of version 2019.3, Triple-DES is no longer a default supported cipher for SSL. See What's Changed - Things to Know Before You Upgrade .
January 2020	Added recommendation to configure TLS for SMTP.
February 2020	Added recommendation to configure SSL for LDAP server.
May 2020	Added TLS v1.3 to the disabled list of TLS ciphers. Added clarification to introduction about topic versioning.
October 2020	Added TLS v1.3 as a default supported cipher.
January 2021	Added clarification: All products enabled by the Data Management license require REST API.
February 2021	Removed recommendation to disable REST API. The API is now used internally by Tableau Server and disabling it may limit functionality.

Configure SMTP Setup

Tableau Server can email server administrators about system failures, and email server users about subscribed views and data-driven alerts. First, however, you need to configure the SMTP server that Tableau Server uses to send email. After configuring SMTP, complete the steps to configure notifications (Configure Server Event Notification), then when you start or restart the server, it will trigger an email notification, which confirms that you have set up notifications correctly.

Configuring SMTP requires that you restart Tableau Server services.

Secure SMTP

To enable and configure TLS for SMTP, you must use the TSM CLI as described in this topic. Tableau Server only supports STARTTLS (Opportunistic or Explicit TLS).

If your organization does not use public certificates for verifying TLS connections, then you can upload a private certificate to Tableau Server to verify trusted connections. For more information, see the `tsm security custom-cert add` command.

You may also configure SMTP TLS for encryption-only by disabling the certificate validation process. For more information, see the section, *Configuration file reference*, in the *Use the TSM CLI* tab below.

Use the TSM web interface

1. Open TSM in a browser:

`https://<tsm-computer-name>:8850`. For more information, see [Sign in to Tableau Services Manager Web UI](#).

2. Click **Notifications** on the **Configuration** tab and click **Email Server**.
3. Enter the SMTP configuration information for your organization:

The screenshot shows the 'CONFIGURATION' tab selected in the top navigation bar. Below it, the 'Notifications' section is active, with 'Email Server' selected. The 'Configure email server' form contains the following fields and values:

- SMTP server address: `smtp.example.lan`
- Username: `tableau-notify@example.lan`
- Password: `*****`
- Port Number: `25 (Default)`
- Send all emails from: `no-reply@example.lan`
- Send server health email to: `tableau-health@example.lan`
- Tableau Server URL: `https://tableau.example.lan`

At the bottom of the form, there are two buttons: 'Cancel' and 'Save Pending Changes'.

4. Click **Save Pending Changes** after you've entered your configuration information.
5. Click **Pending Changes** at the top of the page:



6. Click **Apply Changes and Restart**.
7. Run the `tsm email test-smtp-connection` to view and verify the connection configuration. See `tsm email test-smtp-connection`.

Use the TSM CLI

For the initial configuration of SMTP, we recommend that you use the configuration file template below to create a json file. You can also set any single configuration key listed below with the syntax described in `tsm configuration set`.

1. Copy the following json template to a file.

Important: The template below includes common options for most deployments. After you copy the template to a text file, you must edit the option values for your SMTP server requirements. You may need to remove or add options. See the reference section that follows for more information about all supported SMTP key options.

```
{
  "configKeys": {
    "svcmonitor.notification.smtp.server": "SMTP server host
name",
    "svcmonitor.notification.smtp.send_account": "SMTP user
name",
    "svcmonitor.notification.smtp.port": 443,
    "svcmonitor.notification.smtp.password": "SMTP user
account password",
    "svcmonitor.notification.smtp.ssl_enabled": true,
    "svcmonitor.notification.smtp.from_address": "From email
```

```

address",
    "svcmonitor.notification.smtp.target_addresses": "To
email address1,address2",
    "svcmonitor.notification.smtp.canonical_url": "Tableau
Server URL"
    }
}

```

2. Run the `tsm settings import -f file.json` to pass the json file with the appropriate values to Tableau Services Manager to configure Tableau Server for SMTP. Tableau Services Manager will validate the entity values.
3. Run the `tsm pending-changes apply` command to apply the changes. See `tsm pending-changes apply`.
4. Run the `tsm email test-smtp-connection` to view and verify the connection configuration. See `tsm email test-smtp-connection`.

SMTP CLI configuration reference

This table lists all of the options that can be used to configure SMTP with TSM CLI.

Option	Description
<code>svc-mon-itor.notification.smtp.server</code>	Address of SMTP server. Example: "svc-mon-itor.notification.smtp.server": "mail.example.com"
<code>svc-mon-</code>	User name for SMTP account.

Option	Description
<code>itor.notification.smtp.send_account</code>	
<code>svc-monitor.notification.smtp.port</code>	Port number for SMTP server. The default is 25.
<code>svc-monitor.notification.smtp.password</code>	Password for SMTP server account. Example: <pre>"svc-monitor.notification.smtp.password": "password"</pre>
<code>svc-monitor.notification.smtp.ssl_enabled</code>	Specifies whether the connection to the SMTP server is encrypted. The default is false.
<code>svc-monitor.notification.smtp.ssl_required</code>	If enabled, Tableau Server will refuse to connect to SMTP servers without using TLS. The <code>svc-monitor.notification.smtp.ssl_enabled</code> option must also be set to true. The default is false.
<code>svc-monitor</code>	If set to true, Tableau Server will check the SMTP server identity as specified by RFC 2595 .

Option	Description
<pre>itor.no- tification.smtp.ssl_check_ server_identity</pre>	<p>These additional checks based on the content of the server's certificate are intended to prevent man-in-the-middle attacks.</p> <p>The default is false.</p>
<pre>svc- mon- itor.no- tification.smtp.ssl_trust_ all_hosts</pre>	<p>When using TLS, trust certificates from all mail servers, ignoring the validity of the certificate's chain of trust. By setting this key to true, TLS will be used only to encrypt the traffic to the SMTP host.</p> <p>The default is false.</p>
<pre>svc- mon- itor.no- tification.smtp.ssl_ ciphers</pre>	<p>The default and supported sets of cipher suites is defined by the version of JDK that is installed with Tableau Server. See the section below, TLS ciphers, for a list of supported and default ciphers.</p> <p>To update the cipher suites used by Tableau Server for SMTP TLS connections, enter a white space-separated list of cipher suites for this value. For example, "TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256 TLS_DHE_DSS_WITH_AES_128_GCM_SHA256 TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384".</p>
<pre>svc- mon- itor.no-</pre>	<p>The default TLS versions enabled on this version of Tableau Server are TLSv1, TLSv1.1, TLSv1.2 and TLSv1.3.</p>

Option	Description
<code>tification.smtp.ssl_versions</code>	<p>TLS version support is defined by the version of JDK that is installed with Tableau Server.</p> <p>Supported versions of TLS are <code>SSLv2Hello</code>, <code>SSLv3</code>, <code>TLSv1</code>, <code>TLSv1.1</code>, <code>TLSv1.2</code>, <code>TLSv1.3</code>.</p> <p>To update the versions used by Tableau Server for SMTP TLS connections, enter a white space-separated list of versions for this value. For example, <code>"TLSv1.2 TLSv1.3"</code>.</p>
<code>svc-monitor.notification.smtp.from_address</code>	<p>Email address that will send an notification if there's a system failure. The email address must have valid syntax (for example, <code>ITalerts@bigco.com</code> or <code>noreply@mycompany</code>), but it does not have to be an actual email account on Tableau Server. (Some SMTP servers may require an actual email account, however.)</p> <div data-bbox="751 1140 1349 1360" style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;"> <p>Note: You can override the system-wide email address on a per-site basis. For more information, see What is a site.</p> </div> <p>Example:</p> <pre>"svc-monitor.notification.smtp.from_address": "donot-reply@example.com"</pre>
<code>svc-</code>	<p>Email address to receive notifications. If email</p>

Option	Description
<pre>monitor.notification.smtp.target_ addresses</pre>	<p>notifications are enabled, you need to include at least one address. Separate multiple addresses with commas.</p> <p>Example:</p> <pre>"svc- monitor.notification.smtp.target_ addresses": "iluvdata@example.com"</pre>
<pre>svc- monitor.notification.smtp.canonical_ url</pre>	<p>URL of the Tableau Server. Enter <code>http://</code> or <code>https://</code>, followed by the name or IP address of the Tableau server. Used in the footer of subscription email.</p> <p>Example:</p> <pre>"svc- monitor.notification.smtp.canonical_ url": "http://myserver.example.com"</pre>

TLS ciphers

The following is a list of TLS ciphers that are supported by the JDK that is included with Tableau Server. In this version of Tableau Server, all of these ciphers are enabled by default. You can specify a custom list of ciphers for your SMTP configuration by entering a white-space separated list with the option, `svcmonitor.notification.smtp.ssl_ciphers`, as described in the table above.

<pre>TLS_RSA_WITH_AES_128_CBC_ SHA256</pre>	<pre>TLS_ECDH_RSA_WITH_AES_256_ GCM_SHA384</pre>
---	--

TLS_DHE_DSS_WITH_AES_256_GCM_SHA384	TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256
TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA	TLS_RSA_WITH_AES_256_CBC_SHA256
TLS_RSA_WITH_AES_128_GCM_SHA256	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
TLS_DHE_DSS_WITH_AES_128_CBC_SHA256	TLS_DHE_DSS_WITH_AES_256_CBC_SHA
TLS_DHE_RSA_WITH_AES_128_CBC_SHA256	TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
TLS_RSA_WITH_AES_256_GCM_SHA384	TLS_EMPTY_RENEGOTIATION_INFO_SCSV
TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	TLS_DHE_DSS_WITH_AES_256_CBC_SHA256
TLS_RSA_WITH_AES_256_CBC_SHA	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256	TLS_DHE_RSA_WITH_AES_256_CBC_SHA

TLS_DHE_DSS_WITH_AES_128_GCM_SHA256	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	TLS_ECDH_RSA_WITH_AES_128_CBC_SHA
TLS_DHE_RSA_WITH_AES_256_GCM_SHA384	TLS_DHE_RSA_WITH_AES_128_CBC_SHA
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256	TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256
TLS_ECDH_RSA_WITH_AES_256_CBC_SHA	TLS_DHE_DSS_WITH_AES_128_CBC_SHA
TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
TLS_RSA_WITH_AES_128_CBC_SHA	TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	TLS_AES_256_GCM_SHA384
TLS_AES_128_GCM_SHA256	

Files and Permissions in TSM

This topic covers the permissions requirements needed for Tableau Services Manager (TSM) to access and use files. This information is intended for server administrators. This topic does *not* address permissions used for managing users and content within Tableau Server (permissions for content and users). For information related to these permissions, see [Permissions](#).

During installation of TSM and Tableau Server, an unprivileged user (*tableau*) is created and added to a server authorized group (*tableau*). This user account enables the work done by TSM and Tableau Server processes. You can change the user and group during installation. For more information, see Identity Store.

Permissions requirements for TSM apply to both files, and to the directories in which the files are placed. When TSM creates and manages files, the files get put into specific default locations with the necessary permissions and you don't need to worry about setting permissions. When you create, copy, or move files yourself, or when you put files into non-default locations, you need to be aware of permission requirements so that TSM can properly access the files. Common cases (For information about using non-default locations, see [tsm File Paths.](#))

General rules for permissions and TSM are:

- **Files**—If the *tableau* group has access to a file (if it is the group owner and has read access to the file), the users in the group have access to the file. An alternate approach is to give "other" read access.
- **Directories**—If the *tableau* group has read and execute access to the directory that contains a file, and any parent directories of that directory, the users in the group have access to the file.

Situations that may require you to adjust permissions include server backup files and site import archives that you copy from a different computer or to a non-default location, customization files such as logos or images, and security certificates such as SSL certificates.

For example, if you migrate from Tableau Server on Windows to Tableau Server on Linux, you use a backup created in Windows to restore data to your Linux server. Because this backup file isn't created by TSM, it may not have the correct permissions for the restore process to access it. You need to make sure the backup file and the directory structure you copy it into have the proper permissions. Similarly, if you are copying files like certificates to additional nodes in a cluster, you need to make sure the files and the directories you copy them into have the permissions the *tableau* user needs in order to access them.

Setting permissions for individual files

If you are using a file you copy to one of the default locations created by TSM, you need to make sure the ownership and permissions on the file allow TSM access by giving the *tableau* user read access. You can do this in one of two ways:

- You can give the *tableau* user read access by giving the *tableau* group (in a default installation) read and execute access to a file using the `chgrp` and `chmod` commands. For example:

```
chgrp tableau <backup>.tsbak
```

```
chmod g+rx <backup>.tsbak
```

- Alternately, you can give world read and execute access to the file:

```
chmod o+rx <backup>.tsbak
```

Setting permissions for directories

In addition to setting the proper permissions on the files themselves, TSM also needs permissions for the directory that contains the file, as well as any parent directories. If you are using a non-default location for files that TSM will access, you will need to make sure permissions for the parent directory or directories that contain the file allow read and execute access.

You can address this issue in a couple of ways:

- Change group ownership of the directory to the *tableau* group, and add group read and execute permission to the directory. Doing this makes files in the directory more available to the *tableau* user.

```
chgrp tableau <directory-name>
```

```
chmod g+rx <directory-name>
```

- Alternatively, you can add world read and execute permission to the directory. This

makes files in the directory more available to all users on the system. This approach may require additional steps to ensure security of other files in the directory. For example, you may want to make sure other files in the directory are not world readable so other users cannot read them.

```
chmod o+rx <directory-name>
```

Hint: You can use `namei -mo` command to list an entire permissions tree. This can make it easier to see what directories need to have permissions adjusted to allow access by the *tableau* group. You can find more information on the internet.

Configure Server Event Notification

A Tableau Services Manager (TSM) administrator can configure Tableau Server to allow notifications for the following events:

- Content updates
 - Extract failures (enabled by default)
 - Subscription views for users (disabled by default)
- Server health monitoring
 - Server status changes (disabled by default)
 - Desktop License reporting (disabled by default)
- Drive space
 - Email alerts when disk space crosses or remains below pre-configured thresholds (disabled by default)
 - Recording usage history (enabled by default)

Note: You need to configure SMTP before you can configure subscriptions or notifications. For more information, see [Configure SMTP Setup](#).

Use the TSM web interface

1. Open TSM in a browser:

`https://<tsm-computer-name>:8850.`

For more information, see [Sign in to Tableau Services Manager Web UI](#).

2. Click **Notifications** on the **Configuration** tab and click **Events**.
3. Configure notification settings for your organization:

- Content updates

- **Send emails for extract refresh failures**

When this option is enabled (the default), a server administrator can configure email notifications to be sent when extract refreshes fail. These messages are configured at the site level, so even if this option is enabled, messages are not sent unless the **Send email to data source and workbook owners when scheduled refreshes fail** option is enabled for a site (this is enabled by default). For details, see [Enable Extract Refresh Scheduling and Failure Notification](#).

- **Allow users to receive email for views that they have subscribed to**

When this option is enabled (by default is it disabled), a server administrator can configure a site to send subscription email. These email messages are configured at the site level and can only be configured when this option is enabled. For details, see [Set Up a Site for Subscriptions](#).

When users subscribe to a workbook or view, a snapshot of the view is emailed to them on a scheduled basis, so they can see the latest updates without having to sign into Tableau Server.

To allow users to attach PDF renderings on subscription emails, select **Let users add attachments to subscribed views**.

- Server health monitoring
 - **Send emails for Tableau Server process events (up, down, and failover)**

Tableau Server sends an email message when the data engine, repository, or gateway server processes stop or restart, or when the initial Tableau Server node stops or restarts.

If you are running a single-server installation (all processes on the same computer), health alerts are only sent when Tableau Server is up. No "down" alerts are sent. If you are running a distributed installation that's configured for failover, a DOWN alert means that the active repository or a data engine instance has failed and the subsequent UP alert means that the passive instance (repository) or second instance (data engine) of that process has taken over.

Note: Tableau Server is designed to be self-correcting. If a service or process stops responding or goes down, Tableau Server attempts to restart it. This can take 15 to 30 minutes to complete. Because of this, reacting immediately to service or process alerts can be counter-productive, especially in an installation with redundant services that can handle requests while one restarts.

- **Enable Tableau Desktop License reporting**

License reporting data originates in Tableau Desktop and is sent to Tableau Server. When this option is enabled, Tableau Server will gen-

erate and display the administrative report for Desktop License reporting. For information on the report, see Desktop License Usage.

- Drive space

Enable notifications (alerts) for remaining disk space on your Tableau Server.

- **Send emails when unused drive space drops below thresholds**

You can configure Tableau Server to send email notifications when disk space usage on any node crosses a threshold, or remains below the threshold. And you can configure how often threshold notifications are sent.

There are two thresholds you must set, **Warning threshold** and **Critical threshold**. Thresholds are expressed in percentage of disk space remaining. The critical threshold must be less than the warning threshold.

You also specify the **Send threshold alert every** option. This determines how often, in minutes, warning and critical notifications should be sent. The default value is 60 minutes.

- **Record disk space usage information and threshold violations for use in custom administrative views**

When you configure Tableau Server to record disk space usage, information about free disk space is saved in the repository and you can view the usage history using the Administrative Views.

4. Click **Save Pending Changes** after you've entered your configuration information.
5. Click **Pending Changes** at the top of the page:



6. Click **Apply Changes and Restart**.

Use the TSM CLI

The various notification values described above can be set individually with the `tsm configuration set` command. Alternatively, you can construct a json file and pass all configuration values in one operation. Both methods are described in this section.

Set notification values individually

The following table shows the key/value pairs that map to the notification events described earlier in this topic. Use the `tsm configuration set` command with the following syntax to set a single key/value pair:

```
tsm configuration set -k <config.key> -v <config_value>
```

For example, to enable job failure notifications, run the following command:

```
tsm configuration set -k backgrounder.notifications_enabled -v true
```

Notification option	Key	Value
Extract failures or Flow run failures	<code>backgrounder.notifications_enabled</code>	<code>true false</code>
Enable subscription views for user	<code>subscriptions.enabled</code>	<code>true false</code>
Enable PDF attachments	<code>subscriptions.attachments_enabled</code>	<code>true false</code>

for sub- scriptions		
Maximum attachment size (MB) for subscription notifications	<code>subscriptions.max_attachment_size_megabytes</code>	integer value, default is 150
Server status changes	<code>svcmonitor.notification.smtp.enabled</code>	true false
License reporting	<code>features.DesktopReporting</code>	true false
Remaining space thresholds: enable email notifications	<code>storage.monitoring.email_enabled</code>	true false
Remaining space thresholds: warning percentage	<code>storage.monitoring.warning_percent</code>	integer value, for example, 20
Remaining space thresholds: critical percentage	<code>storage.monitoring.critical_percent</code>	integer value, for example, 15
Set email interval	<code>storage.monitoring.email_interval_min</code>	integer value, in minutes, for example, 25
Record usage history	<code>storage.monitoring.record_history_enabled</code>	true false

Tableau Server on Linux Administrator Guide

After you are done setting values, you must run the following command:

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

Set all notification values with a single json file

To make all notifications settings with a single configuration, you can pass a json file.

Copy and edit the following template to create a file for your configuration.

```
{
  "configKeys": {
    "backgrounder.notifications_enabled": true,
    "subscriptions.enabled": true,
    "subscriptions.attachments_enabled": true,
    "subscriptions.max_attachment_size_megabytes": 150,
    "svcmonitor.notification.smtp.enabled": true,
    "features.DesktopReporting": true,
    "storage.monitoring.email_enabled": true,
    "storage.monitoring.warning_percent": 20,
    "storage.monitoring.critical_percent": 15,
    "storage.monitoring.email_interval_min": 25,
    "storage.monitoring.record_history_enabled": true
  }
}
```

After you have saved the file, pass it with the following command:

```
tsm settings import -f <path-to-file.json>
```

To apply changes, run the following command:

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

Configure Data Cache

Views published to Tableau Server are interactive and sometimes have a live connection to a database. As users interact with the views in a web browser, the data that is queried gets stored in a cache. Subsequent visits will pull the data from this cache if it is available. By default, Tableau Server will cache and reuse data for as long as possible. You can change this behavior by configuring the caching option using the `tsm data-access caching set` command.

1. Run this command:

```
tsm data-access caching set -r <value>
```

Where `<value>` is one of these options:

- **low or empty string ("").** This is the default value and indicates that Tableau Server should configure cache and always use cached data when available.
- `<value>`. "`<value>`" specifies the maximum number of minutes data should be cached.
- **always** or **0** (zero). These values indicates that Tableau Server should always get the latest data and that the cache should be refreshed each time a page is reloaded.

2. Apply changes with the `tsm pending-changes apply` command. This will

restart Tableau Server.

Database Drivers

Tableau connectors require a driver to talk to the database. Before you can connect to data sources from Tableau Server, you must install drivers for the data sources you want to connect to. You can find information about supported data sources for Tableau Server on Linux on the [Tableau Server tech specs page](#). You can find driver links and installation instructions for all the supported connectors on the [Driver Download page](#).

Important: You must install the PostgreSQL driver if you want to use the built-in administrative views.

Install drivers in a cluster

You need to install the drivers for your data sources on the initial node in a Tableau Server cluster. If you install Tableau Server on multiple nodes, you must also install drivers on any node that runs any of the following processes:

- Application Server (Vizportal)
- Backgrounder
- Data Server
- VizQL Server

Server Crash Reporter

The Tableau Server administrator can enable an option to allow logs and related files to be sent to Tableau when the server has an issue that results in a crash. These files are used by Tableau to identify and address issues that cause crashes. By default this option is disabled, and it should only be enabled in organizations that are not subject to regulations related to data privacy.

Important: Do not enable crash reporting if your data is subject to privacy regulations.

If Tableau Server has a problem that results in a crash, log files and dump files are generated. If the crash data upload feature is enabled, these files are automatically gathered and zipped into an encrypted package that is sent in the background, at the scheduled time. The encrypted package is sent in small pieces to limit impact to network performance. Only one crash report is packaged and uploaded at a time (a new crash report is not packaged until the previous package has been uploaded) and is sent in a "first in, first out" order. You can schedule the sending for a low-use window to further reduce any impact to your users.

The encrypted package is made up crash dump files and logs that include the following:

- Crash/core dump files
- Error log files related to the crash
- Manifest files related to the crash

The files can contain data that includes:

- Machine-specific information (for example: hardware, operating system, domain).
- A snapshot of the contents of memory at the time of the crash, including application activity details like information about data connections, actions taken by the user in Tableau, and data being worked on in Tableau.
- Tableau information including customer-identifiable information.

Configure Server Crash Reporter

Server crash reporting is disabled by default. This topic describes how to enable and configure server crash reporting. Crash reports are encrypted and sent to Tableau. See [Server Crash Reporter](#) for more information.

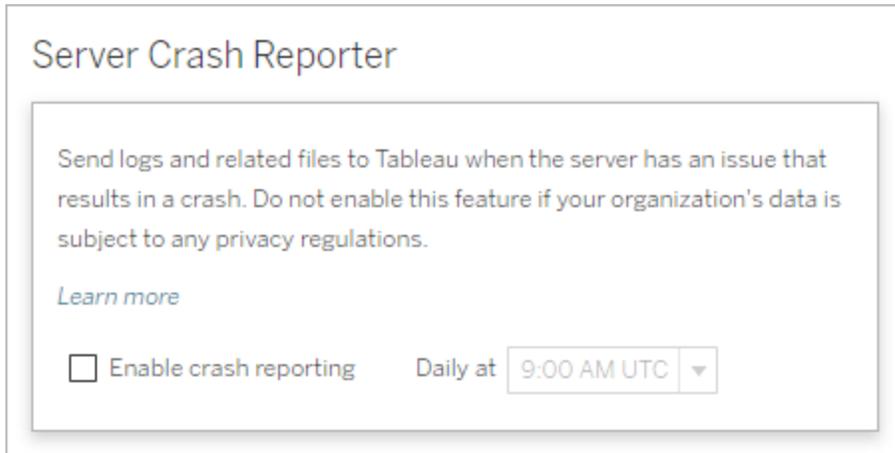
If your organization uses a proxy server to connect to the internet then you must configure server crash reporter to use the proxy. Even if you have already configured Tableau Server to use a proxy, you must also configure server crash reporter separately. To configure proxy for server crash reporter you must use TSM CLI procedure as described in this topic.

Important: Do not enable crash reporting if your data is subject to privacy regulations.

Use the TSM web interface

1. Open TSM in a browser:

https://<tsm-computer-name>:8850. For more information, see Sign in to Tableau Services Manager Web UI.
2. Click the **Maintenance** tab.
3. Under Other Maintenance Tasks, in Server Crash Reporter, select **Enable crash reporting**:



4. Specify the scheduled time of day to upload the crash reports to Tableau.
5. When you are finished, click **Pending Changes**, and then click **Apply Changes and Restart**.

Use the TSM CLI

Use the configuration file template below to create a json file. After you have filled in the options with the appropriate values, pass the json file and apply settings with the following

commands:

```
tsm settings import -f path-to-file.json
```

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

Because the configuration file is using `configKey` class, the values that you pass are not validated by TSM as they are when you use `configEntities` class. You can verify and set individual options by using the `tsm configuration` commands.

Crash reporter settings

The crash reporter settings in the template below specify a range of options for configuring Tableau Server to send crash reports to Tableau.

Configuration template

Use this template to configure the gateway settings.

For more explanation about configuration files, entities, and keys see [Configuration File Example](#).

```
{
  "configKeys": {
    "servercrashupload.enabled": "true",
    "servercrashupload.scheduled_time": "1:00:00 UTC",
    "servercrashupload.proxy_server_host": "",
    "servercrashupload.proxy_server_port": "",
    "servercrashupload.proxy_server_username": "",
    "servercrashupload.proxy_server_password": ""
  }
}
```

Tableau Server on Linux Administrator Guide

```
        "servercrashupload.preserve_upload_packages": "false",  
        "servercrashupload.delete_completed_dumps": "false"  
    }  
}
```

Configuration file reference

This table includes keys that you can set to configure crash reporting.

`servercrashupload.enabled`

Default: `false`.

Set to `true` to enable crash reporting.

`servercrashupload.scheduled_time`

Default: `1:00:00 UTC`

Specifies the scheduled time that crash uploads will begin. Enter time of day in 24 hour format.

`servercrashupload.proxy_server_host`

If your organization uses a proxy server to communicate with the internet, specify the host name.

`servercrashupload.proxy_server_port`

If your organization uses a proxy server to communicate with the internet, specify the port number.

`servercrashupload.proxy_server_username`

If your proxy server requires authentication, specify the user name with this key.

`servercrashupload.proxy_server_password`

If your proxy server requires authentication, specify the password with this key.

```
servercrashupload.preserve_upload_packages
```

Default: `false`.

To save all packages that are created for a crash reporting, set this key to `true`.

By default, packages are saved to `/var/opt/tableau/tableau_server-
/data/tabsvc/clustercontroller/tabcrashreporter`.

```
servercrashupload.delete_completed_dumps
```

Default: `false`.

To delete all dumps after they are sent, set this key to `true`.

Navigate the Admin Areas of the Tableau Web Environment

As an administrator on Tableau Server or Tableau Online, you can access admin settings that aren't available to other users to configure sites, users, projects, and to do other content-related tasks.

The settings in this article refer to the Tableau web environment. Tableau Server administrators with appropriate credentials can also change server settings such as processor, caching, authentication, distributed deployment, and related configurations using the TSM web environment. For information, see [Sign in to Tableau Services Manager Web UI](#).

Access based on site role and number of sites

The menus you get when you sign in to Tableau Server or Tableau Online depend on the following conditions:

- Whether you're a site or server administrator.

Site administrator access is available on Tableau Online and Tableau Server. Server

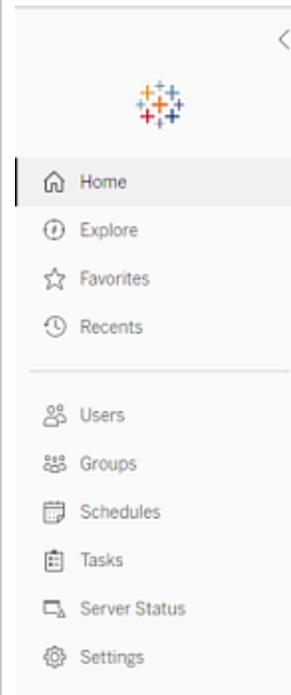
Tableau Server on Linux Administrator Guide

administrator access is only on Tableau Server.

- Whether you have access to only one site or to multiple sites.

Server administrator

On a **single-site** server, the site selector does not appear, and all other menus are the same.

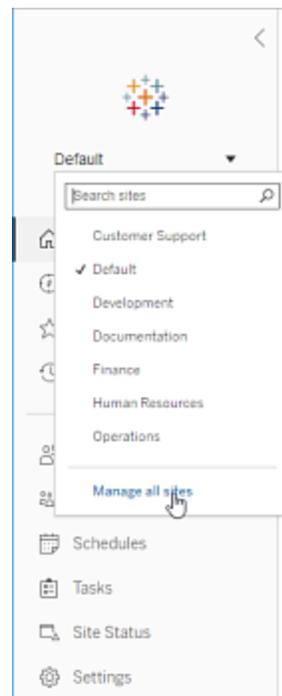
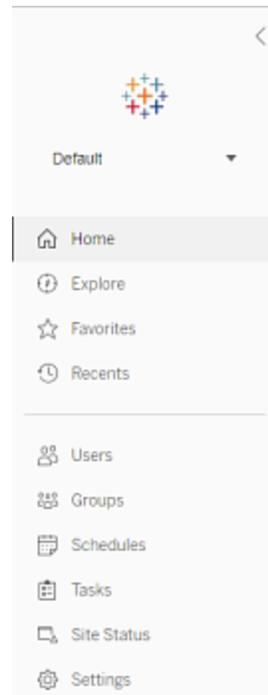


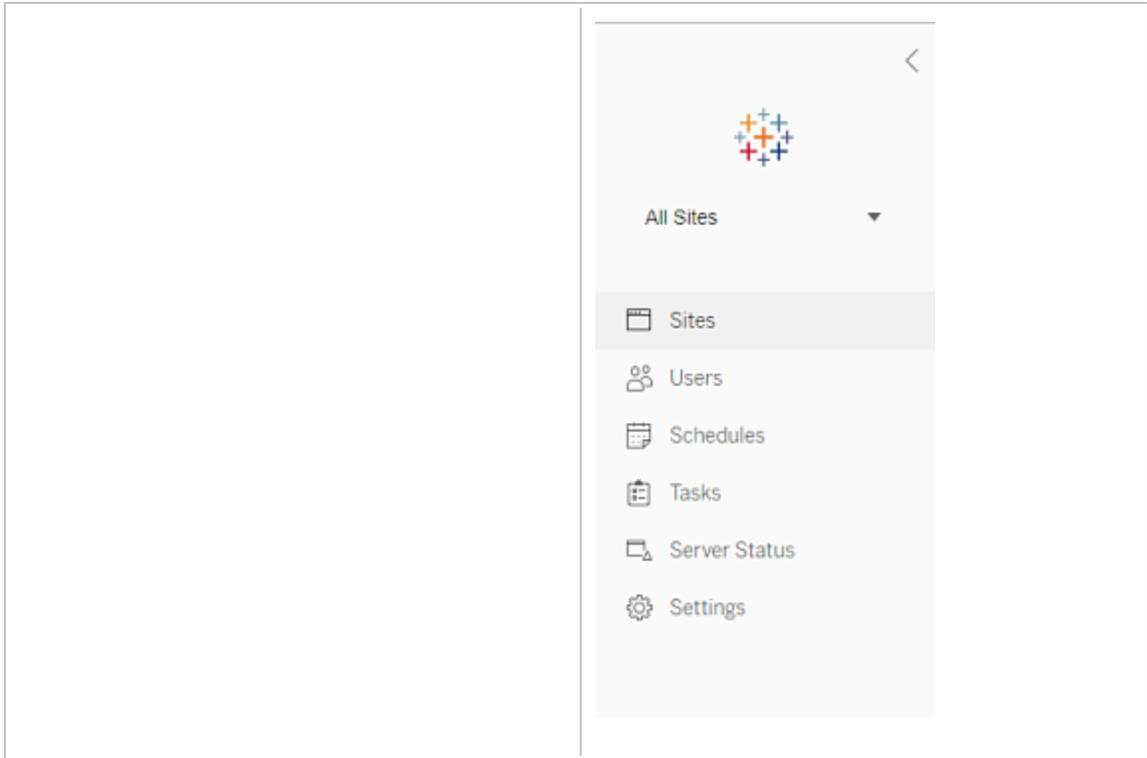
In a **multi-site** environment, menus along the left enable you to modify a specific site or all sites, and to configure users, groups, schedules, tasks, and server settings.

To access server administrator settings that affect all sites, open the site menu by clicking the arrow next to the current site name, and then select **Manage all sites**.

The **Content** and **Group** tabs go away, and the site menu text changes to **All Sites** to let you know you are managing server-wide settings, and options like **Server Status** reflect the server-wide view.

To return to the site administration menus, select **All Sites**, and then select the site you want to manage.





Site administrator

If you are a site administrator for Tableau Online or Tableau Server, and you have access to multiple sites, you'll get menus for selecting which site to manage, and for managing that site's content, users, groups, schedules, and tasks, and for monitoring its status.

The site selector displays the name of the current site. To go to another site, select the site menu, and then select the site name.

If you have access to only one site, the site selector does not appear, but all other menus are the same.

A screenshot of the Tableau Server administrator interface. At the top, there is a back arrow and a Tableau logo. Below the logo is a dropdown menu labeled 'Default'. Below this is a list of navigation options: 'Home' (highlighted with a grey bar), 'Explore', 'Favorites', 'Recents', 'Users', 'Groups', 'Schedules', 'Tasks', and 'Site Status'. Each option has a small icon to its left.

Server administrator tasks

Server administrators (available with Tableau Server Enterprise only) can do the following:

- Monitor server status and activity.
- Generate log files.
- Add sites and edit site settings. Only server administrators can add sites to the server.
- Add users to the server, and assign users to sites.
- Add and manage site groups.

To manage settings only for a specific site, you must first navigate to the site. Within each site, you can do the following:

- Administer content: Create projects, move content from one project to another, assign permissions, change ownership of a content resource, and so on.
- Manage schedules for extract refreshes and subscriptions.
- Monitor site activity and record workbook performance metrics.
- Manage storage space limits for content published by users.
- Allow web authoring.
- Enable revision history.
- Allow site administrators to add and remove users.
- Set the maximum number of licenses that site can consume for each license type (Creator, Explorer, Viewer).
- Allow users to subscribe to workbooks and views, and allow content owners to subscribe others to workbooks and views.
- Enable offline snapshots for favorites (iOS only).

Site administrator tasks

A site administrator on Tableau Online or Tableau Server can do the following tasks:

- Administer content: Create projects, move content from one project to another, assign permissions, change ownership of a content resource, and so on.
- View, manage, and manually run schedules for extract refreshes and subscriptions.
- Add and manage site users (if allowed by the server administrator; see Site Settings Reference).
- Add and manage site groups.
- Monitor site activity.

Distributed and High Availability Tableau Server Installations

The most basic way to run Tableau Server is to install a single node. With this type of installation you have a fully functional Tableau Server, with all Tableau Services Manager (TSM) and Tableau Server processes running on that single node, but this may not be the optimal way to use Tableau Server. You can decide how to install Tableau based on your organization's needs, and your resources, adding additional nodes and configuring Tableau for high availability. Your installation options include:

- **Single-node installation**—This type of installation is reasonable for testing, running trials, and for environments that can handle occasional downtime and system availability due to lack of redundancy. All server processes are running on a single machine. There is less redundancy and fewer safeguards in the event of a problem with one of the server processes. You also need to make sure the computer you install Tableau Server on has adequate resources to handle the processes and the demands of users and data.
- **Distributed installation**—This type of installation is also called a multi-node

installation and requires multiple computers so you can install and run server processes on those distributed nodes. Spreading the server processes out over multiple nodes can extend the reliability and efficiency of Tableau Server by providing redundancy and additional computing power. With the right configuration, a distributed installation can also provide you with automatic repository failover. For more information on failover, see [Repository Failover](#).

- **Highly available (HA) installation**—An HA installation of Tableau Server is a special type of multi-node installation with a minimum of three nodes and multiple instances of key processes (the Repository, File Store/Data Engine (Hyper), Coordination Service, and Client File Service) on different computers. With an HA installation, there is built-in redundancy of those key processes, including multiple File Stores, and automatic Repository failover. The goal is to minimize system downtime by eliminating single points of failure, and enabling detection of failures with failover where possible.

Downtime is still possible, in the event of an initial node failure. Dashboards and views may load more slowly than expected, and timeouts are possible, depending on how your system is configured and being used. For more information about initial node failure, see [If an initial node fails](#) below.

The first computer you install Tableau on, the "initial node," has some unique characteristics. Three processes run only on the initial node and cannot be moved to any other node except in a failure situation, the License Service (License Manager), Activation Service, and TSM Controller (Administration Controller). Tableau Server includes a script that automates moving these processes to one of your other existing nodes so you can get complete access back to TSM and keep Tableau Server running.

Two other processes are initially included on the initial node but can be added or moved to additional nodes, the CFS (Client File Service) and the Coordination Service. Depending on how your installation was configured with CFS and Coordination Service, you may also need to take steps to redeploy these.

For information about moving the License service and TSM Controller from the initial node to another node, see [Recover from an initial node failure](#) below.

Prerequisite

These instructions assume that your cluster meets the Distributed Requirements.

Note: All nodes in a multi-node cluster must have the same type of operating system. You cannot install a multi-node instance of Tableau Server on a combination of Linux and Windows nodes.

Licensing

You must have a valid Tableau Server product key. The type of license you have may determine how many nodes you can install Tableau on. You cannot use a trial license key for a distributed Tableau Server installation. For more information on licensing, see [Licensing Overview](#).

Creating a distributed Tableau Server installation

These are the general steps you follow to create a distributed installation of Tableau Server:

1. Begin by installing Tableau Server on your initial node.

For details, see [Install and Configure Tableau Server](#).
2. Generate a node configuration (bootstrap) file on the initial node.

For details, see [Generate the node bootstrap file](#).
3. Install Tableau Server on an additional node using the node bootstrap file.

For more information, see [Install and initialize an additional node](#).
4. Configure your additional node with the processes you want to run on it.

For more information, see [Configure the additional node](#).

5. Repeat Steps 3 and 4 for any additional nodes you want to install.
6. Deploy a new Coordination Service ensemble.

For more information, see [Deploy a Coordination Service Ensemble](#).

7. Add Client File Service to every node that is running the Coordination Service.

For more information, see [Configure Client File Service](#).

Creating a highly available (HA) Tableau Server installation

A high availability Tableau Server installation is a special type of distributed installation, designed to accommodate failure in key server components without loss of complete server functionality. To create an HA installation, follow the same steps you take to create a distributed deployment but include additional steps to make the deployment highly available. These additional steps include adding at least two additional nodes (for a minimum of three nodes in the cluster), adding a second instance of the repository, and second instances of the data engine/file store, adding additional gateway processes, and deploying a Coordination Service ensemble. You can also add a load balancer to distribute requests among the gateways.

At a high level, these are the steps you follow to create a highly available installation of Tableau Server:

1. Begin by installing Tableau Server on your initial node.

For details, see [Install and Configure Tableau Server](#).

2. Generate a node configuration (bootstrap) file on the initial node.

For details, see [Generate the node bootstrap file](#).

3. Install Tableau Server on at least two additional nodes using the node bootstrap file.

For more information, see [Install and initialize an additional node](#).

4. Configure each additional node with the processes you want to run on it. These must include a second copy of the Tableau Server repository, and a second copy of the data engine and file store, as well as additional instances of the gateway.

For more information, see [Configure the additional node](#).

5. Deploy a Coordination Service ensemble.

For more information, see [Deploy a Coordination Service Ensemble](#).

6. Add Client File Service to every node that is running the Coordination Service.

For more information, see [Configure Client File Service](#).

7. (Optional) Configure a load balancer.

For more information, see [Add a Load Balancer](#).

For details on how to create a three-node HA installation, see [Example: Install and Configure a Three-Node HA Cluster](#).

If an initial node fails

If there is a problem with the initial node and you have redundant processes on your other nodes, there is no guarantee that Tableau Server will continue to run. Tableau Server can continue to run for up to 72 hours after an initial node failure, before the lack of the licensing service impacts other processes. If so, your users may be able to continue to sign in and see and use their content after the initial node fails, but you will not be able to reconfigure Tableau Server because you won't have access to the Administration Controller.

Even when configured with redundant processes, *it is possible that Tableau Server may not continue to function after the initial node fails*. This is true even when an installation is configured for high availability. This means you should make a point of moving the two unique processes to another of your running nodes as soon as possible. If your initial node fails for

reasons that are recoverable in a relatively short amount of time (for example, a hardware failure you can correct), you should first attempt to bring the node back up without using the procedure below.

Recover from an initial node failure

With a Tableau Server installation, the initial node includes two services that are only installed on that node, the License service, and the TSM Controller. If there is a problem with the initial node, Tableau Server may not continue to function, even when configured for high availability. To recover from a situation where the initial node fails, you can move the TSM Controller and the License service to one of your already configured nodes. This allows you to recover from the failure while using resources you already have in the cluster. You do not have to configure a standby initial node in case the initial node fails.

For details on how to recover from a failure on the initial node, see [Recover from an Initial Node Failure](#).

Configure Coordination Service ensemble on additional nodes

Configuring a Coordination Service on multiple nodes provides additional duplication of processes and so reduces the possibility of server downtime due to an issue with one of the Coordination Service nodes. For details on how to deploy a Coordination Service ensemble on your cluster, see [Deploy a Coordination Service Ensemble](#).

Add Client File Service (CFS) to additional nodes

Tableau Server requires at least one instance of Client File Service (CFS). Adding additional instances of CFS to other nodes provides additional duplication of processes and so reduces the possibility of server downtime due to an issue with one of the CFS nodes. We recommend that you configure an instance of CFS on each of the nodes where you deploy

the Coordination Service. For details on how to configure CFS on other nodes, see [Configure Client File Service](#).

Tableau Server service license check

A number of processes are installed when you install Tableau Server. Some of these processes are dependent on the existence of a valid Tableau Server license while other installed processes are not. The subset of Tableau Server that require a valid Tableau Server license are considered "licensed processes."

When a licensed process starts or restarts, the process checks with the Tableau Server License Manager service on the initial node to verify there is a valid license. When the License Manager validates the license, the process is fully functional and able to respond to requests from other Tableau Server processes. Once a licensed process has received confirmation from the License Manager, the process does not need to reconfirm the license for 72 hours, or until the process restarts. If the process is not able to verify that it is licensed (if the primary node is unavailable, for example) it cannot run, but it continues to check for a valid license until it confirms the license. To see when the last licensing check occurred, look at the log files in the `/var/opt/tableau/tableau_server-/data/tabsvc/logs/licenseservice` directory. For more information about licensed processes, see [Licensed processes](#).

Distributed Requirements

Before you start to configure a Tableau Server cluster, make sure you meet the following requirements.

Hardware

While the computers you use in your cluster must meet the requirements described in [Before you install...](#), they do not need to be identical.

Hardware Guidelines for High Availability

Here are some guidelines for the systems you use for **failover and high availability**:

- **Failover—three computers:** To configure a cluster that provides failover support for the file store and repository processes, you need at least three computers or VMs: one for the initial Tableau Server node and two for additional nodes.
- **Multiple gateways—three computers and a load balancer:** Adding multiple Gateway processes to your Tableau Server installation and using a load balancer to automatically distribute requests to those gateways enhances the reliability of Tableau further. To configure a cluster that provides failover support and multiple gateways, you need to add a load balancer to front your three-node cluster.
- **Failover & multiple gateway support—three computers and a load balancer:** To configure a cluster that provides the above plus support for multiple gateways, you need at least three computers or VMs, and a load balancer to front the cluster.
- **High availability—three computers and a load balancer:** To configure for high availability, you need the resources described above.
- **Initial computers:** If you configure for high availability, the initial Tableau Server node may be running few or no Tableau Server processes. Therefore, the computer that serves as the initial node does not need as many cores as the ones running your additional nodes. You will, however, need adequate disk space for backups because the initial computer is used during the backup and restore processes. In addition to the amount of space needed for the backup file, you need temporary disk space. For details on disk space requirements, see [Disk Space Usage for Backup and Restore](#).

Software

- All nodes in a distributed installation must be running the same version of Tableau Server.
- All nodes must be running the same distribution of Linux.

Installation location

When you start Tableau Services Manager (TSM) you can specify a non-default location for the data directory. This location must be the same on all nodes in your distributed Tableau Server installation.

Networking and Ports

- **Ports:** As with any distributed system, the computers or VMs you use need to be able to communicate with one another. For information on how Tableau Services Manager handles port mapping, see [Tableau Services Manager Ports](#).
- **Latency:** Network latency between server nodes can impact Tableau Server performance. Be aware of possible latency issues, especially if you run into performance problems. To reduce network latency, you can take steps such as locating your gateways and data sources in proximity to Tableau Server.
- **Static IP addresses:** Any computer running Tableau Server, whether it's a single server installation or part of a cluster, must have a static IP address.
- **Discoverable:** Each node in the cluster must be discoverable from other node computers using DNS or a local host file.
- **Time zone and time:** Each node in the cluster must be in the same timezone, with their system clocks synchronized. This may happen automatically. For example, if your nodes are all in the Active Directory domain, the domain controller usually handles this. If you are not sure your cluster meets this requirement, consult with your internal IT experts.

Best Practices

Here are some things to keep in mind before you start to install and configure:

- **IP addresses or computer names:** As mentioned above, each computer in the cluster must use a static IP address.
- **Backup:** It's a best practice to create a backup prior to making significant system changes. See [Back up Tableau Server data](#) for steps.

SSL

If you are planning to configure SSL for a highly available Tableau Server cluster with multiple gateways and a load balancer ([learn more](#)), make sure that the SSL certificate you use was issued for the load balancer's host name. See [Configure SSL for External HTTP Traffic to and from Tableau Server](#) for other details.

Distributed Installation Recommendations

When you add nodes to a Tableau Server installation, you must decide how many processes to run on each computer. This page provides some general recommendations that are intended only as a starting point.

In addition to these general recommendations, you should also:

- Understand how your organization uses Tableau Server and tune your configuration for your use case—for example, whether you want to optimize for user response or for extract refreshes.
- Perform thorough performance testing to identify the best places to adjust process configuration.

For more information on tailoring a Tableau Server installation to your organization's needs, see [Performance Tuning Examples](#).

For more information on the requirements for a distributed installation and for information on configuring additional nodes, see [Distributed and High Availability Tableau Server Installations](#).

Recommendations for all installations

Although the computers that make up a Tableau Server cluster do not need to have identical hardware, they must all meet the same minimum system requirements. The recommendations on this page assume that the computers where you install Tableau Server have eight cores or more.

The following recommendations apply to all server configurations:

- Run Backgrounder processes on a dedicated computer if you plan on refreshing extracts frequently. Backgrounder processes are generally the most CPU intensive and can slow down other processes on the same computer.

Note: In versions earlier than 10.5, the backgrounder process spun up temporary instances of Data Engine when refreshing extracts. In 10.5, a single instance of Data Engine is installed with the backgrounder during setup. The backgrounder uses this running instance of Data Engine when refreshing extracts.

- If you plan to refresh extracts frequently or if you plan to refresh large extracts, increase the number of processes for Backgrounder processes.
- Run VizQL processes on a different computer than Backgrounder processes. Having them on the same machine means that extract refreshes can affect user views.
- The instance of Data Engine installed on the node where File Store is installed is used for querying data for view requests. Consider separating the File Store process from the backgrounder processes to help minimize the backgrounder tasks from affecting user views.
- **Optimizing with topology configurations:**
 - Co-locating File Store on the same node as the Administration Controller can reduce the length of time it takes to back up Tableau Server by reducing or eliminating the need to transfer data between nodes during the backup process. This is especially true if your organization uses many extracts.

- Co-locating the repository (pgsql) with the Administration Controller node can also help to reduce back up time, but the time savings is less significant than that of the File Store.

The Administration Controller is usually on the initial node, unless you have had an initial node failure and moved the controller to another node.

Note: In a distributed installation with three or more nodes, you can have a maximum of two repository instances (active and passive). You can also run Tableau Server with one repository, but doing this means there is no failover available for the repository. For more information, see [Tableau Server Repository](#).

Install and Configure Additional Nodes

After you install Tableau Server on one computer (or node), the server is functional and ready for use, but it has no redundancy. If there is a problem with a process or a problem with the computer itself, Tableau Server may be unavailable. In addition, all processes are running on a single computer, so there can be contention for resources on that computer.

You can extend your Tableau Server installation by adding Tableau to additional nodes, creating a distributed installation. This article describes the general steps for installing Tableau Server on additional nodes and assumes you have already installed Tableau on an initial node. For more information on installing Tableau on the initial node, see [Install and Initialize TSM](#).

If you are installing Tableau Server on multiple nodes, you should install and configure one node at a time. This makes it easier to troubleshoot any issues you might run into.

Important: You should add and configure additional nodes when you can fully complete the process by applying pending changes. Adding a node without finishing by applying pending changes can result in users being unable to log into Tableau Server.

Use the TSM web interface

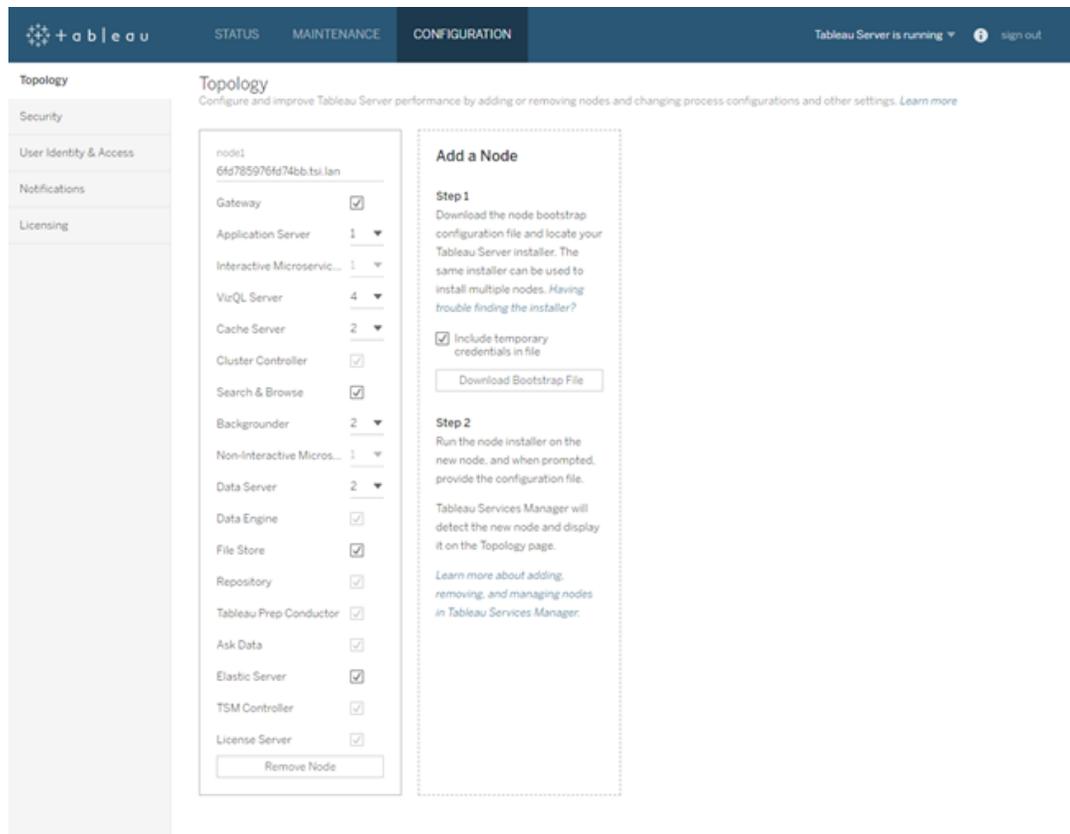
Generate the node bootstrap file

1. Open TSM in a browser:

`https://<tsm-computer-name>:8850`

For more information, see [Sign in to Tableau Services Manager Web UI](#).

2. Click the **Configuration** tab, and in the Add a Node box, click **Download Bootstrap File**.



The bootstrap file is created and copied to your local computer.

Beginning with version 2019.3, embedded credentials are included in the bootstrap file by default. If you don't want credentials embedded in the bootstrap file, clear the

Include temporary credentials in file option. See What's New and Changed in Tableau Server in What's New for more information. If you want to completely disable the ability to include embedded credentials in node bootstrap files, you can set a configuration option for the server. See `features.PasswordlessBootstrapInit` for more details.

Install and initialize an additional node

Before you begin, verify that your node bootstrap file is recent. For example, if you have run `tsm security regenerate-internal-tokens` after you generated a bootstrap file, then initialization will fail.

1. Copy the original installer you used on the first computer along with the bootstrap file you generated and put them in a location accessible from the new computer you are adding Tableau Server to. This could be a mounted network share, or directly on the new computer.
2. If you are running a local firewall, then you need to configure firewall rules for all the nodes in the cluster. For more information, see [Configure Local Firewall](#).
3. On the new node, run the Tableau Server Setup program:

Use the package manager to install the Tableau Server package.

Do not install to a location using a symbolic link or to a directory on a Network File System (NFS) volume.

- On RHEL-like distributions, including CentOS, you have the option to install Tableau to a non-default location.
 - **Default location**—To install to the default location (`/opt/tableau/tableau_server`), run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: 2019-2-5):

```
sudo yum update
```

Tableau Server on Linux Administrator Guide

```
sudo yum install tableau-server-<version>.x86_64.rpm
```

- **Non-default location**—To install to a non-default location, you must use `rpm -i`. You will also need to install all dependent packages. See the note below.

Run the following command:

```
sudo rpm -i --prefix /preferred/install/path tableau-server.rpm
```

Note: When you use `yum` to install Tableau Server, all dependent packages are automatically downloaded and installed. This is the preferred method for installing Tableau. If you want to install to a non-default location, or your organization does not allow you to use `yum` and you must install using `rpm -i`, you must also install all dependent packages separately. For information about installing dependent packages, see [Installing Tableau Server on an Air-Gapped Computer Running Linux](#).

- On Ubuntu, run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: 2019-2-5):

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get -y install gdebi-core
sudo gdebi -n tableau-server-<version>_amd64.deb
```

4. To initialize communications between your new node and the initial node, run the `initialize-tsm` script that is installed when you install Tableau Server.

Navigate to the `scripts` directory:

```
cd /opt/tableau/tableau_server/packages/scripts.<version_
code>/
```

5. Run the initialize-tsm script:

```
sudo ./initialize-tsm -b /path/to/<bootstrap>.json --
accepteula
```

- Use the `-b` flag to provide the path to the bootstrap file that you copied to the computer. If you have encrypted the bootstrap file, then you must pipe the file as described in [Securing secrets for import and export operations](#).
- If the bootstrap file was generated without embedded credentials, use the `-u` flag to specify the user name of the administrative user on the initial node. This is the name of an administrative user on the computer, not the Tableau Server administrator. You will be prompted for the user password. For more information, see [tsm topology nodes get-bootstrap-file](#).

```
sudo ./initialize-tsm -b /path/to/<bootstrap>.json -u
<admin-user-on-first-node> --accepteula
```

Important: You must enter the credentials of the same user that you used during the installation process on the initial node.

- The `--accepteula` flag accepts the Tableau Server End User License Agreement (EULA). The EULA is available in the following location:

```
/opt/tableau/tableau_server/packages/docs.<version_
code>/EULA.rtf
```

Configure the additional node with Cluster Controller

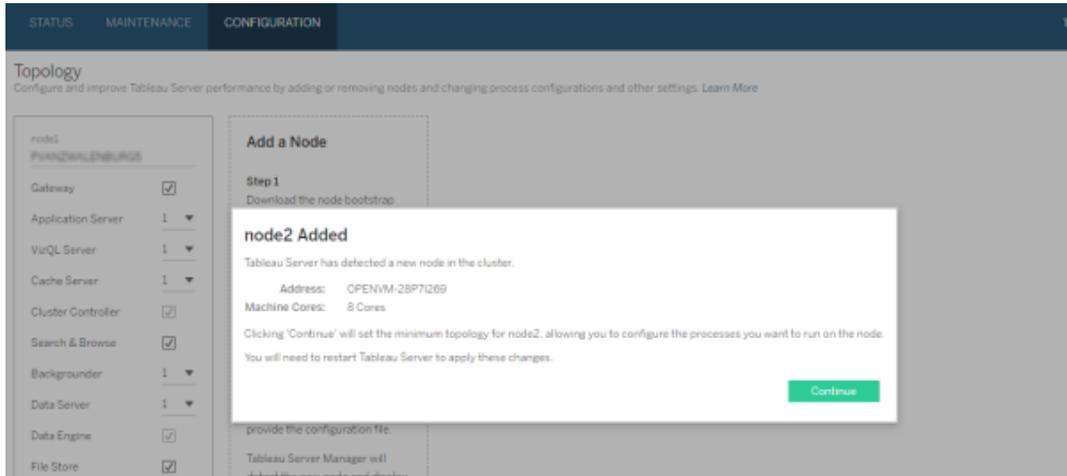
1. Open TSM in a browser:

```
https://<tsm-computer-name>:8850
```

For more information, see [Sign in to Tableau Services Manager Web UI](#).

Tableau Server on Linux Administrator Guide

2. Click the **Configuration** tab. A message should tell you that the new node was added.



Click **Continue** to dismiss the message.

3. Cluster Controller is part of the minimum topology and is already selected. Each node must run Cluster Controller.

If you only want to set the minimum topology for the node, go to Step 4 below. You might do this if you will be adding additional nodes and are not ready to configure this node beyond the minimum.

If you want to add additional processes to the minimum topology, specify the processes that should run on the node. Do this by selecting the processes you want, or specifying the number of processes that should run on the node.

For example, to add a Gateway and two instances of the Backgrounder on the node:

- a. Select **Gateway**.
- b. Set the **Backgrounder** count to 2.

Adding Backgrounder to a node will also add an instance of Data Engine if one is not already on the node.

The specific processes and process counts you set will depend on your organizational environment and needs. Some processes are added automatically when you add another process. For more information, see [Configure Nodes and Tableau Server Processes](#).

Note: The TSM Web UI limits you to a maximum of 8 instances of processes that allow you to select the number of instances. To configure more instances than this, use the command line and the `TSM topology set-process` command. For more information, see `tsm topology set-process`.

4. Click **Pending Changes** at the top of the page:



If you are configuring a cluster with three or more nodes, a Coordination Service ensemble warning displays. You can continue. You will deploy a Coordination Service ensemble in a separate step.

5. Click **Apply Changes and Restart** and **Confirm** to confirm a restart of Tableau Server.

After Tableau Server restarts, the node is included with the minimum topology necessary and any additional processes you configured.

Use the TSM CLI

Generate the node bootstrap file

To install Tableau Server on additional computers you use the same installer you did for the initial node, along with a "bootstrap" file you generate on the initial node.

Important: The bootstrap file contains a copy of the master keystore file used for encrypting the configuration secrets. The file can also embedded credentials which are valid for a pre-determined amount of time (see `tabadmincontroller.auth.expiration.minutes`) and serve as a session cookie. We strongly recommend that you take additional measures to secure the bootstrap file.

The following command set provides an example method to encrypt the bootstrap file output. This method is similar to the encryption process described in more detail at [Securing secrets for import and export operations](#).

Note however, the method here must be passed as separate arguments with trailing `&& \` operators as follows:

```
mkfifo -m 600 /tmp/secure1 && \  
  
tsm topology nodes get-bootstrap-file --file /tmp/secure1 && \  
  
gpg --symmetric --batch --yes --passphrase-file ~/.secret-  
s/pgppassphrase.txt --cipher-algo AES256 --output encrypted.enc  
< /tmp/secure1 && \  
  
rm /tmp/secure1
```

1. After installing Tableau Server on the initial node, generate the node bootstrap file.
2. On the initial node, open a terminal session.
3. Type this command to generate the bootstrap file:

```
tsm topology nodes get-bootstrap-file --file <path\  
file>.json
```

Beginning with version 2019.3, embedded credentials are included in the bootstrap file by default. If you don't want the bootstrap file to include credentials, use the `-nec` or `--no-embedded-credentials` option:

```
tsm topology nodes get-bootstrap-file --file <path\file>.json --no-embedded-credentials.
```

See [What's New and Changed in Tableau Server](#) in [What's New](#) for more information. If you want to completely disable the ability to include embedded credentials in node bootstrap files, you can set a configuration option for the server. See [features.PasswordlessBootstrapInit](#) for more details.

Install and initialize an additional node

1. Copy the original installer you used on the first computer along with the bootstrap file you generated and put them in a location accessible from the new computer you are adding Tableau Server to. This could be a mounted network share, or directly on the new computer.
2. If you are running a local firewall, then you need to configure firewall rules for all the nodes in the cluster. For more information, see [Configure Local Firewall](#).
3. On the new node, run the Tableau Server Setup program:

Use the package manager to install the Tableau Server package.

Do not install to a location using a symbolic link or to a directory on a Network File System (NFS) volume.

- On RHEL-like distributions, including CentOS, you have the option to install Tableau to a non-default location.

- **Default location**—To install to the default location (`/opt/tableau/tableau_server`), run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: 2019-2-5):

```
sudo yum update
```

Tableau Server on Linux Administrator Guide

```
sudo yum install tableau-server-<version>.x86_64.rpm
```

- **Non-default location**—To install to a non-default location, you must use `rpm -i`. You will also need to install all dependent packages. See the note below.

Run the following command:

```
sudo rpm -i --prefix /preferred/install/path tableau-server.rpm
```

Note: When you use `yum` to install Tableau Server, all dependent packages are automatically downloaded and installed. This is the preferred method for installing Tableau. If you want to install to a non-default location, or your organization does not allow you to use `yum` and you must install using `rpm -i`, you must also install all dependent packages separately. For information about installing dependent packages, see [Installing Tableau Server on an Air-Gapped Computer Running Linux](#).

- On Ubuntu, run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: 2019-2-5):

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get -y install gdebi-core
sudo gdebi -n tableau-server-<version>_amd64.deb
```

4. To initialize communications between your new node and the initial node, run the `initialize-tsm` script that is installed when you install Tableau Server.

On the new node:

Navigate to the `scripts` directory:

```
cd /opt/tableau/tableau_server/packages/scripts.<version_
code>/
```

5. Run the initialize-tsm script:

```
sudo ./initialize-tsm -b /path/to/<bootstrap>.json --
accepteula
```

- Use the `-b` flag to provide the path to the bootstrap file that you copied to the computer. If you have encrypted the bootstrap file, then you must pipe the file as described in [Securing secrets for import and export operations](#).
- If the bootstrap file was generated without embedded credentials, use the `-u` flag to specify the user name of the administrative user on the initial node. This is the name of an administrative user on the computer, not the Tableau Server administrator. You will be prompted for the user password. For more information, see [tsm topology nodes get-bootstrap-file](#).

```
sudo ./initialize-tsm -b /path/to/<bootstrap>.json -u
<admin-user-on-first-node> --accepteula
```

Important: You must enter the credentials of the same user that you used during the installation process on the initial node.

- The `--accepteula` flag accepts the Tableau Server End User License Agreement (EULA). The EULA is available in the following location:

```
/opt/tableau/tableau_server/packages/docs.<version_
code>/EULA.rtf
```

Configure the additional node

Note: This basic example illustrates how to set the topology on a node. For a more detailed, working multi-node example, see [Example: Install and Configure a Three-](#)

Node HA Cluster.

On the initial (original) node, set the topology for the newly added node. The topology specifies which processes should run on the node, and how many instances of each process should run. The topology for the node will depend on your environment and organizational needs. The below are just examples of setting the topology.

1. On the initial (original) node, open a terminal session.
2. Get the node-id for the new node:

```
tsm topology list-nodes -v
```

The `-v` option lists the nodes and the processes they are currently running. You can identify the newly added node because it will not have any processes on it.

3. Specify the individual processes that should run on this node.

Do this with the following command:

```
tsm topology set-process -n <nodeID> -pr <processname> -c  
<n>
```

You must add an instance of the Cluster Controller to each node.

For example, to add the Cluster Controller, two instances of the Backgrounder, and a Gateway to node2:

```
tsm topology set-process -n node2 -pr clustercontroller -c  
1  
tsm topology set-process -n node2 -pr backgrounder -c 2  
tsm topology set-process -n node2 -pr gateway -c 1
```

The specific processes and process counts you set will depend on your organizational environment and needs. Some processes are added automatically when you add

another process. For more information, see [Configure Nodes and Tableau Server Processes](#).

4. Apply the node configuration. If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see [tsm pending-changes apply](#).

```
tsm pending-changes apply
```

A warning about deploying a Coordination Service ensemble displays if you have deployed a three- or five-node cluster. If this is the only warning, you can safely override it using the `--ignore-warnings` option to apply the configuration changes in spite of the warning.

```
tsm pending-changes apply --ignore-warnings
```

Install drivers

You need to install drivers so that Tableau Server can connect to data and run extracts.

Install these drivers on nodes that are running any of the following processes:

- VizQL Server (`vizqlserver`)
- Application Server (`vizportal`)
- Data Server (`dataserver`)
- Backgrounder (`backgrounder`)

Drivers and administrative views

If you want to use the built-in administrative views in Tableau Server, you also need to install the PostgreSQL driver on any nodes running any of the above processes.

For more information, see [Database Drivers](#).

Database Drivers

Tableau connectors require a driver to talk to the database. Before you can connect to data sources from Tableau Server, you must install drivers for the data sources you want to connect to. You can find information about supported data sources for Tableau Server on Linux on the [Tableau Server tech specs page](#). You can find driver links and installation instructions for all the supported connectors on the [Driver Download page](#).

Important: You must install the PostgreSQL driver if you want to use the built-in administrative views.

Install drivers in a cluster

You need to install the drivers for your data sources on the initial node in a Tableau Server cluster. If you install Tableau Server on multiple nodes, you must also install drivers on any node that runs any of the following processes:

- Application Server (Vizportal)
- Backgrounder
- Data Server
- VizQL Server

Example: Install and Configure a Three-Node HA Cluster

A multi-node high availability installation of Tableau Server can help to maximize the efficiency and availability of Tableau Server. When configuring a multi-node Tableau Server deployment, the steps you perform are designed to build in redundancy, helping to reduce your potential downtime. In addition to simply improving efficiency by moving or adding server processes to additional nodes, you can create a highly available (HA) installation of Tableau Server by satisfying these requirements:

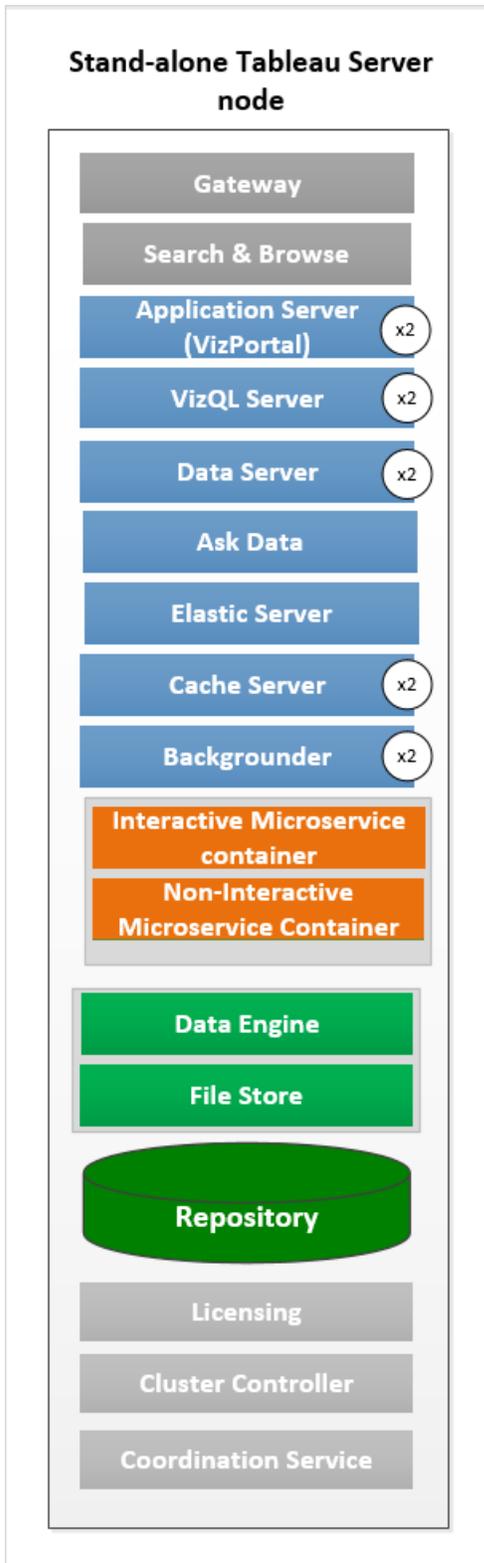
- Add additional nodes for a total of at least three nodes
- Deploy a Coordination Service ensemble on at least three nodes

- Add an instance of Client File Service (CFS) on at least one additional node (we recommend adding an instance on each node running the Coordination Service)
- Add a second instance of the File Store on one of the additional nodes (Data Engine will be installed automatically, if it is not already on the node)
- Add a second instance of the Repository (pgsql) on one of the additional nodes

A Tableau Server installation that includes these additions will have built-in redundancy and can support failover in the event of a problem with the repository. This example shows how to do this, and more.

A Single Server System

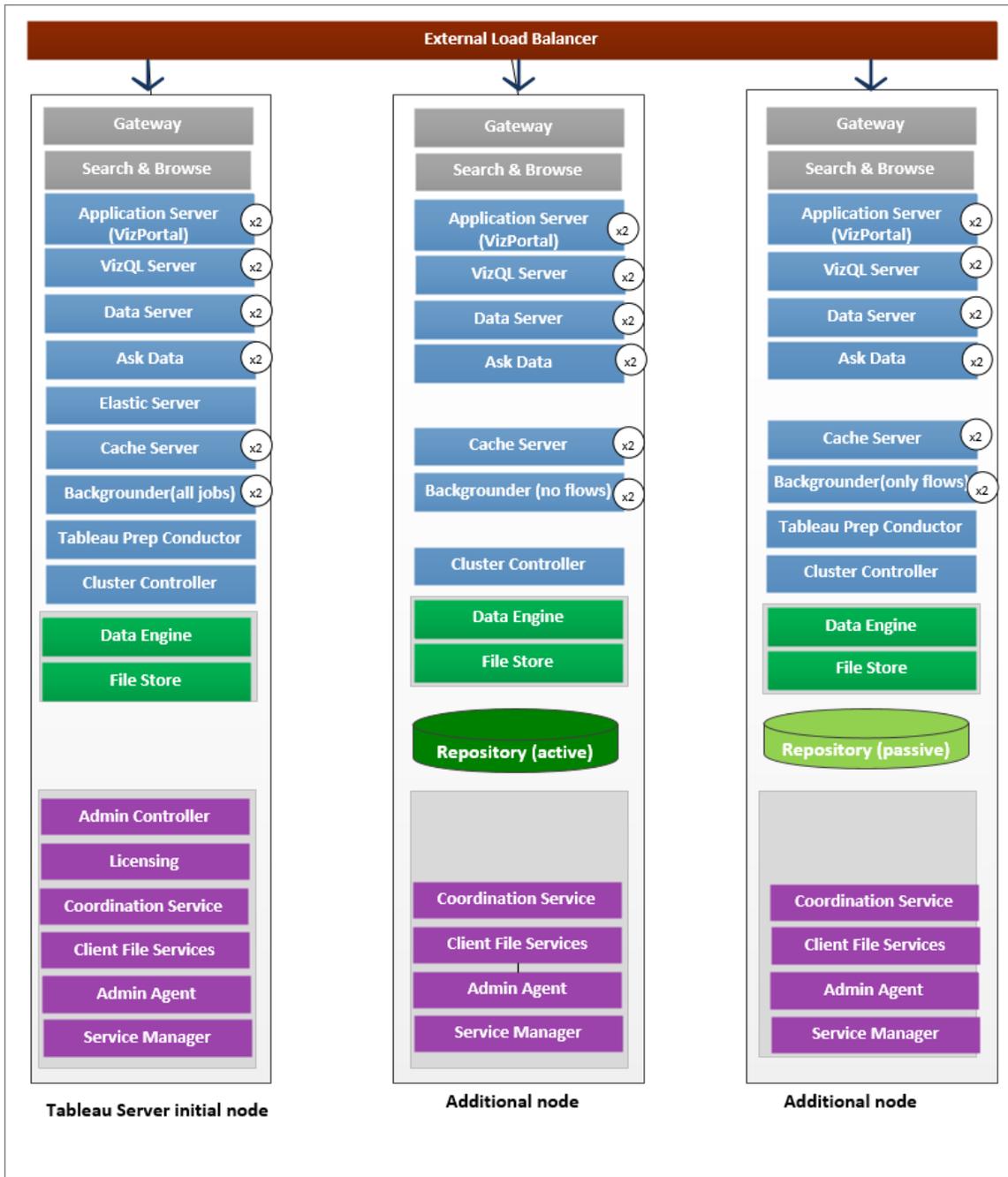
After installing Tableau Server on an initial node, you have a system that is running everything it needs to function. It has at least one instance of all server processes and is the most basic configuration of Tableau Server. It has no redundancy. The server topology looks like this (some TSM-specific processes are not shown):



A Three-Node System

To build in redundancy, you need to add additional nodes to host instances of the repository and the data engine and file store processes. You can add instances of other processes, including multiple instances of a process on a node. In addition, to reduce the system's vulnerability, you can run multiple gateways and additional instances of some of the server processes. The fewest number of computers required to achieve this configuration is three.

In the diagram below, the file store process has been added to both additional nodes. A second, passive instance of the repository has also been added to one of the other new nodes. Finally, the server processes (shown in blue) have been added to the additional nodes to provide redundancy.



Configuration steps

This procedure describes how to configure a three-node HA Tableau Server cluster with two repository instances and two file store/data engine instances on the additional nodes as

pictured above.

Before you begin

Before you install Tableau Server on any additional nodes, ensure that each additional node meets the distributed requirements. See [Distributed Requirements](#) for details.

Use the TSM web interface

Note: This operation includes steps that you may need to perform using the TSM command line.

Step 1: Install the initial node

See [Install and Configure Tableau Server](#).

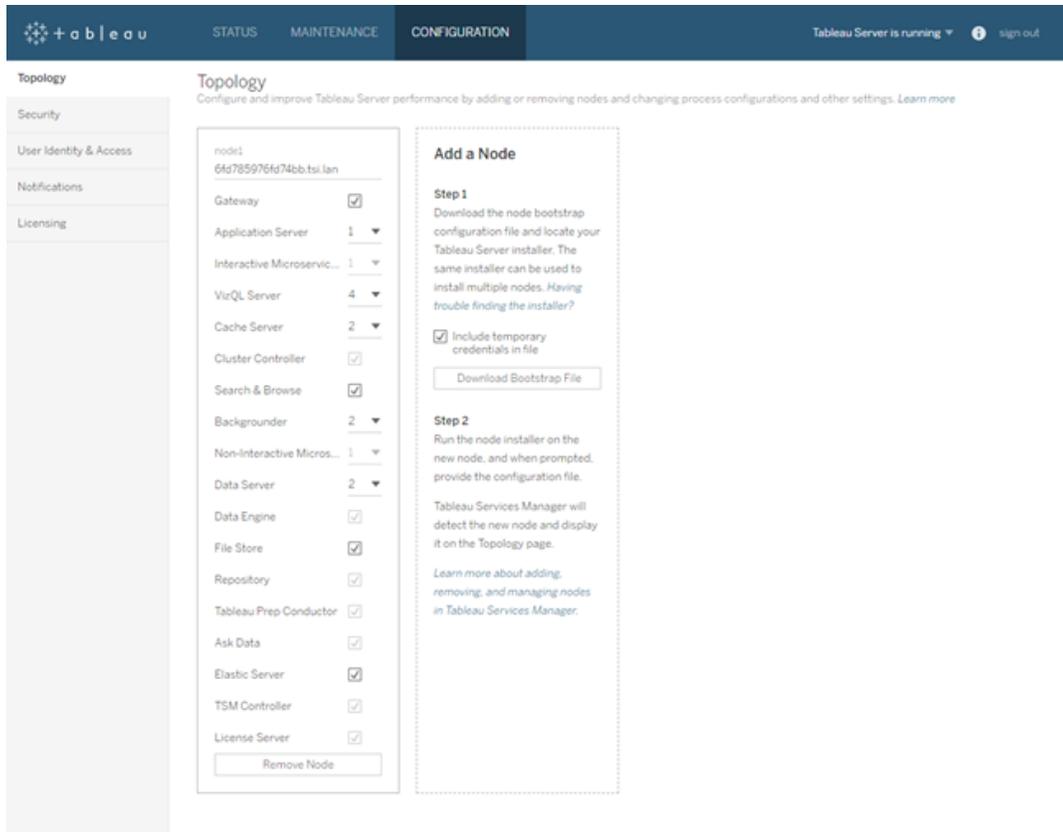
Step 2: Generate a bootstrap file for the additional nodes

1. Open TSM in a browser:

`https://<tsm-computer-name>:8850`

For more information, see [Sign in to Tableau Services Manager Web UI](#).

2. Click the **Configuration** tab, and in the Add a Node box, click **Download Bootstrap File**.



The bootstrap file is created and copied to your local computer.

Beginning with version 2019.3, embedded credentials are included in the bootstrap file by default. If you don't want credentials embedded in the bootstrap file, clear the **Include temporary credentials in file** option. See *What's New and Changed in Tableau Server* in *What's New* for more information. If you want to completely disable the ability to include embedded credentials in node bootstrap files, you can set a configuration option for the server. See `features.PasswordlessBootstrapInit` for more details.

Step 3: Install and initialize an additional node

1. Copy the original installer you used on the first computer along with the bootstrap file you generated and put them in a location accessible from the new computer you are adding Tableau Server to. This could be a mounted network share, or directly on the

new computer.

2. If you are running a local firewall, then you need to configure firewall rules for all the nodes in the cluster. For more information, see [Configure Local Firewall](#).
3. On the new node, run the Tableau Server Setup program:

Use the package manager to install the Tableau Server package.

Do not install to a location using a symbolic link or to a directory on a Network File System (NFS) volume.

- On RHEL-like distributions, including CentOS, you have the option to install Tableau to a non-default location.

- **Default location**—To install to the default location (`/opt/tableau/tableau_server`), run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: 2019-2-5):

```
sudo yum update
```

```
sudo yum install tableau-server-<version>.x86_64.rpm
```

- **Non-default location**—To install to a non-default location, you must use `rpm -i`. You will also need to install all dependent packages. See the note below.

Run the following command:

```
sudo rpm -i --prefix /preferred/install/path
tableau-server.rpm
```

Note: When you use `yum` to install Tableau Server, all dependent packages are automatically downloaded and installed. This is the preferred method for installing Tableau. If you want to install to a non-default location, or your organization does not allow you to use `yum` and you must install using `rpm -i`, you must also install all dependent packages separately. For information about installing dependent packages, see [Installing Tableau Server on an Air-Gapped Computer Running Linux](#).

- On Ubuntu, run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: 2019-2-5):

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get -y install gdebi-core
sudo gdebi -n tableau-server-<version>_amd64.deb
```

4. To initialize communications between your new node and the initial node, run the `initialize-tsm` script that is installed when you install Tableau Server.

Navigate to the `scripts` directory:

```
cd /opt/tableau/tableau_server/packages/scripts.<version_code>/
```

5. Run the `initialize-tsm` script:

```
sudo ./initialize-tsm -b /path/to/<bootstrap>.json --accepteula
```

- Use the `-b` flag to provide the path to the bootstrap file that you copied to the computer. If you have encrypted the bootstrap file, then you must pipe the file as described in [Securing secrets for import and export operations](#).
- If the bootstrap file was generated without embedded credentials, use the `-u`

flag to specify the user name of the administrative user on the initial node. This is the name of an administrative user on the computer, not the Tableau Server administrator. You will be prompted for the user password. For more information, see `tsm topology nodes get-bootstrap-file`.

```
sudo ./initialize-tsm -b /path/to/<bootstrap>.json -u
<admin-user-on-first-node> --accepteula
```

Important: You must enter the credentials of the same user that you used during the installation process on the initial node.

- The `--accepteula` flag accepts the Tableau Server End User License Agreement (EULA). The EULA is available in the following location:

```
/opt/tableau/tableau_server/packages/docs.<version_
code>/EULA.rtf
```

Step 4: Install and initialize a second additional node

1. Copy the original installer you used on the first computer along with the bootstrap file you generated and put them in a location accessible from the new computer you are adding Tableau Server to. This could be a mounted network share, or directly on the new computer.
2. If you are running a local firewall, then you need to configure firewall rules for all the nodes in the cluster. For more information, see [Configure Local Firewall](#).
3. On the new node, run the Tableau Server Setup program:

Use the package manager to install the Tableau Server package.

Do not install to a location using a symbolic link or to a directory on a Network File System (NFS) volume.

- On RHEL-like distributions, including CentOS, you have the option to install Tableau to a non-default location.
 - **Default location**—To install to the default location (`/opt/tableau/tableau_server`), run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: 2019-2-5):

```
sudo yum update

sudo yum install tableau-server-<version>.x86_64.rpm
```
 - **Non-default location**—To install to a non-default location, you must use `rpm -i`. You will also need to install all dependent packages. See the note below.

Run the following command:

```
sudo rpm -i --prefix /preferred/install/path
tableau-server.rpm
```

Note: When you use `yum` to install Tableau Server, all dependent packages are automatically downloaded and installed. This is the preferred method for installing Tableau. If you want to install to a non-default location, or your organization does not allow you to use `yum` and you must install using `rpm -i`, you must also install all dependent packages separately. For information about installing dependent packages, see [Installing Tableau Server on an Air-Gapped Computer Running Linux](#).

- On Ubuntu, run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: 2019-2-5):

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get -y install gdebi-core
sudo gdebi -n tableau-server-<version>_amd64.deb
```

4. To initialize communications between your new node and the initial node, run the initialize-tsm script that is installed when you install Tableau Server.

Navigate to the `scripts` directory:

```
cd /opt/tableau/tableau_server/packages/scripts.<version_code>/
```

5. Run the initialize-tsm script:

```
sudo ./initialize-tsm -b /path/to/<bootstrap>.json --accepteula
```

- Use the `-b` flag to provide the path to the bootstrap file that you copied to the computer. If you have encrypted the bootstrap file, then you must pipe the file as described in [Securing secrets for import and export operations](#).
- If the bootstrap file was generated without embedded credentials, use the `-u` flag to specify the user name of the administrative user on the initial node. This is the name of an administrative user on the computer, not the Tableau Server administrator. You will be prompted for the user password. For more information, see [tsm topology nodes get-bootstrap-file](#).

```
sudo ./initialize-tsm -b /path/to/<bootstrap>.json -u
<admin-user-on-first-node> --accepteula
```

Important: You must enter the credentials of the same user that you used during the installation process on the initial node.

Tableau Server on Linux Administrator Guide

- The `--accepteula` flag accepts the Tableau Server End User License Agreement (EULA). The EULA is available in the following location:

```
/opt/tableau/tableau_server/packages/docs.<version_code>/EULA.rtf
```

Step 5: Configure the new node with a minimum topology

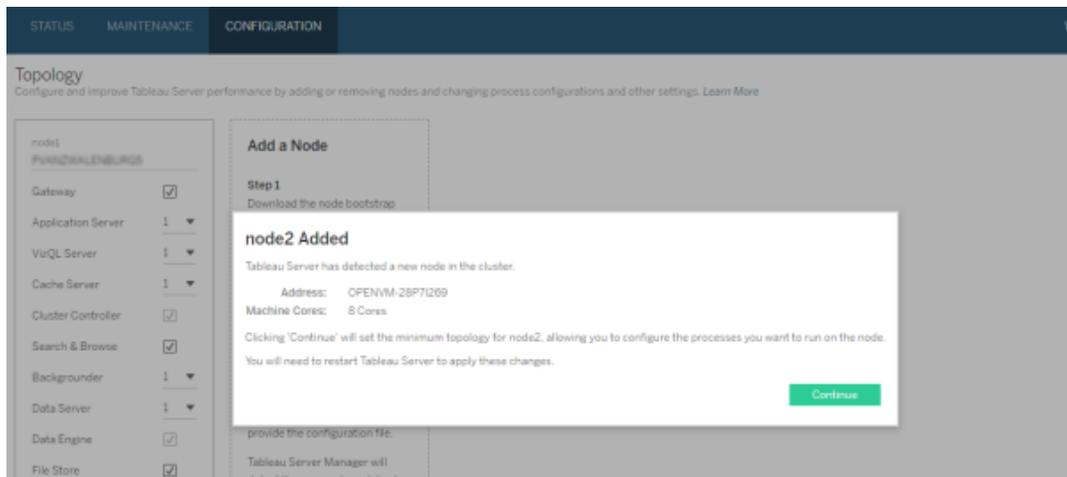
To complete the process of adding new nodes to your cluster, you need to configure them with a minimum topology. With a minimum topology, the only pending change will be the addition of Cluster Controller, which is required on each node. If you want other processes on the nodes you can add most of them at the same time. This example only configures the nodes with Cluster Controller.

1. Open TSM in a browser:

`https://<tsm-computer-name>:8850`

For more information, see [Sign in to Tableau Services Manager Web UI](#).

2. Click the **Configuration** tab. A message should tell you that new nodes were added.



Click **Continue** to dismiss the message.

3. Click **Pending Changes** at the top of the page:



If you are configuring a cluster with three or more nodes, a Coordination Service ensemble warning displays. You can continue. You will deploy a Coordination Service ensemble in a separate step.

4. Click **Apply Changes and Restart** and **Confirm** to confirm a restart of Tableau Server.

When Tableau Server restarts, the nodes are included with the minimum topology necessary.

Step 6: Deploy a Coordination Service ensemble

If you install a total of three or more nodes, you should also deploy a Coordination Service ensemble. If you do not, you will get a warning message every time you make changes to the server configuration or topology. You can ignore this message, but as a best practice you should deploy a multi-node Coordination Service ensemble.

When you install Tableau Server, a single instance of the Coordination Service is installed on the initial node. TSM and Tableau Server depend on the Coordination Service to function properly, so to provide redundancy and ensure availability on multi-node installations, configure additional instances of the Coordination Service by deploying a Coordination Service ensemble. Coordination Service ensembles are installed with one, three, or five instances of the Coordination Service. In a three-node installation of Tableau Server, the recommended number of Coordination Service instances is three, one on each node.

Do not attempt to deploy a Coordination Service ensemble if there are other changes pending. Discard or apply any pending changes before deploying a new Coordination Service ensemble.

Note: This operation includes steps that you may need to perform using the TSM command line.

Tableau Server on Linux Administrator Guide

The following steps illustrate how to deploy a new Coordination Service ensemble on an existing three-node Tableau Server cluster and clean up the old ensemble.

1. On the initial node, open a terminal session.
2. Stop Tableau Server:

```
tsm stop
```

If prompted, sign in as a TSM administrator.

Some TSM processes will continue to run, including the Administration Controller and Administration Agent.

3. Confirm there are no pending changes:

```
tsm pending-changes list
```

If there are pending changes, you need to either discard the changes or apply them.

Applying pending changes will take some time:

- Discard the changes

```
tsm pending-changes discard
```

or

- Apply the changes:

```
tsm pending-changes apply
```

Wait until the command completes and you are returned to the system prompt.

4. Get the node IDs for each node in the cluster:

```
tsm topology list-nodes -v
```

5. Use the `tsm topology deploy-coordination-service` command to add a new Coordination Service ensemble by adding the Coordination Service to specified

nodes. You must specify the node(s) that the Coordination Service should be added to. The command also makes the new ensemble the "production" ensemble (the ensemble in use) and removes the old ensemble, unless the deployment fails. If this happens, see step 6 below.

Note: A "y/n" prompt displays confirming that a server restart will take place. To run the command without input, include the `--ignore-prompt` option.

For example, deploy the Coordination Service to all three nodes of a three-node cluster, where the nodes are node1, node2, and node3:

```
tsm topology deploy-coordination-service -n  
node1,node2,node3
```

Wait until the command completes and you are returned to the system prompt.

6. (Optional) If the deployment fails, you need to run the `tsm topology cleanup-coordination-service` command to remove the unsuccessfully deployed ensemble. For details on running the command, see `tsm topology cleanup-coordination-service`.

7. Start Tableau Server:

```
tsm start
```

For more information and details on deploying a new Coordination Service ensemble, see [Deploy a Coordination Service Ensemble](#).

Step 7: Configure Client File Services (CFS) on additional nodes

Add CFS to additional nodes. We recommend you add CFS to every node running the Coordination Service.

Note: This operation includes steps that you may need to perform using the TSM command line.

Tableau Server on Linux Administrator Guide

1. On the initial node, open a terminal session.
2. Find the node ID for the node you are adding CFS to:

```
tsm topology list-nodes -v
```

3. Add CFS on the node by specifying the node, the process, and a single instance.

For example, this command adds an instance of CFS to node2:

```
tsm topology set-process -n node2 -pr clientfileservice -c  
1
```

If you attempt to add an instance of CFS to a node that already is configured with CFS, an error message will let you know there is already an instance on the node.

To add CFS to additional nodes, repeat this step for each node.

4. Apply the changes:

```
tsm pending-changes apply
```

Step 7: Configure processes for the second node

1. Open TSM in a browser:

```
https://<tsm-computer-name>:8850
```

For more information, see [Sign in to Tableau Services Manager Web UI](#).

2. Click the **Configuration** tab.
3. Specify the processes and number of instances that should run on the first additional node.

In this example:

- a. Select **Gateway**.
- b. Set the **Application Server** (vizportal) count to 2.
- c. Set the **VizQL Server** count to 2.
- d. Set the **Cache Server** count to 2.
- e. Select **Search & Browse**.
- f. Set the **Backgrounder** count to 2.

Adding Backgrounder to a node will also add an instance of Data Engine if one is not already on the node.

- g. Set the **Data Server** count to 2.
- h. Select **File Store**.
- i. Select **Repository** (pgsql).

The specific processes and process counts you set will depend on your organizational environment and needs. Some processes are added automatically when you add another process. For more information, see [Configure Nodes and Tableau Server Processes](#).

Note: The TSM Web UI limits you to a maximum of 8 instances of processes that allow you to select the number of instances. To configure more instances than this, use the command line and the `TSM topology set-process` command. For more information, see `tsm topology set-process`.

Step 8: Configure processes for third node

1. In TSM, on the **Configuration** tab, specify the processes and number of instances that should run on the second additional node.

In this example:

- a. Select **Gateway**.
- b. Set the **Application Server** (vizportal) count to 2.
- c. Set the **VizQL Server** count to 2.
- d. Set the **Cache Server** count to 2.
- e. Select **Search & Browse**.
- f. Set the **Backgrounder** count to 2.

Adding Backgrounder to a node will also add an instance of Data Engine if one is not already on the node.

- g. Set the **Data Server** count to 2.
- h. Select **File Store**.

2. Click **Pending Changes** at the top of the page:



The Pending Changes list displays.

3. Click **Apply Changes and Restart** and **Confirm** to confirm a restart of Tableau Server.

Use the TSM CLI

Step 1: Install the initial node

See [Install and Configure Tableau Server](#).

Step 2: Generate a bootstrap file for the additional nodes

To install Tableau Server on additional computers you use the same installer you did for the initial node, along with a "bootstrap" file you generate on the initial node.

Important: The bootstrap file contains a copy of the master keystore file used for encrypting

the configuration secrets. The file can also embedded credentials which are valid for a pre-determined amount of time (see `tabadmincontroller.auth.expiration.minutes`) and serve as a session cookie. We strongly recommend that you take additional measures to secure the bootstrap file.

The following command set provides an example method to encrypt the bootstrap file output. This method is similar to the encryption process described in more detail at [Securing secrets for import and export operations](#).

Note however, the method here must be passed as separate arguments with trailing `&& \` operators as follows:

```
mkfifo -m 600 /tmp/secure1 && \
tsm topology nodes get-bootstrap-file --file /tmp/secure1 && \
gpg --symmetric --batch --yes --passphrase-file ~/.secret-
s/pgppassphrase.txt --cipher-algo AES256 --output encrypted.enc
< /tmp/secure1 && \
rm /tmp/secure1
```

1. After installing Tableau Server on the initial node, generate the node bootstrap file.
2. On the initial node, open a terminal session.
3. Type this command to generate the bootstrap file:

```
tsm topology nodes get-bootstrap-file --file <path\
file>.json
```

Beginning with version 2019.3, embedded credentials are included in the bootstrap file by default. If you don't want the bootstrap file to include credentials, use the `-nec` or `--no-embedded-credentials` option:

```
tsm topology nodes get-bootstrap-file --file <path\
file>.json --no-embedded-credentials.
```

See [What's New and Changed in Tableau Server](#) in [What's New](#) for more information. If you want to completely disable the ability to include embedded credentials in node bootstrap files, you can set a configuration option for the server. See [features.PasswordlessBootstrapInit](#) for more details.

Step 3: Install an additional node

1. Copy the original installer you used on the first computer along with the bootstrap file you generated and put them in a location accessible from the new computer you are adding Tableau Server to. This could be a mounted network share, or directly on the new computer.
2. If you are running a local firewall, then you need to configure firewall rules for all the nodes in the cluster. For more information, see [Configure Local Firewall](#).
3. On the new node, run the Tableau Server Setup program:

Use the package manager to install the Tableau Server package.

Do not install to a location using a symbolic link or to a directory on a Network File System (NFS) volume.

- On RHEL-like distributions, including CentOS, you have the option to install Tableau to a non-default location.
 - **Default location**—To install to the default location (`/opt/tableau/tableau_server`), run the following commands, where `<version>` is formatted as `major-minor-maintenance` (ex: `2019-2-5`):

```
sudo yum update

sudo yum install tableau-server-<version>.x86_64.rpm
```
 - **Non-default location**—To install to a non-default location, you must use

`rpm -i`. You will also need to install all dependent packages. See the note below.

Run the following command:

```
sudo rpm -i --prefix /preferred/install/path
tableau-server.rpm
```

Note: When you use `yum` to install Tableau Server, all dependent packages are automatically downloaded and installed. This is the preferred method for installing Tableau. If you want to install to a non-default location, or your organization does not allow you to use `yum` and you must install using `rpm -i`, you must also install all dependent packages separately. For information about installing dependent packages, see [Installing Tableau Server on an Air-Gapped Computer Running Linux](#).

- On Ubuntu, run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: 2019-2-5):

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get -y install gdebi-core
sudo gdebi -n tableau-server-<version>_amd64.deb
```

4. To initialize communications between your new node and the initial node, run the `initialize-tsm` script that is installed when you install Tableau Server.

On the new node:

Navigate to the `scripts` directory:

```
cd /opt/tableau/tableau_server/packages/scripts.<version_
code>/
```

5. Run the `initialize-tsm` script:

Tableau Server on Linux Administrator Guide

```
sudo ./initialize-tsm -b /path/to/<bootstrap>.json --accepteula
```

- Use the `-b` flag to provide the path to the bootstrap file that you copied to the computer. If you have encrypted the bootstrap file, then you must pipe the file as described in [Securing secrets for import and export operations](#).
- If the bootstrap file was generated without embedded credentials, use the `-u` flag to specify the user name of the administrative user on the initial node. This is the name of an administrative user on the computer, not the Tableau Server administrator. You will be prompted for the user password. For more information, see [tsm topology nodes get-bootstrap-file](#).

```
sudo ./initialize-tsm -b /path/to/<bootstrap>.json -u <admin-user-on-first-node> --accepteula
```

Important: You must enter the credentials of the same user that you used during the installation process on the initial node.

- The `--accepteula` flag accepts the Tableau Server End User License Agreement (EULA). The EULA is available in the following location:

```
/opt/tableau/tableau_server/packages/docs.<version_code>/EULA.rtf
```

Step 4: Install a second additional node

Install Tableau Server on the second additional node:

1. Copy the original installer you used on the first computer along with the bootstrap file you generated and put them in a location accessible from the new computer you are adding Tableau Server to. This could be a mounted network share, or directly on the new computer.
2. If you are running a local firewall, then you need to configure firewall rules for all the

nodes in the cluster. For more information, see [Configure Local Firewall](#).

3. On the new node, run the Tableau Server Setup program:

Use the package manager to install the Tableau Server package.

Do not install to a location using a symbolic link or to a directory on a Network File System (NFS) volume.

- On RHEL-like distributions, including CentOS, you have the option to install Tableau to a non-default location.

- **Default location**—To install to the default location (`/opt/tableau/tableau_server`), run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: 2019-2-5):

```
sudo yum update
```

```
sudo yum install tableau-server-<version>.x86_64.rpm
```

- **Non-default location**—To install to a non-default location, you must use `rpm -i`. You will also need to install all dependent packages. See the note below.

Run the following command:

```
sudo rpm -i --prefix /preferred/install/path  
tableau-server.rpm
```

Note: When you use `yum` to install Tableau Server, all dependent packages are automatically downloaded and installed. This is the preferred method for installing Tableau. If you want to install to a non-default

location, or your organization does not allow you to use `yum` and you must install using `rpm -i`, you must also install all dependent packages separately. For information about installing dependent packages, see [Installing Tableau Server on an Air-Gapped Computer Running Linux](#).

- On Ubuntu, run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: 2019-2-5):

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get -y install gdebi-core
sudo gdebi -n tableau-server-<version>_amd64.deb
```

4. To initialize communications between your new node and the initial node, run the `initialize-tsm` script that is installed when you install Tableau Server.

On the new node:

Navigate to the `scripts` directory:

```
cd /opt/tableau/tableau_server/packages/scripts.<version_
code>/
```

5. Run the `initialize-tsm` script:

```
sudo ./initialize-tsm -b /path/to/<bootstrap>.json --
accepteula
```

- Use the `-b` flag to provide the path to the bootstrap file that you copied to the computer. If you have encrypted the bootstrap file, then you must pipe the file as described in [Securing secrets for import and export operations](#).
- If the bootstrap file was generated without embedded credentials, use the `-u` flag to specify the user name of the administrative user on the initial node. This

is the name of an administrative user on the computer, not the Tableau Server administrator. You will be prompted for the user password. For more information, see `tsm topology nodes get-bootstrap-file`.

```
sudo ./initialize-tsm -b /path/to/<bootstrap>.json -u
<admin-user-on-first-node> --accepteula
```

Important: You must enter the credentials of the same user that you used during the installation process on the initial node.

- The `--accepteula` flag accepts the Tableau Server End User License Agreement (EULA). The EULA is available in the following location:

```
/opt/tableau/tableau_server/packages/docs.<version_
code>/EULA.rtf
```

Step 5: Add a process to the additional nodes

1. On the initial node, configure a cluster controller instance on each additional node:

```
tsm topology set-process -n <nodeID_second-node> -pr
clustercontroller -c 1
```

```
tsm topology set-process -n <nodeID_third-node> -pr
clustercontroller -c 1
```

2. Apply the node configuration changes:

```
tsm pending-changes apply --ignore-warnings
```

A warning about deploying a Coordination Service ensemble displays because you have deployed a three-node cluster. Use the `--ignore-warnings` option to apply the configuration changes in spite of the warning. You will deploy a new Coordination Service ensemble in the next step.

Step 6: Deploy a Coordination Service ensemble

If you install a total of three or more nodes, you should also deploy a Coordination Service ensemble. If you do not, you will get a warning message every time you make changes to the server configuration or topology. You can ignore this message, but as a best practice you should deploy a multi-node Coordination Service ensemble.

When you install Tableau Server, a single instance of the Coordination Service is installed on the initial node. TSM and Tableau Server depend on the Coordination Service to function properly, so to provide redundancy and ensure availability on multi-node installations, configure additional instances of the Coordination Service by deploying a Coordination Service ensemble. Coordination Service ensembles are installed with one, three, or five instances of the Coordination Service. In a three-node installation of Tableau Server, the recommended number of Coordination Service instances is three, one on each node.

Do not attempt to deploy a Coordination Service ensemble if there are other changes pending. Discard or apply any pending changes before deploying a new Coordination Service ensemble.

Note: This operation includes steps that you may need to perform using the TSM command line.

The following steps illustrate how to deploy a new Coordination Service ensemble on an existing three-node Tableau Server cluster and clean up the old ensemble.

1. On the initial node, open a terminal session.
2. Stop Tableau Server:

```
tsm stop
```

If prompted, sign in as a TSM administrator.

Some TSM processes will continue to run, including the Administration Controller and Administration Agent.

3. Confirm there are no pending changes:

```
tsm pending-changes list
```

If there are pending changes, you need to either discard the changes or apply them.

Applying pending changes will take some time:

- Discard the changes

```
tsm pending-changes discard
```

or

- Apply the changes:

```
tsm pending-changes apply
```

Wait until the command completes and you are returned to the system prompt.

4. Get the node IDs for each node in the cluster:

```
tsm topology list-nodes -v
```

5. Use the `tsm topology deploy-coordination-service` command to add a new Coordination Service ensemble by adding the Coordination Service to specified nodes. You must specify the node(s) that the Coordination Service should be added to. The command also makes the new ensemble the "production" ensemble (the ensemble in use) and removes the old ensemble, unless the deployment fails. If this happens, see step 6 below.

Note: A "y/n" prompt displays confirming that a server restart will take place. To run the command without input, include the `--ignore-prompt` option.

Tableau Server on Linux Administrator Guide

For example, deploy the Coordination Service to all three nodes of a three-node cluster, where the nodes are node1, node2, and node3:

```
tsm topology deploy-coordination-service -n  
node1,node2,node3
```

Wait until the command completes and you are returned to the system prompt.

6. (Optional) If the deployment fails, you need to run the `tsm topology cleanup-coordination-service` command to remove the unsuccessfully deployed ensemble. For details on running the command, see `tsm topology cleanup-coordination-service`.

7. Start Tableau Server:

```
tsm start
```

For more information and details on deploying a new Coordination Service ensemble, see [Deploy a Coordination Service Ensemble](#) .

Step 7: Configure Client File Services (CFS) on additional nodes

Add CFS to additional nodes. We recommend you add CFS to every node running the Coordination Service.

Note: This operation includes steps that you may need to perform using the TSM command line.

1. On the initial node, open a terminal session.
2. Find the node ID for the node you are adding CFS to:

```
tsm topology list-nodes -v
```

3. Add CFS on the node by specifying the node, the process, and a single instance.

For example, this command adds an instance of CFS to node2:

```
tsm topology set-process -n node2 -pr clientfileservice -c
1
```

If you attempt to add an instance of CFS to a node that already is configured with CFS, an error message will let you know there is already an instance on the node.

To add CFS to additional nodes, repeat this step for each node.

4. Apply the changes:

```
tsm pending-changes apply
```

Step 7: Configure processes for the second node

1. On the initial node, set the topology for the second node. The topology specifies which processes should run on the node, and how many instances of each process should run. For details about configuring nodes, see [Configure Nodes](#). Data engine will be added automatically when you add other processes. For details on when data engine is added, see [Tableau Server Processes](#).

- a. Get the node-id for the new node: `tsm topology list-nodes -v`

- b. Specify individual processes that will run on the new node:

```
tsm topology set-process -n <node-id> -pr gateway -c 1
```

```
tsm topology set-process -n <node-id> -pr vizqlserver
-c 2
```

```
tsm topology set-process -n <node-id> -pr vizportal -c
2
```

```
tsm topology set-process -n <node-id> -pr backgrounder
-c 2
```

Tableau Server on Linux Administrator Guide

```
tsm topology set-process -n <node-id> -pr cacheserver -c 2
```

```
tsm topology set-process -n <node-id> -pr searchserver -c 1
```

```
tsm topology set-process -n <node-id> -pr dataserver -c 2
```

```
tsm topology set-process -n <node-id> -pr filestore -c 1
```

```
tsm topology set-process -n <node-id> -pr pgsql -c 1
```

2. Apply the node configuration changes. You will be prompted with a message that Tableau Server will restart.

```
tsm pending-changes apply
```

Step 8: Configure processes for third node

On the initial node, set the topology for the new node. The topology specifies which processes should run on the node, and how many instances of each process should run. Data engine will be added automatically when you add other processes. For details on when data engine is added, see [Tableau Server Processes](#).

1. Get the node-id for the new node:

```
tsm topology list-nodes -v
```

2. Specify individual processes that will run on the new node:

```
tsm topology set-process -n <node-id> -pr gateway -c 1
```

```
tsm topology set-process -n <node-id> -pr vizqlserver -c 2
```

```
tsm topology set-process -n <node-id> -pr vizportal -c 2
```

```
tsm topology set-process -n <node-id> -pr backgrounder -c  
2
```

```
tsm topology set-process -n <node-id> -pr cacheserver -c 2
```

```
tsm topology set-process -n <node-id> -pr searchserver -c  
1
```

```
tsm topology set-process -n <node-id> -pr dataserver -c 2
```

```
tsm topology set-process -n <node-id> -pr filestore -c 1
```

3. Apply the node configuration. You will be prompted with a message that Tableau Server will restart.

```
tsm pending-changes apply
```

4. Start the server:

```
tsm start
```

Step 9: Configure firewall rules (optional)

If you are running a local firewall, then you need to configure firewall rules for all the nodes in the cluster. For more information, see [Configure Local Firewall](#). For more information, see [Local firewall configuration](#).

Add a Load Balancer

At this point, all three nodes have gateways, which are used to route requests to available server processes. All gateways are active, but to further reduce the potential for downtime in the cluster, you can configure a load balancer. For more information, see [Add a Load Balancer](#).

Add a Load Balancer

You can enhance the reliability of Tableau Server by running gateways on multiple nodes, and configuring a load balancer to distribute requests across the gateways. Unlike the

repository process, which can be active or passive, all gateway processes are active. If one gateway in a cluster becomes unavailable, the load balancer stops sending requests to it. The load balancer algorithm you choose determines how the gateways will route client requests.

Notes: If you will be using Kerberos authentication, you need to configure Tableau Server for your load balancer before you configure Tableau Server for Kerberos. For more information, see [Configure Kerberos](#).

Windows SSPI authentication is not supported with a load balancer.

Note the following as you configure your load balancer to work with Tableau Server:

- **Tested load balancers:** Tableau Server clusters with multiple gateways have been tested with Apache and F5 load balancers.

If you are using an Apache load balancer and creating custom administrative views, you need to connect directly to the Tableau Server repository. You cannot connect through the load balancer.

- **Tableau Server URL:** When a load balancer is in front of a Tableau Server cluster, the URL that's accessed by Tableau Server users belongs to the load balancer, not the initial Tableau Server node.
- **Single load balancer endpoint:** You must configure your load balancers for a single URL endpoint. You cannot configure different endpoint hosts to redirect to the same Tableau Server deployment. The single external URL is defined in `gateway.public.host` when you configure Tableau Server, as described in the procedure below.
- **Trusted host settings:** The computer running the load balancer must be identified to Tableau Server as a trusted host. See the procedure below for how to configure

Tableau Server.

- **Proxy server configurations:** The settings used to identify a load balancer to Tableau Server are the same ones that are used to identify a proxy server. If your Tableau Server cluster requires both a proxy server and a load balancer, both must use a single external URL defined in `gateway.public.host` and all proxy servers and load balancers must be specified in `gateway.trusted` and `gateway.trusted_hosts`. For more information, see [Configuring Proxies for Tableau Server](#).

Configure Tableau Server to work with a load balancer

Use the following key-value pairs to configure Tableau Server for a load balancer. You can set the values individually with the `tsm configuration set` command.

Or, use the configuration file template example at the end of this topic to create a `configKey.json` file. After you have filled in the options with the appropriate values, pass the `json` file and apply settings with `tsm` commands.

Load balancer configuration key-value pairs

The key-value pairs described in this section can be set with the `tsm configuration set` command. After you have set all values, run the following command:

```
tsm pending-changes apply
```

```
gateway.public.host
```

Value: `name`

The value, `name` is the URL that will be used to reach Tableau Server through the load balancer.

For example, if Tableau Server is reached by entering `tableau.example.com` in a browser address bar, enter this command:

Tableau Server on Linux Administrator Guide

```
tsm configuration set -k gateway.public.host -v "tableau-  
.example.com"
```

gateway.public.port

Value: port_number

By default, Tableau assumes that the load balancer is listening on port 80 for external communications. Use this key to specify a different port.

For example, if your load balancer is configured for SSL and listening on port 443, enter the following command:

```
tsm configuration set -k gateway.public.port -v "443"
```

gateway.trusted

Value: server

Specify the load balancer(s) IPv4 address or computer name.

The value for `server` can be a comma-separated list, for example:

```
tsm configuration set -k gateway.trusted -v "10.32.139.45,  
10.32.139.46, 10.32.139.47"
```

or

```
tsm configuration set -k gateway.trusted -v "proxy1, proxy2,  
proxy3"
```

gateway.trusted_hosts

Value:

Specify alternate names for the load balancer(s), such as its fully-qualified domain name, any non-fully-qualified domain names, and any aliases. These are the names a user might type in a browser. Separate each name with a comma:

```
tsm configuration set -k gateway.trusted_hosts -v "name1, name2,
name3"
```

For example:

```
tsm configuration set -k gateway.trusted_hosts -v "lb.example.com,
lb, ftp.example.com, www.example.com"
```

Configuration template example

You can configure all of the settings in one operation by customizing the following configuration template example and creating a configKey json file.

Refer to the key-value pair descriptions above to customize the values for your organization.

```
{
  "configKeys": {
    "gateway.public.host": "tableau.example.com",
    "gateway.public.port": "443",
    "gateway.trusted": "proxy1, proxy2, proxy3",
    "gateway.trusted_hosts": "lb.example.com, lb, ftp.example.com,
www.example.com"
  }
}
```

When you are finished, save the json file.

Pass the json file with tsm to set the configuration, and then apply settings by running the following commands:

```
tsm settings import -f /path/to/file.json
```

```
tsm pending-changes apply
```

For more information about using configKey schemes, see Configuration File Example.

Deploy a Coordination Service Ensemble

The Coordination Service is built on [Apache ZooKeeper](#), an open-source project, and coordinates activities on the server, guaranteeing a quorum in the event of a failure, and serving as the source of "truth" regarding the server topology, configuration, and state. The service is installed automatically on the initial Tableau Server node, but no additional instances are installed as you add additional nodes. Because the successful functioning of Tableau Server depends on a properly functioning Coordination Service, we recommend that for server installations of three or more nodes, you add additional instances of the Coordination Service by deploying a new Coordination Service ensemble. This provides redundancy and improved availability in the event that one instance of the Coordination Service has problems.

Important: The process to deploy a Coordination Service ensemble changed as of version 2020.1.0. If you are running an earlier version of Tableau Server, see the documentation for that version. You can find documentation for all supported versions here: [Tableau Help](#)

- Hardware requirements
- The Coordination Service Quorum
- Deploy a new Coordination Service ensemble

Hardware requirements

The hardware you use for Tableau Server can have an effect on how well the Coordination Service runs. In particular:

- **Memory.** The Coordination Service maintains state information in memory. By design, the memory footprint is small, and is typically not a factor in overall server performance.
- **Disk speed.** Because the service stores state information on disk, it benefits from fast

disk speed on the individual node computers.

- **Connection speed** between nodes. The service communicates continuously between cluster nodes; a fast connection speeds between nodes helps with efficient synchronization.

Because the Coordination Service is I/O intensive, if you are running Tableau Server on computers that meet or just exceed the minimum hardware requirements, you may want to configure a Coordination Service ensemble that puts the service on nodes that are not being used for other server processes. This reduces the chance of delays due to I/O contention between server processes. For information on how to deploy an ensemble on dedicated Coordination Service-only nodes, see [Configure Tableau Server for High Availability with Coordination Service-Only Nodes](#).

The Coordination Service Quorum

To ensure that the Coordination Service can work properly, the service requires a *quorum*—a minimum number of instances of the service. This means that the number of nodes in your installation impacts how many instances of the Coordination Service you want to configure in your ensemble.

Number of Coordination Service instances to use

The maximum number of Coordination Service instances you can have in an ensemble on Tableau Server depends on how many Tableau Server nodes you have in your deployment. Configure a Coordination Service ensemble based on these guidelines:

Total number of server nodes	Recommended number of Coordination Service nodes in ensemble (must be 1, 3, or 5)	Notes
1-2 nodes	1 node	This is the default and requires no changes unless you want to move the Coordination Service off your initial node and onto your additional node.

Total number of server nodes	Recommended number of Coordination Service nodes in ensemble (must be 1, 3, or 5)	Notes
3-4 nodes	3 nodes	
5 or more nodes	3 nodes or 5 nodes	<p>Five is the maximum number of Coordination Service instances you can install. A 3-node Coordination Service ensemble allows for one of the ensemble nodes to fail without causing Tableau Server to fail. A 5-node ensemble allows for two of the ensemble nodes to fail without causing Tableau Server to fail.</p> <p>For most installations, three Coordination Service nodes are adequate, and because of the I/O-intensive nature of the Coordination Service, this is the most performant configuration.</p> <p>If high availability is your absolute priority, you may want to consider deploying a 5-node Coordination Service ensemble. This provides the most redundancy in the event that one or more nodes fail but will require more system resources. A maximum of two of the ensemble nodes can fail without impacting Tableau Server (as long as any other services on the node also exist on still-functioning nodes).</p> <p>To reduce performance impact, locate the Coordination Service on nodes that are running fewer other services or consider using Coordination Service-only nodes. For details, see Configure Tableau Server for High Availability with Coordination Service-Only Nodes.</p>

Deploy a new Coordination Service ensemble

If you install a total of three or more nodes, you should also deploy a Coordination Service ensemble. If you do not, you will get a warning message every time you make changes to the server configuration or topology. You can ignore this message, but as a best practice you should deploy a multi-node Coordination Service ensemble.

When you install Tableau Server, a single instance of the Coordination Service is installed on the initial node. TSM and Tableau Server depend on the Coordination Service to function properly, so to provide redundancy and ensure availability on multi-node installations, configure additional instances of the Coordination Service by deploying a Coordination Service ensemble. Coordination Service ensembles are installed with one, three, or five instances of the Coordination Service. In a three-node installation of Tableau Server, the recommended number of Coordination Service instances is three, one on each node.

Do not attempt to deploy a Coordination Service ensemble if there are other changes pending. Discard or apply any pending changes before deploying a new Coordination Service ensemble.

Note: This operation includes steps that you may need to perform using the TSM command line.

The following steps illustrate how to deploy a new Coordination Service ensemble on an existing three-node Tableau Server cluster and clean up the old ensemble.

1. On the initial node, open a terminal session.
2. Stop Tableau Server:

```
tsm stop
```

If prompted, sign in as a TSM administrator.

Some TSM processes will continue to run, including the Administration Controller and Administration Agent.

3. Confirm there are no pending changes:

```
tsm pending-changes list
```

If there are pending changes, you need to either discard the changes or apply them.

Applying pending changes will take some time:

- Discard the changes

```
tsm pending-changes discard
```

or

- Apply the changes:

```
tsm pending-changes apply
```

Wait until the command completes and you are returned to the system prompt.

4. Get the node IDs for each node in the cluster:

```
tsm topology list-nodes -v
```

5. Use the `tsm topology deploy-coordination-service` command to add a new Coordination Service ensemble by adding the Coordination Service to specified nodes. You must specify the node(s) that the Coordination Service should be added to. The command also makes the new ensemble the "production" ensemble (the ensemble in use) and removes the old ensemble, unless the deployment fails. If this happens, see step 6 below.

Note: A "y/n" prompt displays confirming that a server restart will take place. To run the command without input, include the `--ignore-prompt` option.

For example, deploy the Coordination Service to all three nodes of a three-node cluster, where the nodes are node1, node2, and node3:

```
tsm topology deploy-coordination-service -n  
node1,node2,node3
```

Wait until the command completes and you are returned to the system prompt.

6. (Optional) If the deployment fails, you need to run the `tsm topology cleanup-coordination-service` command to remove the unsuccessfully deployed ensemble. For details on running the command, see `tsm topology cleanup-coordination-service`.
7. Start Tableau Server:

```
tsm start
```

Configure Client File Service

The Client File Service (CFS) manages most shared files in a multinode cluster. For example, authentication related certificates, keys, and files (OpenID, mutual SSL, SAML, and Kerberos), and customization files are managed by CFS.

A single instance of CFS is configured on the initial node when you first install Tableau Services Manager (TSM) and Tableau Server. You can and should configure additional instances on other nodes.

In a cluster, if a node that is running your only instance of CFS fails, any files being managed by CFS will be lost, and you will need to repopulate CFS those files by reimporting certs and custom images, and making any related configuration changes. For a list of files managed by CFS, see [Tableau Server Client File Service](#).

Keep in mind these limitations and recommendations:

- There must be at least one instance of CFS for any Tableau Server installation. You cannot remove all instances of CFS.

Tableau Server on Linux Administrator Guide

- We recommend you do not configure more than one instance on any one node. Additional instances would not provide any benefit and would be a waste of resources.
- We recommend you configure a CFS instance on each node where you deploy the Coordination Service. This provides redundancy and helps safeguard Tableau Server from unexpected errors in case a node fails.

This article explains how to add or remove an instance of CFS. You cannot configure CFS from the Web UI. You must use the CLI to add or remove CFS. The procedure is the same one you use for adding a process to a node.

Configure CFS on additional nodes

1. On the initial node, open a terminal session.
2. Find the node ID for the node you are adding CFS to:

```
tsm topology list-nodes -v
```

3. Add CFS on the node by specifying the node, the process, and a single instance.

For example, this command adds an instance of CFS to node2:

```
tsm topology set-process -n node2 -pr clientfileservice -c  
1
```

If you attempt to add an instance of CFS to a node that already is configured with CFS, an error message will let you know there is already an instance on the node.

To add CFS to additional nodes, repeat this step for each node.

4. Apply the changes:

```
tsm pending-changes apply
```

Repository Failover

In a Tableau Server installation, the repository (pgsql) database is one of the key required processes. The Tableau Server repository stores information about Tableau Server users, groups and group assignments, permissions, projects, data sources, and extract metadata and refresh information. Because it is critical to the server functioning, Tableau Server has a built-in automatic "failover" for the repository when server is installed in a distributed environment that meets certain requirements.

Automatic repository failover

Automatic repository failover means that if there is a problem with the active Tableau Server repository, the server will automatically switch to using the passive repository. This does not happen immediately, to protect against momentary issues with the repository that don't justify a switch, but if the repository is unavailable for more than five minutes, failover occurs.

For automatic repository failover to work, your Tableau Server installation needs:

- A minimum of three nodes
- Two instances of the repository installed

Optional but highly recommended:

- A multi-node Coordination Service ensemble deployed

With these conditions satisfied, repository failover will occur if the active repository becomes unavailable, either due to a problem with the process, or a problem with the node the process is running on. If the original repository becomes available again (if, for example, the node is restarted and all processes come up properly), it is made the passive repository, available for failover if necessary.

Manual repository failover

There may be reasons you want to shift back to the original repository after failover occurs. One reason would be if that instance of the repository is installed on a computer with more

resources. To do this, use the `tsm topology failover-repository` command to manually switch back to the original repository. For more information, see `tsm topology failover-repository`.

Preferred active repository

When you configure Tableau Server you have the option to specify a node as the preferred active repository. When Tableau Server is configured for repository failover, the preferred active repository node is the one used for the active repository. This is an optional step, and if you do not specify a preferred active repository node, Tableau Server will select the active repository node on startup.

To configure the preferred active repository, use the `tsm configuration set` command to configure the `pgsql.preferred_host` option:

```
tsm configuration set -k pgsql.preferred_host -v "<host_name>"
```

Note: The `host_name` is case-sensitive and must match the node name shown in the output of `tsm status -v`.

Configure a preferred active repository node if you want Tableau Server to select a specific node on startup. You might want to do this if you have a particular server you want to use for your active repository (a computer with more disk space or memory for example), or if you are using custom administrative views. Custom administrative views have embedded connection information that refers to the repository for which you created the views. For more information on connecting to the Tableau Server repository, see [Collect Data with the Tableau Server Repository](#)

Recover from an Initial Node Failure

The first computer you install Tableau on, the "initial node," has some unique characteristics. Three processes run only on the initial node and cannot be moved to any other node except in a failure situation, the License Service (License Manager), Activation Service, and TSM

Controller (Administration Controller). Tableau Server includes a script that automates moving these processes to one of your other existing nodes so you can get complete access back to TSM and keep Tableau Server running.

Two other processes are initially included on the initial node but can be added or moved to additional nodes, the CFS (Client File Service) and the Coordination Service. Depending on how your installation was configured with CFS and Coordination Service, you may also need to take steps to redeploy these.

If an initial node fails

If there is a problem with the initial node and you have redundant processes on your other nodes, there is no guarantee that Tableau Server will continue to run. Tableau Server can continue to run for up to 72 hours after an initial node failure, before the lack of the licensing service impacts other processes. If so, your users may be able to continue to sign in and see and use their content after the initial node fails, but you will not be able to reconfigure Tableau Server because you won't have access to the Administration Controller.

Even when configured with redundant processes, *it is possible that Tableau Server may not continue to function after the initial node fails*. This is true even when an installation is configured for high availability. This means you should make a point of moving the two unique processes to another of your running nodes as soon as possible. If your initial node fails for reasons that are recoverable in a relatively short amount of time (for example, a hardware failure you can correct), you should first attempt to bring the node back up without using the procedure below.

Note: The steps in this article require server downtime and can be disruptive, and should only be used in the event of a catastrophic failure of the initial node. If you are unable to get your initial node running again, use the following steps to move key TSM processes to another node in your cluster.

General requirements

The 2021.1 version of Tableau Server has been updated with improved recovery functionality. The procedure in this topic has been written for Tableau Server 2021.1.

If you are attempting to recover a failed node from an earlier version of Tableau Server, you must follow the procedure for that version. To view archived versions of Tableau online help, see [Tableau Help](#).

- As part of the process for setting up a multi-node Tableau Server installation you should have deployed a Coordination Service ensemble. The process below assumes there was a Coordination Ensemble deployed before there was a problem with the initial node. For more information about deploying a Coordination Service ensemble, see [Deploy a Coordination Service Ensemble](#).
- This process assumes that you have configured instances of Client File Service (CFS) on every node that is running the Coordination Service. If you did not add additional instances of CFS, your only instance was on the initial node, and you will need to add at least one instance of CFS to another node. You will also need to repopulate CFS. Tableau Server requires at least one instance of the CFS. For more information, see [Configure Client File Service](#) and [Tableau Server Client File Service](#).

Note: This operation includes steps that you may need to perform using the TSM command line.

Move the TSM Controller, License Service, and Activation Service to another node

If there is a problem with the initial node, the TSM Controller, the Licensing Service, and Activation Service need to be started on another node. Follow these steps to use the provided `move-tsm-controller` script and get the TSM Controller, Licensing Service, and Activation Service working on another node.

1. On a node that is still working, run the Controller recovery script. At a terminal prompt on a working node, type the following command:

```
sudo /opt/tableau/tableau_server-
/packages/scripts.<version_code>/move-tsm-controller -n
<nodeID>
```

where "nodeID" is the ID for the node you want the TSM Controller to run on. For example:

```
sudo /opt/tableau/tableau_server-
/packages/scripts.10400.17.0802.1319/move-tsm-controller -
n node2
```

2. Verify the Administration Controller is running on the node:

```
tsm status -v
```

3. Stop Tableau Server.

The remainder of this procedure includes some commands with the `--ignore-node-status` option. When a command is run with the `--ignore-node-status` option, the command will run without consideration of the status of the specified node. To use `--ignore-node-status`, specify the failed node:

```
tsm stop --ignore-node-status <nodeID>
```

For example, if node1 has failed, run the command as follows:

```
tsm stop --ignore-node-status node1
```

4. Add the License Service to the node:

```
tsm topology set-process -pr licenseservice -n <nodeID> -c
1
```

5. Remove the old License Service from the original node, where "nodeID" is the initial

Tableau Server on Linux Administrator Guide

node that has failed:

```
tsm topology set-process -pr licenseservice -n <nodeID> -c 0
```

6. If you're using Server ATR, add the Activation Service to the node:

```
tsm topology set-process -pr activationsservice -n <nodeID> -c 1
```

7. If you're using Server ATR, remove the old Activation Service from the original node, where "nodeID" is the initial node that has failed:

```
tsm topology set-process -pr activationsservice -n <nodeID> -c 0
```

Important: In a cluster, if a node that is running your only instance of CFS fails, any files being managed by CFS will be lost, and you will need to repopulate CFS those files by reimporting certs and custom images, and making any related configuration changes. For a list of files managed by CFS, see [Tableau Server Client File Service](#).

8. If the initial node had been running the Messaging Service, add the Messaging Service to this node:

```
tsm topology set-process -pr activemqserver -n node2 -c 1
```

9. (Optional) You can also add other processes that had been running on the initial node but are not running on this node. For example, to add an cache server:

```
tsm topology set-process -pr cacheserver -n node2 -c 1
```

10. Apply the changes:

```
tsm pending-changes apply --ignore-node-status <nodeID>
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

11. Restart the TSM Administration Controller (as *tableau* system account):

```
sudo su -l tableau -c "systemctl --user restart tabadmincontroller_0.service"
```

Note: It may take a few minutes for `tabadmincontroller` to restart. If you attempt to apply pending changes in the next step before the controller has fully restarted, TSM will not be able to connect to the controller. You can verify that the controller is running by using the `tsm status -v` command. Tableau Server Administration Controller should be listed as "is running".

12. Apply pending changes (there may not appear to be any, but this step is required):

```
tsm pending-changes apply --ignore-node-status <nodeID>
```

13. Activate the Tableau Server license on the new Controller node:

```
tsm licenses activate -k <product-key>
```

14. Verify the license is properly activated:

```
tsm licenses list
```

15. If the initial node was running the Coordination Service, you need to deploy a new Coordination Service ensemble that does not include that node. If you have a three node cluster and the initial node was running the Coordination Service, you must

deploy a new, single-instance Coordination Service ensemble on a different node and clean up the old ensemble. In this example, a single instance of the Coordination Service is being deployed to the second node:

```
tsm topology deploy-coordination-service -n node2 --ignore-node-status node1
```

16. If the initial node was running a File Store instance, you need to remove that instance:

```
tsm topology filestore decommission -n <nodeID> --delete-filestore
```

Where `nodeID` is the initial node that has failed.

17. Apply pending changes, using the `--ignore-warnings` flag if the new Coordination Service ensemble you deployed above is a single node ensemble:

```
tsm pending-changes apply --ignore-node-status node1 --ignore-warnings
```

18. Remove the initial node, where `nodeID` is the initial node that has failed:

```
tsm topology remove-nodes -n <nodeID>
```

19. Apply pending changes, using the `--ignore-warnings` flag if the new Coordination Service ensemble you deployed above is a single node ensemble:

```
tsm pending-changes apply --ignore-warnings
```

20. Start Tableau Server:

```
tsm start
```

At this point your server should start, and you will be able to use TSM to configure it. The next step is to replace your initial node so your cluster has the original number of nodes. How you do this depends on whether or not you want to reuse the node that

failed. We recommend that you only reuse that node if you are able to identify the reason it failed, and take steps to keep the failure from recurring.

21. If you plan to reuse the original node, you first need to completely remove Tableau from it. Do this by running the `tableau-server-obliterate` script. For details on doing this, see [Remove Tableau Server from Your Computer](#).
22. On a fresh computer, or on your original computer after completely removing Tableau, install Tableau using your original Setup program and a bootstrap file generated from the node that is now running the Administration Controller and Licensing Service. This creates an additional node you can configure as part of your cluster. For details on how to add the node, see [Install and Configure Additional Nodes](#).

A best practice is to configure any processes you lost when the original node failed, to make sure your cluster is fully redundant. You may want to move processes from your new initial node to the newly added additional node to duplicate your original configuration. For example, if your initial node was only running gateway and File Store, you may want to configure the new initial node the same way.

23. You should also redeploy a new Coordination Service ensemble, once you have your nodes up and running the way you want. For details, see [Deploy a Coordination Service Ensemble](#).
24. Finally, if you have not already done this, add an instance of CFS to every node that is running the Coordination Service. For more information, see [Configure Client File Service](#)

In a cluster, if a node that is running your only instance of CFS fails, any files being managed by CFS will be lost, and you will need to repopulate CFS those files by reimporting certs and custom images, and making any related configuration changes. For a list of files managed by CFS, see [Tableau Server Client File Service](#).

Recover from a Node Failure

If there is a problem with one of your server nodes, and you have redundant processes on your other nodes, Tableau Server can continue to run. Your users can continue to sign in and see and use their content after the node fails, but they may experience performance degradation as a result of the failed node. In addition, your server will be at greater risk of catastrophic failure if the bad node was running processes that are no longer redundant. This means you should make a point of removing the bad node and replacing it as soon as you can. If your node fails for reasons that are recoverable in a relatively short amount of time (for example, a hardware failure you can correct), you should first attempt to bring the node back up without using the procedure below.

Note: If the failed node is your *initial* node, there are larger implications for your Tableau Server installations. For details on how to recover from the failure of an initial node, see [Recover from an Initial Node Failure](#).

General requirements

The 2020.1 version of Tableau Server has been updated with improved recovery functionality. The procedure in this topic has been written for Tableau Server 2020.1.

If you are attempting to recover a failed node from an earlier version of Tableau Server, you must follow the procedure for that version. To view archived versions of Tableau online help, see [Tableau Help](#).

- There is at least one functioning node with an instance of the File Store on it.
- There is at least one functioning node with a Repository on it.
- There is at least one functioning node with the Client File Service (CFS) on it.

Note: This operation includes steps that you may need to perform using the TSM command line.

Removing a Failed Node

To remove a failed node from a Tableau Server cluster:

1. Identify the failed node:

```
tsm status -v
```

The failed node will have a status of "ERROR" and processes will show as unavailable. The node ID is listed as "node<n>" with the machine name following it. For example, node3:

```
node3: WIN-00915SFASVH
                                     Status: ERROR
                                     'Tableau Server Gateway 0' status
```

2. Stop Tableau Server.

The remainder of this procedure includes some commands with the `--ignore-node-status` option. When a command is run with the `--ignore-node-status` option, the command will run without consideration of the status of the specified node. To use `--ignore-node-status`, specify the failed node:

```
tsm stop --ignore-node-status <nodeID>
```

For example, if node3 has failed, run the command as follows:

```
tsm stop --ignore-node-status node3
```

3. Determine any key processes that were running on the node:
 - If the failed node was running the Messaging Service, you need to remove the service from the failed node and add it to a new node.

Remove it from the failed node:

Tableau Server on Linux Administrator Guide

```
tsm topology set-process -pr activemqserver -n <nodeID>
-c 0
```

Add it to a new node:

```
tsm topology set-process -pr activemqserver -n <nodeID>
-c 1
```

- If the failed node was running the Coordination Service, you need to deploy a new ensemble before you can remove the node:

```
tsm topology deploy-coordination-service -n <good_
nodeID> --ignore-node-status <failed_nodeID>
```

- If the failed node was running the only instance of Client File Service (CFS), you need to configure a new instance of CFS on a working node. We recommend that you configure CFS on every node that is running the Coordination Service. For detail steps, see [Configure Client File Service](#) .
- If the failed node was running File Store, you need to force-decommission File Store and remove it before you can remove the node.

```
tsm topology filestore decommission -n <nodeID> --
delete-filestore
```

Apply pending changes (use `--ignore-warnings` option if you had a three node cluster and a single Coordination Service instance):

```
tsm pending-changes apply --ignore-warnings --ignore-
node-status <nodeID>
```

4. If the cluster was a three-node cluster and there are repositories on the remaining working nodes, you need to either remove one repository, or add a new node. This is because you are limited to a single instance of the repository when you have fewer than three nodes.

To remove one repository:

```
tsm topology set-process -n <nodeID> -pr pgsql -c 0
```

5. Run the command to remove the failed node. This adds the change to the pending changes list:

```
tsm topology remove-nodes -n <nodeID>
```

6. Verify the node removal is pending:

```
tsm pending-changes list
```

7. Apply pending changes to remove the node:

```
tsm pending-changes apply
```

8. Start Tableau Server:

```
tsm start
```

9. Install Tableau Server on a new node and configure the node with the processes that the old, failed node had been running.
10. On a fresh computer, or on your original computer after completely removing Tableau, install Tableau using your original Setup program and a bootstrap file generated from the initial node. For details on how to do this, see [Install and Configure Additional Nodes](#).

A best practice is to configure any processes you lost when the original node failed, to make sure your cluster is fully redundant.

11. You should also redeploy a new Coordination Service ensemble, once you have your nodes up and running the way you want. For details, see [Deploy a Coordination Service Ensemble](#).
12. Finally, if you have not already done this, add an instance of CFS to every node that is running the Coordination Service. For more information, see [Configure Client File Service](#).

Configure Nodes

Use the Tableau Services Manager (TSM) Web UI or CLI commands to configure the topology of a node. The initial node is configured with a default that includes all the processes used by TSM and Tableau Server. When you add additional nodes you need to specify which processes will run on those nodes, and how many instances of those processes will run. You may also want to change the topology of the initial node, either adding instances of existing processes, or moving some of those processes to your additional nodes (this is common when setting up a distributed installation of Tableau Server).

Note: You cannot remove the File Store or Repository (pgsql) if this is the only instance of that process in your cluster. You must have at least one instance of each of the processes in your Tableau Server installation, and you must add the second instance and allow it to synchronize with the first before you remove the process on the original node. For more information see [Move the Repository Process](#) and [Move the File Store Process](#).

Use the TSM web interface

In most cases, you can make multiple changes to your server configuration. Exceptions are if you are moving or removing a File Store instance or the Repository.

To configure your Tableau Server nodes using the TSM web interface, do the following:

1. Open TSM in a browser:

`https://<tsm-computer-name>:8850`

For more information, see [Sign in to Tableau Services Manager Web UI](#).

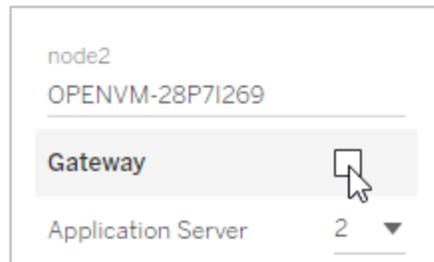
2. Click the **Configuration** tab.

Your next steps depend on the configuration changes you want to make.

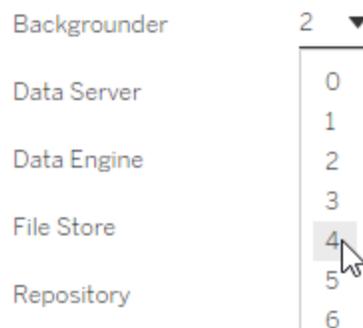
- **Add processes**—Add processes to a node by specifying the number of instances you want on the node, or selecting the box for the process.

For example, to add four instances of the Backgrounder and a Gateway to a node:

- Select the **Gateway** box:



- Set the **Backgrounder** count to 4:



Adding Backgrounder to a node will also add an instance of Data Engine if one is not already on the node.

Note: The TSM Web UI limits you to a maximum of 8 instances of processes that allow you to select the number of instances. To configure more instances than this, use the command line and the `TSM`

`topology set-process` command. For more information, see `tsm topology set-process`.

- **Change process count**—Change the number of processes on a node by specifying the new number, or selecting the box for the process.
- **Remove a process completely**—Remove all instances of a process from a node by clearing the box for the process, or setting the count to 0 (zero).

In most cases you move a process from one node to another by setting the process instance count on the first node to 0 (zero), and setting the count to a non-zero value on the second node.

If you are attempting to make a configuration that is not allowed (if, for example, you try to remove a File Store that has not been decommissioned), a message displays to let you know this.

3. Click **Pending Changes** at the top right, and **Apply Changes and Restart** to commit the changes and restart Tableau Server.

Use the TSM CLI

- Adding processes to a node
- Changing the number of processes on a node
- Removing all instances of a process from a node
- Moving all instances of a process from one node to another node

To configure nodes, run commands from the initial node and use the node ID to specify which node you are configuring. To determine the node ID, use the `tsm topology list-nodes` command. Use the `tsm topology set-process` command to add, update or remove a process on a node. You need to specify the node you are configuring,

the process you are adding, updating, or removing, and the number of instances of the process. After setting the topology for a node you need to apply the changes to Tableau Server.

Apply changes using the `tsm pending-changes apply` command. After the changes are applied, Tableau Server is returned to the state it was in before the command was run. This means that if it was running, it will be restarted, and if it was stopped it will remain stopped after pending changes have been applied. In most cases, if Tableau Server is running when you apply pending changes, the server is stopped so that changes can be applied, and then restarted. The exception is if you are changing the number of instances of Backgrounder, or VizQL Server on an existing node. With changes to those processes on an existing node, Tableau Server does not have to be stopped if it is running.

You need the node ID for a node in order to configure the node. To determine the node ID, use this command:

```
tsm topology list-nodes -v
```

Note: Examples here show some process names. For a complete list, see [Tableau Server Processes](#).

Adding processes to a node

Use the `tsm topology set-process` command to add a process to a node. You need to specify the node you are configuring, the process you are adding, and the number of instances of the process.

1. On the initial node, open a terminal session.
2. Find the node ID for the node you are changing:

```
tsm topology list-nodes -v
```

3. Add processes on the node by specifying the process and the number of instances.

Tableau Server on Linux Administrator Guide

For example, this command adds two instances of backgrounder to node1:

```
tsm topology set-process -n node1 -pr backgrounder -c 2
```

4. Apply the changes:

```
tsm pending-changes apply
```

Changing the number of processes on a node

Change the number of processes on a node by specifying an already configured process and providing a new value for the number of instances.

1. On the initial node, open a terminal session.
2. Find the node ID for the node you are changing:

```
tsm topology list-nodes -v
```

3. Change the number of processes on the node by specifying an already configured process and providing a new value for the number of instances.

For example, on a node (node1) that is already running backgrounder, this command changes the number of instances to four:

```
tsm topology set-process -n node1 -pr backgrounder -c 4
```

4. Apply the changes:

```
tsm pending-changes apply
```

Removing all instances of a process from a node

1. On the initial node, open a terminal session.
2. Find the node ID for the node you are changing:

```
tsm topology list-nodes -v
```

3. Remove a process from a node by specifying a count of 0 instances for that process

on the node.

For example, this command removes the backgrounder process from node1:

```
tsm topology set-process -n node1 -pr backgrounder -c 0
```

4. Apply the changes:

```
tsm pending-changes apply
```

Moving all instances of a process from one node to another node

In most cases you move a process from one node to another by setting the process instance count on the first node to zero, and setting the count to a non-zero value on the second node.

1. On the initial node, open a terminal session.
2. Find the node ID for the node you are changing:

```
tsm topology list-nodes -v
```

3. Move a process from one node to another node by specifying a count of 0 instances for that process on the first node and specifying a count of 1 or greater to the second node.

For example, these commands remove Backgrounder from node0 and add two instances of it to node1:

```
tsm topology set-process -n node0 -pr backgrounder -c 0
```

```
tsm topology set-process -n node1 -pr backgrounder -c 2
```

4. Apply the changes:

```
tsm pending-changes apply
```

Workload Management through Node Roles

Using node roles, you can configure where certain types of workloads are processed on your Tableau Server installation. The node roles features allows you to dedicate and scale resources to specific workloads. You can configure node roles for Backgrounder and File Store.

The Backgrounder node role specifies the type of background tasks that should run on a node, whereas the File Store node role specifies the type extract workload that should run on a node. Both node roles are specified at a node level. Although these node roles can work independently to optimize selected workload, the two node roles in combination can be used to specialize server nodes to preferentially execute selected workloads to optimize performance extract heavy workloads. This combination is discussed in more detail later in the File Store Node Roles section.

Backgrounder Node Roles

The Backgrounder process runs Tableau Server tasks, including extract refreshes, subscriptions, flow tasks, 'Run Now' tasks, and tasks initiated from *tabcmd*. Running all these tasks can use a lot of machine resources. If you have more than one Backgrounder node in your cluster, you can manage your Backgrounder workload by specifying the type of tasks a Backgrounder can run on a node using the Backgrounder node role feature.

This configuration option is currently available only through TSM CLI commands and is only useful on multi-node clusters. If you have only one node, the Backgrounder is set to run all tasks by default and that cannot be changed.

Using Backgrounder Node Roles

The Backgrounder node role feature is intended to give you more control and governance over where certain type of Backgrounder workloads are processed in your Tableau Server installation and allows you to dedicate and scale resources to specific workloads.

For example, if your deployment is heavy on extract and users are running a lot of extract refreshes or encryption jobs, it could be beneficial to dedicate a node to extract refreshes.

Similarly, in the case of subscriptions, if your Tableau Server installation processes a lot of subscriptions and you want to ensure that other jobs do not take resources from subscriptions, then you can dedicate a node to subscriptions. In these cases, you would also want to dedicate other background nodes to workloads other than extract refreshes or subscriptions.

To support high availability, Tableau recommends having multiple nodes that are dedicated towards a specific workload. For example, if you dedicate a node to extract refreshes, you should also configure a second node to process extract refresh workload. This way if a node dedicated to extract refreshes becomes unavailable, extract refreshes can still be processed by the other node.

Configuration Options

Configuration	Jobs
all-jobs (default)	All Tableau Server jobs
flows	Flow run jobs.
no-flows	All jobs except flows.
extract-refreshes	Jobs that are created for: Incremental refreshes, full refreshes, encryption and decryption of all extracts including extracts that flow outputs create.
subscriptions	Subscription jobs
extract-refreshes-and-subscriptions	Extract-refreshes, encryption and decryption of all extracts including extracts that flow outputs generate, and subscription jobs.
no-extract-refreshes	All jobs except extract-refreshes, extract encryption and decryption of all extracts including extracts created from flow outputs.
no-subscriptions	All jobs except subscriptions.
no-extract-	All jobs except extract-refreshes, encryption and decryption of all

refreshes-and-subscriptions	extracts including extracts created from flow outputs, and subscriptions.
-----------------------------	---

For more information on how to use the tsm commands to set the node role, see tsm topology.

Note: Making configurations to node roles require a restart of the server and will require some downtime. For more information, see tsm pending-changes.

License Requirements

Configuring a node to do only a specific type of tasks, like, flows, extract refreshes, and subscriptions, you must have one of the following licenses activated on your Tableau Server:

- To configure a node to run flows, you must have a valid Data Management Add-on license activated on your server, and have Tableau Prep Conductor running on that node. To learn more about Tableau Prep Conductor, see [Tableau Prep Conductor](#).
- To configure a node to run all jobs except flows, you must have Tableau Prep Conductor running on the node. Tableau Prep Conductor requires a valid Tableau Data Management Add-on license activated on your server. For more information, see [Licensing Tableau Prep Conductor](#).
- To configure a node to run extract refreshes, subscriptions, and any combination related to extract refreshes and subscriptions you must have a valid Advanced Management license activated on your Tableau Server. If the license expires or is deactivated, you will see an error any time you make a change to the Server configuration. For more information on Advanced Management. see [About Tableau Advanced Management](#).

Important!

While flows, extract refreshes, and subscriptions can be expensive and resource heavy, they are not the only jobs that may require dedicated resources. In the **all jobs** group, there are a

variety of System jobs that the Backgrounder executes, such as thumbnail generation for workbooks. Make sure that the nodes that run jobs other than extract refreshes, subscriptions, or flows have enough machine resources.

For more information on configuring node roles using TSM commands, see `tsm topology set-node-role`.

Considerations

There are some rules you must consider when configuring Backgrounder node roles, which are listed below:

- Only one node role configuration can be set for a node at a time. You cannot configure multiple node roles on a node.
- To configure a node role, there must be at least one Backgrounder process on that node.
- If you have only one Backgrounder node, you must configure this node to run all jobs. This is the default configuration and does not require additional licensing.
- If you have more than one Backgrounder node, combined, they must be configured to handle all jobs. This can be achieved in the following ways:
 - Configure one of the nodes to run all jobs using the all jobs option. This is the easiest and most straightforward way.
 - Using one of the exception configurations on one of the nodes:
 - no-flows
 - no-subscriptions
 - no-extract-refreshes
 - no-extract-refreshes-and-subscriptions

For example, in a cluster where there are three backgrounders, you could have one node configured to run flows, one to run subscriptions and extract refreshes, and one to run all jobs except flows, subscription and extract refreshes.

Note: The ability to specify node roles to run flows, or run all jobs except flows, or run all jobs was introduced in 2019.1.

File Store Node Roles

The Tableau Server File Store controls the storage of extracts. There are three broad categories of workloads that are extract dependent.

Extract Workload	Execution Service
Refresh	Backgrounder
Query	Data Engine
Backup/Restore	Backup/Restore

File Store node role management in combination with Backgrounder node role management gives server admins the ability to specialize server nodes to preferentially execute selected workloads to optimize performance of all categories of extract heavy workloads.

It is possible to specialize a node to execute extract query workloads through a topology that has only stand-alone Data Engine nodes. For more information, see [Optimize for Extract Query Heavy Environments](#). However, this is at the expense of extract refresh workloads, which are executed by Backgrounder nodes. With the topology-based isolation approach, extract refresh heavy Backgrounder workloads can get slower as none of the Backgrounder nodes have a File Store and thus all extract refresh traffic goes over the network.

With the File Store Node Role configuration option, it is possible to designate certain server nodes that process extract queries to be preferentially selected from the list of server nodes that can do so. This helps speed up workloads such as backup and extract refreshes by allowing server admins to enable File Store on Backgrounder server nodes, which prevents extract queries from running on these nodes. This feature is useful if you have an extract-

heavy query workload and an extract-heavy refresh workload and want to achieve optimal extract query and refresh performance.

Guidelines to Optimize for Extract Refresh and Backup/Restore Workloads.

Start from a topology with specialized Data Engine nodes (see Optimize for Extract Query Heavy Environments).

Process	Initial Node	Additional Node 1	Additional Node 2	Additional Node 3
Cluster Controller	✓	✓	✓	✓
Gateway	✓	✓		
Application Server	✓	✓		
VizQL Server	✓ ✓	✓ ✓		
Cache Server	✓ ✓	✓ ✓		
Search & Browse	✓	✓		
Backgrounder	✓ ✓ ✓ ✓	✓ ✓ ✓ ✓		
Data Server	✓ ✓	✓ ✓		
Data Engine	✓	✓	✓	✓
File Store			✓	✓
Repository	✓	✗		

Topology 1 - Dedicated Data Engine Nodes

1. Add File Store to Node 0.

```
tsm topology set-process -n node0 -pr filestore -c 1
```

2. Designate Node 2 and Node 3 to preferentially execute extract-query workloads

```
tsm topology set-node-role -n node2, node3 -r extract-queries
```

3. Designate Node 0 to preferentially execute extract-refresh workloads.

```
tsm topology set-node-role -n node0 -r extract-refreshes
```

4. Designate Node 1 to preferentially execute non-extract-refresh workloads.

```
tsm topology set-node-role -n node1 -r no-extract-refreshes
```

5. Apply pending changes.

```
tsm pending-changes apply
```

Tableau Server on Linux Administrator Guide

Process	Initial Node	Additional Node 1	Additional Node 2	Additional Node 3
Cluster Controller	✓	✓	✓	✓
Gateway	✓	✓		
Application Server	✓	✓		
VizQL Server	✓ ✓	✓ ✓		
Cache Server	✓ ✓	✓ ✓		
Search & Browse	✓	✓		
Backgrounder	✓ ✓ ✓ ✓	✓ ✓ ✓ ✓		
Data Server	✓ ✓	✓ ✓		
Data Engine	✓	✓	✓	✓
File Store	✓		✓	✓
Repository	✓	✗		

Topology 2 - Extra File Store Node

Note: In your Tableau Server deployment, adding File Store roles to existing nodes will temporarily increase network I/O between all File Store nodes while the new File Store is being synchronized. The duration of this operation is dependent on the volume of data on the File Store and the network bandwidth capacity. The status of synchronization can be monitored using the TSM Web GUI. If you are adding more than one File Store to your deployment, it is recommended to add them consecutively and wait for the initial synchronization to complete in between each File Store addition.

Fine Tune Extract Query Workload Management

When extract queries for email subscriptions and metric alerts are running at the same time that users are interactively viewing extract-based visualizations, users may experience slower than normal viz load times. Use the following node roles to fine tune how these workloads are prioritized.

Node role to use	Type of extract query workload	Example
<code>extract-queries</code>	scheduled	email subscriptions and metric alerts

<code>extract-queries-interactive</code>	<code>interactive</code>	users viewing an extract-based visualization
--	--------------------------	--

If your server deployment is seeing growth in email subscriptions and metric alerts, you can add nodes and assign the `extract-queries` node role, which makes them more available to handle subscriptions and alerts.

If your server deployment is seeing growth in users viewing extract-based visualizations, you can add nodes and assign the `extract-queries-interactive` node role, which makes them prioritize interactive extract queries to reduce extract-based viz load times. The `extract-queries-interactive` node role is a preference and not strict isolation. This means that queries will be routed to nodes that have the `extract-queries-interactive` node role assigned. If you have multiple nodes with the `extract-queries-interactive` role, queries will be routed based on node health.

For example, add a node and designate it to preferentially execute `extract-queries-interactive` workloads.

- `tsm topology set-node-role -n node4 -r extract-queries-interactive`

Configuration Options

Configuration	Jobs
<code>all-jobs</code> (default)	All Tableau Server jobs
<code>extract-queries</code>	Jobs that are created for extract queries. The nodes selected will run as <code>all-jobs</code> and will prioritize the processing of extract queries.
<code>extract-queries-interactive</code>	Jobs that are created for extract queries. The nodes selected will run as <code>all-jobs</code> and will prioritize the processing of interactive extract queries, such as those that run when a user is looking at their screen and waiting for an extract-based dashboard to load. This is an advanced setting and it should only be used if the cluster has a heavy sub-

	scription and alert job workload that causes users to experience degraded performance on viz load times that run around the same time as scheduled loads.
--	---

For more information on configuring node roles using TSM commands, see `tsm topology set-node-role`.

License Requirements

To configure a node to run extract queries you must have a valid Advanced Management license activated on your Tableau Server.

Who can do this

Tableau Server Administrators can configure node roles and activate any required product keys.

Install Tableau Server on a Two-Node Cluster

When you install Tableau Server on a two-node cluster, you can install server processes on one or both nodes. A two-node cluster can improve the performance of Tableau Server, because the work is spread across multiple machines.

Note the following about two-node clusters:

- A two-node cluster does not provide failover or support for high availability.
- You can't install more than one instance of the repository on a two-node cluster, and the repository must be on the initial node.

If you need failover or high availability, or want a second instance of the repository, you must install Tableau Server on a cluster of at least three computers. In a cluster that includes at least three nodes, you can configure two instances of the repository, which gives your cluster failover capability.

Restart Multi-Node Tableau Server Computers

Restarting the computers running a multi-node installation of Tableau Server requires that you follow a few specific steps.

To start computers running a multi-node installation of Tableau Server:

1. Stop Tableau Server. To do this:
 - a. On the initial node, open a terminal session.
 - b. Run this tsm command:

```
tsm stop
```
 - c. Wait until Tableau Server is stopped.
2. Restart the additional node computers. These are all the other computers *except* for the one running TSM.
3. Wait until each of the additional node computers has completely restarted, including the Tableau Server software installed there.
4. Restart the initial node computer (the computer running TSM). When it has completely restarted and Tableau Server is running, connections to each additional node should be restored.

Maintain a Distributed Environment

After you set up an initial node and one or more additional nodes for a distributed installation, you can perform all subsequent configuration and updates from the initial node, using the TSM CLI, or from any computer using a browser and the TSM Web interface.

When you install additional nodes, they are added by computer name. If the computer name of a node changes, you will need to remove and reinstall the node. For details on removing a node, see [Remove a Node](#).

You can monitor the status of the Tableau Server cluster on the TSM Status page. See [View Server Process Status](#) for details.

Additional actions you may need to maintain your distributed environment include:

Move the Repository Process

Tableau Server relies on the PostgreSQL repository to store server data. There must always be at least one active instance of the repository in any Tableau Server installation, and you can have a maximum of two instances (one active, one passive) if you have at least three nodes in your installation. You cannot remove a repository instance if it is the only instance.

This means that if you want to move the only instance of your repository from one node to another node, you need to add a second instance and synchronize the new repository with the old one before you remove the old one. Synchronize repository instances by starting server. If you've added a new repository, it will automatically synchronize with the existing instance.

If you are deleting a node from your server cluster and that node is hosting the only instance of the repository, you must add a second instance of the repository and synchronize the instances before removing the node.

If you are also moving the file store, you can move the repository at the same time. See [Move the File Store Process](#).

Before making a change to the repository, create a full backup of Tableau Server. For more information, see [tsm maintenance backup](#).

Important: You cannot add a second repository instance and remove the first one in the same step. You must have both running so the contents of the first is synchronized with the second, before you remove the original instance.

The steps for moving the repository are:

1. Add a new instance of the repository to another node, start server, and wait for it to synchronize with the first repository.
2. Remove the instance of the repository from the original node.

Use the TSM web interface

To move a repository you need to first add a second instance on a second node, and then after the two instances have synchronized all the data in the original repository, remove the original instance. These steps must be done separately to allow for the synchronization of content between the two instances.

Add a new instance of the repository.

1. Open TSM in a browser:

```
https://<tsm-computer-name>:8850
```

For more information, see [Sign in to Tableau Services Manager Web UI](#).

2. Click the **Configuration** tab.
3. For the node you are adding the repository to:
 - Select **Repository** (pgsql).
4. Click **Pending Changes** at the top of the page:



The Pending Changes list displays.

If you are configuring a three- or five-node cluster and have not deployed a Coordination Service ensemble, a warning will display. You can continue, and deploy a Coordination Service ensemble in a separate step. For details on deploying a Coordination Service ensemble, see [Deploy a Coordination Service Ensemble](#).

5. Click **Apply Changes and Restart** and **Confirm** to confirm a restart of Tableau Server.
6. After Tableau Server has restarted, on the **Status** tab, verify that all processes are active.

Remove an instance of the repository.

1. In TSM, on the **Status** tab, verify that all processes are active. When both repositories show as Active, you can remove the first one.
2. Click the **Configuration** tab.
3. For the node you're removing the repository from, clear the **Repository** box.
4. Click **Pending Changes** at the top of the page.

If you are configuring a three- or five-node cluster and have not deployed a Coordination Service ensemble, a warning will display. You can continue, and deploy a Coordination Service ensemble in a separate step. For details on deploying a Coordination Service ensemble, see [Deploy a Coordination Service Ensemble](#).

5. Click **Apply Changes and Restart** and **Confirm** to confirm a restart of Tableau Server.

Use the TSM CLI

To move a repository you need to first add a second instance on a second node, and then after the two instances have synchronized all the data in the original repository, remove the

original instance. These steps must be done separately to allow for the synchronization of content between the two instances.

Add a new instance of the repository.

1. Add the repository (pgsql) to another node:

```
tsm topology set-process -n <nodeID> -pr pgsql -c 1
```

2. Apply the changes. If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

```
tsm pending-changes apply
```

3. Wait for the new repository on the second node to synchronize with the repository on the first node.

```
tsm status -v
```

Wait until the new repository status shows as "passive".

Remove an instance of the repository.

Once the new instance of the repository is fully synchronized and shows as "passive" you can remove the original instance:

1. Remove the repository from the first node by setting the process count to 0 (zero):

```
tsm topology set-process -n <nodeID> -pr pgsql -c 0
```

2. Apply the change. If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does

not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

```
tsm pending-changes apply
```

Move the File Store Process

This topic describes File Store process when configured to run locally on Tableau Server. However, File Store can be run locally as well as external to Tableau Server. For more information on Tableau Server External File Store, see [Tableau Server External File Store](#).

You cannot remove an instance of File Store if it is the only instance on the server. You cannot run Tableau Server without at least one instance of File Store. This means if you need to move the File Store, or if you are deleting a server node that is hosting the only instance of the File Store, you must first move File Store to another node.

Moving the File Store is a two-part process:

- Adding a second instance of File Store (if there is not an existing second instance).
- Decommissioning and removing the original instance of File Store.

This article assumes you have installed Tableau Server on an initial node and at least one additional node. For more information on adding nodes to Tableau Server, see [Install and Configure Additional Nodes](#).

Use the TSM web interface

Adding a second instance of File Store

This procedure assumes you have added an additional node. For more information on adding nodes to Tableau Server, see [Install and Configure Additional Nodes](#).

1. Open TSM in a browser:

```
https://<tsm-computer-name>:8850
```

For more information, see [Sign in to Tableau Services Manager Web UI](#).

2. Click the **Configuration** tab.
3. For the node you want to add an instance of File Store on, select **File Store**.

Adding File Store to a node will also add an instance of Data Engine if one is not already on the node.

4. Click **Pending Changes** at the top of the page:



The Pending Changes list displays.

A Coordination Service ensemble warning displays because you are configuring a three-node cluster. You can continue. You will deploy a Coordination Service ensemble in an upcoming step.

5. Click **Apply Changes and Restart** and **Confirm** to confirm a restart of Tableau Server.

Decommissioning and removing an instance of file store

1. In TSM, on the **Status** tab, verify that all processes are active.
2. Click the **Configuration** tab.
3. For the node you're removing File Store from, clear the **File Store** box.

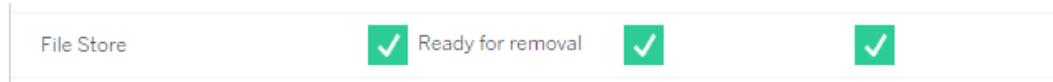
This will generate a warning about decommissioning the File Store before removing it. You cannot remove an instance of File Store unless it has been decommissioned. Click **OK** to decommission and remove the file store. Data Engine will also be removed unless an instance of one of these processes is installed on the node: VizQL Server, Application Server (Vizportal), Data Server, or Backgrounder.

4. If you are removing File Store from a node that also has the TSM Administrative Controller (usually the initial node), a warning will caution you about impact to server

backup performance.

Click **Continue** to decommission File Store.

5. Click the **Status** tab to see the status of the decommission. When the instance of File Store is marked "Ready for removal" you can continue.



6. Click **Pending Changes** at the top of the page.

Note: If you are configuring a three- or five-node cluster, a Coordination Service ensemble warning will display. You can continue to apply pending changes and deploy a Coordination Service ensemble in a separate step. For information on deploying a Coordination Service ensemble, see [Deploy a Coordination Service Ensemble](#).

7. Click **Apply Changes and Restart** and **Confirm** to confirm a restart of Tableau Server.

Use the TSM CLI

Adding a second instance of file store

1. Create a full backup of Tableau Server. For more information, see [Back up Tableau Server data](#).
2. Add the File Store to a second node.

```
tsm topology set-process -n <nodeID> -pr filestore -c 1
```

The File Store is automatically added. Data engine is also added if it is not already on the node.

Apply the configuration changes:

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

3. Check the status of the new file store instance.

```
tsm status -v
```

Wait for the new File Store to synchronize with the File Store on the first node. When synchronization is complete the new File Store has a status of "running" instead of "synchronizing".

Decommissioning and removing an instance of File Store

Once you have a second instance of File Store installed and synchronized you can decommission and remove the original instance. You must decommission the original instance before you remove it. Doing this guarantees that any unique files on the File Store node are duplicated to another file store node.

1. Decommission the original File Store:

```
tsm topology filestore decommission -n <nodeID> --override
```

2. When the decommission command completes, remove the File Store from the node by applying the pending configuration changes. The File Store is automatically removed. Data Engine is also removed unless an instance of one of these processes is installed on the node: VizQL Server, Application Server (Vizportal), Data Server, or Backgrounder.

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt

displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

Move the Messaging Service Process

This article assumes you have installed Tableau Server on an initial node and at least one additional node. For more information on adding nodes to Tableau Server, see [Install and Configure Additional Nodes](#).

You cannot remove an instance of the Messaging Service if it is the only instance on the server. You cannot run Tableau Server without one instance of the Messaging Service. This means if you need to move the Messaging Service, or if you are deleting a server node that is hosting the Messaging Service, you must first move the service to another node.

Moving the Messaging Service is straightforward process of three steps:

- Remove the original instance of the Messaging Service.
- Add a new instance of the Messaging Service.
- Apply the pending changes.

Use the TSM web interface

Moving the Messaging Service

1. In TSM, on the **Status** tab, verify that all processes are active.
2. Click the **Configuration** tab.
3. For the node you're removing the Messaging Service from, clear the **Messaging Service** box.

This will activate the **Pending Changes** button, but in Pending Changes an error will tell you that the Messaging Service (activemqserver) is not on any node. Until you add it to another node, you cannot apply pending changes.

4. For the node you are adding the Messaging Service to, select the **Messaging Service** box.
5. Click **Pending Changes** at the top of the page.
6. Click **Apply Changes and Restart** and **Confirm** to confirm a restart of Tableau Server.

Use the TSM CLI

Moving the Messaging Service

1. On the initial node, open a terminal session.
2. Find the node ID for the nodes you are changing:

```
tsm topology list-nodes -v
```

3. Remove the Messaging Service from one node:

```
tsm topology set-process -n <nodeID> -pr activemqserver -c
0
```

4. Add the Messaging Service to another node:

```
tsm topology set-process -n <nodeID> -pr activemqserver -c
1
```

5. Apply the configuration changes:

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt

displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

Remove a Node

If your Tableau Server installation includes a node you no longer need, you can remove it to simplify your installation, and to free up the hardware resources on that node.

Prerequisites for removing a node

There are prerequisites you must satisfy before you can remove a node from your Tableau Server cluster. If your node has one of a number of limitations, you need to address these before you can remove the node. The limitations include confirm that the node has been added with at least one process, and that the node does not include any process that is not also installed on another node.

If one of the following configuration limitations applies, you must take action before you can remove the node:

- If the node was just added using the Web UI, you need to apply pending changes before you can remove it. If you added it using the command line, you need to configure it with at least one process before you can remove it.
- If the node includes the only Repository instance, you need to move the Repository to another node. See [Move the Repository Process](#).
- If the node is running the only instance of the File Store, you need to move the File Store to another node. See [Move the File Store Process](#).
- If the node is running an instance of the Coordination Service, you must deploy a new Coordination Service ensemble that does not include the node. See [Deploy a Coordination Service Ensemble](#).

- If the node is running the Messaging Service, you need to move the Messaging Service to another node. See [Move the Messaging Service Process](#).

Important: Do not use the `tableau-server-obliterate` script to remove a node. First remove the node using the TSM UI or the `tsm topology remove-nodes` command. This leaves the rest of the cluster in a good state. Later, if you want to re-add the node to the cluster, run the `tableau-server-obliterate` script on the node to completely remove Tableau. After removing Tableau from the node, restart the computer and then re-add the node using the normal steps for adding and configuring a node. For details on running the script, see [Remove Tableau Server from Your Computer](#).

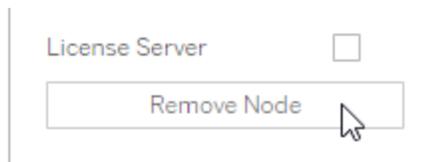
Use the TSM web interface

1. Open TSM in a browser:

`https://<tsm-computer-name>:8850`

For more information, see [Sign in to Tableau Services Manager Web UI](#).

2. Click the **Configuration** tab and, under the node you want to remove, click **Remove Node**:



If a configuration limitation does not allow you to remove the node (if, for example, it includes a File Store that must first be decommissioned), a message will display letting you know this. See **Prerequisites** above for more information.

Note: If you just added the node, and have not configured it, you must first apply pending changes before you can remove the node.

3. Click **Pending Changes** at the top right, and **Apply Changes and Restart**.

Use the TSM CLI

Use the `tsm topology remove-nodes` command to remove a node from a cluster.

To remove a node from a cluster it must have been configured with a process at some point in the past. If you added a node using the CLI and want to remove it but have not configured any processes, you must add a process on it, run the `tsm pending-changes apply` command, and then remove the node. For example, you might add one instance of Cluster Controller to the node: `tsm topology set-process -n <nodeID> -pr cluster-controller -c 1`.

If you are removing a node you added using the Web UI, the Cluster Controller process is automatically added so you do not need to add it before removing the node.

1. On the initial node, open a terminal session.
2. Find the node ID for the node you are changing:

```
tsm topology list-nodes -v
```

3. Remove a node using the `remove-nodes` command.

For example, to remove `node2` from an existing cluster:

```
tsm topology remove-nodes --node-names "node2"
```

If a configuration limitation does not allow you to remove the node (if, for example, it includes a File Store that must first be decommissioned), a message will display letting you know this. See **Prerequisites** above for more information.

4. Apply the changes:

```
tsm pending-changes apply
```

Configure Tableau Server for High Availability with Coordination Service-Only Nodes

The Coordination Service is built on [Apache ZooKeeper](#), an open-source project, and coordinates activities on the server, guaranteeing a quorum in the event of a failure, and serving as the source of "truth" regarding the server topology, configuration, and state. The service is installed automatically on the initial Tableau Server node, but no additional instances are installed as you add additional nodes. Because the successful functioning of Tableau Server depends on a properly functioning Coordination Service, we recommend that for server installations of three or more nodes, you add additional instances of the Coordination Service by deploying a new Coordination Service ensemble. This provides redundancy and improved availability in the event that one instance of the Coordination Service has problems.

The Coordination Service can generate a large amount of I/O as it communicates with other components of the server, so if you are running Tableau Server on computers that meet or just exceed the minimum hardware requirements, you may want to install Tableau Server in a configuration that uses Coordination Service-only nodes. This means installing Coordination Service on nodes that run no other server processes, and removing Coordination Service from any nodes that are running other server processes. This procedure explains how to do this. You can also run the Coordination Service ensemble on the same nodes running other Tableau Server processes. For details on how to do that, see [Deploy a Coordination Service Ensemble](#) .

Important: The process to deploy a Coordination Service ensemble changed as of version 2020.1.0. If you are running an earlier version of Tableau Server, see the documentation for that version. You can find documentation for all supported versions here: [Tableau Help](#)

Prerequisite

Before proceeding with the procedures in this topic, complete the following prerequisites:

- Install and Configure Tableau Server - Install Tableau on your initial node.
- Install and Configure Additional Nodes - Install Tableau on at least two additional nodes.

Note: This operation includes steps that you may need to perform using the TSM command line.

Deploy an ensemble on Coordination Service-only nodes

One way to accommodate the high I/O impact of the Coordination Service is to deploy an ensemble on nodes that only run the Coordination Service and the Cluster Controller. The following steps illustrate how to deploy a Coordination Service ensemble on an existing multi-node Tableau Server cluster.

Note: For a core-based Tableau Server license, Coordination Service-only nodes do not count against the total count of licensed cores.

1. Add additional nodes to your cluster.

See [Install and Configure Additional Nodes](#).

2. If you added the new nodes using the TSM CLI, you need to configure the nodes with Cluster Controller (this step is not necessary if you added the nodes using the TSM Web UI because Cluster Controller is automatically added when you add a node with the Web UI).

On the initial node, open a terminal session.

3. From the initial node of the cluster, configure the new nodes with an instance of the

Cluster Controller:

```
tsm topology set-process -pr clustercontroller -n <node4>
-c 1
```

```
tsm topology set-process -pr clustercontroller -n <node5>
-c 1
```

```
tsm topology set-process -pr clustercontroller -n <node6>
-c 1
```

If prompted, sign in as a TSM administrator.

4. Apply the configuration changes. If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

```
tsm pending-changes apply
```

A warning about deploying a Coordination Service ensemble displays because you have deployed a multi-node cluster. If this is the only warning, you can safely override it using the `--ignore-warnings` option to apply the configuration changes in spite of the warning.

```
tsm pending-changes apply --ignore-warnings
```

5. Confirm that all nodes are up and running:

```
tsm status -v
```

6. On the initial node of the cluster, open a terminal session and type this command to

stop Tableau Server:

```
tsm stop
```

7. Get the node IDs for each node in the cluster:

```
tsm topology list-nodes -v
```

8. Use the `tsm topology deploy-coordination-service` command to add a new Coordination Service ensemble by adding the Coordination Service to specified nodes. You must specify the node(s) that the Coordination Service should be added to. The command also makes the new ensemble the "production" ensemble (the ensemble in use) and removes the old ensemble.

Note: A "y/n" prompt displays confirming that a server restart will take place. To run the command without input, include the `--ignore-prompt` option.

For example, deploy the Coordination Service to three nodes of a six-node cluster:

```
tsm topology deploy-coordination-service -n  
<node4,node5,node6>
```

Wait until the command completes and you are returned to the system prompt.

9. Start Tableau Server:

```
tsm start
```

Add a Load Balancer

You can enhance the reliability of Tableau Server by running gateways on multiple nodes, and configuring a load balancer to distribute requests across the gateways. Unlike the repository process, which can be active or passive, all gateway processes are active. If one gateway in a cluster becomes unavailable, the load balancer stops sending requests to it. The load balancer algorithm you choose determines how the gateways will route client requests.

Notes: If you will be using Kerberos authentication, you need to configure Tableau Server for your load balancer before you configure Tableau Server for Kerberos. For more information, see [Configure Kerberos](#).

Windows SSPI authentication is not supported with a load balancer.

Note the following as you configure your load balancer to work with Tableau Server:

- **Tested load balancers:** Tableau Server clusters with multiple gateways have been tested with Apache and F5 load balancers.

If you are using an Apache load balancer and creating custom administrative views, you need to connect directly to the Tableau Server repository. You cannot connect through the load balancer.

- **Tableau Server URL:** When a load balancer is in front of a Tableau Server cluster, the URL that's accessed by Tableau Server users belongs to the load balancer, not the initial Tableau Server node.
- **Single load balancer endpoint:** You must configure your load balancers for a single URL endpoint. You cannot configure different endpoint hosts to redirect to the same Tableau Server deployment. The single external URL is defined in `gateway.public.host` when you configure Tableau Server, as described in the procedure below.
- **Trusted host settings:** The computer running the load balancer must be identified to Tableau Server as a trusted host. See the procedure below for how to configure Tableau Server.
- **Proxy server configurations:** The settings used to identify a load balancer to Tableau Server are the same ones that are used to identify a proxy server. If your Tableau Server cluster requires both a proxy server and a load balancer, both must use a single external URL defined in `gateway.public.host` and all proxy

Tableau Server on Linux Administrator Guide

servers and load balancers must be specified in `gateway.trusted` and `gateway.trusted_hosts`. For more information, see [Configuring Proxies for Tableau Server](#).

Configure Tableau Server to work with a load balancer

Use the following key-value pairs to configure Tableau Server for a load balancer. You can set the values individually with the `tsm configuration set` command.

Or, use the configuration file template example at the end of this topic to create a `configKey.json` file. After you have filled in the options with the appropriate values, pass the `json` file and apply settings with `tsm` commands.

Load balancer configuration key-value pairs

The key-value pairs described in this section can be set with the `tsm configuration set` command. After you have set all values, run the following command:

```
tsm pending-changes apply
```

```
gateway.public.host
```

Value: `name`

The value, `name` is the URL that will be used to reach Tableau Server through the load balancer.

For example, if Tableau Server is reached by entering `tableau.example.com` in a browser address bar, enter this command:

```
tsm configuration set -k gateway.public.host -v "tableau-  
.example.com"
```

```
gateway.public.port
```

Value: `port_number`

By default, Tableau assumes that the load balancer is listening on port 80 for external communications. Use this key to specify a different port.

For example, if your load balancer is configured for SSL and listening on port 443, enter the following command:

```
tsm configuration set -k gateway.public.port -v "443"
```

`gateway.trusted`

Value: `server`

Specify the load balancer(s) IPv4 address or computer name.

The value for `server` can be a comma-separated list, for example:

```
tsm configuration set -k gateway.trusted -v "10.32.139.45,
10.32.139.46, 10.32.139.47"
```

or

```
tsm configuration set -k gateway.trusted -v "proxy1, proxy2,
proxy3"
```

`gateway.trusted_hosts`

Value:

Specify alternate names for the load balancer(s), such as its fully-qualified domain name, any non-fully-qualified domain names, and any aliases. These are the names a user might type in a browser. Separate each name with a comma:

```
tsm configuration set -k gateway.trusted_hosts -v "name1, name2,
name3"
```

For example:

```
tsm configuration set -k gateway.trusted_hosts -v "lb.example.com,
lb, ftp.example.com, www.example.com"
```

Tableau Server on Linux Administrator Guide

Configuration template example

You can configure all of the settings in one operation by customizing the following configuration template example and creating a configKey json file.

Refer to the key-value pair descriptions above to customize the values for your organization.

```
{
  "configKeys": {
    "gateway.public.host": "tableau.example.com",
    "gateway.public.port": "443",
    "gateway.trusted": "proxy1, proxy2, proxy3",
    "gateway.trusted_hosts": "lb.example.com, lb, ftp.example.com,
www.example.com"
  }
}
```

When you are finished, save the json file.

Pass the json file with tsm to set the configuration, and then apply settings by running the following commands:

```
tsm settings import -f /path/to/file.json
```

```
tsm pending-changes apply
```

For more information about using configKey schemes, see Configuration File Example.

Migrate

The topics in this section provide information on migrating between hardware, operating system, and cloud platforms for Tableau Server on Linux.

Looking for Tableau Server on Windows? See [Migrate](#)

To migrate a site from one Tableau Server to another, see Export or Import a Site.

To copy or migrate content between Tableau Server projects using the Tableau Content Migration Tool, see [About Tableau Content Migration Tool](#).

Changing the Identity Store

Infrastructure or business changes may require you to change the identity store on Tableau Server. There are two kinds of identity stores: local and external. When you installed Tableau Server you configured either a local identity store or an external identity store.

When you configure Tableau Server with a local identity store, all user and group information is stored and managed in the Tableau Server repository. In the local identity store scenario, there is no external source for users and groups.

When you configure Tableau Server with an external store, all user and group information is stored and managed by an external directory service. Tableau Server must synchronize with the external identity store so that local copies of the users and groups exist in the Tableau Server repository, but the external identity store is the authoritative source for all user and group data. Examples of external identity stores are OpenLDAP and Active Directory.

For more information about the Tableau identity store, see [Identity Store](#).

You can change from local store to an external store, or you can change from an external store to a local store. In either case, to change the identity store type, you complete these steps:

1. Uninstall (including manual deletion of directories) and then reinstall of Tableau Server. The procedure for full uninstall and clean install are at the end of this topic.
2. Restore content and permissions.

In these steps the term "restore" does not refer to using the `TSM maintenance restore` command to restore the backup you are making. You cannot restore a backup (`.tsbak`) created on a Tableau Server instance that uses a different identity

store than the target Tableau Server. The backup is a best practice safeguard, in case you need to go back to your original Tableau Server configuration.



Warning

Changing the installation type on Tableau Server can be a complicated and time-consuming process. To avoid data loss or orphaning of content or users, you'll need to plan this process carefully. In all cases, user filters applied to workbooks and data sources will need to be updated manually after the change.

Most importantly, determine how you will transition content and permissions to the new identity store after you reinstall Tableau Server.

Methods for restoring content and permissions

The following list describes two methods for restoring content and permissions after you reinstall Tableau Server. Select the method that best fits with your environmental requirements.

- **Method 1: Use site export and import**—In this method, you start by exporting each site in your existing deployment. Then, you install the new server and configure it for the new identity store type. You then create new users in the default site on the new server. Finally, you import all the original sites. During the import stage, you can map the original identities to the new users that you created in the default site.

You must use the same version of Tableau Server for the export and import operations. Importing a site to a version of Tableau Server that is different from the exported site version is not supported.

Because this method exports all content and permissions at each site, it is the best method for organizations that require a high fidelity replica of the content and permissions after the identity store change is complete. Some organizations require an identity store change as the result of an authentication change. In these cases, a different user name syntax is a often a requirement in the new model. This method,

which includes a process of mapping original user names to new names, provides flexibility for such scenarios.

- **Method 2: Fresh installation; users republish content**—In this method, you install a new version of Tableau Server and select the new identity store type during setup. You also create new sites. You then create users and give them access, and they republish their workbooks and data sources. Unlike the other method, in this one, you do not reuse any of your existing Tableau Server infrastructure.

This method is most appropriate for smaller deployments with fairly autonomous and data savvy users. From an administrative perspective, this method is the simplest, since you're not actively porting over content. However, because you rely entirely on users to republish content, this method may not be successful for large organizations or for those where centralized oversight of content is required.

User filters

User filters are domain-specific. Therefore, when the domain of Tableau Server changes or authentication type changes, filters no longer function as expected. Although the user filters are generated by Tableau Server, after they are set by the user, the filters are stored in the workbooks and data sources. Neither of these methods for changing the identity store modifies the contents of the workbooks or data sources.

As you plan the identity store change, you must also include a final task to correct user filtering in all workbooks and data sources with Tableau Desktop.

User names and the Tableau Identity store

If you are using Method 1, it's helpful to understand how Tableau Server stores user names in the Tableau identity store. Tableau stores all user identities in the repository, which coordinates content permissions and site membership with various services in Tableau Server. Generally, an identity store configured for Active Directory store user names in the format, `domain\username`. Some organizations use a UPN (`jsmith@domain.lan`).

On the other hand, organizations that configure Tableau Server with local identity store usually create standard, truncated user names, such as `jsmith`.

In all cases, these user names are literal strings that must be unique in the Tableau identity store. If you are changing from one identity store type to another, then your target authentication, SSO, or user provisioning solution may require a specific user name format.

Therefore, to maintain all permissions, content, and user viability, one of the following must be true after you change the identity store type:

- The new user names must match the original user names, or
- The original user names must be updated to match a new format.

If an authentication change is driving the identity store change, then the target authentication scheme will likely impose a user name syntax that is different than your original user names. Method 1 includes a process where you can map original user names to new user names.

It's possible that the original user name format will work with the new authentication type. For example, if you used UPN names in a local identity store deployment, you might be able to use the same user names in an Active Directory deployment. You could also use the `domain\username` format for local identity store, as long as users continue to use that format to sign in to Tableau Server.

If you are changing from local identity store to an external Active Directory store, review the topic, [User Management in Deployments with External Identity Stores](#), as part of your planning process.

Method 1: Use site export and import

You must use the same version of Tableau Server for the export and import operations.

1. Export all sites on your server. See [Export or Import a Site](#).
2. Back up, remove, and then reinstall .
3. Create new users on Tableau Server. You should have a new user that corresponds to each user on the original server.
4. Import the sites that you exported in Step 1. See [Export or Import a Site](#). During import, you will be prompted to map the new users to the original users.

Method 2: Fresh installation—users republish content

Even if you do not plan to port content as part of your identity store change, we recommend that you back up the server.

1. Back up, remove, and then reinstall .
2. Create users, sites, and groups.
3. Inform your users of the new Tableau Server, provide them with credentials, and allow them to republish their content.

Back up, remove, and then reinstall

Both methods include the following steps:

1. Back up Tableau Server
2. Remove Tableau Server.
3. Reinstall Tableau Server with the new identity store type.

Step 1: Back up Tableau Server

As a best practice, you should back up the server before proceeding.

Follow the procedure, Create a backup using the TSM command line interface (CLI). Run the `backup` command with the `-d` option. The `-d` option adds the datestamp.

When you are finished, copy the backup file (`.tsbak`) to a safe location that is not a part of your Tableau Server installation.

Step 2: Remove Tableau Server

You must completely remove Tableau Server from the computer. See [Remove Tableau Server from Your Computer](#).

Step 3: Reinstall Tableau Server with new authentication type

1. Go to the Tableau Customer Portal, sign in with your Tableau user name and password, and then download Tableau Server.

2. Install Tableau Server. See [Install and Configure Tableau Server](#) more information. During installation, you will select the new identity store type. See [Configure Initial Node Settings](#).

Migrate to New Hardware

Use the following procedure to migrate Tableau Server from one computer to another. You might do this if you are upgrading the computer hardware that Tableau is running on.

Specifically, these steps describe how to move Tableau Server data from your in-production computer to a new computer where Tableau Server is installed. Before you start, make sure you have followed the steps in [Preparing for Upgrade](#), including creating a backup and gathering any assets that require manual actions. For details, see [Perform a Full Backup and Restore of Tableau Server](#). You'll need these to restore your Tableau Server data and configuration.

You can only restore from a backup that has the same type of identity store as the running server. For example, a backup from a server using local authentication can be restored to a Tableau Server initialized with local authentication, but a backup from a server using Active Directory authentication cannot be restored to a server initialized with local authentication.

Note: This operation includes steps that you may need to perform using the TSM command line.

1. Deactivate your product key on your original installation of Tableau Server. You need to do this so you can activate it on the new computer. For more information, see ["Tableau Server product key activation"](#) below.
2. Install Tableau Server on the new computer, activate your license, configure initial settings, and create your admin user. For details, see [Install and Configure Tableau Server](#).
3. Copy your backup file `.tsbak` to the folder location specified by the

`basefilepath.backuprestore` variable. By default this is: `/var/opt/tableau/tableau_server/data/tabsvc/files/backups`.

Note: You can change the location for the backup file. For more information, see [tsm File Paths](#).

4. Next, stop Tableau Server:

```
tsm stop
```

5. Restore your in-production data to your new Tableau Server installation using the `tsm maintenance restore` command:

```
tsm maintenance restore -f <filename>
```

where `<filename>` is the name of the `.tsbak` file you copied in step 3.

For more information about restoring Tableau Server data, see [Restore from a backup](#).

6. Start the server:

```
tsm start
```

7. **Distributed installations only:** Install Tableau Server on the new computers you want to add to your new Tableau Server cluster. See [Install and Configure Additional Nodes](#) for steps.
8. If you have not deactivated your product key on the old computer, do that after you test your new Tableau Server installation and confirm that it's ready for production. For details, see ["Tableau Server product key activation"](#) below.

Note: If you do not have an internet connection, you are prompted to create an offline activation file to complete the deactivation process. See [Activate Tableau Server Offline - Adding a License for steps](#).

Tableau Server product key activation

You can activate the same Tableau Server product key up to three times. This allows you to test Tableau Server (in a sandbox or QA environments, for example), as well as use Tableau in production. To maximize your activations, you should deactivate your product key when you remove Tableau Server from a computer, unless you will be reinstalling Tableau on the same computer. Doing this gives you the opportunity to use the activation on a different computer. For example, if you move Tableau Server from one computer or VM to another, deactivate the product key, then remove Tableau from the original computer. When you install Tableau on the new computer, you can activate the key there without any conflict. If you use role-based licensing, be sure to activate a Creator or Explorer key or you may lose administrator access to Tableau. If you are removing Tableau Server to reinstall it on the same computer, you don't need to deactivate the key. Tableau will use the key when reinstalled. For example, if you are moving Tableau from one drive on a computer to a different drive on the same computer. For information on how to deactivate a product key, see [tsm licenses deactivate](#).

Migrate Tableau Server from Windows to Linux

Customers running Tableau Server on Windows can migrate to Linux by taking a backup of their existing Tableau installation and restoring it to a fresh installation on Linux. This topic describes the steps necessary to do this migration. You cannot migrate or upgrade from a beta version of Tableau Server to an officially released version.

The basic steps to migrate from Tableau Server on Windows to Tableau Server on Linux include:

1. Step 1: Plan your migration—Plan for your migration, including gathering all the information you'll need to be successful. During this step you should familiarize yourself with potential differences between Tableau Server on Windows and Tableau Server on Linux.
2. Step 2: Create a backup—Create a backup of Tableau Server on Windows.
3. Step 3: Install Tableau Server on Linux and restore the Windows backup—Install a fresh instance of Tableau Server on Linux in a test environment so you can test out the migration, then restore your Windows backup. The restore of your Windows backup will restore the Tableau content (users, projects, sites, workbooks and data sources), but will not restore customizations, so you will need to spend some time configuring Tableau Server on Linux to match the expectations in your organization. If you have a multi-node installation you will need to add nodes and configure them separately.
4. Step 4: Test Tableau Server on Linux—Try Tableau Server on Linux to make sure content is there as you expect, and users are able to perform all the actions they do on Windows. Look specifically at any changes identified as potential differences between Tableau Server on Windows and Tableau Server on Linux. Include key stakeholders in the testing both to leverage their knowledge and to help communicate the upcoming changes.
5. Step 5: Install Tableau Server on Linux in your production environment and restore the Windows backup—Once you're satisfied that Tableau Server on Linux gives you the functionality you need, install Tableau in your production Linux environment and restore the Windows backup.

Step 1: Plan your migration

A successful migration from Windows to Linux requires some preparation beforehand. You will need to satisfy the following requirements:

- **Identity store:** You can only restore from a backup that has the same type of identity store as the running server. For example, a backup from a server using local authentication can be restored to a Tableau Server initialized with local authentication, but a backup from a server using Active Directory authentication cannot be restored to a server initialized with local authentication. To change the identity store from a local identity store to an external identity store, see [Changing the Identity Store](#).
- **Server administrator:** You must have password for at least one of the user accounts with Server Administrator privileges from the Tableau Server source (Windows) deployment. You must use the same user as the Server Administrator in the migrated deployment to Linux.
- **Clean installation of Tableau Server for Linux:** When you install Tableau Server on Linux later in this topic, be sure to use the same identity store type as you are using on Windows, and do not create users or content. When you restore the Windows backup file to the Linux deployment of Tableau Server, all user data and content will be replaced from the Windows back up file.
- **Differences between Windows and Linux:** Review the differences between Tableau Server on Windows and on Linux so you are aware of them and can investigate in your test environment.
 - If you are migrating from a version of Server on Windows prior to 10.5, approach the migration like an upgrade and familiarize yourself with any changes between your existing version and 10.5 by reading [What's Changed](#) in the Server on Windows help.
 - Any custom fonts you use may need to be installed on your Linux computer, and may render differently there than on Windows.
 - Connection options for Linux are a subset of those available for Tableau Server on Windows. Review the connection types that are available and make sure the ones you need are supported.

- TSM replaces `tabadmin`. Understand the TSM commands and how they compare to older `tabadmin` commands. For a list of `tabadmin` commands and their TSM equivalents, see [Migrate from Tabadmin to the TSM CLI](#).

Step 2: Create a backup

The steps to create a backup depend on what version of Tableau Server on Windows you are running. If you have version 2018.1.x or earlier, you need to use `tabadmin` to create your backup. If you are running version 2018.2.0 or later, you use TSM to create the backup.

Migrate from 2018.1 or earlier

If you are migrating from Tableau Server on Windows version 2018.1.x or earlier:

1. Log on to the computer running Tableau Server on Windows.
2. Open a command prompt as an administrator.
3. Navigate to the bin directory. For example:

```
cd "C:\Program Files\Tableau\Tableau Server\10.4\bin"
```

Run the following command:

4. `tabadmin backup tabserver -v -d`

Include the `-v` flag to verify the integrity of the backup. Include the `-d` flag to include the date in the file name. In the example, `tabserver` will be used as the base file name of the resulting backup file. The date of the backup will be appended to the file name, for example, `tabserver-2017-12-20.tsbak`.

For more information, see [Back Up Tableau Server Data](#) in the Tableau Server on Windows help (version 2018.1).

Migrate from 2018.2 or later

If you are migrating from Tableau Server on Windows version 2018.2.0 or later:

Tableau Server on Linux Administrator Guide

1. Log on to the computer running Tableau Server on Windows.
2. Open a command prompt as an administrator.
3. Run the following command:

```
tsm maintenance backup -f <filename> -d
```

Include the `-d` flag to include the date in the backup file name.

For more information, see [Back up Tableau Server data](#) in the Tableau Server on Windows help.

Step 3: Install Tableau Server on Linux and restore the Windows backup

In a test environment, install Tableau Server on Linux:

- Install Tableau Server for Linux according to the procedure, [Install and Configure Tableau Server](#). Use the same identity store as on your Windows deployment, and do not create users or content.

Restore the Tableau Server on Windows backup:

1. Copy the Windows backup file to the computer running Tableau Server on Linux. By default the restore process will look for the file in this location:

```
/var/opt/tableau/tableau_server/data/tabsvc/files/backups/
```

You can change the location. For more information, see [tsm File Paths](#).

2. Run the following command to stop Tableau Server:

```
tsm stop
```

3. Run the following command to restore from the backup file:

```
tsm maintenance restore -f <filename.tsbak>
```

Where `<filename.tsbak>` is the name of your backup file, for example, `tab-server-2017-10-20.tsbak`.

For more information, see [tsm maintenance restore](#)

4. Run the following command to start Tableau Server:

```
tsm start
```

After restoring your Tableau content, you may need to configure Tableau Server. For example, if you are migrating from a multi-node installation, you will need to add and configure the additional nodes. You will also need to configure any customizations you made on Windows, including copying over image or logo files if applicable.

Step 4: Test Tableau Server on Linux

Once you've installed Tableau Server on Linux and restored your Windows backup, you can test the new version of Tableau Server. Because the restore process only restores content to Tableau, you may need to update configuration, topology (adding additional nodes for example), and any customizations you have on your Windows installation.

You'll want to be familiar with the known differences between Windows and Linux, as well as any potential problem areas. Test basic functionality, along with any special aspects of server that your organization relies on. For example, there may be key data sources that your organization uses with Tableau. Test these to make sure you're seeing what you expect.

These are some areas of testing to consider:

- **User access.** Confirm that Tableau Server users, including administrators, can sign in. Test your normal user sign in process. Have some of your users participate in the testing to make sure they are able to sign in as expected, and that they can get to the same content that they have access to in your production environment.
- **Viewing built-in administrative views.** With this release of Tableau Server on Linux, you must install PostgreSQL drivers manually, and the administrative views depend on these. Confirm that you have installed the drivers necessary by accessing the built-in administrative views. For more information, see [Administrative Views](#) .
- **Data source availability.** Tableau Server on Linux supports a subset of the data

sources on Tableau Server on Windows. You need to confirm that the data sources used by your organization are supported on Linux, and install any drivers required. For details on which data sources are supported, see the [Tableau Server tech specs](#). For information about installing drivers, see Database Drivers.

- **Access to file-based data sources on shared drives.** Data sources such as Excel files on network drives will require special actions in order to be accessible from Linux. You or your IT department will need to mount the drives and update any workbooks using these data sources. For more information on connecting to network drives, see the [Tableau Community](#).
- **Viewing published workbooks.** Have users who are familiar with the content try to view published workbooks to make sure they appear as expected. Test views embedded in web pages (for example, in SharePoint pages). Be especially aware of how fonts may differ between Windows and Linux, and some custom fonts may need to be added to your Linux computer, or replaced with other fonts if they are not available on Linux. Dashboard layouts may appear different as well, due to differences in fonts.
- **Server processes.** Sign in to Tableau Server as a server administrator, and then open the Server Status page to confirm that all services and processes are running as expected. Be aware of intentional changes due to changes in version 10.5. For more information, see Tableau Server Data Engine.
- **Publishing workbooks and data sources.** Have users publish workbooks and data sources from Tableau Desktop to make sure this goes as you expect. You may need to install drivers to support the data source connections you are using. See Database Drivers.
- **Subscriptions and extract refreshes.** Manually run some extract refreshes to confirm that they complete successfully. Run some key scheduled extract refreshes to confirm that they complete as expected.
- **Permissions.** Confirm that permissions are still set as expected for users and

content.

- **Command-line utilities and APIs.** If applicable, test the command line utilities (tsm and tabcmd) and programmatic access via APIs. With Tableau Server on Linux, tabadmin is replaced by TSM. For more information, see [Migrate from Tabadmin to the TSM CLI](#).

Step 5: Install Tableau Server on Linux in your production environment and restore the Windows backup

When you have completed testing and have identified those areas that require additional changes on your part, or communication to your users, you are ready to install Tableau Server on Linux in your production environment and restore the Windows backup. To do this, follow the same steps described above.

Important: *If you are migrating from Tableau Server version 2018.1 or earlier on Windows - You can take a backup from Tableau Server on Linux and restore that backup to Tableau Server on Windows version 2018.2 or later, but you cannot restore that backup to Tableau Server on Windows version 2018.1 or earlier. This means that if you migrate to Linux, you cannot easily switch back to Tableau Server on Windows version 2018.1 or earlier. Keep this in mind as you test your migration, and consider keeping your installation on Windows until you are certain that you no longer need it.*

Migrate from Tabadmin to the TSM CLI

The Tableau Services Manager (TSM) command-line interface (CLI) replaces the tabadmin CLI in Tableau Server on Linux, and in Tableau Server on Windows version 2018.2. This page maps tabadmin commands to TSM commands to help you to migrate to the TSM CLI.

To learn more about the TSM CLI, see [tsm Command Line Reference](#).

Looking for tabadmin commands for Tableau Server on Windows version 2018.1 and earlier? See [tabadmin Commands](#).

Tabadmin commands with a corresponding TSM CLI command

The following table shows which tabadmin commands correspond to commands available in the TSM CLI.

Command Description	Tabadmin Command(s)	Comparable TSM CLI Command
Activate a license	<code>tabadmin activate --activate</code>	<code>tsm licenses activate</code>
Deactivate licenses	<code>tabadmin activate --return</code>	<code>tsm licenses deactivate</code>
Activate a trial license	<code>tabadmin activate --trial</code>	<code>tsm licenses activate -trial</code>
Create a backup of the data managed by Tableau Server	<code>tabadmin backup</code>	<code>tsm maintenance backup</code> A backup created using TSM does not include any server configuration data. There is no option to include server configuration data.
Clear the server cache	<code>tabadmin clearcache</code>	<code>tsm maintenance cleanup -r</code>
Clean up temporary files and old log files	<code>tabadmin cleanup</code>	<code>tsm maintenance cleanup</code> Note: This command was added in version 10.5.1
Update the	<code>tabadmin configure</code>	<code>tsm pending-changes</code>

server configuration with any changes you've made		apply
Customize the server name and logos	tabadmin customize	tsm customize
Enable access to the repository	tabadmin dbpass	tsm data-access repository-access enable
Disable access to the repository	tabadmin dbpass --disable	tsm data-access repository-access disable
Set a file store instance to read-only mode	tabadmin decommission	tsm topology filestore decommission
Delete one or more Web Data Connectors (WDCs) from Tableau Server	tabadmin delete_web-dataconnector	tsm data-access web-data-connectors delete To learn more, see Web Data Connectors in Tableau Server.
Add a Web Data Connector (WDC) to Tableau Server	tabadmin import_web-dataconnector and tabadmin whitelist_web-dataconnector	tsm data-access web-data-connectors add Note: TSM does not support importing WDCs, instead it lets you add WDCs to an allowlist. To learn more, see Web Data Connectors in Tableau Server.
List Web Data	tabadmin list_web-	tsm data-access web-

Tableau Server on Linux Administrator Guide

Connectors (WDCs) used by Tableau Server	<code>dataconnectors</code>	<code>data-connectors list</code> To learn more, see Web Data Connectors in Tableau Server.
Export a site from Tableau Server	<code>tabadmin exportsite</code>	<code>tsm sites export</code>
Initiate a repository failover	<code>tabadmin fail-overrepository</code>	<code>tsm topology failover-repository</code>
Get a configuration option	<code>tabadmin get</code>	<code>tsm configuration get</code>
Get the OpenID redirect URL	<code>tabadmin get_openid_redirect_url</code>	<code>tsm authentication openid get-redirect-url</code>
Import site .csv files into Tableau Server	<code>tabadmin importsite</code>	<code>tsm sites import</code>
Import a site into Tableau Server using .csv files	<code>tabadmin importsite_verified</code>	<code>tsm sites import-verified</code>
Display license information for Tableau Server	<code>tabadmin licenses</code>	<code>tsm licenses list</code> Note: For more information about the output of this command, see View Server Licenses .
Move a file store from read-only mode to an active	<code>tabadmin recommission</code>	<code>tsm topology filestore recommission</code>

read/write state		
Regenerate internal security tokens	<code>tabadmin regenerate_internal_tokens</code>	<code>tsm security regenerate-internal-tokens</code>
Register Tableau Server	<code>tabadmin register</code>	<code>tsm register</code>
Rebuild the search index for Tableau Server	<code>tabadmin reindex</code>	<code>tsm maintenance reindex-search</code>
Reset the Tableau Server administrator account	<code>tabadmin reset</code>	<code>tsm reset</code> Note: Added in version 2018.1.
Stop and restart all Tableau Server processes	<code>tabadmin restart</code>	<code>tsm restart</code>
Restore from a Tableau Server backup file	<code>tabadmin restore</code>	<code>tsm maintenance restore</code> The restore command does not restore any server configuration data. This is true whether you are using a backup created with TSM or a backup created with <code>tabadmin</code> .
Set a configuration option	<code>tabadmin set</code>	<code>tsm configuration set</code>

Tableau Server on Linux Administrator Guide

Activate or suspend a site	<code>tabadmin sitestate</code>	<code>tsm sites unlock</code>
Start all Tableau Server processes	<code>tabadmin start</code>	<code>tsm start</code>
Get the status of Tableau Server and server processes	<code>tabadmin status</code>	<code>tsm status</code>
Stop all Tableau Server processes	<code>tabadmin stop</code>	<code>tsm stop</code>
Create an archive (.zip) file with Tableau Server log files	<code>tabadmin ziplogs</code>	<code>tsm maintenance ziplogs</code> The default behavior of the ziplogs command has changed: with tsm, the command collects up to the last two days of log files by default. The tabadmin ziplogs command collected up to seven days of log files. For more information, see <code>tsm maintenance ziplogs</code> .

Tabadmin commands with no corresponding TSM CLI command

The following table lists the tabadmin commands for which a comparable TSM CLI command is not available.

Command Description	Tabadmin Command	Notes
Add or remove	<code>tabadmin administrator</code>	You can use the Tableau Server

a user from the system administrator group		REST API Add User to Group and Remove User from Group methods to add or remove a user from the system administrator group.
Create a new key to encrypt sensitive information stored in the repository	<code>tabadmin assetkeys</code>	Use the <code>tsm security regenerate-internal-tokens</code> command to create or regenerate secrets and master keys.
Specify whether Tableau Server starts at system start-up time	<code>tabadmin autostart</code>	Tableau Server returns to the state it was in prior to a system restart. If it was running, it will restart. If it was stopped it will be stopped after the system starts.
Identify a second server node for backup	<code>tabadmin fail-overprimary</code>	TSM does not have primary nodes, so a TSM equivalent to this command is not needed.
Manage credentials for delegated data access on Tableau Server	<code>tabadmin manage_global_credentials</code>	We recommend that you use Kerberos delegation to Apache Impala for global credential management. To learn more, see Kerberos and Enable Kerberos Delegation for Hive/Impala in the Tableau Community.
Reset the password for a Tableau Server account	<code>tabadmin passwd</code>	If your server uses local authentication, you can use the Tableau Server REST API Update User method to reset the password for a user account.
Reset binding	<code>tabadmin reset_openid_</code>	

Tableau Server on Linux Administrator Guide

between Tableau Server user ID and Open ID Connect identity provider	<code>sub</code>	
Determine whether your environment meets the minimum requirements to run Tableau Server	<code>tabadmin validate</code>	
Verify that a backup of the Tableau Server repository will restore successfully	<code>tabadmin verify_data-base</code>	The <code>tsm maintenance backup</code> command automatically verifies that a backup will restore correctly unless you use the <code>--skip-verification</code> parameter.
Prepare VizQL processes for fast load times after a Tableau Server restart	<code>tabadmin warmup</code>	The <code>tabadmin warmup</code> command is no longer necessary, as Tableau Server is now optimized to automatically provide fast load times after a server restart.

Migrate Tableau Server from an On-Premises Computer to a VM in the Cloud

You can migrate Tableau Server from a computer in your data center to a virtual machine (VM) in the cloud. As a part of this migration, you'll need to move various Tableau Server data and configuration settings from your on-premises computer to a VM in the cloud where Tableau Server is installed.

Note: This operation includes steps that you may need to perform using the TSM command line.

Prerequisites

Before you migrate Tableau Server from an on-premises computer to a VM in the cloud, be sure to review the following prerequisites:

- Have an account and an environment set up on your cloud provider. For more information, see:
 - [Install Tableau Server on Amazon Web Services](#)
 - [Install Tableau Server on the Google Cloud Platform](#)
 - [Install Tableau Server on Microsoft Azure](#)
 - [Install Tableau Server in the Alibaba Cloud](#)
- Read [Preparing for Upgrade and Back up Tableau Server data](#) to prepare for the migration.

To migrate Tableau Server to a VM in the cloud

1. Ensure that there are no background tasks running, such as, extract refreshes, workbook publishing, etc. When Tableau Server is idle, note any data source connection types in use so that you can install the appropriate drivers on the new Tableau Server on the VM in the cloud.

Tableau Server on Linux Administrator Guide

2. Stop Tableau Server. At a command prompt, type:

```
tsm stop
```

3. After Tableau Server is stopped, make a complete backup, following the steps in [Preparing for Upgrade](#) and [Back up Tableau Server data](#). The backup is what you will use to restore your Tableau Server data. You'll use this backup to restore your Tableau Server data on the VM in the cloud.
4. Move the backup off the server and store the backup securely. For example, you can store your backup on cloud storage such as Amazon S3, Google Drive, etc.
5. If you are using SSL on the on-premises server, make a copy of the SSL folder. Be sure to note any security rules, for example, firewall rules, ports, etc. to avoid communication issues between different elements of your infrastructure and your new Tableau Server on the VM in the cloud.
6. Remove any DNS entries using the IP address of the on-premises Tableau Server, so you can reassign them to the Tableau Server on the VM in the cloud.
7. Launch a VM into the cloud that meets the Tableau Server recommendations.
8. Install Tableau Server on the VM in the cloud. For more information, see:
 - [Install Tableau Server on Amazon Web Services](#)
 - [Install Tableau Server on the Google Cloud Platform](#)
 - [Install Tableau Server on Microsoft Azure](#)
 - [Install Tableau Server in the Alibaba Cloud](#)
9. After installation is complete, copy any SSL files to Tableau Server on the VM in the cloud, and restore the backup on your new server.
10. Configure Tableau Server on the VM in the cloud and install all SSH and port related stuff.

Note: If you change the IP address/port of the database, you might need to update all connection information.

11. Copy your backup file `.tsbak` to the folder location specified by the

`basefilepath.backuprestore` variable. By default this is: `/var/opt/tableau/tableau_server/data/tabsvc/files/backups`.

Note: You can change the location for the backup file. For more information, see [tsm File Paths](#).

12. Next, stop Tableau Server. At a command prompt, type:

```
tsm stop
```

13. Restore your in-production data without configuration information to your new Tableau Server installation. At a command prompt, type:

```
tsm maintenance restore -f <filename>
```

where `<filename>` is the name of the `.tsbak` file you copied in step 11.

For more information about restoring Tableau Server data, see [Restore from a backup](#).

14. Start the server. At a command prompt, type:

```
tsm start
```

15. **Distributed installations only:** Install Tableau Server on the new VMs you want to add to your new Tableau Server cluster in the cloud. For more information, see:

- [Install and Configure Additional Nodes](#)
- [Self-Deploy Tableau Server on AWS in a Distributed Environment](#)
- [Self-Deploy Tableau Server on the Google Cloud Platform in a Distributed Environment](#)
- [Self-Deploy Tableau Server on Microsoft Azure in a Distributed Environment](#)
- [Self-Deploy Tableau Server on Alibaba Cloud in a Distributed Environment](#)

16. Test your Tableau Server on your VM in the cloud to ensure that it works as expected. If your Tableau Server is working fine in the cloud, you can deactivate your on-premises Tableau Server product key and use this same product key to activate your Tableau Server on the VM in the cloud.

Note: If you do not have an internet connection, you are prompted to create an off-line activation file to complete the deactivation process. For more information, see [Deactivate Tableau Server Offline](#).

17. The same Tableau Server product key can be activated three times: once for a production environment, once for a test environment, and once for a QA environment. After you have tested your new Tableau Server installation and confirmed that it's ready for production, you must deactivate your earlier production version of Tableau Server, and then you must uninstall it. To deactivate the earlier version, see [tsm licenses deactivate](#).

Note: If you do not have an internet connection, you are prompted to create an off-line activation file to complete the deactivation process. See [Deactivate Tableau Server Offline](#) for steps.

Upgrade Tableau Server Overview

The articles in this section help you upgrade an existing installation of Tableau Server on Linux. They describe the recommended steps of planning and testing *before* actually performing the upgrade. There's information about best practices, and when you're ready to actually perform your upgrade, steps for upgrading a single node server and a multi-node installation. Where possible, we call out possible pitfalls and help you to avoid these.

Note: If you are looking for documentation about upgrading to a version earlier than 2018.2.0, go to the [main Tableau Help web site](#) and use the Version dropdown to select the version you are upgrading to, then click the help for that version.

Looking for Tableau Server on Windows? See [Upgrade Tableau Server Overview](#).

Choose your upgrade path

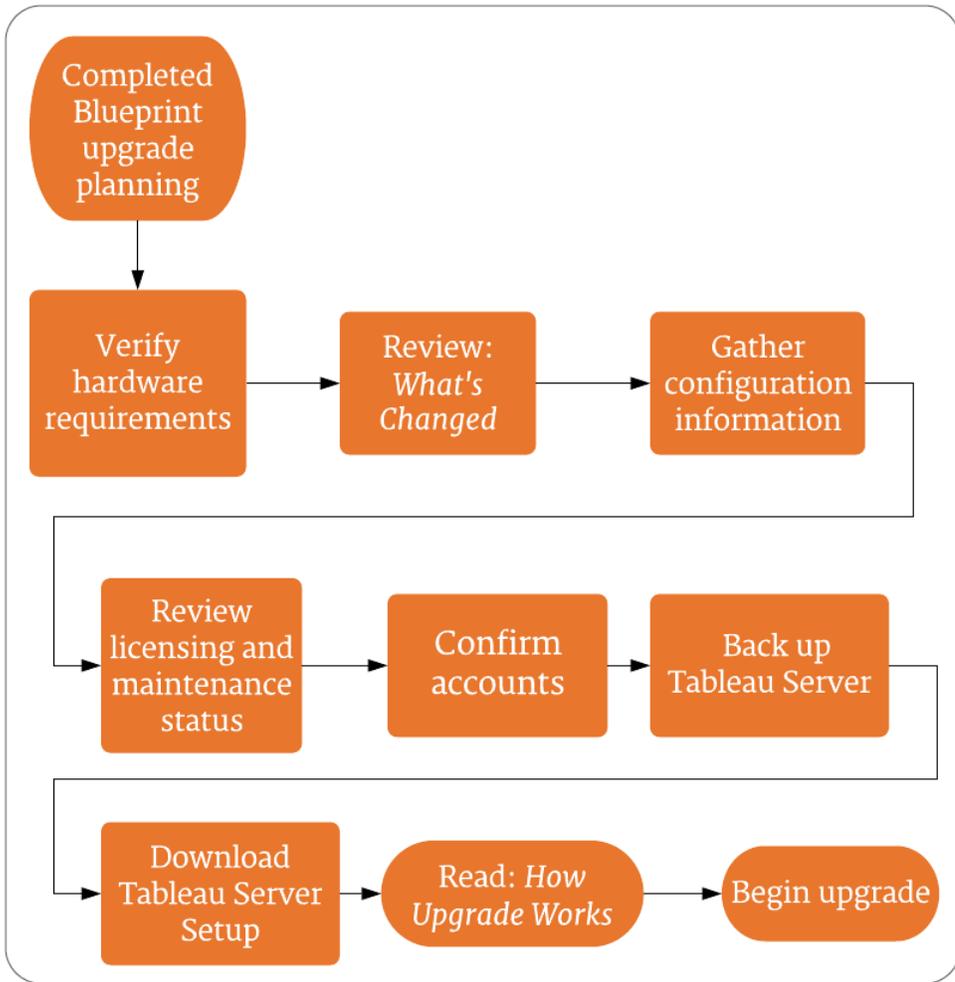
Important: Your upgrade steps depend on which version you are upgrading from. When you are ready to actually upgrade, be sure you follow the procedure that applies to your installation:

- Upgrading from Tableau Server on Linux version 10.5 (10.5.0 or later) requires special steps. If you are upgrading from version 10.5, use the Upgrade Tableau Server on Linux from 10.5 procedure.
- Upgrading from Tableau Server on Linux version 2018.1 or later is more straightforward. If you are upgrading from version 2018.1.0 or later, follow the steps in Upgrading from 2018.1 and Later (Linux).

For instructions on how to determine your version of Tableau Server, see View Server Version.

Preparing for Upgrade

Follow the flow chart and the linked topics to prepare your environment for Tableau Server upgrade.



Server Upgrade - Minimum Hardware Recommendations

For production use, the computer on which you upgrade Tableau Server should meet or exceed the minimum hardware recommendations. These recommendations are general. Actual system needs for Tableau Server installations can vary based on many factors, including number of users and the number and size of extracts. If the Setup program determines that your computer does not meet the following recommendations, you will get a warning, but you can continue with the setup process. The minimum recommendations listed here are intended as general guidance. However the recommendations for your environment may

vary. For more information, see the [Hardware recommendations section](#) of the [Recommended Baseline Configurations](#) topic.

<i>Install Type</i>	<i>Processor</i>	<i>CPU</i>	<i>RAM</i>	<i>Free Disk Space</i>
Single node	<ul style="list-style-type: none"> • 64-bit • Must support SSE4.2 and POPCNT instruction sets • ARM-based processors are not supported 	8-core, 2.0 GHz or higher	32 GB	50 GB

If you are adding Tableau Prep Conductor to your Tableau Server installation, we recommend you add a second node and dedicate this to running Tableau Server Prep Conductor. This node should have a minimum of 4 cores (8 vCPUs), and 16 GB of RAM.

Multi-node and enterprise deployments	<p>Contact Tableau for technical guidance.</p> <p>Nodes must meet or exceed the minimum hardware recommendations, except:</p> <ul style="list-style-type: none"> • Nodes running backgrounder, where 4 cores may be acceptable. • Dedicated node for Tableau Prep Conductor: Minimum of 4 cores (8 vCPUs), and 16 GB of RAM.
---------------------------------------	--

Important: The disk space requirement cannot be checked until you initialize TSM. If you don't have enough space, you won't be told this until after you install the Tableau Server package.

50 GB disk space available, with a minimum of 15 GB allocated to the `/opt` directory, and the remainder allocated to the `/var` directory for data storage.

- Free disk space is calculated after the Tableau Server Setup program is unzipped. The Setup program uses about 1 GB of space. You may need to allocate additional disk space depending on various factors like whether you will be using extracts.

The core Tableau Server bits must be installed in a directory with at least 15 GB of free disk space. If you attempt to install Tableau Server on a computer that does not have enough space, the Tableau Server package will install, but you will be unable to continue with setup. By default the install location is the `/opt` directory. You can change the installation path for Tableau Server on RHEL distros.

If you plan to make heavy use of extracts then you may need to allocate additional disk space. You can specify a different directory for data (extract) storage during installation.

- **Network attached storage space requirements for External File Store:** If you are planning to configure [Tableau Server with External File Store](#), you will need to estimate the amount of storage space to dedicate on your network attached storage.

Estimating the storage size: You must take into account the amount of storage needed for publishing and refreshing extracts. In addition, you must also take into account the repository backup size unless you specifically choose the option to do your repository backup separately as described in the [Option 2: Backup Repository Separately](#) topic.

- Extracts:
 - Consider the number of extracts that will be published to Tableau Server and the size of each extract. Test your needs by publishing several extracts to Tableau Server, and then checking the disk space used. You can use this amount of disk space to help you figure out how many

extracts will be published to Tableau Server over time as well as how each existing extract will increase in size.

- Consider the space needed by the temp directory during an extract refresh. The temp directory, which is where an extract is stored to during a refresh, may require up to three times the final file size of the extract.
- Repository Backup:
 - To obtain an estimate of the repository data, check the size of `<data directory>/pgsql/data/base` directory.
 - To obtain the exact size of the repository data, open the backup file and use the size of the `workgroup.pg_dump` file.
- Core count is based on "physical" cores. Physical cores can represent actual server hardware or cores on a virtual machine (VM). Hyper-threading is ignored for the purposes of counting cores.
- RAM shown is the minimum recommended for a single-node installation. Your installation may function better with more RAM, depending on activity, number of users, and background jobs, for example.

For hardware recommendations for Tableau Server in the cloud, see the following:

- Selecting an AWS Instance Type and Size in the [Tableau Server on Linux in the AWS Cloud Administrator Guide](#)
- Selecting a Google Compute Engine Virtual Machine Type and Size in the [Tableau Server on Linux in the Google Cloud Platform Administrator Guide](#)
- Selecting a Microsoft Azure Virtual Machine Type and Size in the [Tableau Server on Linux in Microsoft Azure Administrator Guide](#)

Continue to Server Upgrade - Review What's Changed.

Server Upgrade - Review What's Changed

This topic includes a list of important product changes in Tableau Server, beginning with version 2018.1. The changes described in this topic may impact the upgrade process itself, or they may impact functionality after you have upgraded. Read these changes carefully and make note of the changes and remediation steps that you'll need to take. Include these remediation steps as part of your upgrade process or post-upgrade configuration plan.

This list is cumulative, so if you are upgrading from an early version (for example, 2018.3), read the list of changes for every version between your version and the version you are upgrading to.

This topic does not include all of the new features that have been added in each release. To develop a more robust verification and testing plan, we recommend that you also review new features before you upgrade. To see all of the new features, by release, see [What's New and Changed in Tableau Server](#).

[Tableau Server Release Notes](#) | [Known Issues](#)

Changed in 2021.2

Version 2021.2 includes the following changes you should know about before upgrading.

Encrypted communication channel with Active Directory is required

Beginning with version 2021.2, Tableau Server on Linux requires an encrypted LDAP channel when you use Active Directory as an identity store. You must install a valid SSL/TSL certificate before upgrading to 2021.2. Although not recommended, you can also disable the default encrypted LDAP channel. For more information about installing certificate and enabling or disabling encryption for Active Directory communication, see [Configure Encrypted Channel to LDAP External Identity Store](#).

Secure SAML configuration is enforced

Beginning with version 2021.2, Tableau Server enforces secure SAML configuration for Server-wide and Site SAML.

Signature algorithm The certificate that you configure for SAML must use a secure signature algorithm, for example SHA-256. If you attempt to upgrade Tableau Server with a SAML certificate that uses SHA-1 signature hash, upgrade will fail.

We recommend that you upload a new certificate with SHA-256. See [Update SAML Certificate](#).

Alternatively, you can configure Tableau Server to accept the less-secure SHA-1 hash by setting the `tsm wgserver.saml.blocklisted_digest_algorithms` configuration key.

RSA key and ECDSA curve sizes The certificate used for SAML must either have an RSA key strength of 2048, or ECDSA curve size of 256. If you attempt to upgrade Tableau Server with a SAML certificate that uses less-secure key or curve sizes, upgrade will fail.

We recommend that you upload a new certificate with more secure key or curve size. See [Update SAML Certificate](#).

Alternatively you can configure Tableau Server to accept the less-secure sizes by setting the respective configuration keys, `wgserver.saml.min.allowed.rsa_key_size` and `wgserver.saml.min.allowed.elliptic_curve_size`.

Site SAML secure configuration verification

This release includes an updated `tabcmd` utility with the `validateidpmetadata` command. The `validateidpmetadata` command will scan your existing SAML sites and identify the signature algorithms and key sizes that are configured for each site. See [validateidpmetadata](#).

Tableau Prep Flow Service has been renamed to Tableau Prep Minerva Service

Beginning with version 2021.2, the Tableau Prep Flow Service (`flowqueryservice`) has been renamed to the Tableau Prep Minerva Service (`flowminerva`). This process appears on the TSM Status page and in the output of the TSM command `tsm status -v`. For more information, see [Tableau Server Processes](#).

New configuration key for setting maximum time for single subscription job

Beginning in version 2021.2, you now set the maximum time to complete a single subscription job using the new `backgrounder.timeout.single_subscription_notify` configuration key. This key replaces use of the `backgrounder.timeout_tasks` key for configuring timeouts on single subscription jobs. As of version 2021.2 the `backgrounder.timeout_tasks` key no longer uses the `single_subscription_notify` option. For more information, see `backgrounder.timeout.single_subscription_notify`.

Changed in 2021.1

Version 2021.1 includes the following changes you should know about before upgrading.

Change in default behavior for snapshot backups

Note: If you have both External File Store and External Repository enabled and use scripts to do backups, this can potentially result in not having a backup of the repository.

Beginning with Tableau Sever 2021.1, if you have both External File Store and External Repository enabled, Tableau Server backup process no longer includes creating a backup file of the repository by default. If you are upgrading from a version earlier than 2021.1, and you have scripts to run or schedule your backups, and want to continue to include the repository backup, see [Option 1: Include Repository in the Backup](#).

SHA-256 hash algorithm enforced for SAML assertions with IdP

Beginning with Tableau Sever 2021.1, Tableau Server will hash message signatures and digests with SHA-256 in SAML assertions to the IdP. To change this behavior, see `wgserver.saml.sha256`.

New SAML configuration keys

Beginning with Tableau Sever 2021.1, Tableau Server includes three new SAML configuration keys:

- `wgserver.saml.blocklisted_digest_algorithms`: Specifies the digest algorithms that are not allowed as part of certificates used in SAML configuration or for signing SAML assertions. When set, assertions that are signed with a blocklisted algorithm will fail and certificates containing a blocklisted digest algorithm will fail. We recommend blocking the SHA-1(SHA1) algorithm. Before blocking SHA-1, confirm that your SAML certificates digest algorithms are not SHA-1, and that your SAML idP does not sign messages with SHA-1.
- `wgserver.saml.min.allowed.elliptic_curve_size`: Specifies the minimum elliptic curve size required for ECDSA certificates that uploaded and configured for Tableau Server. If Tableau Server is configured with an ECDSA certificate that does not satisfy this minimum curve size, Tableau Server may go into a degraded state. If you attempt to upload a certificate that does not meet the minimum curve size, Tableau Server will reject the certificate. We recommend a minimum curve size of 256.
- `wgserver.saml.min.allowed.rsa_key_size`: Specifies the minimum key size that required for RSA certificates that are uploaded and configured for Tableau Server. If Tableau Server is configured with an RSA certificate that does not satisfy this minimum key size, Tableau Server may go into a degraded state. If you attempt to upload a certificate that does not meet the minimum key size, Tableau Server will reject the certificate. . We recommend a minimum key size of 2048.

Tableau Server on Linux Administrator Guide

In future releases of Tableau Server, these keys will be configured with secure minimum defaults, as specified in the recommendations above. In future releases, if your SAML configuration is not configured with the minimum requirements, SAML functionality or upgrade may fail. In the meantime, we recommend that you update your SAML configuration to be in compliance with the recommendations above.

Client and CA certificates used for mutual SSL must meet minimum security requirements

Beginning with Tableau Server 2021.1, Tableau Server configured for mutual SSL authentication will enforce secure certificate requirements.

Client and certificate authority (CA) certificates used for mutual SSL must either have an RSA key strength of 2048, or ECDSA curve size of 256.

Tableau Server will fail mutual authentication requests from client certificates that do not meet these requirements.

Additionally, upgrade of Tableau Server will fail if the certificate authority (CA) certificate that is configured for mutual SSL does not meet these requirements.

- To update the CA certificate file for Tableau Server mutual SSL, upload the certificate as described in [Configure Mutual SSL Authentication](#).
- To disable the certificate security enforcement, see the Tableau KB topic, [Tableau Server Using Mutual SSL Authentication Fails to Start or Rejects Login After Upgrade to Tableau Server 2021.1 because Certificates Use Insecure Key Attribute](#).

Improvements to JVM maximum heap size scaling for Tableau Catalog ingestion and elastic server

Beginning with Tableau Sever 2021.1, JVM maximum heap size scales automatically for Catalog ingestion and elastic server. For more information, see `non-interactivecontainer.vmopts` and `elasticserver.vmopts`.

Old notifications do not appear in new notifications center

When you upgrade to Tableau Server 2021.1, old notifications will not appear in the new notifications center. All existing email notifications will continue. To continue tracking older

notifications, make sure the server setting for Notifications for Extract Jobs and Flow Runs is enabled before upgrading. For more information, see [What's New and Changed in Tableau Server](#).

Changed in 2020.4

Version 2020.4 includes the following changes you should know about before upgrading.

SHA1 certificates disabled for SSL mutual authentication

Beginning with Tableau Server 2020.4, Tableau Server configured for mutual SSL authentication will block authentication of users with client certificates that use the SHA-1 signing algorithm.

Users who attempt to log in with SHA-1 client certificates encounter an "Unable to sign in" error, and the following error will be visible in the VizPortal logs:

```
Unsupported client certificate signature detected: [certificate
Signature Algorithm name]
```

If the clients in your organization are connecting to Tableau Server using SSL mutual authentication and the clients are using certificates with the SHA-1 signing algorithm, then you must perform one of the following actions before upgrading:

- Update the certificates used by your clients. The certificates must use the SHA-256 (or stronger) signing algorithm, or
- Disable the SHA-1 blocklist before upgrading to Tableau Server 2020.4. To disable the blocking functionality, run the following TSM commands before you upgrade:

```
tsm configuration set -k "ssl.client_certificate_login.b-
locklisted_signature_algorithms" -v ""
```

Note that the value (-v) includes an empty set of quotes to specify a null value.

For more information about how to reset this value after you have updated client certificates, see the Knowledge Base article, [Mutual SSL Fails After Upgrading if Certificates Signed with SHA-1](#).

For more information about why this change has been implemented, see the Tableau Community post, [Retiring Default Support for SHA-1 Signed Certificates Used During Mutual SSL Authentication in Customer-Managed Tableau Server 2020.4](#), and the National Institute of Standards and Technology policy statement, [NIST Policy on Hash Functions](#).

`wgserver.domain.whitelist` config key has been deprecated

Beginning with version 2020.4, the `wgserver.domain.whitelist` configuration key has been deprecated. Use the new key, `wgserver.domain.accept_list` to specify allowed domains.

All functionality of the older key has been transferred to the new key. As of 2020.4.0, existing installations configured with the old key will not be impacted, and the key can still be used, but you must include the `--force-keys` option when setting the key, and the key will be permanently removed in a future release. For all new configurations, use `wgserver.domain.accept_list` instead. For more information about the new configuration key, see `wgserver.domain.accept_list`.

CPU requirement update

Beginning with version 2020.4, Tableau Server requires all nodes to have CPUs that support SSE4.2 and POPCNT instruction sets. These CPUs have been common for more than 10 years and are installed on most newer computers. If you attempt to install or upgrade Tableau on a computer that does not support these instruction sets, the installer will not allow you to continue.

Checkpoint Upgrade for Tableau Server

Version 2020.4 of Tableau Server includes updates to the `upgrade-tsm` script that give you the ability to rerun the script when an upgrade fails. The script keeps track of successful progress, and when you rerun the script after an upgrade failure, restarts at the latest

successful "checkpoint." This saves time because successful steps do not need to be repeated on rerunning the script. As a first troubleshooting step, rerun the `upgrade-tsm` script. Doing this can result in a successful upgrade if a momentary environmental problem or timing issue caused the initial failure.

Checkpoint upgrades also give experienced IT professionals the option to further troubleshoot the failure, and make corrections (for example, if there is a problem with disk space, or a permissions issue on a particular file or directory) before rerunning the upgrade script. For troubleshooting details, see [Upgrade fails on 2020.4.0 or later](#).

Upgrades from 10.5 must be done in stages

Beginning with version 2020.4, version 10.5 of Tableau Server on Linux is no longer officially supported. This means you cannot upgrade directly from 10.5.x to version 2020.4.0 or later. To upgrade from version 10.5.x you must first upgrade to a version between 2018.1 and 2020.3, and then upgrade to the newest version. For instructions on upgrading from 10.5, see [Upgrade Tableau Server on Linux from 10.5](#).

PostgreSQL upgrade

Version 2020.4 of Tableau Server includes a major-version update of the PostgreSQL database used for the Tableau repository. During the upgrade a temporary `pgsql-only` backup is created and this increases the length of time the upgrade takes as well as the amount of temporary disk space used.

If you have External Repository configured, before upgrading your Tableau Server to 2020.4, you must make sure you are using PostgreSQL version 12 or later. For more information, see [Upgrade Tableau Server with External Repository for a New Major Version of PostgreSQL](#)

PostgreSQL database connections require updated driver

Beginning with version 2020.4, PostgreSQL connections use a new JDBC driver. You must download and install this driver on Tableau Server on Linux to enable existing content rely-

ing on Postgres connections to properly display. This includes Administrative Custom Views. Follow the instructions on the [Driver Download](#) page to get and install the new driver.

Visual improvements to user, group, and site dialogs

The 2020.4 release adds visual updates to administrator menus and dialogs to make managing users, groups, and sites easier in Tableau Server.

Changes to default JVM heap size for Tableau Catalog ingestion

Beginning with Tableau Server 2020.4, the default JVM heap size for Catalog ingestion is determined by and then scales based on system memory. To review the default values for Catalog ingestion, see `noninteractivecontainer.vmopts`.

Changed in 2020.3

Version 2020.3 includes the following changes you should know about before upgrading.

Data Acceleration for Workbooks that use the Object Model Framework

Administrators can now also enable data acceleration for workbooks that use the Object Model framework. See [Data Acceleration](#).

Changed in 2020.2

Version 2020.2 includes the following changes you should know about before upgrading.

Analytics extensions (external service) connections must be reconfigured

Tableau supports a set of functions that your users can use to pass expressions to analytics extensions for integration with R and Python. Previously, this feature was referred to "external services."

In previous versions of Tableau Server, the analytics extensions configuration applied to the Tableau Server global configuration and was administered with TSM.

Beginning with version 2020.2, you can configure a different analytics extensions connection for each site. Administration of the analytics extensions connection has moved out of TSM and to Tableau Server administrator. Additionally, configuring an analytics extensions connection no longer requires a restart of Tableau Server.

However, if you have configured Tableau Server with an analytics extensions connection, then you must reconfigure that connection after you upgrade Tableau Server to version 2020.2.

Before upgrading, document the existing analytics extensions configuration. On previous versions (2019.1-2020.1) you can run the `tsm security vizql-extsvc-ssl list` command to list the existing connection details. To retrieve the password that is stored for the analytics extensions connection (if any), run the following `tsm` command:

```
tsm configuration get -k vizqlserver.rserve.password
```

After you have upgraded, configure the analytics extension connection. See [Configure Connections with Analytics Extensions](#).

Oracle data source connections require updated driver

Beginning with version 2020.2, Oracle data sources use a new JDBC driver. You must download and install this driver on Tableau Server to enable existing content relying on Oracle connections to properly display. Until this driver is downloaded and installed on Tableau Server, existing content will not display or refresh. Follow the instructions on the [Driver Download](#) page to get and install the new driver. For details on the changes and how they impact Tableau authors, see [Oracle](#) in the Tableau Desktop and Web Authoring Help.

Changed in 2020.1

Version 2020.1 includes the following changes you should know about before upgrading.

- Server Upgrade - Review What's Changed
- `tableau-server-obliterate` script now preserves backup and log files

Tableau Server on Linux Administrator Guide

- Improvements to the `tsm topology deploy-coordination-service` command
- Large statistical file extracts consume system resources

`tableau-server-obliterate` script now preserves backup and log files

In previous versions of Tableau Server, running the `tableau-server-obliterate` script deleted all content from the Data directory. Beginning with version 2020.1, the default behavior of the `tableau-server-obliterate` script has changed: the script copies and saves Tableau Server backup and log files to the `logs-temp` directory. The default location for the `logs-temp` directory is at `/var/opt/tableau/logs-temp`. You can set options on the script to disable this new functionality.

See [Remove Tableau Server from Your Computer](#).

Improvements to the `tsm topology deploy-coordination-service` command

Beginning with version 2020.1, the command to deploy a new coordination service ensemble waits for the ensemble to be properly deployed and removes the old ensemble before returning control to the command prompt. This simplifies the process for deploying a new coordination service ensemble and eliminates the chances of continuing on with other commands or actions before Tableau Server is properly configured for the new ensemble. You no longer need to run the `tsm topology cleanup-coordination-service` command after deploying a new ensemble in version 2020.1.0 and later.

Note: With this change, all coordination service ensemble commands require input for a "y/n" prompt confirming that a server restart will take place. To run these commands without input, include the `--ignore-prompt` option.

For more information about deploying a coordination service ensemble, see [Deploy a Coordination Service Ensemble](#).

Large statistical file extracts consume system resources

Beginning with Tableau Server 2020.1, a change to the Statistical File (SAS) library may cause large SAS-based extract refreshes to fail. The following connection error will be visible in the backgrounder logs, along with spikes in system memory consumption.

"exceeded per-process memory threshold"

To work around the issue, run the following TSM commands after upgrading Tableau Server:

```
tsm configuration set -k features.ReadStat -v false
tsm pending-changes apply
```

For more information and steps to work around the issue in Tableau Desktop, see the Knowledge Base article, [Large Sas File Extract Refresh Failed And Huge Memory Consumption Observed After Upgraded Tableau Server To 2020.3.1.](#)

Changed in 2019.4

Version 2019.4 includes the following changes you should know about before upgrading.

- Server Upgrade - Review What's Changed
- New tabcmd installer for Windows
- Updates to Tableau Server Job management
- Updates to tsm maintenance cleanup command
- Changed behavior for Copy Link when sharing a view

New tabcmd installer for Windows

The 2019.4 release of Tableau Server features a newly rewritten Setup program for installing tabcmd on Windows.

This update changes some of the command line options. If you have automated the installation of tabcmd, be sure to review the current command line options for the tabcmd Setup program. For more information, see [tabcmd and Install Switches and Properties for tabcmd \(Windows\)](#)

Updates to Tableau Server Job management

The Job management page has several updates and new options including the following:

Tableau Server on Linux Administrator Guide

- The **Jobs** page now includes **Average Run** and **Average Queue** time information for each job.
- Improved job cancellation experience. Server and Site admins can now enter customized notes as well as pick recipients of the e-mail when cancelling a job.
- More information is provided in the job detail dialog box, such as: the last time the job was run successfully, and the name of the job creator.
- Canceling flow run jobs is now supported. Previously, in Tableau Server version 2019.3, only extract refreshes and subscriptions were supported.
- New **Task Type** has been added: Encryption jobs.
- Failed jobs now includes jobs that have been suspended. Suspended jobs are a sub-status of the failed jobs and have their own icon to distinguish them from other failed jobs.

For more information, see [Managing Background Jobs in Tableau Server](#).

Updates to tsm maintenance cleanup command

The `tsm maintenance cleanup` command now includes an option to clear the image cache.

In addition, the `--verbose` option has been removed from the command. Running the command with the `--verbose` option will result in an error. If your organization has automated task using the `tsm maintenance cleanup` command with the `--verbose` option, update your automation scripts.

Changed behavior for Copy Link when sharing a view

Beginning with version 2019.4, the behavior of the **Copy Link** option of the **Share** button in a view no longer includes the "embed=y" parameter by default. To include this parameter you need to edit the copied URL before adding it to your web page code. For more information sharing views by embedding them in web pages, see [Embed Views into Webpages](#) in the Tableau Desktop and Web Authoring Help. For details about the parameter, see [URL parameters for iframe tags](#) in the Tableau Desktop and Web Authoring Help.

Changed in 2019.3

Version 2019.3 includes the following changes you should know about before upgrading.

- Default cipher suite no longer includes default support for old ciphers
- Tableau Server Service Manager has been removed
- Server Upgrade - Review What's Changed
- Upgrade without needing to provide a password
- Some tsm command options are no longer configurable

Default cipher suite no longer includes default support for old ciphers

The 2019.3 version of Tableau Server no longer includes default support for Triple-DES, IDEA, and CAMELIA ciphers for SSL connections. These ciphers are no longer considered adequately secure. In the previous versions of the Security Hardening Checklist, we included a recommendation to disable the Triple-DES cipher, which was enabled by default.

Depending on your environment, you may need to make additional configurations before our after upgrading:

- If you disabled the Triple-DES cipher as documented in the Checklist, then we recommend running the following commands before you upgrade to 2019.3:

```
tsm configuration set -k ssl.ciphersuite -d
```

```
tsm pending-changes apply
```

This command reverts Tableau Server to the 2019.2 (and older) default cipher suite configuration. After you have upgraded to 2019.3, Triple-DES, IDEA, and CAMELIA will be removed from the default supported ciphers.

Do not run the command if you made other changes to the `ssl_ciphersuite` key.

- If you did not change the `ssl_ciphersuite` key, then Setup will update the key

Tableau Server on Linux Administrator Guide

value to the new default as part of the upgrade process. The new default value for the `ssl_ciphersuite` key is
`HIGH:MEDIUM:!aNULL:!MD5:!RC4:!3DES:!CAMELLIA:!IDEA:!SEED.`

- If your users still require support for Triple-DES, then we recommend upgrading to more recent browsers that support more secure ciphers. If that is not an option, then you can run the following commands after you have upgraded. The first command sets the `ssl_ciphersuite` key to the pre-2019.3 configuration, which supports Triple-DES:

```
tsm configuration set -k ssl.ciphersuite -v  
"HIGH:MEDIUM:!aNULL:!MD5:!RC4"  
  
tsm pending-changes apply
```

Tableau Server Service Manager has been removed

The Tableau Server Service Manager (`tabsvc`) has been removed. Instead, `systemd` manages the Tableau business processes that were previously managed by `tabsvc`:

- Gateway
- Repository & Cluster Controller
- File Store
- Application Service
- SAML Service
- Backgrounder
- Data Server
- VizQL Server
- Data Engine
- Cache Server
- Search & Browse

You can use the following scripts to start and stop the business services that `tabsvc` managed in previous versions:

```
start-administrative-services  
  
stop-administrative-services
```

The scripts are located at `/opt/tableau/tableau_server-
/packages/scripts.<version>/`.

Usage data is sent to Tableau

Beginning with Tableau Server 2019.3, server usage data is sent to Tableau by default. This is the case for new installs and upgrades. You have the ability to opt out during installation, upgrade, or at any time after Tableau Server has been installed. For more information about usage data, see [Server Usage Data](#).

To disable the sending of server usage data:

- If you are installing Tableau Server for the first time, you can disable this feature during the configuration of initial node settings. For details, see [Configure Initial Node Settings](#).
- If you are upgrading Tableau Server, the `upgrade-tsm` script will let you know about the option and how to disable it.
- If you want to have installed or upgraded Tableau Server and want to disable the sending of server usage data, you can do this from the TSM Maintenance page, or from the command line. For details, see [Server Usage Data](#).

Upgrade without needing to provide a password

Beginning with Tableau Server 2019.3, when you upgrade from version 2019.2 or later, the `upgrade-tsm` script no longer requires you to specify a user and password. Version 2019.3 and later take advantage of the [What's New and Changed in Tableau Server](#) feature and uses the account you have logged in as to authenticate the upgrade. You can override this behavior with the `--user` option, to specify an administrative user. When you do this, you can also provide a password using the `--password` option, or enter a password when prompted. Upgrades from versions earlier than 2019.2.0 will still prompt you for a password.

Some tsm command options are no longer configurable

Beginning with Tableau Server 2019.3, the following options are enabled on install and are no longer configurable:

- `backgrounder.enable_sort_jobs_by_job_rank`
- `backgrounder.enable_task_run_time_and_job_rank`

Changed in 2019.2

Version 2019.2 includes the following changes you should know about before upgrading.

- Minimum hardware requirements have changed
- Upgrade requires at least 15% free disk space on each node
- "Referrer-Policy" HTTP header added
- Map-related internet addresses have changed

Minimum hardware requirements have changed

Beginning with version 2019.2, Tableau Server requires a minimum of 4 cores and 16 GB of RAM in order to install. The installation program will check that the computer you are installing on satisfies these minimums before you can install. If the computer does not meet these hardware minimums, Tableau cannot be installed. For details, see [Minimum Hardware Requirements and Recommendations for Tableau Server](#). For more information about this change, see this Tableau blog post: [Updated machine requirements for Tableau Server trials](#).

Upgrade requires at least 15% free disk space on each node

Upgrading to version 2019.2 includes a version update to the PostGRES database that Tableau uses internally. This update means that a backup of the database is done before upgrading, and that requires a minimum of 15% free space on each node.

"Referrer-Policy" HTTP header added

Beginning in 2019.2, Tableau Server includes the ability to configure Referrer-Policy HTTP header behavior. This policy is enabled with a default behavior that will include the origin URL for all "secure as" connections (policy no-referrer-when-downgrade). In previous versions, the Referrer-Policy header was not included in responses sent by Tableau Server. This change is not likely to impact user scenarios for most browsers. However, we recommend testing browser-based user scenarios after you upgrade to 2019.2. In the event a user scenario is effected, you can disable or change this header behavior. See [HTTP Response Headers](#).

Map-related internet addresses have changed

Beginning in 2019.2, Tableau Server needs access using port 443 to two new map locations:

- mapsconfig.tableau.com
- api.mapbox.com

These replace the address used by previous versions: maps.tableau.com. For details on configuring Tableau Server for proper internet access, see [Configuring Proxies for Tableau Server](#).

Changed in 2019.1

Version 2019.1 includes the following changes you should know about before upgrading.

- Install to non-default locations (RHEL-like distributions only)
- High-resolution thumbnails
- Project card images

Install to non-default locations (RHEL-like distributions only)

Beginning with version 2019.1, on RHEL-like distributions, you can install Tableau Server to a location other than the default `/opt`. This is not supported on Ubuntu due to limitations of

Tableau Server on Linux Administrator Guide

the deb package. For details, see install instructions in [Install and Initialize TSM](#) or upgrade instructions in [Upgrading from 2018.1 and Later \(Linux\)](#).

High-resolution thumbnails

In version 2019.1 and later, workbooks and views use higher resolution thumbnails. When you upgrade to Tableau Server version 2019.1 or later, the Backgrounder runs the [Enqueue Thumbnails Upgrade](#) job and the [Upgrade Thumbnails](#) job, which refreshes thumbnails for any views that have not been edited or published since Tableau version 2018.2 and converts them from low resolution thumbnail images (192 x192 pixels) to high resolution thumbnail images (300 x 300 pixels), as shown in the figure below.



When the [Upgrade Thumbnails](#) job refreshes the thumbnail images, the modified date of the view is changed to match the thumbnail update date. The [Upgrade Thumbnails](#) job does not change the modified date of workbooks when it updates the contained view's thumbnails.

Thumbnails may appear blurry until the [Upgrade Thumbnails](#) job is complete. Depending on the number of workbooks you have, this could a few minutes to a few hours. You may also notice that thumbnail images may appear in low resolution for up to a week after upgrading to Tableau Server 2019.1. For more information, see [Upgrade Thumbnails Job](#).

Project card images

In version 2019.1 and later, project card images will not appear on the project card unless they are added as the last item in the project description and encapsulated with !

(exclamation marks). For more information, see [Add a Project Image](#).

Changed in 2018.3

Version 2018.3 includes some changes you should know about before upgrading.

- Connecting to secondary trusted Active Directory domains may result in error
- Floating worksheet captions become transparent
- Changes to some Data Engine tsm configuration set options
- SSL Off-loading and SAML

Connecting to secondary trusted Active Directory domains may result in error

A secondary domain is one that Tableau Server connects to for user synchronization, but is a domain where Tableau Server is not installed. Tableau Server will attempt to connect to secondary domains for user and group synchronization. But in some cases, Tableau Server will be unable to connect to the secondary domain, which will result in the error, "Domain not in whitelist (errorCode=101015)."

Setting the `wgserver.domain.whitelist` option in TSM is required by a fix for the security vulnerability, [\[Important\] ADV-2020-003: Tableau Server Forced Authentication](#). As of February 2020, the fix for this vulnerability is included in all latest versions and maintenance releases of Tableau Server. See `wgserver.domain.whitelist`.

Floating worksheet captions become transparent

To make a floating caption visible again, content authors will need to change the caption background to a color.

Changes to some Data Engine tsm configuration set options

Two new configuration tsm set configuration options have been added: `hyper.hard_concurrent_query_thread_limit` and `hyper.soft_concurrent_query_thread_limit`

The new options replace the `hyper.num_job_worker_threads` and `hyper.num_task_worker_threads` options available in Tableau Server versions 2018.3 and earlier. The `hyper.num_job_worker_threads` and `hyper.num_task_worker_threads` have been deprecated and will no longer work in 2019.1 and later.

SSL Off-loading and SAML

Beginning with version 2018.3, Tableau Server validates SAML response message returned from the IdP. If your organization terminates SSL connections from the IdP at a proxy server before sending the authentication request to Tableau Server, then users may be unable to login with SAML after you upgrade from 2018.2.

In this scenario, SSL is "off-loaded" at the proxy server, which means the https request is terminated at the proxy server and then forwarded to Tableau Server over http. Since SSL is off-loaded at the proxy, Tableau Server will validate with the protocol that it receives (http), but the IdP response is formatted with https, so validation will fail unless your proxy server includes the X-Forwarded-Proto header set to `https`. See [Configuring Proxies for Tableau Server](#).

Changed in 2018.2

Version 2018.2 includes some significant changes you should know about before upgrading. The following sections summarize the significant changes to Tableau Server 2018.2 and provide links for additional information.

- [Updates to Tableau Server Apache Logs](#)
- [The Read Only site role is now deprecated](#)
- [The `tabcmd initialuser` command prompts for password](#)
- [New ports added](#)

Moving the repository in a two-node cluster not supported

Updates to Tableau Server Apache Logs

Apache log files stored in `/logs/httpd` now contain records with several formatting improvements, including the following:

- Server names are shown according to the `UseCanonicalName` setting, which makes it easier to distinguish between nodes in a multi-node server deployment.
- Date and time are now tracked in a single date-and-time string that can be readily recognized as a single value by a variety of data analysis tools, including Tableau Desktop.
- Time zones are now easier to parse
 - On Linux, the numerical offset from UTC is provided.
 - On Windows, the name of the timezone is enclosed in quotes

Date, time and time zone formatting changes are illustrated by an example log entry from Tableau Server version 2018.1 and earlier releases:

```
:::1 - - 2018-05-09 08:51:48.872 GMT Daylight Time 80 "POST
/vizql/w/Regional/v/Obesity/sessions/A971A92AF77A4F4A9813065BDCEB24D6-0:0/-
commands/tabsrv/render-tooltip-server HTTP/1.1" "-" 200 445
"593" 135021 WvKolPH4UcM5KrzUblI8EwAAA@E
```

For purposes of comparison, here is a log entry from Tableau Server version 2018.2:

```
10.210.24.3 127.0.0.1 - 2018-05-09T16:07:58.120 "GMT Daylight
Time" 80 "POST /vizql/w/Super-
store/v/Overview/-
bootstrapSession/sessions/185CCDC854A44765BB0298E93B403879-0:3
HTTP/1.1" "-" 200 136026 "784" 2370951 WvMOzgKIh-
fzh9kFWO@ow2gAAA1Y
```

The Read Only site role is now deprecated

The **Read Only** site role that was introduced in Tableau Server version 2018.1 is now deprecated. Any users assigned to the **Read Only** site role in version 2018.1 are reassigned to the **Viewer** site role after an upgrade to version 2018.2. To learn more about site roles, see Set Users' Site Roles.

The `tabcmd initialuser` command prompts for password

In previous versions, the `tabcmd initialuser` command required a `--password` parameter to set the initial administrator password. In the `tabcmd` included with this version of Tableau Server, you can leave the `--password` parameter out of the command to be prompted in the shell instead. See `initialuser`.

New ports added

We've added new service ports in 2018.2.

Automatic port assignment

For installations where automatic port assignment is enabled (this is recommended and the default) there should be no impact to upgrading. The new ports will be automatically selected on upgrade.

Manual port assignment

If you have disabled automatic port assignment, *you will need to manually specify the new ports in your configuration before upgrading*. For more information about manually assigning ports, see Manual port assignment.

Ports added in 2018.2

Port names	Description
You must define each of these for every node if you disable dynamic port mapping.	
hyper.port	Data engine primary port.

Port names	Description
hyper.connection.port	Data engine connection port.
licenseservice.vendor_daemon.port	License service vendor daemon port.
samlservice.port	SAML service port.
tabadminagent.columbo.port	Administration Agent service discovery port
tabadminagent.filetransfer.port	TSM Agent file transfer port.
vizportal_0.microservice.extensions.port	Application server extensions port.
vizportal_0.monolith_grpc.port	Application server GRPC port.

Changed in 2018.1

Version 2018.1 includes some changes you should know about before upgrading. The following sections summarize the significant changes to Tableau Server 2018.1 and provide links for additional information.

- Removed tabadmin command options
- Updated tsm commands
- Updated upgrade-tsm script

Removed tabadmin command options

The previously deprecated `--license` and `--administrator` options are no longer available when using the `tabcmd createsiteusers`, `tabcmd createusers` and

Tableau Server on Linux Administrator Guide

`tabcmd syncgroup` commands. These are breaking changes for any scripts that still use the `--license` or `--administrator` options. To update your scripts, use the `--role` option with these commands to specify the license levels or site roles. For details, see `tabcmd` Commands.

Updated tsm commands

As of version 2018.1 we have added or expanded the following tsm commands:

- `tsm maintenace cleanup`—The `tsm` maintenance cleanup command was added in version 10.5.1. For version 2018.1 it has been enhanced with additional options to give you more flexibility. For details, see `tsm` maintenance cleanup.
- `tsm reset` —Added in version 2018.1, this command allows you to reset the initial administrative user. For details, see `tsm` reset.

Updated upgrade-tsm script

As of version 2018.1 the `upgrade-tsm` script you run when you upgrade from a previous version now requires you to include the name of a TSM administrator when you run it. For example:

```
sudo /opt/tableau/tableau_server/packages/scripts.<version_code>/upgrade-tsm -u <tsm_system_admin> -p <tsm_system_admin_password> --accepteula
```

where `tsm_system_admin` is a user with administrative permissions on the initial node computer.

Continue to Server Upgrade - Gather Configuration Details.

Server Upgrade - Gather Configuration Details

If you are running an in-place upgrade (you're not updating hardware as part of your upgrade), then nearly all the configuration data is preserved. Strictly-speaking, you do not need to gather all of the configuration information in this case. However, we recommend gathering the information as detailed in this topic. In the worse case scenario, should

upgrade fail, then you will have a record of all configuration information should you need to restore. In any case, you can use the configuration details you collect to verify the upgrade when it's complete.

Take screen shots

A relatively quick way to capture the basics of your configuration is to take screen shots of the TSM web interface pages and the Tableau Server admin area.

Click through all visible pages and take screen shots:

- Sign in to Tableau Services Manager Web UI
- Sign in to the Tableau Server Admin Area

Record object counts

When you are in Tableau Server admin area, count and record the following in each site:

- Projects, workbooks, views, data sources
- Users and groups

Record firewall configuration

If you have configured a local firewall for Tableau Server on Linux, then it's a good idea to copy the configuration for your records.

Our setup documentation describes how to use [Firewalld](#) to configure the firewall on single and multi-node deployments of Tableau Server running on RHEL/CentOS distributions. See [Configure Local Firewall](#).

Run the following command to retrieve the firewall configuration:

```
sudo firewall-cmd --list-all
```

Verify TSM Controller certificate expiry

Verify the certificate for the TSM Controller is still valid.

To verify your TSM Controller SSL certificate's expiration date:

Tableau Server on Linux Administrator Guide

1. Open a terminal session.
2. Type the following commands to display the dates when the certificate is valid:

```
openssl s_client -connect <tsm_servername>:8850  
  
echo | openssl s_client -connect <tsm_servername>:8850  
2>/dev/null | openssl x509 -noout -dates
```

3. If the certificate is expired, [open a case](#) with our Support team, and they can provide guidance.

Gather asset files

Many of the supporting files (certificates, IdP metadata, logos, etc) that you upload to Tableau Server are not accessible with TSM after you upload them. Specifically, files that are uploaded and managed by the Client File Service are renamed and obfuscated before they are distributed across the deployment. This process also parametrizes the file attributes that are required by Tableau services. As a result, files are not mapped to a single file location on the file system for the following files. If you have uploaded any of the following files, be sure to have copies of them saved off of the computer that is running Tableau Server:

- SAML certificate file
- SAML key file
- SAML IdP metadata file
- The custom certificate installed by `tsm security custom-cert add`
- OpenID.static.file
- Kerberos.keytab file
- LDAP Kerberos keytab file
- LDAP Kerberos conf file
- Mutual SSL certificate file
- Mutual SSL revocation file
- Customization header logo file
- Customization sign-in logo file
- Customization compact logo file

Gather custom configuration information

Some configuration information is not displayed in the TSM or Tableau Server web pages. This section includes configuration details that you may need to gather depending on how you've customized your Tableau deployment.

Secure SMTP

If you have configured TLS for Tableau Server, then you will need to record the TLS-related configurations, which are not included in the **Email Server** configuration of the TSM Web UI.

To gather the TLS-related configurations, you must run `tsm configuration get` with the following key values:

- `svcmonitor.notification.smtp.ssl_enabled`
- `svcmonitor.notification.smtp.ssl_required`
- `svcmonitor.notification.smtp.ssl_check_server_identity`
- `svcmonitor.notification.smtp.ssl_trust_all_hosts`
- `svcmonitor.notification.smtp.ssl_ciphers`
- `svcmonitor.notification.smtp.ssl_versions`

For example, to retrieve the list of ciphers that are configured for SMTP TLS, run the following command:

```
tsm configuration get -k svcmonitor.notification.smtp.ssl_ciphers
```

You can find more information about each of the keys above in the TSM CLI section of *Configure SMTP Setup*.

Analytics extensions

If you have configured analytics extensions (formerly referred to as "external services"), you will need to record your configuration information. Upgrading to Tableau Server 2020.2 or later will remove all configuration for this feature.

Tableau Server on Linux Administrator Guide

To retrieve the analytics extensions configuration from Tableau Server versions 2019.1 through 2020.1, run the following command:

```
tsm security vizql-extsvc-ssl list
```

To retrieve the password that is stored for the analytics extensions connection (if any), run the following tsm command:

```
tsm configuration get -k vizqlserver.rserve.password
```

You should also have a copy of the certificate for analytics extensions if you've configured SSL.

External Repository

This applies only if you are using the External Repository configuration with Tableau Server. If you are not sure if this applies to you, see [Tableau Server External Repository](#).

Tableau Server 2021.2.x requires a major version change in PostgreSQL version for the repository, PostgreSQL version 12 or later. If your repository is local and installed on Tableau Server, the Tableau Server upgrade process will automatically upgrade the PostgreSQL database and there are no specific steps that you need to take. If however, you have configured the repository external to Tableau Server, there are some steps you must take prior to the upgrade as described below:

1. Create a new instance of PostgreSQL DB. For more information, see:
 1. [Create a PostgreSQL DB Instance on AWS Relational Database Service \(RDS\)](#)
 2. [Create a Azure Database PostgreSQL Instance on Azure](#)
2. Create a configuration file and download the SSL certificate file for the new instance that you created in Step 1. For more information, see [Upgrade Tableau Server with External Repository for a New Major Version of PostgreSQL](#).

During upgrade, you will need to point Tableau Server to the new instance using the configuration file. The upgrade process will migrate the content from your current external repository to your new instance. For more details, see [Upgrade Tableau Server with External Repository for a New Major Version of PostgreSQL](#).

Port customization

If you have changed the dynamic port range or have configured ports manually for Tableau Server, record the changes you have made.

Run the following command:

```
tsm topology list-ports
```

For more information see [Tableau Services Manager Ports](#).

Continue to [.Server Upgrade - Verify Licensing Status](#)

Server Upgrade - Verify Licensing Status

Prior to upgrading Tableau Server, review the product keys that are currently installed and compare them to the product keys, maintenance expiration, and license expiration end dates that are listed in the Tableau Customer Portal.

It is important that all expiration and end dates associate with product keys are up-to-date before you upgrade.

Step 1: View license expiration information.

Run the following command to view all licenses that are installed on your Tableau Server installation:

```
tsm licenses list
```

The `tsm licenses list` command returns all licenses, each with a number of fields. For upgrade, you need to make note of two fields:

- For each subscription license, make note of the date in the LIC EXP field. LIC EXP displays the date that the license expires and Tableau Server will stop working.
- For each legacy perpetual key, make note of the MAINT EXP field. MAINT EXP displays the date that the maintenance contract for the Tableau Server deployment expires.

Step 2: Compare license expiration dates from TSM to your licenses in [Tableau Customer Portal](#).

Step 3: If required, update licenses.

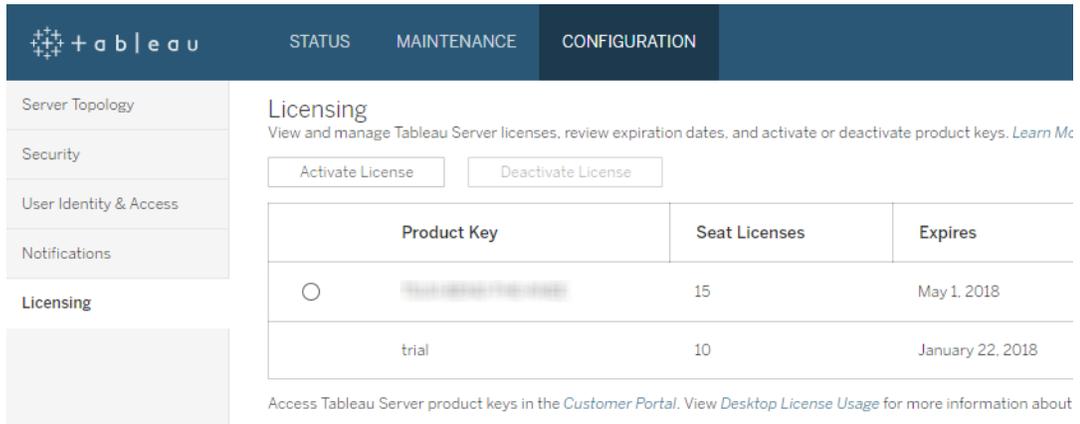
If your TSM maintenance date is not current or the expiration date will occur soon or is not the same as the date listed in the Tableau Customer Portal, refresh the license.

To refresh the license:

1. Open TSM in a browser:

`http://<tsm-computer-name>:8850`

2. Click **Configuration** and **Licensing** and click **Refresh All**:



If this does not update the maintenance or expiration end date but the Tableau Customer Portal does show more current end dates, please contact [Tableau Technical Support](#).

Reactivating the product key will be part of the upgrade process.

Continue to Server Upgrade - Verify Accounts.

Server Upgrade - Verify Accounts

Before you upgrade Tableau Server, verify that the account you will use for running Tableau

Server Setup has `sudo` access.

If your organization uses an external identity store (LDAP or Active Directory), then it's a best practice to have the credentials or the keytab file for the account that is used to bind with LDAP.

Continue to Server Upgrade - Back Up Tableau Server.

Server Upgrade - Back Up Tableau Server

We recommend that you make a backup of your installation of Tableau Server before beginning the upgrade process. This provides data that you'll need to set up a test version of the upgraded environment. It also lets you recover if the upgrade process fails.

Notes:

- We recommend you disable subscriptions and scheduling in your production environment immediately before taking the backup, and reenabling them after the backup is complete. Doing this will help avoid having your users receive duplicate subscriptions and email messages when you restore your backup in your test environment.
- The full backup can take a while if you have a large installation or a lot of extracts.
- Any changes made between the time you took the backup and the time you do the upgrade are lost because they aren't included in the backup.
- Remove Unneeded Files.

To back up server configuration data, use the `tsm settings` command. When you use the `tsm maintenance backup` command, the current date is appended to the backup file:

```
tsm maintenance backup -f <backup_file> -d
```

For more information, see `tsm maintenance backup`.

Continue to Server Upgrade - Download Setup.

Server Upgrade - Download Setup

You must download and copy the Tableau Server Setup program to the computer running Tableau Server. If you are running a multi-node deployment of Tableau Server, copy the Setup program to each node in the cluster.

To get the latest version of Tableau Server software, go to the [Customer Portal](#). When you purchase Tableau, you get a user name and password to sign in to the Customer Portal.

Important: Always download and install the latest maintenance release for the version you are upgrading to.

Continue to [How Tableau Server Upgrade Works](#).

How Tableau Server Upgrade Works

When upgrading from Tableau Server 2018.2 or later, you do not uninstall the previous version. The upgrade process is designed to install a new version side-by-side with your existing version, and then switch from the earlier version to the later one.

When doing a TSM-to-TSM upgrade (version 2018.2.x to later version), you must leave your existing version in place and running until just before you upgrade using the upgrade script. If you uninstall your existing version before upgrading, you will not be able to upgrade. For your end-users, this has the impact of reducing downtime because you install your new version while the existing version is running. The only time the server is not running is during the period the upgrade script is actually upgrading to the new version.

After installing the new version and upgrading, you can uninstall the previous version when you choose to do so. You can leave the older version installed indefinitely, though one reason you might choose to uninstall it is to free up disk space used by files specific to the older version. To understand the difference between uninstalling Tableau Server and completely removing all aspects of Tableau Server, see [Uninstall Tableau Server](#).

Upgrading Tableau Server requires a stop and start the server as part of the upgrade process. During this stop/restart Tableau Server is unavailable.

What's Changed - Things to Know Before You Upgrade

Beginning with version 2020.2, this topic has been merged with What's New in Tableau Server, to provide a single location for new and changed features and behavior in Tableau Server. See What's New and Changed in Tableau Server for both new features and for changes introduced in this and earlier versions.

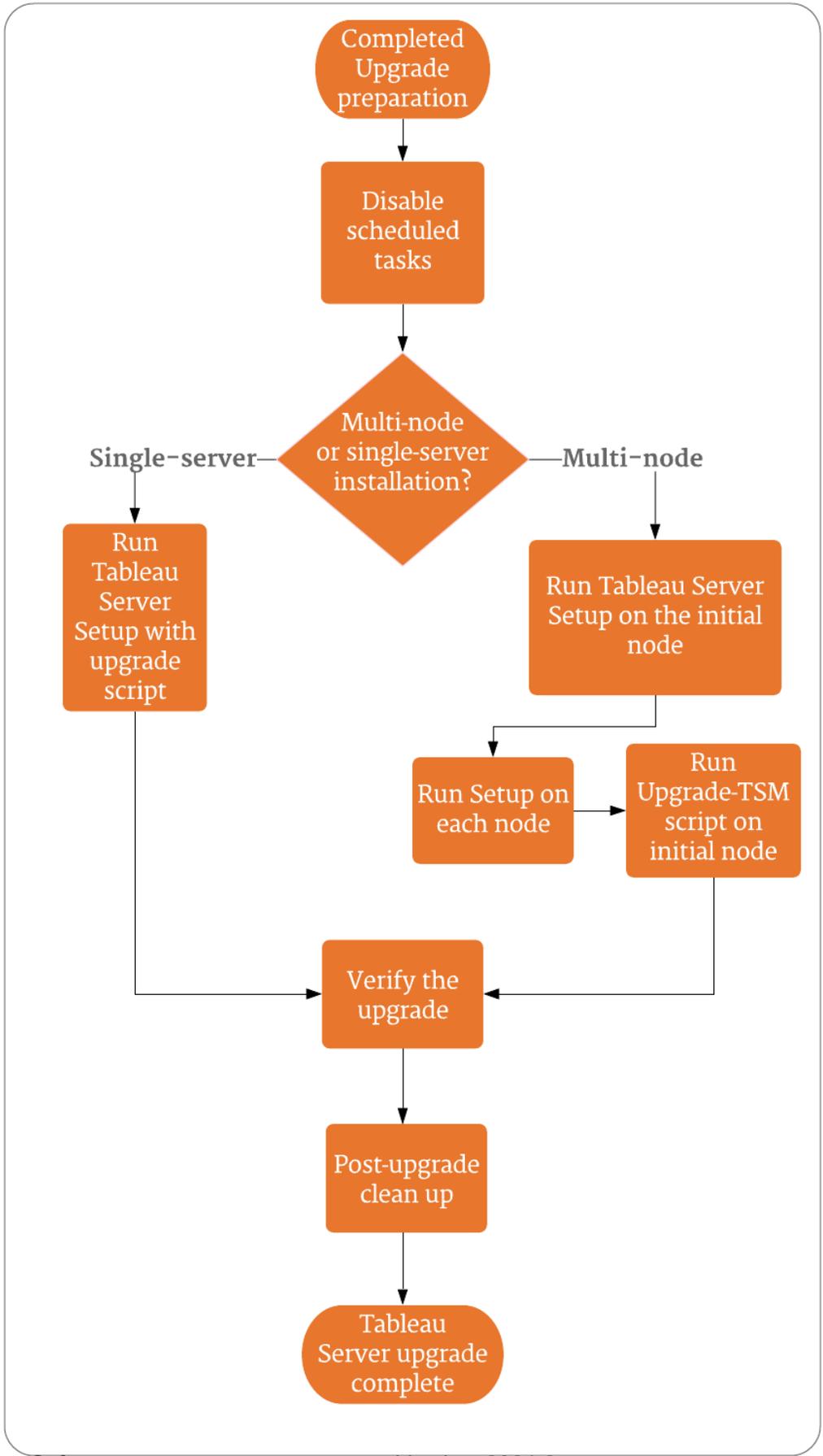
Upgrading from 2018.1 and Later (Linux)

Follow the flow chart and the linked topics to perform an in-place upgrade of Tableau Server.

If you are migrating Tableau Server to new hardware as part of your upgrade, see [Migrate to New Hardware](#).

This topic is part of the Tableau Server in-place upgrade scenario for upgrading from Tableau Server on Linux 2018.1 or later.

Upgrading Tableau Server on Linux from version 10.5 requires special steps that are documented separately. For instructions for how to upgrade from 10.5, see [Upgrade Tableau Server on Linux from 10.5](#).



Support and services to help with Tableau Server upgrades

Tableau Server upgrades require planning and testing. Upgrades are generally done during times when users are not on Tableau Server. If you plan an upgrade over a weekend and anticipate needing Tableau Technical Support, see the [Technical Support Programs](#) information on the Tableau web site. Availability depends on the level of support you have. If you would like Tableau to assist with the planning and upgrading of your Tableau Server installation, see our [Tableau Global Services Server Upgrade](#) page for details.

Server Upgrade - Disable Scheduled Tasks

You must disable scheduled tasks before you perform the upgrade.

Run the following procedure to disable all scheduled extract refreshes, flows, and subscriptions.

1. Sign in to the Tableau Server Admin Area as the Tableau Server administrator.
2. Go to the server-wide Schedules page:
 - On a Tableau Server where only a single (Default) site exists, click **Schedules** in the left pane.
 - On a Tableau Server with more than one site, click **Manage all sites** in the drop-down menu on the top of the left pane, and then click **Schedules**.
3. On the Schedules page, click **Select All**.
4. On the **Actions** menu, select **Disable**, and then click **Disable** in the resulting prompt.

Continue to: [Single-Server Upgrade -- Run Setup](#), or [Multi-node Upgrade -- Run Setup](#)

Single-Server Upgrade -- Run Setup

This topic is part of the Tableau Server in-place upgrade scenario for upgrading from Tableau Server on Linux 2018.1 or later.

Tableau Server on Linux Administrator Guide

Upgrading Tableau Server on Linux from version 10.5 requires special steps that are documented separately. For instructions for how to upgrade from 10.5, see Upgrade Tableau Server on Linux from 10.5.

Run Setup

Follow these steps to upgrade a single-node installation of Tableau Server version 2018.2 or later.

1. Log on as a user with sudo access to the computer you are upgrading.
2. Navigate to the directory where you copied the `.rpm` or `.deb` Tableau Server package.
3. Use the package manager to install the Tableau Server package.

Do not install to a location using a symbolic link or to a directory on a Network File System (NFS) volume.

- On RHEL-like distributions, including CentOS, you have the option to install Tableau to a non-default location.
 - **Default location**—To install to the default location (`/opt/tableau/tableau_server`), run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: 2019-2-5):

```
sudo yum update

sudo yum install tableau-server-<version>.x86_64.rpm
```
 - **Non-default location**—To install to a non-default location, you must use `rpm -i`. You will also need to install all dependent packages. See the note below.

Run the following command:

```
sudo rpm -i --prefix /preferred/install/path
tableau-server.rpm
```

Note: When you use `yum` to install Tableau Server, all dependent packages are automatically downloaded and installed. This is the preferred method for installing Tableau. If you want to install to a non-default location, or your organization does not allow you to use `yum` and you must install using `rpm -i`, you must also install all dependent packages separately. For information about installing dependent packages, see [Installing Tableau Server on an Air-Gapped Computer Running Linux](#).

- On Ubuntu, run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: 2019-2-5):

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get -y install gdebi-core
sudo gdebi -n tableau-server-<version>_amd64.deb
```

4. Run the upgrade script.

If Tableau Server is not stopped when you run the script, the script will let you know, and will offer to stop the server. You can also choose to stop the server before running the script using the `tsm stop` command. Tableau Server must be stopped to complete the upgrade.

The options you need to include depend on the version you are upgrading to:

- Version 2019.3 or later:

```
sudo /opt/tableau/tableau_server-
/packages/scripts.<version_code>/upgrade-tsm --
accepteula
```

where `<version_code>` is the long form of new version you are upgrading to, for example `scripts.20183.18.1128.2033`.

Starting with version 2019.3.0, when you upgrade from 2019.2.x or later, the script runs using the account you are logged in with. If you are prompted, enter your password. For more information, see [What's Changed - Things to Know Before You Upgrade](#). You can specify a different user with administrative permissions using the `-u` option and specifying a user with administrative permissions on the computer where the initial node is installed. You will be prompted for the password for the administrative user.

- Version 2018.1 through version 2019.2.x:

```
sudo /opt/tableau/tableau_server-  
/packages/scripts.<version_code>/upgrade-tsm -u <sys-  
tem_admin> --accepteula
```

where `<version_code>` is the long form of new version you are upgrading to, for example `scripts.20183.18.1128.2033`, and `<system_admin>` is a user with administrative permissions on the computer where the initial node is installed. You will be prompted for the password for the administrative user.

The `-u` option was added as of 2018.1. For more information, see [What's Changed - Things to Know Before You Upgrade](#).

To see all the options available for the `upgrade-tsm` script, use the `-h` option. For example:

```
sudo /opt/tableau/tableau_server/packages/scripts.<version_  
code>/upgrade-tsm -h
```

5. After the upgrade is completed, exit the terminal session and log in again. This ensures that your session will be using the updated TSM version.
6. Start Tableau Server:

```
tsm start
```

Continue to Verify Tableau Server Upgrade.

Multi-node Upgrade -- Run Setup

This topic is part of the Tableau Server in-place upgrade scenario for upgrading from Tableau Server on Linux 2018.1 or later.

Upgrading Tableau Server on Linux from version 10.5 requires special steps that are documented separately. For instructions for how to upgrade from 10.5, see Upgrade Tableau Server on Linux from 10.5.

Run Setup

1. Log on as a user with sudo access to the initial node in the cluster.
2. Navigate to the directory where you copied the `.rpm` or `.deb` Tableau Server package.
3. Use the package manager to install the Tableau Server package.

Do not install to a location using a symbolic link or to a directory on a Network File System (NFS) volume.

- On RHEL-like distributions, including CentOS, you have the option to install Tableau to a non-default location.

- **Default location**—To install to the default location (`/opt/tableau/tableau_server`), run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: 2019-2-5):

```
sudo yum update
```

```
sudo yum install tableau-server-<version>.x86_64.rpm
```

- **Non-default location**—To install to a non-default location, you must use `rpm -i`. You will also need to install all dependent packages. See the note below.

Run the following command:

```
sudo rpm -i --prefix /preferred/install/path  
tableau-server.rpm
```

Note: When you use `yum` to install Tableau Server, all dependent packages are automatically downloaded and installed. This is the preferred method for installing Tableau. If you want to install to a non-default location, or your organization does not allow you to use `yum` and you must install using `rpm -i`, you must also install all dependent packages separately. For information about installing dependent packages, see [Installing Tableau Server on an Air-Gapped Computer Running Linux](#).

- On Ubuntu, run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: 2019-2-5):

```
sudo apt-get update  
sudo apt-get upgrade  
sudo apt-get -y install gdebi-core  
sudo gdebi -n tableau-server-<version>_amd64.deb
```

Continue to Multi-node Upgrade -- Run Setup on Each Node.

Multi-node Upgrade -- Run Setup on Each Node

This topic is part of the Tableau Server in-place upgrade scenario for upgrading from Tableau Server on Linux 2018.1 or later.

Upgrading Tableau Server on Linux from version 10.5 requires special steps that are documented separately. For instructions for how to upgrade from 10.5, see Upgrade Tableau Server on Linux from 10.5.

Run Setup

On each additional node, navigate to the folder where you copied the Tableau Server Setup program and run the Setup program as you did on the initial node. This will install the new version of Tableau Server on each node, side-by-side with your existing, running version.

Continue to Multi-node Upgrade -- Run Upgrade Script.

Multi-node Upgrade -- Run Upgrade Script

This topic is part of the Tableau Server in-place upgrade scenario for upgrading from Tableau Server on Linux 2018.1 or later.

Upgrading Tableau Server on Linux from version 10.5 requires special steps that are documented separately. For instructions for how to upgrade from 10.5, see Upgrade Tableau Server on Linux from 10.5.

Run Upgrade script

1. After you have installed the new version on *every node in the cluster*, run the upgrade script on the initial node.

If Tableau Server is not stopped when you run the script, the script will let you know, and will offer to stop the server. You can also choose to stop the server before running the script using the `tsm stop` command. Tableau Server must be stopped to complete the upgrade.

The options you need to include depend on the version you are upgrading to:

Tableau Server on Linux Administrator Guide

- Version 2019.3 or later:

```
sudo /opt/tableau/tableau_server-  
/packages/scripts.<version_code>/upgrade-tsm --  
accepteula
```

where `<version_code>` is the long form of new version you are upgrading to, for example `scripts.20183.18.1128.2033`.

Starting with version 2019.3.0, when you upgrade from 2019.2.x or later, the script runs using the account you are logged in with. If you are prompted, enter your password. For more information, see [What's Changed - Things to Know Before You Upgrade](#). You can specify a different user with administrative permissions using the `-u` option and specifying a user with administrative permissions on the computer where the initial node is installed. You will be prompted for the password for the administrative user.

- Version 2018.1 through version 2019.2.x:

```
sudo /opt/tableau/tableau_server-  
/packages/scripts.<version_code>/upgrade-tsm -u <sys-  
tem_admin> --accepteula
```

where `<version_code>` is the long form of new version you are upgrading to, for example `scripts.20183.18.1128.2033`, and `<system_admin>` is a user with administrative permissions on the computer where the initial node is installed. You will be prompted for the password for the administrative user.

The `-u` option was added as of 2018.1. For more information, see [What's Changed - Things to Know Before You Upgrade](#).

To see all the options available for the `upgrade-tsm` script, use the `-h` option. For example:

```
sudo /opt/tableau/tableau_server-
/packages/scripts.<version_code>/upgrade-tsm -h
```

2. After the upgrade is completed, exit the terminal session and log in again. This ensures that your session will be using the updated TSM version.
3. Start Tableau Server:

```
tsm start
```

Continue to Verify Tableau Server Upgrade.

Verify Tableau Server Upgrade

Work through the sections of this topic to verify if server upgrade was successful.

Verify Tableau Service processes

Start your verification process by signing into TSM with the command line and verifying the status of Tableau Server processes.

1. Open a command prompt with an account that is a member of the `tsmadmin` group.

For more information on using the `tsm` command line, see [Using the tsm CLI](#).

2. Run the following command to view the status of all the Tableau Server processes:

```
tsm status -v.
```

What if I get an access denied error when I attempt to run TSM commands?

The account that you use to configure the rest of the installation must be a member of the `tsmadmin` group that was created during initialization. To view the user accounts in the `tsmadmin` group, run the following command:

```
grep tsmadmin /etc/group
```

If the user account is not in the group, run the following command to add the user to the `tsmadmin` group:

Tableau Server on Linux Administrator Guide

```
sudo usermod -G tsmadmin -a <username>
```

Verify TSM global settings

As part of the preparation for upgrade, you should have taken screen shots of the TSM web UI settings. Sign in to Tableau Services Manager Web UI and compare the settings with the screen shots.

Enable subscriptions and scheduling

1. Sign in to the Tableau Server Admin Area as the Tableau Server administrator.
2. Go to the server-wide Schedules page:
 - On a Tableau Server where only a single (Default) site exists, click **Schedules** in the left pane.
 - On a Tableau Server with more than one site, click **Manage all sites** in the drop-down menu on the top of the left pane, and then click **Schedules**.
3. On the Schedules page, click **Select All**.
4. On the **Actions** menu, select **Enable**, and then click **Enable** in the resulting prompt.

Verify user access

With a user account, sign in to Tableau Server with Tableau Desktop and a browser.

If Tableau Server is available from outside your organization, verify connectivity from the internet.

If your organization supports mobile connectivity to Tableau Server, sign in to server with a mobile device.

View published workbooks

With a user account, sign in to Tableau Server and browse published workbooks. Open workbooks to view them in a browser

Verify publishing workbooks and data sources

Using an account with a Creator license, verify web authoring functionality to connect to data, and to create and publish workbooks and data sources:

- [Connect to Data on the Web](#)
- [Publish a Data Source on the Web](#)
- [Upload Workbooks to a Tableau Site](#)

Verify Tableau Prep Builder

If your organization is running Tableau Prep Builder, verify that you can connect to data, open a flow, and other Prep tasks. See [Tableau Prep: Connect to Data](#).

Verify count of Tableau objects

As part of upgrade preparation, you should have noted how many projects, workbooks, views, data sources, users, and groups are in your deployment. Verify that all objects still appear.

Sign in to the Tableau Server Admin Area as the Tableau Server administrator.

- Verify count of projects, workbooks, views, data sources
- Verify count of users and groups

Verify API functionality

If your organization has deployed or developed content with Tableau APIs, verify that these are operational.

REST API

Verify REST API access and authentication by running the sample cURL from [Get Started Tutorial: Introduction and Set Up](#):

```
curl "https://MY-SERVER/api/3.8/auth/signin" -X POST -d
@signin.xml
```

Content of signin.xml:

Tableau Server on Linux Administrator Guide

```
<tsRequest>
  <personalAccessTokenName="MY_PAT_NAME"
    <credentials name="username" password="password" >
      <site contentUrl="MarketingSite" />
    </credentials>
  </tsRequest>
```

Example response:

```
<tsResponse version-and-namespace-settings>
  <credentials token="12ab34cd56ef78ab90cd12ef34ab56cd">
    <site id="9a8b7c6d-5e4f-3a2b-1c0d-9e8f7a6b5c4d"
      contentUrl="MarketingSite"/>
  </credentials>
</tsResponse>
```

Compatibility testing

For other developed features, perform compatibility testing to verify that the upgraded version of Tableau Server operates as expected with your existing solutions:

- Dashboard Extensions
- JavaScript API for embedded Tableau
- Connectors (Web Data Connector, Tacos)

Go to Post Upgrade Cleanup.

Post Upgrade Cleanup

As discussed in *How Tableau Server Upgrade Works*, the Tableau Server upgrade process installs a new version side-by-side with your existing version. Now that upgrade is complete and verified, you can remove the older version of Tableau Server to free up disk space. This is an optional step.

Uninstall previous version

Use this procedure to free up disk space by uninstalling packages for previous Tableau

Server versions after you have upgraded to a newer version of Tableau Server.

1. Look at the `environment.bash` file to confirm which version of Tableau Server is currently in use. At a command prompt, type:

```
grep TABLEAU_SERVER_DATA_DIR_VERSION /etc/-
opt/tableau/tableau_server/environment.bash
```

2. Determine which versions of the Tableau Server package are installed on your computer.

- On RHEL-like distributions, including CentOS, run the following command:

```
yum list installed tableau-server"*"
```

- On Ubuntu, run the following command:

```
apt list --installed tableau-server"*"
```

3. Remove the Tableau Server package with your package manager.

- On RHEL-like distributions, including CentOS, run the following command:

```
sudo yum remove tableau-server-<version>.x86_64
```

- On Ubuntu, run the following commands:

```
sudo apt-get purge tableau-server-<version>
```

Upgrade Tableau Server on Linux from 10.5

Important: Beginning with version 2020.4.0, if you are running version 10.5 of Tableau Server on Linux, you cannot upgrade directly to the latest version. You must upgrade to a version between 2018.1 and 2020.3 before upgrading to 2020.4 or later. Support for version 10.5 ended in July 2020 so direct upgrades to version 2020.4 or later are not supported. For information about supported versions, see [the Tableau web site](#).

When you upgrade Tableau Server on Linux from version 10.5, you need to take several unique steps to complete the upgrade. These are necessary because of a change made after version 10.5.0 released, related to sudo privileges. For more information, see [System User, sudo Privileges, and systemd](#). You only need to do these extra steps once, during the upgrade to 2018.1 or later. This topic describes how to upgrade from version 10.5.0 or 10.5.x (10.5.1 or later) to version 2018.1 or later.

If you attempt to upgrade from 10.5.0 or 10.5.x without following these instructions, warnings are displayed and the upgrade is canceled. You will not break your existing Tableau Server installation, but you cannot continue the upgrade.

To identify the version of your installation, see [View Server Version](#).

Follow these steps to upgrade from 10.5.0:

1. Upgrade to 10.5.x—If you are running version 10.5.0, you must first upgrade to 10.5.x (10.5.1 or higher) by installing 10.5.x and running the `upgrade-tsm` script in the 10.5.x scripts directory on your initial node.
2. Install 2018.x or later, up to 2020.3.x—With 10.5.x installed and running as expected, install 2018.x or later, but do not upgrade to this version yet.
3. Run TSM commands—Use TSM to stop the server and run three additional commands.
4. Migrate 10.5.x to single user—Run the migration script in the new version (2018.x or later) scripts directory. Do this on every node in your cluster.
5. Upgrade to 2018.x or later, up to 2020.3.x— Upgrade Tableau Server by running the `upgrade-tsm` script from the new version scripts directory on your initial node.
6. Upgrading from 2018.2 and Later— After upgrading to a version between 2018.x and 2020.3.x, you can upgrade Tableau Server to 2020.4 or later by following the instructions for here: [Upgrading from 2018.1 and Later \(Linux\)](#).

Upgrade to 10.5.x

If you are running version 10.5.0 of Tableau Server on Linux, the first step you need to take is to upgrade to a later version of 10.5. Beginning with version 10.5.1 changes were made that are needed in order to upgrade to 2018.1 or later. (If you are already on a version of 10.5 that is higher than 10.5.0, you can skip to the Install 2018.x or later, up to 2020.3.x step.)

To upgrade from 10.5.0 to a later version of 10.5:

1. On each node in your cluster:
 - a. Copy the Tableau Server version 10.5.x .rpm or .deb package to location accessible from the computer you are upgrading.

If you are upgrading a distributed deployment of Tableau Server, copy the .rpm or .deb package to each node in the cluster or to a location accessible from each node.

- b. Log on as a user with sudo access to the computer you are upgrading.
- c. Navigate to the directory where you copied the .rpm or .deb Tableau Server package.
- d. Use the package manager to install the Tableau Server package.

- On RHEL-like distributions, including CentOS, run the following command:

```
sudo yum install tableau-server-<version>.x86_64.rpm
```

- On Ubuntu, run the following commands:

```
sudo gdebi -n tableau-server-<version>_amd64.deb
```

2. Stop Tableau Server. If you are upgrading a cluster, do this after you have installed

the new package on every node in your cluster.

```
tsm stop
```

3. With Tableau Server stopped, run the following command on your initial node. Do not run this command on any additional nodes:

```
sudo /opt/tableau/tableau_server/packages/scripts.<version_
code>/upgrade-tsm --accepteula
```

where `<version>` is the 10.5.x version you are upgrading to.

To see all the options available for the `upgrade-tsm` script, use the `-h` option. For example:

```
upgrade-tsm -h
```

4. After the upgrade is completed, ensure your session is using the updated TSM version by doing one of the following:

- Use the source command:

```
source /etc/profile.d/tableau_server.sh
```

- Exit the terminal session on the initial node and log in again.

5. Start Tableau Server:

```
tsm start
```

Install 2018.x or later, up to 2020.3.x

Install the new Tableau Server package but do not upgrade to this version yet. Before you do so, you need to run several commands and a migration script. You can install the new version package without stopping the server. When you install the new package you are copying the software to your computer but not changing anything about the currently running version.

To install the new version package, on each node in your cluster:

1. Copy the Tableau Server .rpm or .deb package to location accessible from the computer you are upgrading.

If you are upgrading a distributed deployment of Tableau Server, then copy the .rpm or .deb package to each node in the cluster or to a location accessible from each node.

2. Log on as a user with sudo access to the computer you are upgrading.
3. Navigate to the directory where you copied the .rpm or .deb Tableau Server package.
4. Use the package manager to install the Tableau Server package.

- On RHEL-like distributions, including CentOS, run the following command:

```
sudo yum install tableau-server-<version>.x86_64.rpm
```

- On Ubuntu, run the following commands:

```
sudo gdebi -n tableau-server-<version>_amd64.deb
```

Run TSM commands

Using version 10.5.x of Tableau Server that is installed and running:

1. Stop the server:

```
tsm stop
```

2. Run these three commands:

```
tsm configuration set -k service.linux.privileged_user -v  
'tableau'
```

```
tsm configuration set -k install.username -v 'tableau'
```

Tableau Server on Linux Administrator Guide

```
tsm pending-changes apply
```

where 'tableau' is the user name you specified with the `initialize-tsm --unprivileged-user` option when you first installed 10.5.x. If you did not specify a user, the default is 'tableau'.

Migrate 10.5.x to single user

Run this script from the 2018.x or later scripts directory:

```
sudo /opt/tableau/tableau_server/packages/scripts.<version_code>/migrate-to-single-user
```

where <version_code> is the long form of your new version number.

Important: If you have a multi-node installation, you must run this script on every node in your cluster.

At this point Tableau Server is running 10.5.x but configured to work with a single user. This is an interim stage. You should complete the upgrade to version 2018.x or later before using Tableau.

Upgrade to 2018.x or later, up to 2020.3.x

After completing the above steps:

1. With Tableau Server stopped, run the upgrade script on the initial node. Do not run the script on any additional nodes. The options you need to include depend on the version you are upgrading to:

- Version 2019.3 or later:

```
sudo /opt/tableau/tableau_server-  
/packages/scripts.<version_code>/upgrade-tsm --  
accepteula
```

where `<version_code>` is the long form of new version you are upgrading to, for example `scripts.20183.18.1128.2033`.

Starting with version 2019.3.0, when you upgrade from 2019.2.x or later, the script runs using the account you are logged in with. If you are prompted, enter your password. For more information, see [What's Changed - Things to Know Before You Upgrade](#). You can specify a different user with administrative permissions using the `-u` option and specifying a user with administrative permissions on the computer where the initial node is installed. You will be prompted for the password for the administrative user.

- Version 2018.1 through version 2019.2.x:

```
sudo /opt/tableau/tableau_server-
/packages/scripts.<version_code>/upgrade-tsm -u <system_admin> --accepteula
```

where `<version_code>` is the long form of new version you are upgrading to, for example `scripts.20183.18.1128.2033`, and `<system_admin>` is a user with administrative permissions on the computer where the initial node is installed. You will be prompted for the password for the administrative user.

The `-u` option was added as of 2018.1. For more information, see [What's Changed - Things to Know Before You Upgrade](#).

To see all the options available for the `upgrade-tsm` script, use the `-h` option. For example:

```
sudo /opt/tableau/tableau_server-
/packages/scripts.<version_code>/upgrade-tsm -h
```

2. After the upgrade is completed, ensure your session is using the updated TSM version by doing one of the following:

Tableau Server on Linux Administrator Guide

- Use the source command:

```
source /etc/profile.d/tableau_server.sh
```

- Exit the terminal session on the initial node and log in again.

3. Start Tableau Server:

```
tsm start
```

When desired, you can remove Tableau Server on Linux version 10.5 from your server. Unlike most other programs that run on Linux, previous Tableau Server versions are not automatically removed as part of a successful upgrade. To learn more, see [Remove Tableau Server from Your Computer](#).

Related topics

- [Common Tableau Server Upgrade Issues](#)

Test the Upgrade

The best way to learn what impact a Tableau Server upgrade will have to your current environment is to test it. Knowing how an upgrade will affect your users and your server helps you plan and communicate before the actual upgrade, ensuring that your users will not be caught by surprise.

If you have a Tableau Server test environment this is a great place to test out the upgrade.

We recommend the following sequence for testing a Tableau Server upgrade:

1. [Prepare a test environment](#)
2. [Upgrade the test environment](#)
3. [Confirm that existing functionality works](#)
4. [Performance and user acceptance testing](#)

5. [Test new features](#)
6. [Communicate about the upgrade](#)

Prepare a test environment

To start, create a test environment that mirrors your production environment as closely as possible. The closer your test environment is to the actual environment you will be upgrading, the more accurate a representation you will have of how the upgrade will impact you. This includes identical or similar hardware and operating systems, as well as the same authentication options and network access.

When you've got a test computer or virtual machine ready, follow these steps for creating a test environment.

1. On the existing production environment, create a backup of Tableau Server using the `tsm maintenance backup` command.

For more information, see [Create a pre-upgrade backup](#).

2. On your test environment, install a copy of the same version of Tableau Server as you have in your production environment.

Note: You can download the setup program for your current version from the [Alternate Downloads Site](#).

3. Restore your existing database data using the `tsm maintenance restore` command.

For more information, see [Restore from a backup](#).

4. Manually replicate your existing Tableau Server configuration.

You need to manually configure certain aspects of your environment because when you restore the Tableau database it doesn't include configuration details and customizations.

Upgrade the test environment

Follow the appropriate steps for upgrading the test environment, based on your environment:

- Upgrading from 2018.1 and Later (Linux)
- Upgrade Tableau Server on Linux from 10.5

Confirm that everything works as expected

After you have the new version of Tableau Server installed and configured in your test environment, you are ready to test. You should test basic functionality, along with any special aspects of server that your organization relies on. For example, if there are key subscriptions that your organization relies on, make sure that you test those.

These are some areas of testing to consider:

- **Server processes.** Sign in to Tableau Server as a server administrator, and then open the Server Status page to confirm that all services and processes are running as expected (including on all additional nodes if this is a distributed installation).
- **User access.** Confirm that Tableau Server users can sign in. Test your normal user sign in process. Have some of your users participate in the testing to make sure they are able to sign in as expected, and that they can get to the same content that they have access to in your production environment.
- **Publishing workbooks and data sources.** Have users publish workbooks and data sources from Tableau Desktop to make sure this goes as you expect.
- **Viewing published workbooks.** Have users who are familiar with the content try to view published workbooks to make sure they appear as expected. Test views

embedded in web pages (for example, in SharePoint pages).

- **Subscriptions and extract refreshes.** Manually run some extract refreshes to confirm that they complete successfully. Run some key scheduled extract refreshes to confirm that they complete as expected.
- **Permissions.** Confirm that permissions are still set as expected for users and content.
- **Command-line utilities and APIs.** If applicable, test the command line utilities (tsm and tabcmd) and programmatic access via APIs.

Performance and user acceptance testing

Use tools like Tabjolt, Replayer, and Scout to do performance and user acceptance testing on your test environment. For more information about these and other performance testing tools, see [Performance Monitoring Tools](#).

Test new features

Take a look at the new features that come with the version you are upgrading to, and at any features that were added between the version you currently have and the new version. Think about how to help your users understand the benefits of the features that apply to your environment.

For more information on new features, see [What's New in the Tableau Server Help](#).

Communicate about the upgrade

The best way to make an upgrade go smoothly is by letting your organization know ahead of time about the upgrade and how it might impact them. If you've had users help test, take advantage of their experience by having them help communicate the changes they saw while testing. You can also provide user access to the test environment if there are key people who should see the upgraded version before the actual upgrade.

Troubleshoot Tableau Server Install and Upgrade

Follow the suggestions in this topic to resolve common issues with Tableau Server. For additional troubleshooting steps based on process status viewed on the Status page, see [Troubleshoot Server Processes](#).

General Troubleshooting Steps

Many Tableau Server issues can be addressed with some basic steps:

1. Make sure there is enough disk space on each computer running Tableau Server. Limited disk space can cause a failure to install, a failure to upgrade, or problems running Tableau Server.
2. Restart Tableau Server. Issues related to processes not fully started can be resolved by restarting Tableau Server in a controlled way. To restart Tableau Server, use the `tsm restart` command. This will stop all the processes associated with Tableau Server and then restart them.
3. Reindex Tableau Server. Issues related to indexing can be resolved by reindexing Tableau Server. To reindex Tableau Server, use the `tsm maintenance reindex-search` command. For more information, see [Reindexing Tableau Server Search & Browse](#) below.
4. Restart the computer on which Tableau Server is running. Some issues, such as those related to data source connectivity, can be resolved by restarting the server computer.

Common Tableau Server Install Issues

Installation logs location

The install log, `app-install.log`, is located in `/var/opt/tableau/tableau_`

server/logs.

The upgrade log, `app-upgrade.log`, is located in `/var/opt/tableau/tableau_server/logs`.

Unable to log into TSM or Tableau Server (sign in screen redisplay after entering credentials)

Using Internet Explorer or Edge, if you enter your credentials into the TSM or Tableau Server sign-in screen and the page redisplay without signing you in, verify that the hostname or domain in your URL does not include an underscore (`_`). If the hostname or domain of the Tableau Server computer includes an underscore (`_`) Internet Explorer or Edge browsers will not set a cookie, so the page will redisplay without signing you in. To work around this, use "localhost" or the IP address of the computer in the URL. For example:

`https://localhost:8850`.

For more information, see the [Tableau Knowledge Base](#).

Multiple install attempts fail

If you attempt to install Tableau Server and the install fails, any subsequent installation attempts are likely to fail unless you run the `tableau-server-obliterate` script to clean Tableau off the computer.

A failed install attempt can leave the computer in a state that causes subsequent attempts to also fail with errors that don't seem directly related to a previous install attempt. One possible error is:

```
Enabling and starting all services
+ services=(appzookeeper* tabadmincontroller* tabsvc*
licenseservice* fnplicenseservice* tabadminagent* cli-
entfileservice*)
+ systemctl_user enable appzookeeper_0.service
'tabadmincontroller*' 'tabsvc*' 'licenseservice*' fnpli-
censeservice_0.service 'tabadminagent*' 'clientfileservice*'
++ id -ru a_tabadminpoc
```

Tableau Server on Linux Administrator Guide

```
+ local unprivileged_uid=222954
+ su -l a_tabadminpoc -c 'XDG_RUNTIME_DIR=/run/user/222954 systemctl --user enable appzookeeper_0.service tabadmincontroller* tabsvc* licenseservice* fnplicenseservice_0.service tabadminagent* clientfileservice*'
Failed to execute operation: No such file or directory
```

To fix this problem, run the `tableau-server-obliterate` script to clean up any left over remnants of the previous install attempt and then restart the computer. For more information, see [Running the tableau-server-obliterate script](#).

Important: If you created a backup of Tableau (<file>.tsbak) you want to keep (for example, to restore to your new installation), copy that file to a safe location on another computer to guarantee it is not removed when you clean up your Tableau computer.

Install fails due to hardware requirements

Tableau Server cannot install if the computer you are installing on does not meet the minimum hardware requirements. The requirements apply to all computers on which you are installing Tableau Server. For details on minimum hardware requirements, see [Minimum Hardware Requirements and Recommendations for Tableau Server](#).

Install or upgrade fails due to CPU requirements

Beginning in version 2020.4.0 Tableau Server requires CPUs that support SSE4.2 and POPCNT instruction sets. You cannot install or upgrade Tableau Server 2020.4.0 or later on computers that have CPUs which do not support these instruction sets.

You may see this error message when installing a new installation, or in preparation for upgrading an existing installation:

```
Your computer's processor doesn't meet the minimum requirements that Tableau requires to install the software. If you are using a VM, make sure Processor compatibility mode is off.
```

The SSE4.2 and POPCNT instruction sets have been common for more than 10 years and most newer CPUs support them, but if you get an error related to processor minimum requirements when attempting to install or upgrade Tableau Server on a Virtual Machine (VM), Processor compatibility mode may be enabled on the VM. To successfully install or upgrade Tableau on a VM, make sure the Processor compatibility mode is turned off.

Common Tableau Server Upgrade Issues

Maps do not display or display incompletely after upgrading

Beginning with Tableau version 2019.2, the internet access requirements changed for maps. If you are upgrading from version 2019.1.x or earlier to version 2019.2.x or later, and maps are not displaying as expected, confirm that your environment is configured to allow access on port 443 to `mapsconfig.tableau.com` and `api.mapbox.com`.

In version 2019.1.x or earlier, access was necessary to `maps.tableausoftware.com`.

For more details on internet access requirements, see [Communicating with the Internet](#).

Upgrade script error: "Tableau Server Version change validation failed."

When upgrading, if you run the `upgrade-tsm` script from the `scripts.<version_code>` directory for the earlier version, the upgrade will fail with an error:

```
Tableau Server Version change validation failed.
Tableau Server <version> is already installed.
```

If you get this error, change to the `scripts.<version_code>` directory for the version you just installed and run the script from there.

Upgrade logs location

The upgrade log, `app-upgrade.log`, is located in `/var/opt/tableau/tableau_server/logs`.

Tableau Server on Linux Administrator Guide

Upgrade multi-node, initializing additional node fails with "Enter your credentials again" error

If you attempt to initialize an additional node when upgrading Tableau Server and see this error:

```
Enter your credentials again. The credentials you enter must provide administrative access to the computer where you generated the configuration file.
```

this is an indication that the node is unable to connect to or communicate with the initial node.

This can happen for multiple reasons:

- The credentials you entered are not valid or you mistyped them. The credentials must be for a user who has administrative permissions on the computer where Tableau Server was first installed. You do not need to use the credentials of the user who created the bootstrap file but doing so will ensure you are using valid credentials.
- The local firewall of the computer you are trying to add is not allowing communication to the initial node. For more information, see [Local firewall configuration](#).

Upgrading fails due to lack of disk space

If there is not enough disk space for the Tableau Server Setup program to run and do the upgrade, the installation will fail. The amount of disk space required will depend on the size of your repository database and the number and size of your extracts.

To free up disk space:

1. Create a log archive snapshot using the `tsm maintenance ziplogs` command.

After you create the ziplogs file, save it to a safe location that is not part of your Tableau Server installation.

2. Clean up unnecessary files using the `tsm maintenance cleanup` command. For more information, see [Remove Unneeded Files](#).

Upgrade fails on RebuildSearchIndex job

Beginning with version 2020.1.x, the final step in an upgrade is to rebuild the search index. At this point all services have been upgraded, so if this job fails, you can manually reset the search server by running the `tsm maintenance reset-searchserver` command. You do not need to obliterate and start over.

The error will be:

```
An error occurred while rebuilding search index.
```

To reset the search server :

1. On the initial node, open a terminal session.

This must be a new terminal session because the upgrade script updates system environment for the new version.

2. Rebuild the search index using the `tsm maintenance reset-searchserver` command.

Upgrade fails on 2020.4.0 or later

Beginning with version 2020.4.0, the Checkpoint Upgrade feature allows you to retry a failed upgrade. In general, this is most useful for experienced server administrators and IT professionals who are comfortable with Tableau Server log files and are willing to search through them. But the feature can help in all failed upgrades because it allows you to rerun the `upgrade-tsm` script, and the script is run from the last successful step, saving time. For those with experience, it may be possible to identify problems like disk space problems, or permissions issues, correct them, and rerun the upgrade.

If you are upgrading to version 2020.4.0 or later and the upgrade fails, the following steps may help you to complete the upgrade:

- Rerun the `upgrade-tsm` script. Upgrade failures are sometimes a result of timeouts during the upgrade process, and rerunning the script can allow the upgrade to get beyond intermittent or occasional timing issues. This is also a step that is safe

to do, and easy. Rerunning the script will do no harm, and at worst, the upgrade will fail again at the same point, but without needing to go through any previous steps.

The script is located in the `\scripts` directory:

```
opt/tableau/tableau_server/packages/scripts.<version_
code>/upgrade-tsm
```

If your Tableau Server upgrade isn't successful when you rerun the `upgrade-tsm` script, and you are comfortable with Tableau Server logs, you can take these additional troubleshooting steps:

- Look at the output of the script in the command window. Useful error messages may help you identify the cause of the upgrade failure and give you some ideas for how to correct the issue.
- Look in the `app-upgrade.log` file. Any errors that are displayed at the command line will also appear in the `app-upgrade.log` file, often with more details.
- Look in the `tabadmincontroller.log` file. Upgrade problems that aren't easily identifiable in the above two instances are likely the result of an issue in a job. The `tabadmincontroller.log` file may have more information that helps you diagnose the issue.

Note: For information about log file locations, see [Tableau Server Logs and Log File Locations](#).

Common Settings Import Issues

Import of settings file causes "not present on any node" validation error due to missing services

If you are upgrading by installing a new version of Tableau Server and importing a settings file from an earlier version, you may encounter topology validation errors when running the `tsm settings import` command.

This can happen when you export a settings file from an older version of Tableau Server and import it into a new version, and new services have been added to Tableau between the two versions.

Errors will be similar to this (the specific service may be different):

```
>tsm settings import -f 20183-export.json
```

```
Pending topology set.
```

```
There are 1 topology validation errors/warnings.
```

```
Service 'elasticserver' is not present on any node in the cluster.
```

```
Service: Elastic Server
```

To resolve this issue, add any missing services to Tableau Server:

1. For any service that generated a validation error, add the service with an instance count of 1.

For example, if the Elastic Server is not present in the cluster, set the process instance count to 1 using the service name that appears in the first line of the validation error message:

```
tsm topology set-process -n node1 -pr elasticserver -c 1
```

Repeat this step for each service that results in an error.

2. When you have no more warnings or errors, apply the pending changes:

```
tsm pending-changes apply
```

Your settings should be imported successfully.

Import of settings file causes "configuration value you specified does not match" error

If you are installing a new version of Tableau Server and import a settings file from an earlier version, you may encounter configuration validation errors when running the `tsm`

Tableau Server on Linux Administrator Guide

`settings import` command. These can occur when a settings file includes a configuration value that has since been removed from Tableau.

The error will look similar to this (the configuration key may be different):

```
>tsm settings import -f 20183-export.json
Configuration error: At least one configuration value you specified does not match a known configuration key. This applies to the following keys: '[features.TsmConfigFileService]'
Use this parameter to override unknown key error: --force-keys
```

To resolve this issue, edit the settings file you are importing to remove the reference to the configuration key or keys in the error:

1. Copy the JSON settings file and save the copy for backup.
2. Open the JSON settings file in a plain text editor.
3. Locate and delete the entire line that includes the key. In this example, `features.TsmConfigFileService`:

```
"configKeys" : {
  "config.version" : 19,
  "tabadmincontroller.port" : "8850",
  "endpoints.enabled" : false,
  "endpoints.health.enabled" : true,
  "features.TsmConfigFileService" : true,
  "tableau_projects.language" : "en",
```

The above is an example of a small section of an exported settings file and is not intended to represent the entire contents of the file.

4. Save the settings file and import it again.

You may encounter additional errors related to topology validation. For information about solving those errors, see [Import of settings file causes "not present on any node" validation error due to missing services above](#).

"You cannot directly modify instances of the Coordination Service" error

This error can occur in two situations:

- When you import a Tableau Server settings file into an installation that has a different Coordination Service topology than the settings file does
- When you attempt to configure the Coordination Service using the `tsm topology set-process` command

If you see this error after importing a settings file:

The Tableau Server settings file has a different Coordination Service topology than the target server does. This can happen if you are upgrading Tableau Server by installing a new version and importing a settings file from an earlier version. If you have not explicitly deployed a Coordination Service ensemble on the target server, it has a single instance of Coordination Service, on the initial node.

To correct this error you can take either correct the mismatch from the command line, or by editing the settings import file. You can also discard all pending changes, deploy the Coordination Service on the target computer to match the settings in the import file, and reimport the settings file.

To correct the mismatch from the command line, for each node that generates an error, use the `tsm topology set-process` command to revert the instance count of Coordination Service.

1. Run the `tsm pending-changes list` command. The output shows you which nodes have changes.
2. Find the node or nodes where the Coordination Service count is changed.

For example, if the settings file had a Coordination Service instance on node2, but the target system did not have any Coordination Service instance on that node, the count for node 2 would show as changed from 0 to 1 by the import of the settings file:

Tableau Server on Linux Administrator Guide

```
C:\Windows\system32>tsm pending-changes list
Configuration
There are no pending configuration changes.
Topology
node2:
    Coordination Service
        New Instance Count:1
        Old Instance Count:0
```

3. Use the `tsm topology set-process` command to set the count back to the "Old Instance" value.

For the example above:

```
tsm topology set-process -n node2 -c 0 -pr "Coordination
Service"
```

4. Once you have reset any Coordination Service instance count that was changed, apply pending changes:

```
tsm pending-changes apply
```

If you see the error when setting the process count for Coordination Service manually:

This error can also occur if you attempt to update the Coordination Service directly, using the `tsm topology set-process` command instead of the `tsm topology` commands for managing the Coordination Service. If you tried this:

1. Use the `tsm pending-changes discard` command to discard the pending changes.
2. Use the correct commands for configuring the Coordination Service. For more information, see [Deploy a Coordination Service Ensemble](#) .

Starting Tableau Server

Tableau Server cannot determine if it fully started

In some instances Tableau Server may report that it could not determine if all components started properly on startup. A message displays: "Unable to determine if all components of the service started properly."

If you see this message after starting, verify that Tableau Server is running as expected by using a `tsm status -v` command.

If the status shows as running ("Status: RUNNING"), then the server successfully started and you can ignore the message. If the status is DEGRADED or STOPPED, see "Tableau Server doesn't start" in the next section.

Tableau Server doesn't start

If Tableau Server does not start or is running in a degraded state, run the `tsm restart` command from a command prompt. This will shut down any processes that are running, and restart Tableau Server.

Reindexing Tableau Server Search & Browse

Problems that can be solved by rebuilding Search & Browse index

Symptoms of an index that needs to be rebuilt include:

- A blank list of sites when a user attempts to log in
- A blank list of projects when a user tries to select a project
- Missing content (workbooks, views, dashboards)
- Unexpected or inaccurate alerts (for example, an "refresh failed" alert on a workbook that does not include an extract)

If you see any of these behaviors, reset and rebuild the Search & Browse index using the `tsm maintenance reset-searchserver` command.

Activating Tableau Server

Tableau Server license activation fails

In some instances Tableau Server license activation may fail. Error messages can range from a very generic one:

- `An error has occurred`

To more specific messages:

- `Function flxActCommonLicSpcPopulateFromTS returned error 50030, 71521,`
- `No license found for 'Tableau Server'`

To resolve this issue, try these solutions in the order listed:

Confirm you can access the licensing server

The Tableau licensing service was moved to a new data center on October 6, 2018. This means any environments that required special configuration (static IP safe listing for example) to access `licensing.tableau.com` or `licensing.tableau.com` will need to be updated before you can activate, refresh, or deactivate a Tableau product key.

To test access, type the URL and the port of the licensing server in a browser:

```
https://licensing.tableau.com:443
```

and:

```
https://atr.licensing.tableau.com/_status/healthz
```

If you are able to access the server, a "Test success" message displays for the first server, and an "OK" message displays for the second.

Tableau Server needs to make a connection to the following internet locations for licensing purposes:

- `licensing.tableau.com:443`
- `atr.licensing.tableau.com:443`
- `s.ss2.us`
- `ocsp.rootg2.amazontrust.com`
- `ocsp.rootca1.amazontrust.com`
- `ocsp.sca1b.amazontrust.com`
- `crt.sca1b.amazontrust.com`
- `crt.rootca1.amazontrust.com`
- `ocsp.sca0a.amazontrust.com`
- `crt.sca0a.amazontrust.com`
- `ocsp.sca1a.amazontrust.com`
- `crt.sca1a.amazontrust.com`
- `ocsp.sca2a.amazontrust.com`
- `crt.sca2a.amazontrust.com`
- `ocsp.sca3a.amazontrust.com`
- `crt.sca3a.amazontrust.com`
- `ocsp.sca4a.amazontrust.com`
- `crt.sca4a.amazontrust.com`
- `crl.rootca1.amazontrust.com`
- `crl.rootg2.amazontrust.com`

Tableau Server on Linux Administrator Guide

- `cr1.sca1b.amazontrust.com`

Requests to the above domains may be on port 80 or 443. Port 80 is used for certificate validation (revocation, certificate chain, etc). Port 443 is used for SSL connections.

Verify the date and time

Verify the date and time on the initial Tableau Server computer is correct. If the clock is set to a time and date earlier than the current date, Tableau Server cannot be activated.

Force the product key to be read again

1. On the initial Tableau Server computer, log on as a user with sudo access.
2. Change to the Tableau Server bin directory. By default this is:

```
/opt/tableau/tableau_server/packages/bin.<version_code>/
```

3. Type the following commands:

```
tsm stop
```

```
./lmreread
```

```
tsm start
```

Send the contents of trusted storage to Tableau Support

If FlexNet Licensing Services is installed and running but you're still seeing an error, there might be a problem with the Tableau product key information. To resolve this issue, complete the following steps to create a file of the key information located in trusted storage.

1. On the initial Tableau Server computer, log on as a user with sudo access.
2. Type the following command:

```
serveractutil -view > <machine_name>-LicResults.txt
```

This creates the `<machine_name>-LicResults.txt` file in your current directory. If you don't have write permissions for that location and see an error, change to a location where you do have permission to create a file and run the command again.

3. Contact Tableau Support (<http://www.tableau.com/support/request>) and include the `<machine_name>-LicResults.txt` file that you created.

tabcmd Installation Problems

Installing tabcmd separately

tabcmd is automatically installed on the initial Tableau Server node when you install Tableau Server, but if you want to run it on another computer, you need to download and install tabcmd separately. For details, see [Install tabcmd](#).

Problems installing tabcmd on Linux

tabcmd requires Java 8 (also known as Java 1.8) to run properly. On RHEL-like systems, this will be installed as a dependency when installing tabcmd. On Ubuntu systems, you need to install Java 8 (1.8) separately if it is not already installed.

As of July 2022, Debian distributions are no longer supported. For more information, see [this Tableau Community post](#).

Java is not installed

If you see errors similar to this when installing tabcmd, confirm that Java 8 is installed on your Linux computer:

```
Cannot find 'java' in your PATH. Install 'java' and make sure it is in your PATH to continue.
```

Incorrect version of Java is installed

If you see errors similar to these, confirm that Java 8 is installed:

Tableau Server on Linux Administrator Guide

```
Exception in thread "main" java.lang.UnsupportedClassVersionError: com/tableausoftware/tabcmd/Tabcmd :  
Unsupported major.minor version 52.0
```

or.

```
*** Uncaught exception NoClassDefFoundError: javax/xml/  
bind/JAXBException  
*** See the logs for the stacktrace.
```

Uninstall Tableau Server

Do not uninstall Tableau before upgrading. For details on upgrading, see [Upgrading from 2018.1 and Later \(Linux\)](#).

You can have multiple versions of Tableau Server installed at the same time. This allows you to run most of an upgrade while an existing version is running, and reduces downtime and impact to users. Once you have upgraded, you can uninstall your previous version. Doing this frees up disk space. You do not have to uninstall the previous version.

This article explains how to uninstall previous versions, after you've upgraded to a newer version.

Uninstalling and completely removing Tableau Server

There are two primary "uninstall" scenarios that Tableau Server on Linux supports:

- **Uninstall Tableau Server:** *After you upgrade to a new version of Tableau Server you can uninstall your previous version to free up disk space. Continue reading for information about uninstalling Tableau.*
- **Remove Tableau Server:** *If you want to completely remove Tableau Server from a computer, you can use a script provided by Tableau to remove Tableau Server and all related files. This removes all data as well as server components, so should only be done if you know you want to reset the computer to a pre-Tableau state. You might*

need to do this if Technical Support recommends this step when troubleshooting an installation problem. We recommend you create a backup of your data before removing Tableau. Save the backup file to a safe location on a computer that is not part of your Tableau installation. Completely remove Tableau Server without uninstalling any version first. The script will uninstall all existing versions found on the computer. If you have already uninstalled your existing version and now want to completely remove Tableau, you can find the script to do so in a temporary location. For more details, see [Remove Tableau Server from Your Computer](#).

Uninstall a Tableau Server package

Use this procedure to free up disk space by uninstalling packages for previous Tableau Server versions after you have upgraded to a newer version of Tableau Server.

1. Look at the `environment.bash` file to confirm which version of Tableau Server is currently in use. At a command prompt, type:

```
grep TABLEAU_SERVER_DATA_DIR_VERSION /etc/
opt/tableau/tableau_server/environment.bash
```

2. Determine which versions of the Tableau Server package are installed on your computer.

- On RHEL-like distributions, including CentOS, run the following command:

```
yum list installed tableau-server"*"
```

- On Ubuntu, run the following command:

```
apt list --installed tableau-server"*"
```

3. Remove the Tableau Server package with your package manager.

- On RHEL-like distributions, including CentOS, run the following command:

```
sudo yum remove tableau-server-<version>.x86_64
```

- On Ubuntu, run the following commands:

```
sudo apt-get purge tableau-server-<version>
```

Reinstall a Tableau Server package that was accidentally uninstalled

Do *not* uninstall the package for your currently running version of Tableau Server. Doing so will make the server unusable. To completely remove Tableau Server and all its files, see [Remove Tableau Server from Your Computer](#).

When you uninstall the Tableau Server package for the current instance of Tableau Server the following operations run:

- All files under `/opt/tableau/tableau_server` are removed. These files are the unmodified installation files.
- Tableau Server services are stopped and disabled
- Service files for all Tableau Server services are persisted
- Data files are left in place

If you accidentally uninstall the package for your currently running version of Tableau Server, follow this procedure to correct the situation.

To reinstall after uninstalling the running instance of Tableau Server:

1. Reinstall the package for the version you accidentally uninstalled.
2. Run `initialize-tsm`.

Remove Tableau Server from Your Computer

Warning: The steps below *completely remove* Tableau Server on Linux, and delete users and groups created by `initialize-tsm`, all related data, and configuration information. This includes any files in `/tmp` or `/var/tmp` that are owned by users configured in `/etc/opt/tableau/tableau_server/environment.bash` as privileged and unprivileged users (by default, `tsmagent` and `tableau`). Tableau Server

licenses are also deactivated, unless you omit the `-l` option when running the command shown below and the computer is connected to the internet.

If you want to uninstall a particular Tableau Server package to free up disk space (after upgrading, for example), see [Uninstall Tableau Server](#).

As part of the regular installation of Tableau Server, a script is installed that provides you a way to completely remove Tableau and all associated files from your computer. This is something you would only do if you did not care about your Tableau data, configuration, or log files, or if you are working with Tableau Technical Support and need to reinstall Tableau Server after an installation or upgrade attempt fails. The `obliterate` script will not remove any drivers you installed separately, even those you installed to use with Tableau Server.

The `tableau-server-obliterate` script is intended for when you want to completely remove Tableau Server from your computer. You might want to do this for a couple of different reasons:

- You no longer want Tableau Server installed on the computer. Use the `tableau-server-obliterate` script to remove Tableau Server completely. If this is the case, and the computer is connected to the internet, you can include a `-l` option to deactivate the Tableau Server license.
- Troubleshooting Tableau Server installation problems—If you run into issues installing Tableau, you may need to use the `tableau-server-obliterate` script to completely remove Tableau Server from your computer before reinstalling. Doing this will clean up any older settings or states (such as the `/etc/opt/tableau/tableau_server/environment.bash` file) and allow you to reinstall on a "clean" computer. If you are doing this, you can leave off the `-l` option to preserve licensing information on the computer. When you omit the `-l` option, you will not need to activate your license when you reinstall Tableau Server.

What `tableau-server-obliterate` does

The intent of the `tableau-server-obliterate` script is to completely remove Tableau Server from your computer. This includes files, system settings and configurations, and, if you specify, licensing information.

When you run `tableau-server-obliterate`, the following steps are taken:

- Uninstall is run for all installed versions of Tableau Server (`yum erase` or `apt-get remove`).
- Most contents of the data directory is removed (by default `/var/opt/tableau/tableau_server`). Backup and log files are preserved by default. See the section below, "Preserving Tableau Server backup and log files."
- All semaphores and shared memory segments are deleted.
- All temp files owned by the "tableau" user are deleted from `/tmp` and `/var/tmp`.
- All users and groups created during install are deleted.
- `/etc/opt/tableau` is deleted.
- Trusted certificates are removed from `/etc/pki/ca-trust/source/anchors/TableauServer` and `/usr/share/ca-certificates/tableau`
- Configuration files are removed from:
 - `/etc/sysctl.d/99-tableau-server.conf`
 - `/etc/profile.d/tableau_server*`
 - `/etc/security/limits.d/99-tableau_server*`
 - `/etc/systemd/logind.conf.d/tableau_server*`
 - `/usr/share/bash-completion/completions/tsm`
 - `etc/bash_completion.d/tsm`
 - `/usr/share/bash-completion/completions/tabcmd`
 - `/etc/bash_completion.d/tabcmd`
 - `/run/tableau`
 - `/usr/lib/tmpfiles.d/tableau-server.conf`
- All server licenses (except trial licenses and those activated by the Tableau authorization-to-run (ATR) service) are deactivated if you use the `-l` option and the computer is connected to the internet. This option does not work in offline situations.

Preserving Tableau Server backup and log files

Prior to version 2020.1 of Tableau Server, running the `tableau-server-obliterate` script deleted all content from the Data directory. Beginning with version 2020.1, the default behavior of the `tableau-server-obliterate` script has changed: the script copies and saves Tableau Server backup and log files to the `logs-temp` directory. The default location for the `logs-temp` directory is at `/var/opt/tableau/logs-temp`. You can set options on the script to disable this new functionality.

To change the default behavior and to remove backup or log files, include one of the following options when running the `tableau-server-obliterate` script:

- `-k` Do not copy backups to `logs-temp` directory.
- `-g` Do not copy logs to `logs-temp` directory.
- `-a` Do not copy anything to `logs-temp` directory.

Running the `tableau-server-obliterate` script

You can completely remove Tableau Server from a computer, either preserving the licensing information, or removing the licensing information along with all other aspects of Tableau Server. You might want to preserve licensing if you are going to reinstall Tableau Server on the same computer.

An older version of `tableau-server-obliterate` may miss files from new versions of Tableau Server. Always run the `obliterate` script for the version of Tableau Server that is installed on the computer.

To completely remove Tableau Server without removing server licensing

The example script in this procedure also includes the `-a` option to remove Tableau Server backup and log files.

1. On the initial node, open a terminal session.
2. Run the `tableau-server-obliterate` script:

Tableau Server on Linux Administrator Guide

```
sudo /opt/tableau/tableau_server/packages/scripts.<version_
code>/tableau-server-obliterate -a -y -y -y
```

Note: If you have already uninstalled Tableau Server and now you want to remove it, run the copy of the `tableau-server-obliterate` script located in the `/var/tmp` directory. If the script is not in that directory, reinstall the Tableau Server package and then run the `tableau-server-obliterate` script.

If you have a multi-node (distributed) installation of Tableau Server, run the `tableau-server-obliterate` script on each node in the cluster.

3. Restart each computer you ran the `tableau-server-obliterate` script on.

To completely remove Tableau Server and licensing

The example script in this procedure also includes the `-a` option to remove Tableau Server backup and log files.

1. On the initial node, open a terminal session.
2. Deactivate any active product keys.

```
tsm licenses deactivate -k <product_key>
```

3. Run the `tableau-server-obliterate` script:

```
sudo /opt/tableau/tableau_server/packages/scripts.<version_
code>/tableau-server-obliterate -a -y -y -y -l
```

The `-l` option removes all licensing files from the computer if the computer is connected to the internet. The script first attempts to deactivate any active licenses, but it will remove all licensing information whether or not deactivation was successful. This is why we recommend you run the `tsm licenses deactivate` command before running the `obliterate` script.

Note: If you have already uninstalled Tableau Server and now you want to remove it, run the copy of the `tableau-server-obliterate` script located in the `/var/tmp` directory. If the script is not in that directory, reinstall the Tableau Server package and then run the `tableau-server-obliterate` script.

If you have a multi-node (distributed) installation of Tableau Server, run the `tableau-server-obliterate` script on each node in the cluster. You do not need to deactivate licenses on any additional nodes.

4. Restart each computer you ran the `tableau-server-obliterate` script on.

Help Output for `tableau-server-obliterate` Script

The following help content is the output when you run the following command:

```
sudo ./tableau-server-obliterate -h
```

The `./tableau-server-obliterate` script is installed to `/opt/tableau/tableau_server-
/packages/scripts.<version_code>/`.

Output

Remove Tableau Server from this computer.

```
This script will stop and remove all Tableau Services from this
computer. It also removes data and configuration files. It
leaves
licensing in place. It also preserves logs and backup files,
which
are moved to a temp directory under the Tableau data folder.
You can
force removal of these files, and licensing, using optional
parameters.
```

Tableau Server on Linux Administrator Guide

This script is destructive and not reversible. It should only be used to clean Tableau Server from a computer. For multi-node installations, you must run the script separately on each node.

This script must be run as the root user.

-y	Required. Yes, remove Tableau Server from this computer.
	Must be specified three times to confirm.
-l	Optional. Delete licensing files and data. This command
	will attempt to deactivate licenses before deleting licensing data. Internet access is required for license deactivation. Offline deactivation is not supported. To deactivate license before removing Tableau Server, run 'tsm licenses deactivate' before running this script.
-k	Optional. Do not copy backups to logs_temp directory.
-g	Optional. Do not copy logs to logs_temp directory.
-a	Optional. Do not copy anything to logs_temp directory.

Manage Individual Sites

In addition to planning your sites in Tableau, you can manage users and groups, manage projects and control content access, manage your site data, and interact with views on the web.

What is a site	591
Planning a Site	594
Site Settings Reference	597
Manage Users and Groups	608
Dashboard-based Custom Portals	672
Manage Content Access	675

What is a site

You might be used to using the term *site* to mean “a collection of connected computers,” or perhaps as the short form of “website.” In Tableau-speak, we use site to mean a collection of users, groups, and content (workbooks, data sources) that’s walled off from any other groups and content on the same instance of Tableau Server. Another way to say this is that Tableau Server supports multi-tenancy by allowing server administrators to create sites on the server for multiple sets of users and content.

All server content is published, accessed, and managed on a per-site basis. Each site has its own URL and its own set of users (although each server user can be added to multiple sites). Each site’s content (projects, workbooks, and data sources) is completely segregated from content on other sites.

If you are a server administrator on your Tableau Server deployment, you can learn more about sites, when to use them (vs. projects), and more in [Sites Overview](#), in the **Manage Server** section.

Note: This article pertains to configuring sites on Tableau Server deployments. For Tableau Online, see [Site Administrator Role and Tasks](#).

Site administrator tasks

Where the Server Administrator site role gives a user unrestricted access to the entire Tableau Server deployment, the Site Administrator site roles give a user unrestricted or minimally restricted access at the site level. The differences between Site Administrator Creator and Site Administrator Explorer are in the level of data connection and publishing access. Both site roles allow administering the site itself and managing site users. For more information, see [Set Users' Site Roles](#).

Although a server administrator can work at both the server and site levels, we make a distinction between the two levels of task. The site administrator is typically in charge of creating and maintaining the framework that enables Tableau users in the organization to publish, share, manage, and connect to data sources and workbooks. In this vein, site administrator tasks include any of the following (and both site roles allow this level of access):

- Creating project hierarchies to organize the site's data sources and workbooks.

This can include delegating project-level management to project leaders.

- Creating groups and assigning permissions that allow users to access only the content they need.

- Adding and removing users, assigning their site roles.

This is allowed by default on a site; however, a server administrator can restrict this access to the server level only.

- Managing the site's extract and subscription schedules.
- Monitoring site activity.

For more information about the distinction between server administrator and site administrator, see Administrator-level access to sites, in the **Manage Server** section.

Steps for setting up your site

The table below shows a loose sequence of steps for setting up a site, along with links to topics where you can get more information. You can complete the steps in any order that makes sense for you.

However, before you perform the steps to configure the site, we recommend spending some time with the articles in this section, learning about site authentication, site roles, projects, and permissions. Ideally you would document a plan for your projects, groups, and overall permissions strategy. Then set up a few projects and add a preliminary set of users, to test the plan and resolve issues before you add the remaining users. You can change many site settings after your users are working with the site, but try to go in with the intention of minimizing post-production changes.

Plan	To supplement the recommendations above this table, get an overview of how the site components work together in Planning a Site .
Configure access	<p>Work with the server administrator to determine how users sign in to the site, and configure the site appropriately.</p> <p>For example, if the server is configured for single sign-on using SAML, you might configure SAML authentication at the site level as well.</p>
Create projects and the permissions structure	Projects help you organize content, delegate project-level content management, and manage permissions effectively. To get started, see Use Projects to Manage Content Access .

Add users	Determine the users who can sign in to the site. See Add Users to a Site .
Get your data to Tableau Server	<p>After you create your projects and permissions structure, designate approved users for publishing and managing vetted data sources to the appropriate projects on the site.</p> <p>In some organizations, people serve in multiple Tableau roles. Site administrators commonly also are data stewards. By that, we mean they create, publish, and manage the Tableau data connections. If this is you, make sure you are assigned the Site Administrator Creator site role.</p> <p>After content is published to the site, you can maintain connection information (credentials, access tokens) and refresh schedules. For more information, see Refresh Data on a Schedule.</p>
Analyze site usage and performance	Monitor usage, performance, and other metrics. See Administrative Views .

Planning a Site

Before you add users and content to a site, we recommend that you plan the following aspects of the site.

- [Projects](#)
- [Users and groups](#)

- [Site roles and permissions](#)
- [Extract refresh schedules](#)

The subsequent sections go over these site components, assuming that you are familiar with

Note: This article and section apply only to self-managed Tableau Server deployments on-premises or in the cloud. If you use Tableau Online, see [Manage Content Access](#).

Projects

You can create projects on a site, which act as containers in which you can organize related content assets (such as data sources and workbooks). For example, you might set up a project to contain all of the certified data sources and workbooks your organization uses for mission-critical decisions. Or you might set up projects by department.

Projects are also useful for managing permissions. Once you know how your users need to access content, it's usually easier to create projects based on those the type of content, and maintain permissions at the project level.

Every site has a default project named **Default**. When you create projects, the new projects get their initial set of permissions from the default project. In effect, the default project is a template for new projects. As we explain in related articles, for most environments, we recommend that you use the Default project only as a permissions template, and not as a container for published content.

For more information, see [Use Projects to Manage Content Access](#).

Users and groups

Any user who will publish content or access published content on a site must be able to sign in to the site. If the user already has an account on the server, you'll need to add that user to the appropriate site. You can add a user to more than one site as well. If the user doesn't

already exist, you need to create a user account. Either way, make a list of the users who will need to be able to sign in to each site.

Note: The server license might restrict how many users you can have, or what level of access they can have. Check with the server administrator to make sure that you'll be able to have an account for all your users.

In general, we recommend that you create groups on the server and then add users to the groups. This helps to make permissions much easier to manage. You can assign permissions on groups, to give those permissions to all users in the group. (See the next section.)

A typical strategy is to create groups for users who use content in similar ways. For example, you might create a group named SalesWBPublishers for all the users in the Sales department who publish workbooks, and a separate group named SalesDSPublishers for people in Sales who publish data sources. Each of these sets of users needs its own set of capabilities, so it makes sense to have a group for each for these needs.

Site roles and permissions

Each user has a *site role* that determines the maximum permissions that they can have on the site. As part of your site planning, you need to decide each user's site role. A user with a site role that's too restrictive might not be able to do the work they need. By the same token, a security best practice is to limit users' capabilities to only those that they need to do their work. This is referred to as following the principle of *least privilege*.

You or a site administrator you delegate this task to must also determine the permissions a user needs to work with content. Each content asset (workbook, data source, project) supports a set of *capabilities*. For example, you can **View** or **Add Comments** to a workbook. Before a user can perform tasks on a workbook, their permissions must allow those capabilities. A recommended practice is to sketch out a mapping of permissions to users outside of Tableau before you try to set this up on the server.

Permissions determine what a user can do *within the context of the site role*. A user whose site role is **Viewer** can never publish to the site, regardless of the permissions you grant them. A user whose site role is **Creator** can publish a workbook to the site, but only if that user has permission to save and view workbooks.

Extract refresh schedules

If users publish data sources or workbooks that include extracts, you usually want to make sure that the extracts are refreshed, so that they contain the latest data. Users can manually refresh an extract, but this isn't always a good idea if the extract is large, and the refresh takes a long time. Instead, you can set up schedules for when an extract should be refreshed. Another planning task for a site administrator, therefore, is to think about when extracts should be refreshed and to work out schedules.

Site Settings Reference

As a Tableau administrator, you can customize a site for your organization using the settings below. The settings available to you depend on your site configuration and whether you use Tableau Online or Tableau Server. To view and edit site settings, you must be a Site Administrator on Tableau Online or Server Administrator on Tableau Server.

Note: To find a specific setting below, press Ctrl+F (Windows) or Command+F (macOS). Tableau Online and Tableau Server present these settings in a different order.

Accessing site settings

- (Tableau Online) From the Home page, expand the side pane, and click **Settings** at bottom.

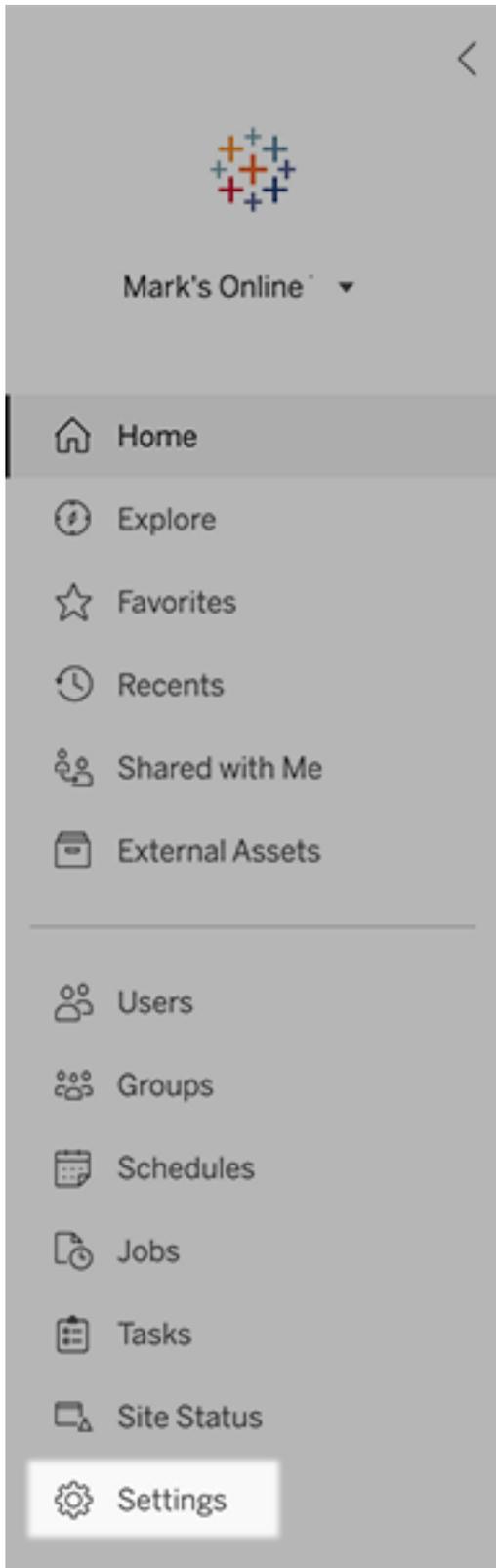
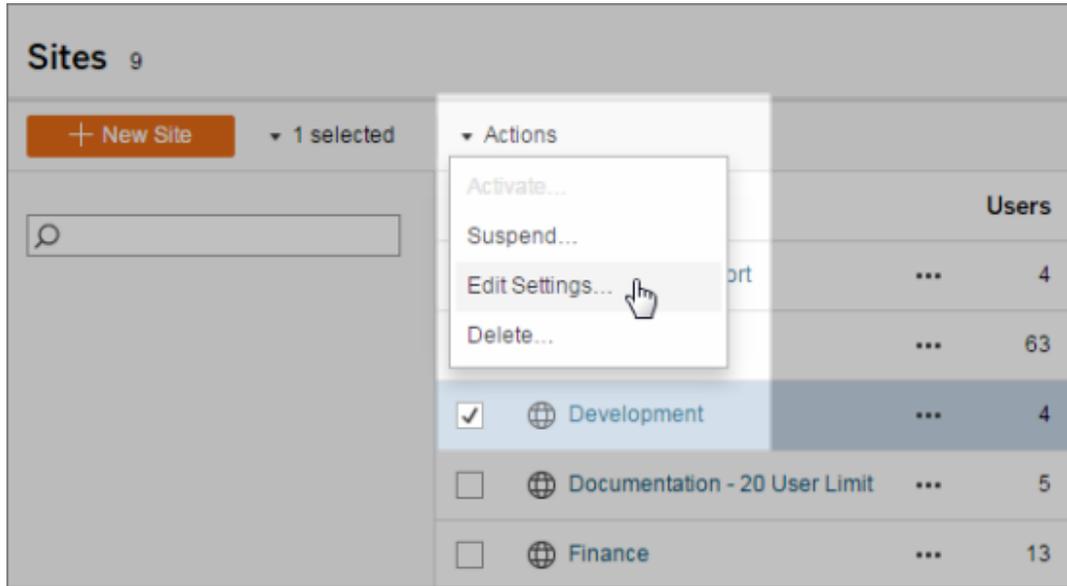


Tableau Server on Linux Administrator Guide

- (Tableau Server) If you're editing an existing site, select it on the Sites page, and then select **Edit Settings**. Or, in a single-site deployment, click **Settings** at the bottom of the side pane.



General section

Setting

Description

Site Name and ID (Tableau Server Administrators only)

Specifies the site name seen in the user interface and the ID seen in the site URL. (If you are editing the Default site, you cannot change the ID.)

You can't change the “#/site” portion of the URL (for example, http://localhost/#/site/sales). In multi-site server environments, these segments appear in the URL for sites other than the Default site.

Storage (Tableau Server Administrators only)

Select either **Server Limit** or **GB**, and for the latter enter the number of gigabytes you want as a limit for storage space for published workbooks, extracts, and other data sources.

If you set a server limit and the site exceeds it, publishers will be prevented from uploading new content until the site is under the limit

again. Server administrators can track where the site is relative to its limit using the Max Storage and Storage Used columns on the Sites page.

Revision History (Tableau Server Administrators only)	Specifies the number of previous versions of workbooks, flows, and data sources that are stored on the server.
Tableau Prep Conductor	Controls whether users with appropriate permissions can schedule and monitor flows. Tableau Prep Conductor is part of the Tableau Data Management Add-on. For more information, see About Tableau Prep Conductor .
Web Authoring	Controls whether browser-based authoring is enabled for the site. When web authoring for workbooks is disabled, users can't create or edit published workbooks from the server web environment but instead must use Tableau Desktop to republish the workbook. When web authoring for flows is disabled, users can't create or edit published flows from the server web environment but instead must use Tableau Prep Builder to re-publish the flow. For more information, see Set a Site's Web Authoring Access and Functions in Tableau Online Help.
Managing Users (Tableau Server Administrators only)	Determines whether only server administrators can add and remove users and change their site roles, or whether site administrators can too. If you allow site administrators to manage users, specify how many users they can add to the site by selecting one of the following: <ul style="list-style-type: none"> • Server Limit adds the number of available server seat licenses. For a server with core-based licensing, there is no limit. • Site Limit lets site administrators add users up to a limit you specify.

- Site Role Limit lets site administrators add users of each site role up to the license limit you specify for the site.

For more information, see [View Server Licenses](#).

Guest Access

Lets people who lack a Tableau Server account see views that have guest access permissions.

Note: If you use Tableau Server, your administrator can disable Guest Access.

Tableau Catalog

Turns off Catalog capabilities when Tableau Server or a Tableau Online site is licensed with the Data Management Add-on. For more information, see [Disable Catalog](#).

Workbook Performance after a Scheduled Refresh (Tableau Server Administrators only)

Pre-computes recently viewed workbooks with scheduled refreshes to open them faster. For more information, see [Configure Workbook Performance after a Scheduled Refresh](#).

Workbook Performance Metrics (Tableau Server Administrators only)

Lets site users collect metrics on how workbooks perform, such as how quickly they load To initiate recording, users must add a parameter to the workbook's URL. For more information, see [Create a Performance Recording](#).

Managed Keychain Clean Up

Lets site administrators manage saved credential keychains for OAuth connections on the site. For more information, see [OAuth Connections](#).

Automatically Suspend Extract Refresh Tasks

To save resources, Tableau can automatically suspend extract refresh tasks for inactive workbooks. This feature applies only to refresh schedules that run weekly or more often. For more inform-

ation, see [Automatically Suspend Extract Refreshes for Inactive Workbooks](#) in Tableau Online Help.

Email Settings (Tableau Server Administrators only)	Specifies the From address and message footer seen in automatic emails for alerts and subscriptions.
Site Invite Notification (Tableau Online only)	For sites with single-sign-on authentication, sends an invite email when new users are added to the site.
Site Logo	Specifies the image that appears with the site name.
Start Page	Controls which site page appears when users sign in. By default, the Home page appears, but you can instead open All Projects, All Workbooks, or other pages. For more information, see Set the Default Start Page in Tableau Online Help.
<p>Note: If you use Tableau Server, your administrator can override this site setting.</p>	
User Visibility	Controls what user and group names are visible to other users. For more information, see Manage User Visibility in Tableau Online Help.
Availability of Ask Data	Controls whether Ask Data lenses are enabled or disabled for data sources. Ask Data lets users query data using conversational language and automatically see visualizations. For more information, see Automatically Build Views with Ask Data in Tableau user Help.
Automatic Access to Metadata about Databases and Tables	Automatically grants users certain capabilities to external assets using derived permissions. For more information, see Turn off derived permissions in Tableau Online Help.

Tableau Server on Linux Administrator Guide

Sensitive Lineage Data	Specifies whether sensitive lineage data should be obfuscated or filtered when users don't have the appropriate permissions to related metadata. For more information, see Sensitive lineage data .
Cross-Database Joins	Determines where the join process happens when joining data from multiple sources. For more information, see Combine Tables from Different Databases in Tableau user Help.
Notifications for Extract Jobs and Flow Runs	Sends email notifications to inform data source, flow, and workbook owners when there are updates to extract jobs or scheduled refreshes. Also sends encryption failure notices if the site encryption mode is set to Enable. For more information, see in Tableau Online Help and Enable Extract Refresh Scheduling and Failure Notification in Tableau Server Help.
Extract Encryption at Rest (Tableau Server Administrators only)	Lets you encrypt .hyper extracts while they are stored on Tableau Server. Server administrators can enforce encryption of all extracts on their site or allow users to encrypt all extracts associated with particular published workbooks or data sources. For more information, see Extract Encryption at Rest .
Tableau Mobile - Enable offline favorites (Tableau Server Administrators only)	Controls whether offline previews of users' favorite views are generated for display when users access the site on Tableau Mobile. For more information, see Manage Tableau Mobile Data on Devices in the Tableau Mobile Deployment Guide.
Tableau Mobile - Enable app lock (Appears under Authentication tab for Tableau Online)	Requires a biometric method or device passcode for users to open the site on Tableau Mobile. For more information, see Enable App Lock for Added Security in the Tableau Mobile Deployment Guide.
Tableau Support Access (Tableau	Allows Tableau Support technicians access to the site to help troubleshoot support cases. By default, this feature is disabled. For

Online only)	more information, see Enable Support Access .
Sharing	Allows users to share items directly with other users. When an item is shared, the recipients get a notification and the item is added to their Shared with Me page. If this is not enabled, users can only copy a link to share. For more information, see Share Web Content in Tableau user Help.
Comments	Controls whether users can add remarks in a Comments side pane for each view and @mention other Tableau users to notify them via email. For more information, see Comment on Views in Tableau user Help.
Data-Driven Alerts	Lets users automatically receive emails when data reaches key thresholds. For more information, see Send Data-Driven Alerts in Tableau user Help.
Subscriptions	Lets site users subscribe to views and receive regular emails of them. On Tableau Server, these options are available only if you first configure subscription settings .
Recommendations for Views	Controls whether recommendations show on the site and whether the names of users who have looked at recommended items show on recommendation tooltips.
<p>Note: If you use Tableau Server, your administrator can disable Recommendations.</p>	
Request Access	Lets users send access requests to content or project owners. For more information, see Let Site Users Request Access to Content in Tableau Online Help.
Metrics Content Type	Controls whether metrics are available on the site. When enabled, users can create metrics from views and metrics appear as a content type. When disabled, metrics won't appear on the site or con-

tinue to sync; however, you can re-enable the feature to bring back previously created metrics. For more information, see "Set Up for Metrics" in [Tableau Online Help](#) or [Tableau Server Help](#).

Web Page Objects and Web Images	Controls whether these Web Page and Image objects can display target URLs. For more information, see Security for Web Page objects in Tableau user Help.
Site Time Zone for Extracts	The default time zone for extract-based data sources in a site is Coordinated Universal Time (UTC). Site administrators can set a different time zone. For more information, see Set the Site Time Zone for Extracts in Tableau Online Help.
Extract Quota Limit Notifications	Sends email alerts to all site administrators when extract refresh jobs are canceled because of extract job capacity issues.
Run Now	Controls who can run jobs manually using the Run Now option from the web, Rest API, and Tabcmd. By default, this option is selected to allow users to run jobs manually. Clear the check box if only administrators should be allowed to run jobs manually.

Note: If you use Tableau Server, your administrator can disable this site setting.

Manage Notifications	Controls how site users can receive notifications for events such as extract jobs, flow runs, or when another user shares content with them or mentions them in a comment. Notifications can be seen in their Tableau site via the notification center or sent by email. When a notification is enabled, users can configure their notification preferences on their Account Settings page.
----------------------	---

Note: If you use Tableau Server, your server administrator can disable this site setting.

OAuth Clients Registry	For a subset of connectors, you can register a custom OAuth client for the site to override an OAuth client that has been configured for the server. By registering a custom OAuth client, you enable new and existing connections to use the site-level OAuth client instead of the server-wide OAuth client. For more information, see Configure Custom OAuth .
------------------------	---

Authentication section (Tableau Online)

Setting	Description
Authentication Types	Specifies how users can sign in to the site, and how they access it after signing in the first time. Authentication verifies a user's identity. For more information, see Authentication .
Default Authentication Type for Embedded Views	Specifies how users can sign in to embedded views. By default, Tableau authentication is selected.
Manage Users	Lets you add new users to the site or change the site role and authentication method for existing users.
Automatic Provisioning and Group Synchronization (SCIM)	Allows you to manage users on the site through a third-party identity provider (IdP). When enabled, the Base URL and Secret boxes are populated with values to use in the IdP SCIM configuration. For more information, see Automate User Provisioning and Group Synchronization through an External Identity Provider .
Connected Clients	Allows Tableau clients such as Tableau Mobile, Tableau Bridge, and others to stay authenticated to the server after a user provides sign-in credentials the first time. When turned off, users are required to sign in explicitly each time they visit Tableau Online. For more information, see Access Sites from Connected Clients .
App Lock for Tableau Mobile	Requires a biometric method or device passcode for users to open this site on Tableau Mobile. For more information, see Enable App

[Lock for Added Security](#) in the Tableau Mobile Deployment Guide.

Bridge section (Tableau Online)

Setting	Description
Client Not Running Notifications	Sends email alerts to data source owners when a client appears to be disconnected from the site.
Allow Load Balancing (Pooling)	Distributes live queries and refresh jobs across all clients in a pool. For more information, see Enable pooling .

Extensions section

Setting	Description
Dashboard Extensions	Manage and control dashboard extensions. Dashboard extensions are web applications that run in custom dashboard zones and can interact with the rest of the dashboard. For more information, see "Manage Dashboard Extensions" in Tableau Online Help or Tableau Server Help .
Analytics Extensions	Enables a set of functions that your users can use to pass expressions to analytics extensions for integration with R and Python. For more information, see "Configure Connection with Analytics Extensions" in Tableau Online Help or Tableau Server Help .
Tableau Prep Extension	When authoring flows on the web, enables users to apply Einstein Discovery-powered models to their flows to bulk score predictions for their data. For more information, see Configure Einstein Discovery Integration in the Tableau Server help.

Manage Users and Groups

You can add users to your Tableau sites and set their site roles, which determines each user's level of access. In addition, you can create groups of users, and enable guest access to your sites.

Add Users to a Site

Everyone who needs to access Tableau Server—whether to browse, publish, edit content or administer the site—must be added as a user. Administrators have the following options for adding users:

- Add a local user account or a user account from Active Directory, described later in this article.

You can also add users by importing an Active Directory group. See [Create Groups via Active Directory](#).

For Tableau Server on Linux, all external directory communication is configured and managed with a LDAP identity store. In the context of user and group synchronization, Tableau Server configured with LDAP identity store is equivalent to Active Directory. Active Directory synchronization features in Tableau Server function seamlessly with properly configured LDAP directory solutions.

- Import Users via a CSV file that you create using the [CSV Import File Guidelines](#).

Site administrator access to user management

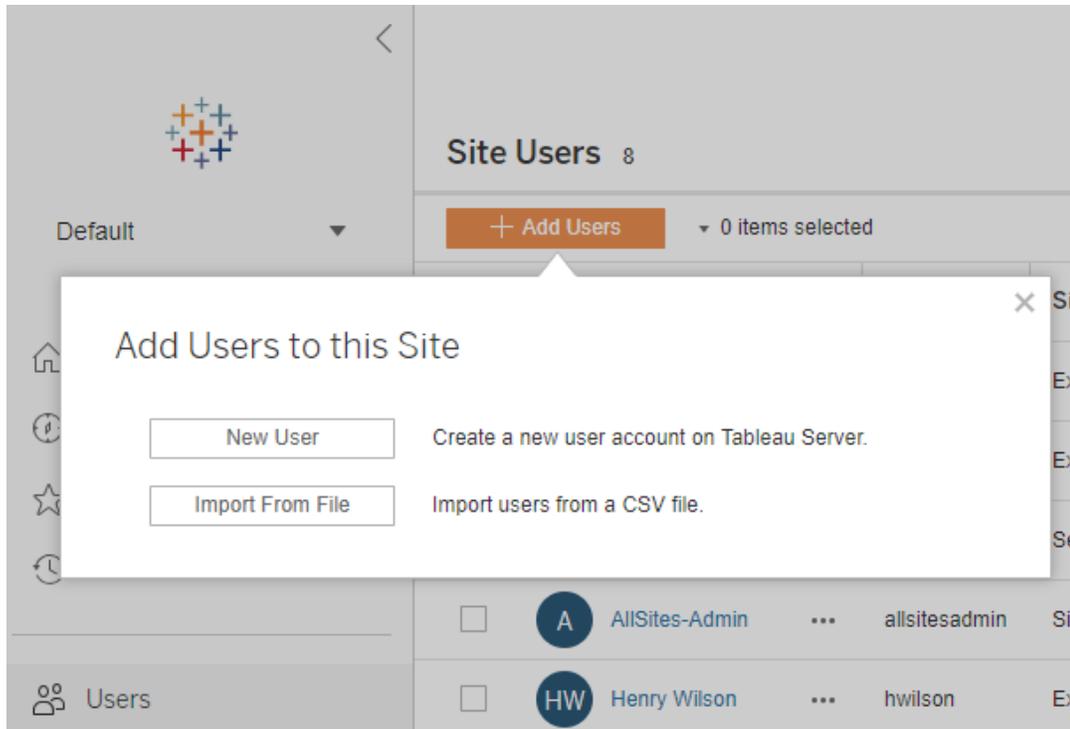
By default site administrators can add and remove users on a site. On the site's Settings page, server administrators can revoke that capability, so that only server administrators can manage the site's users.

A site administrator can edit an existing local user account only if the administrator has access to all of the sites the user is a member of. For example, if User1 is a member of

sites A and B, an administrator of only site B cannot edit User1's full name or reset the password.

Add local users to a site

1. Sign in to Tableau Server as an administrator, and if applicable select the site.
2. Select **Users**. On the Users page, click **Add Users**, and then click **New User**.



3. Enter a user name. With local authentication, using an email address for the user name is the best way to avoid user name collisions (for example, *Irodriguez@example.com* instead of *Irodriguez*).

New User

Username:

Username available

Display name:

Password:

Confirm password:

Email (optional):

Site role: 

User names are not case sensitive. Characters not allowed in user names include the semi-colon (;) and colon (,).

Also enter information in the following fields:

- **Display Name**—Type a display name for the user (e.g., *Laura Rodriguez*).
- **Password**—Type a password for the user.
- **Confirm password**—Retype the password.
- **Email**—This is optional and can be added at a later time in the user profile settings.

4. Select a site role.

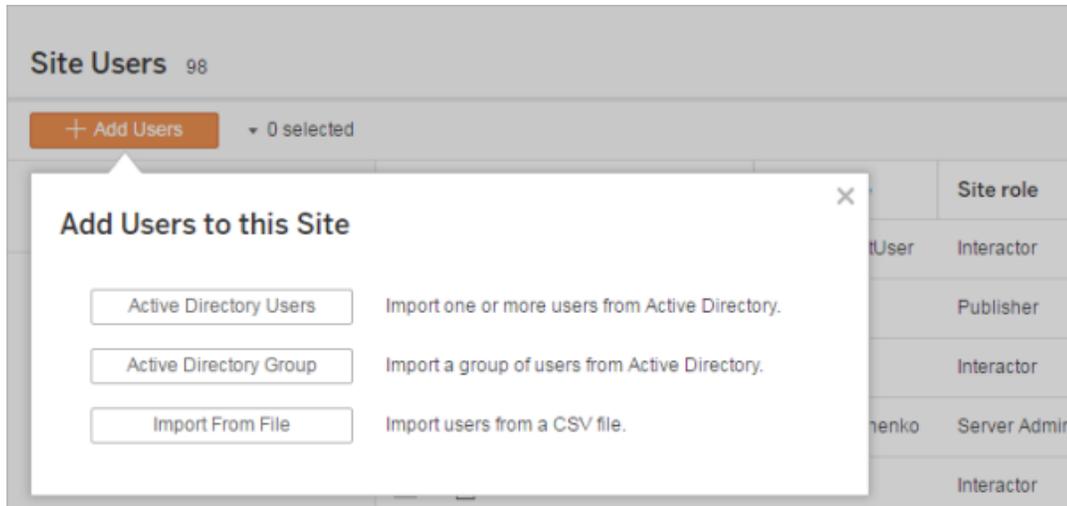
For site role definitions, see [Set Users' Site Roles](#).

5. Click **Add User**.

Add Active Directory users to a site

Before adding users to a site, be sure to review User Management in Deployments with External Identity Stores to understand how multiple domains, domain naming, NetBIOS, and Active Directory user name format influence Tableau user management.

1. In a site, click **Users**, and then click **Add Users**, and then click **Active Directory User**.



2. Enter one or more user names (separated by semicolons).

For example, *tdavis;jjohnson;hwilson*

If you are adding a user that is from the same Active Directory domain that the server is running on, you can type the AD user name without the domain. The server's domain will be assumed.

Note: Do not enter the user's full name in this field; it can cause errors during the importing process.

3. Select a site role.

For site role definitions, see [Set Users' Site Roles](#).

4. Click **Import Users**.

Remove local users

1. Sign in to Tableau Server as an administrator, select the site, and open the **Users** page.
2. Select the check boxes next to the users' names, and on the **Actions** menu, select **Remove**.

If a user is a member only of the current site, and they do not own any content, the user is removed from the server.

If a user you remove from the current site is a member of other sites on the server, their status remains unchanged on those sites.

Related information

- You can also [Add Users to Tableau Server](#), without site affiliation.
- Upgrading Tableau Server to version 2018.1 or later from a pre-2018.1 version, without activating user-based licenses, affects users who were assigned the **Viewer** site role in the pre-2018.1 server version.

To learn more, see the section “User-based licenses” in the [Licensing Overview](#) and see [Set Users' Site Roles](#).

Set Users' Site Roles

When you add users to a site on Tableau Server, independent of their license type, you must apply a *site role* to them. The site role signifies the maximum level of access a user can have on the site. Along with content permissions, the site role determines who can publish, interact with, or only view published content, or who can manage the site's users and administer the site itself.

Looking for Tableau Server on Windows? See [Set Users' Site Roles](#).

How user licenses, site roles, and content permissions work together

The intersection of a user's license type, site role, and content permissions determines the level of access a user has on the Tableau site.

Note: The license level count at the top of the header on the **Server Users** tab may differ than the count under the **Max User Role** filter due to the fact that some users have different roles across sites.

1. The license type is associated with the user. The site role you want to assign to the user determines the license type they will require.

In a multi-site environment on Tableau Server, a user's license applies to all sites the user is a member of.

2. The site role is also set at the user level. In a multi-site environment, you assign site roles on each site. For example, the same user can have the Site Administrator Creator site role on one site, and Viewer site role on another site.

The site role defines the maximum capabilities the user can have.

3. Whether the site role's maximum capabilities are available to the user depends on the permissions set on the content resources (projects, data sources, workbooks).

For example, let's say that a user has the following access on a site:

- Creator license (due to their access on another site)
- Explorer site role (on this site)
- Save permission capability on a project (on this site)

In this scenario, even though the license allows connecting to and creating new data sources in the web editing environment or Tableau Desktop, and a permission rule allows them to save in a project, their site role prevents them from being able to save so their effective permissions do not include the save capability. The user can't publish content to the site.

Similarly, even if a user has a creator license and a creator site role, if they do not have the save capability on at least one project, they can't publish anything to the site.

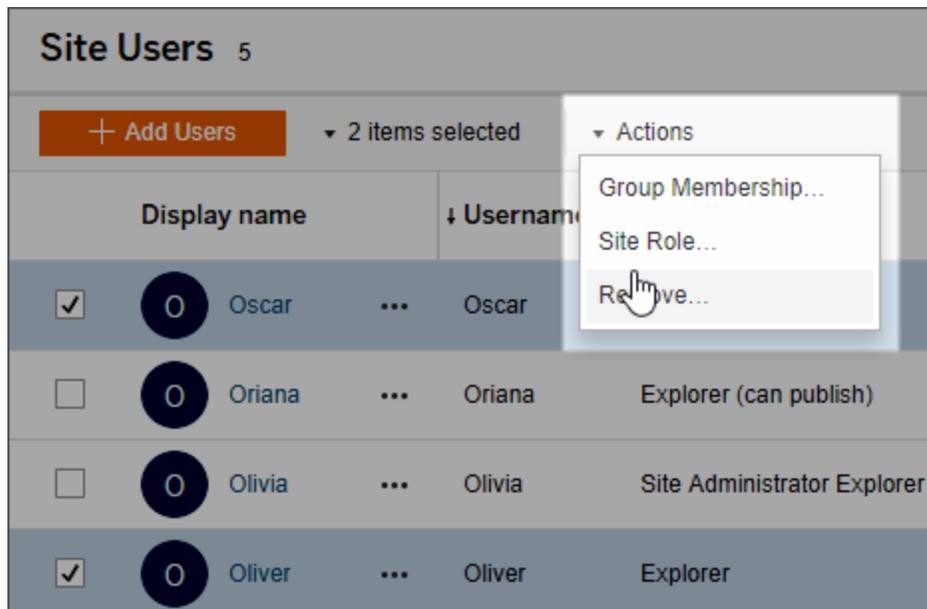
For more information, see [Permissions](#).

Change a user's site role

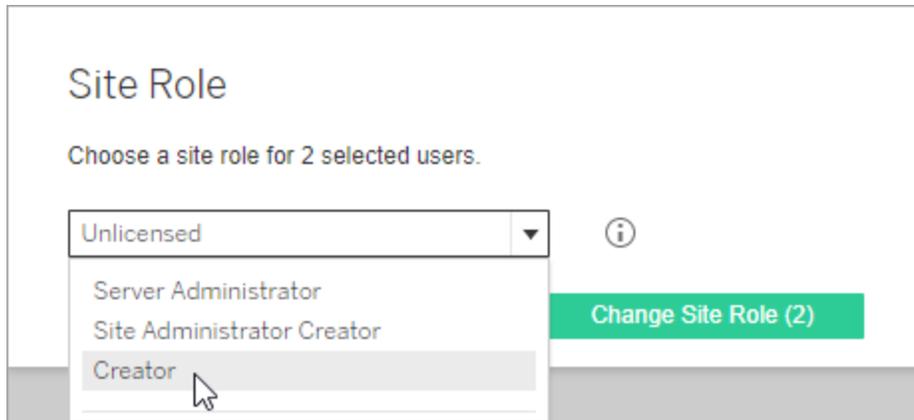
1. Sign in to the site as a server or site administrator, and go to the **Users** area.

If you are a site administrator and do not see the **Users** area, check with your server administrator on whether they have denied user management capabilities to site administrators.

2. Select the users, and then select **Actions > Site Role**.



3. Select the new site role, and then click **Change Site Role**.



You can hover the pointer over the information icon to display a matrix that shows the maximum level of general capabilities each site role allows. For more information, continue to [General capabilities allowed with each site role](#).

General capabilities allowed with each site role

The following table lists the license types as of version 2018.1, the highest level of site role allowed with each, how each site role maps to its pre-2018.1 equivalent; and summarizes the maximum capabilities each site role allows.

What this article covers and where to find what's not covered here

- This information focuses on *site* roles and is more generalized. For a list of common specific tasks available per *license* role, see the matrix on the [For Teams & Organizations](#) tab on the Tableau pricing page.
- This information describes site roles as of version 2018.1. To learn more about how core-based licensing relates to user-based licensing, how licenses transfer, or other specific licensing transition scenarios, start with the following topics:

[Migrate from Core-Based to Role-Based Licensing](#)

[Troubleshoot Licensing](#)

Tableau site roles as of version 2018.1

Site role name as of version 2018.1	Previous site role name	Maximum capabilities this site role allows
Site roles that use a Creator license		
—Users with these site roles have access to Tableau clients such as Tableau Prep, Tableau Desktop, Tableau Bridge, and Tableau Mobile.		
Server Administrator	Server Administrator	<p>Available on Tableau Server only; not applicable to Tableau Online.</p> <p>This site role always occupies the highest license activated on the server between Creator and Explorer. It allows unrestricted access to the configuration settings for the Tableau Server browser environment, all sites on the server, users and groups, and all content assets, such as flows, projects, data sources (including connection information), and workbooks.</p> <p>Connect to Tableau published data sources or external data, from the browser, Tableau Desktop, or Tableau Prep; create and publish new data sources; author and publish workbooks.</p>
Site Administrator Creator	--	<p>This is the highest level of access for Tableau Online.</p> <p>Unrestricted access to content as described above, but at the site level. Connect to Tableau or external data in the browser, Tableau Desktop, or Tableau Prep; create new data sources; build and</p>

Site role name as of version 2018.1	Previous site role name	Maximum capabilities this site role allows
		<p>publish content.</p> <p>On Tableau Server, server administrators can determine whether or not to allow site administrators to manage users and assign site roles and site membership. By default, on Tableau Server, and always on Tableau Online, site administrators are allowed these capabilities.</p>
Creator	--	<p>This is similar to the former Publisher site role, but allows new features. This site role offers non-administrators the maximum level of <i>content</i> access.</p> <p>Connect to Tableau or external data in the browser, build and publish flows, data sources and workbooks, have access to Dashboard Starters, and use interaction features on published views. Can also connect to data from Tableau Prep or Tableau Desktop, publish (upload/save) and download flows, workbooks and data sources.</p>
Site roles that use an Explorer license		
—Users with these site roles can access the server from the browser or Tableau Mobile.		
Server Administrator	N/A	<p>Tableau Server only; not applicable to Tableau Online.</p> <p>If Explorer is the highest license type activated on the server when a new server administrator user is created, the user’s site role is Server Administrator;</p>

Site role name as of version 2018.1	Previous site role name	Maximum capabilities this site role allows
		<p>however, the user will not have the full connecting and publishing capabilities that come only with the Creator license.</p> <p>With the Explorer license a Server Administrator has unrestricted access to the configuration settings for the Tableau Server browser environment, all sites on the server, users and groups, and all content assets, such as projects, flows, data sources (including connection information), and workbooks.</p> <p>However, with the Explorer license, a Server Administrator can't connect to external data from the browser to create a new data source. They can author or publish workbooks and data sources from Tableau Desktop. (they function as an Explorer (can publish) site role with regards to publishing).</p>
Site Administrator Explorer	Site Administrator	<p>Same access to site and user configuration as Site Administrator Creator, but can't connect to external data from the web editing environment.</p> <p>Can connect to Tableau published data sources to create new workbooks, and edit and save existing workbooks.</p>
Explorer (can publish)	Publisher	<p>Can publish workbooks from the web using existing data sources, browse and interact with published views, and use all interaction features.</p> <p>In the web editing environment, can edit and save</p>

Site role name as of version 2018.1	Previous site role name	Maximum capabilities this site role allows
		existing workbooks. Cannot save new standalone data sources from data connections embedded in workbooks, and cannot connect to external data and create new data sources.
Explorer	Interactor	Can browse and interact with published views. Can subscribe to content, create data driven alerts, connect to Tableau published data sources and open workbooks in the web authoring environment for ad-hoc queries, but they can't save their work.
Read Only	Viewer	<p>This site role is available only in version 2018.1, for transitioning users to the user-based Viewer (or other) license and site role. Any users in the Read Only site role prior to upgrading to version 2018.2 or later are reassigned to the Viewer site role.</p> <p>In 2018.1 versions, Read Only users can see and subscribe to published views others have created. Can't use other interaction features or save custom views.</p>
Site roles that use a Viewer license		
Viewer	N/A	<p>Can see published views others have created and use most interaction features. Can subscribe to views and download as images or summary data. Can't connect to data, create, edit, or publish content, or set data alerts.</p> <p>For a list of specific capabilities, see the Viewer</p>

Site role name as of version 2018.1	Previous site role name	Maximum capabilities this site role allows
		<p>column in the matrix on the Tableau pricing page.</p> <p>Note: Although the Viewer site role existed in previous versions, the new Viewer site role has additional capabilities.</p>
Other site roles		
Unlicensed	Unlicensed	<p>Unlicensed users can't sign in to Tableau Server or Tableau Online. Users are assigned the Unlicensed role in the following circumstances:</p> <ul style="list-style-type: none"> • You import users from a CSV file and their license level is set to unlicensed. • The number of available licenses is reached at the time you add or import users. • You remove a user who owns content on the site. The user will still own the content but not be able to do anything with it. • A product key(s) has expired. See Refresh Expiration Date for the Product Key.

Who can publish content

The following site roles allow the specified level of publishing access.

- **Server Administrator** (Tableau Server only); **Site Administrator Creator**; and **Creator** allow full connecting and publishing access.

This includes connecting to data and publishing new flows, new workbooks and new data sources from Tableau Desktop and the web editing environment. The site roles also allow editing and saving existing published workbooks, or publishing updates to existing data sources.

- **Explorer (Can Publish)** and **Site Administrator Explorer** have limited publishing capabilities, as described in General capabilities allowed with each site role.
- **Explorer**, **Viewer**, **Read Only**, and **Unlicensed** do not allow publishing.

Site roles and Active Directory import and synchronization

When you import users from an external directory like Active Directory, you can specify the site role. If a user is not yet a member of any site on the server, the user is added to the site with the assigned role. When you synchronize groups from an external directory, the site role is applied through the **Minimum Site Role** setting on the **Groups - Details** page.

Note: In the context of user and group synchronization, Tableau Server configured with LDAP identity store is equivalent to Active Directory. Active Directory synchronization features in Tableau Server function seamlessly with properly configured LDAP directory solutions.

If a user already exists in a Tableau Server site, the site role assigned during the import or sync process will be applied if it gives the user more access in a site. Importing or synchronizing AD users and groups can promote a user's site role, but does not demote a user's site role.

If a user already has the ability to publish, that ability is maintained.

The matrix below shows the rules applied for site roles on import.

Note: The **Import Site Role** row abbreviated headers indicate the site role specified for import. The **Current Site Role** column headers represent the current user site role. The table values represent the abbreviated resulting site role.

- Site Administrator: SA
- Site Administrator Creator: SC
- Site Administrator Explorer: SE
- Creator: C
- Explorer: E
- Explorer (Can Publish): EP
- Viewer: V
- Unlicensed: U

	Current Site Role						
Import Site Role	SC	C	SE	EP	E	V	U
Site Administrator Creator (SC)	SC	SC	SC	SC	SC	SC	SC
Site Administrator Explorer (SE)	SE	SE	SE	SE	SE	SE	SE
Creator (C)	SC	C	SE	C	C	C	C
Explorer (Can Publish) (EP)	SC	C	SE	EP	EP	EP	EP
Explorer	SE	C	SE	EP	E	E	E

	Current Site Role						
Import Site Role	SC	C	SE	EP	E	V	U
(E)							
Viewer (V)	SE	C	SE	EP	E	V	V
Unlicensed (U)	SE	C	SE	EP	E	V	U

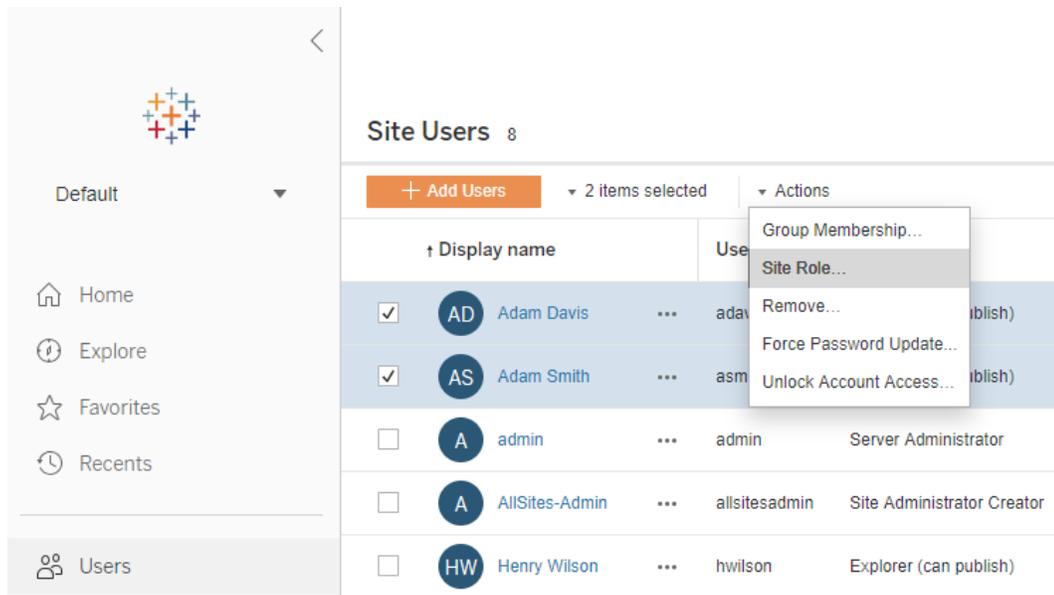
View, Manage, or Remove Users

Administrators can manage a site’s users such as adding and removing users, setting the groups they’re members of, setting their site roles, and so on. On Tableau Server, server administrators can manage users on multiple sites at a time on the **All Sites** page.

View and manage users on a site

Sign in to a site as an administrator, and then select **Users**. On this page you can do any of the following to manage users:

- Set group membership, set site role, or remove the user from the site. If you’ve configured the site for SAML single sign-on, you can set the selected users’ authentication type. If your server is configured for password policies and account access lockout, you can unlock access to a user’s locked out account.

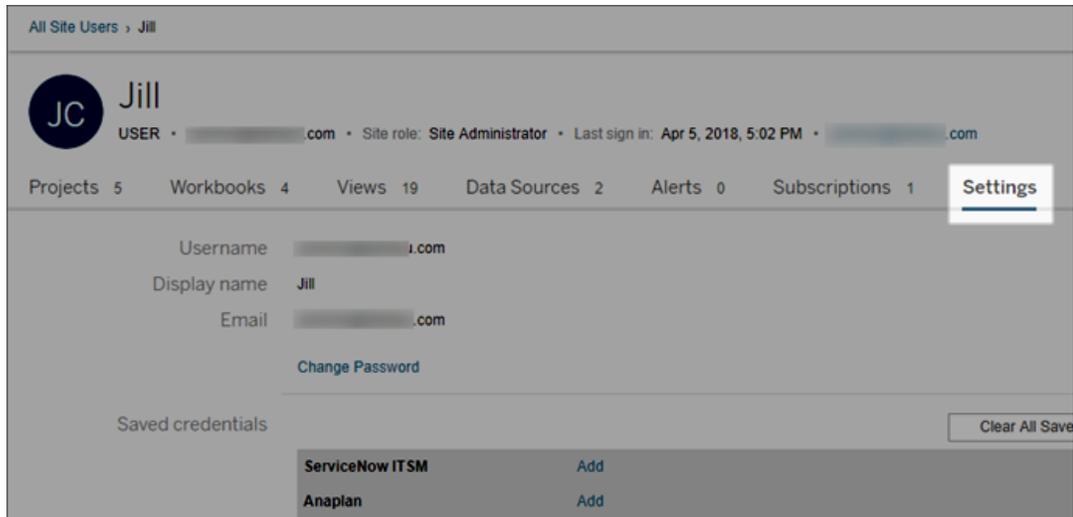


- Select a user name to see details about them, such as content they own, views they subscribe to, and their account settings.

The user **Settings** page is available when the following conditions are true:

- The user is a member only of sites that the site administrator controls
- Site administrators can manage users. This is always the case for Tableau Online and is the default for Tableau Server; Tableau Server administrators can change this access for site admins.

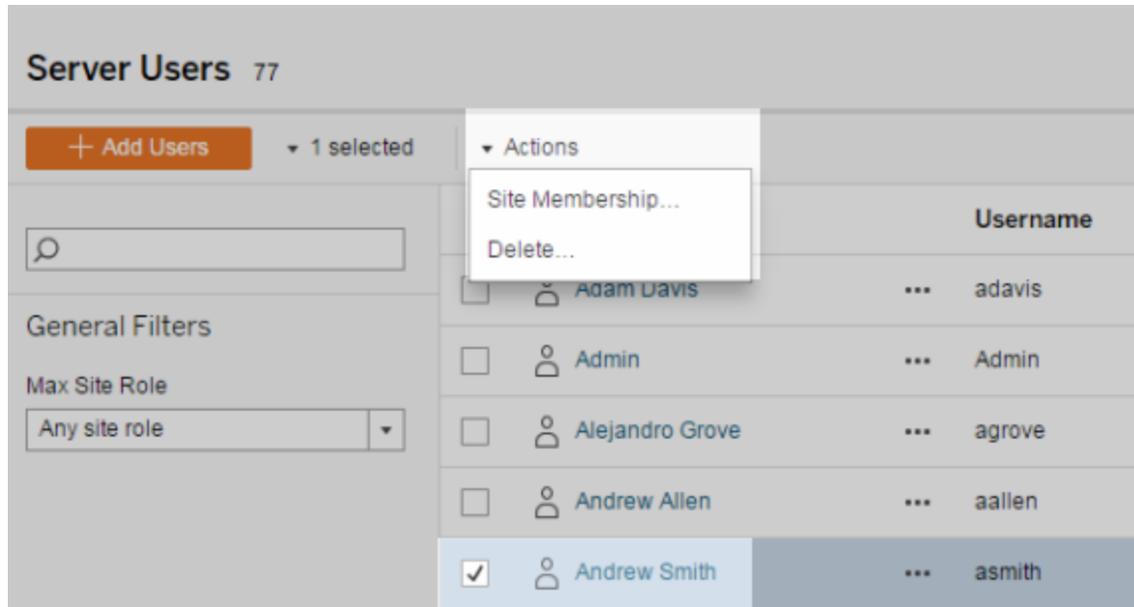
Tableau Server on Linux Administrator Guide



If the server is configured to use the internal user management system (Local Authentication), you can edit the **Display Name**, **Email**, and **Password** for users after they have been added. If you are making many changes, you may find it easier to import the changes from a CSV file. For details, see [Import Users and CSV Import File Guidelines](#).

View and edit server users

Sign in to Tableau Server as a server administrator. On the site menu, select **Manage All Sites**, and then select **Users**.



Manage users' site membership

By default, server and site administrators can manage users at the individual site level. Server administrators can also manage users and their site roles on multiple sites. You do this at the **All Sites** level (at the server level).

1. In the site menu, select **Manage All Sites**, and then select **Users**.
2. On the Server Users page, select the check boxes next to the users, and then select **Actions > Site Membership**.
3. Select one or more sites, and a site role for each site, and then click **Save**.

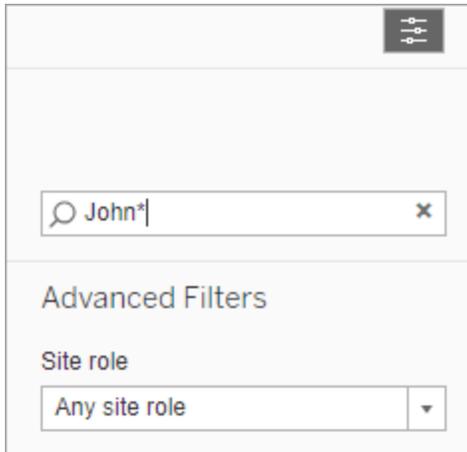
Search for users

To search for a specific user

Use the filter toggle in the upper right to display the search box and site role filter. Then use the search box or filters to find the users you want.

The search operation checks the display name and user name attributes.

You can use the asterisk (*) character as a search wildcard. For example, searching for `John*` will return all user names that start with *John*.



The screenshot shows a search interface with a search bar containing the text "John*" and a search icon. Below the search bar is an "Advanced Filters" section with a "Site role" dropdown menu set to "Any site role".

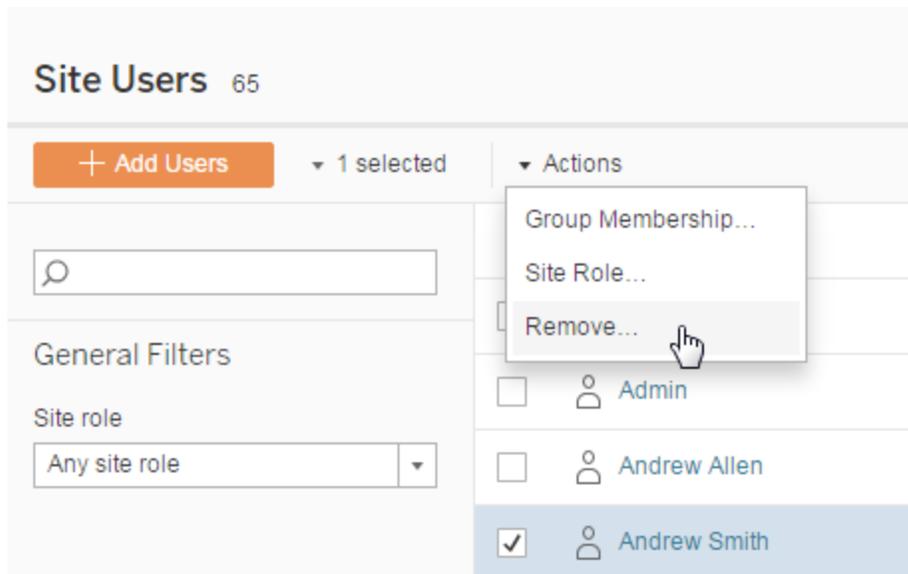
Remove users from a site

You can remove a user only if the user does not own any content (projects, workbooks, views, or data sources). If you attempt to remove a user who owns content, the user site role will be set to Unlicensed, but the user will not be removed.

If the default All Users group has enabled Grant site role on sign in, that user's content must be reassigned to another user or removed before they can be unlicensed or removed. For more information on Grant role on sign in, see [Removing users affected by Grant role on sign in](#). For more information on changing content ownership, see [Manage Content Ownership](#).

Note: On Tableau Server, when an administrator removes a user from a site (and the user belongs only to that one site), the user is also deleted from the server.

1. Sign in to a site as an administrator, and go to the **Users** area. Select one or more users to remove, and then select **Actions > Remove**.



2. Click **Remove** in the confirmation dialog.

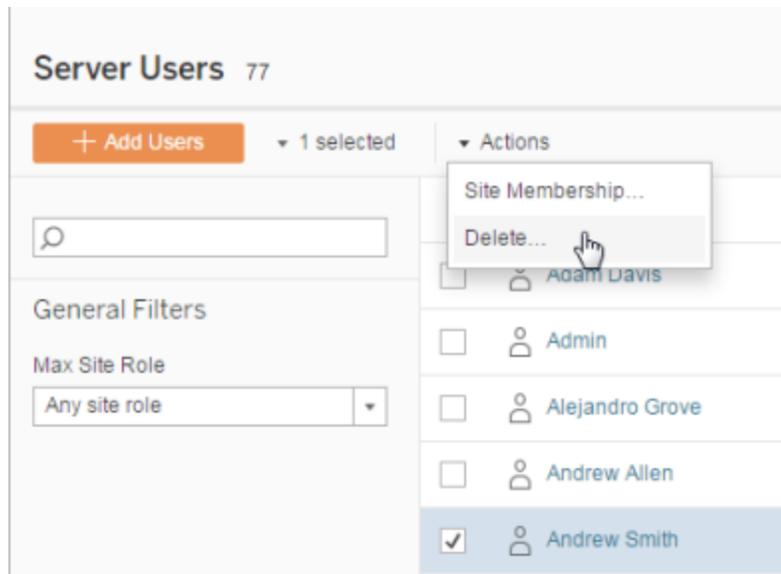
Remove users from the server

You can remove a user only if the user does not own any content (projects, workbooks, views, or data sources). If you attempt to remove a user who owns content, the user site role will be set to Unlicensed, but the user will not removed.

If a user is a member of multiple sites, and they own content on any of those sites, they are removed from the sites on which they don't own content. The user remains a member on sites where they own content, but demoted to the Unlicensed site role.

1. In the site menu, click **Manage All Sites**, and then click **Users**. In a single-site environment, click **Users**.

Select one or more users to delete, and then click **Actions > Delete**.



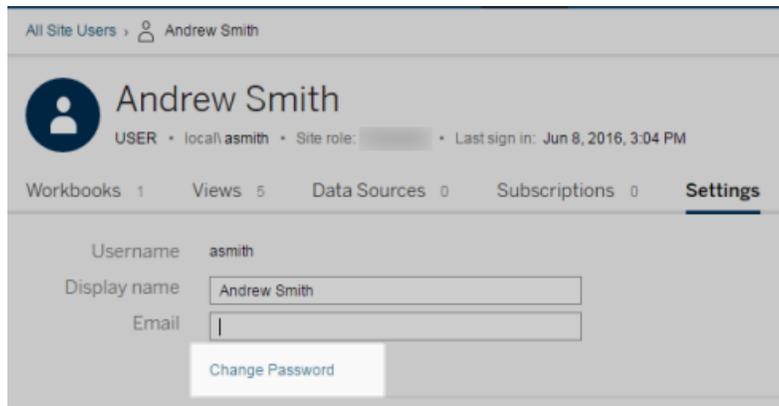
2. Click **Delete** in the confirmation dialog box.

Change passwords for users of a single site

To change the password for a user with membership to a single site, sign in to Tableau Server as a site administrator or a server administrator.

1. Ensure that the correct site is selected in the menu.
2. Click **Users**.
3. Click the display name of a user.
4. Click **Settings**.

5. Click the **Change Password** link, edit the password, and then click **Save Password**.

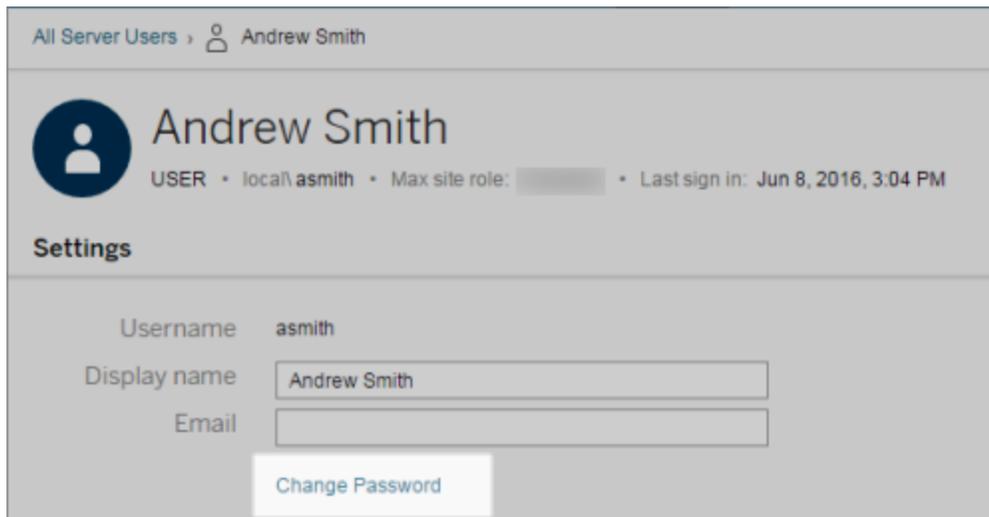


Change passwords for users of multiple sites

To change the password of a user with membership to multiple sites, sign in to Tableau Server as a server administrator.

1. In the site menu, click **Manage All Sites**.
2. Click **Users**.
3. Click the display name of a user.
4. Click the **Change Password** link, edit the password, and then click **Save**

Password.



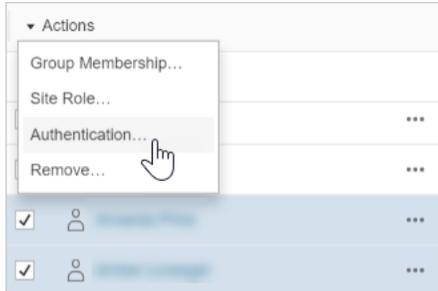
The screenshot shows the user settings page for Andrew Smith. At the top, it says "All Server Users > Andrew Smith". Below that is a profile card with a blue circular icon, the name "Andrew Smith", and details: "USER • local\asmith • Max site role: [redacted] • Last sign in: Jun 8, 2016, 3:04 PM". Underneath is a "Settings" section with fields for "Username" (asmith), "Display name" (Andrew Smith), and "Email" (empty). A "Change Password" button is located at the bottom of the settings area.

Set the User Authentication Type for SAML

On a site that has been configured for site-specific SAML, administrators can specify users' authentication type. For example, if Tableau Server was configured for site-specific SAML and server-wide SAML, administrators can specify which users authenticate with site-specific SAML and which users authenticate with server-wide SAML.

You can assign authentication type at the time you add users to Tableau Server, as well as any time afterward.

1. When you're signed in to the Tableau Server site, select **Users**.
2. On the **Site Users** page, select the check boxes next to the users you want to assign an authentication type.
3. On the **Actions** menu, select **Authentication**.



4. In the Authentication dialog box, select **Site SAML** or **Server Default**.

Notes

- Users that authenticate with site-specific SAML can only belong to one site. If a user needs to belong to multiple sites, set their authentication type to the server default. Depending on how site-specific SAML was configured by the server administrator, the server default is either local authentication or server-wide SAML.
- If you change users' authentication to site-specific SAML, the next time they sign in, they will be directed to your identity provider's site to provide their credentials.

Import Users

To automate the process of adding users to a site, you can create a CSV file that contains user information, and then import the file.

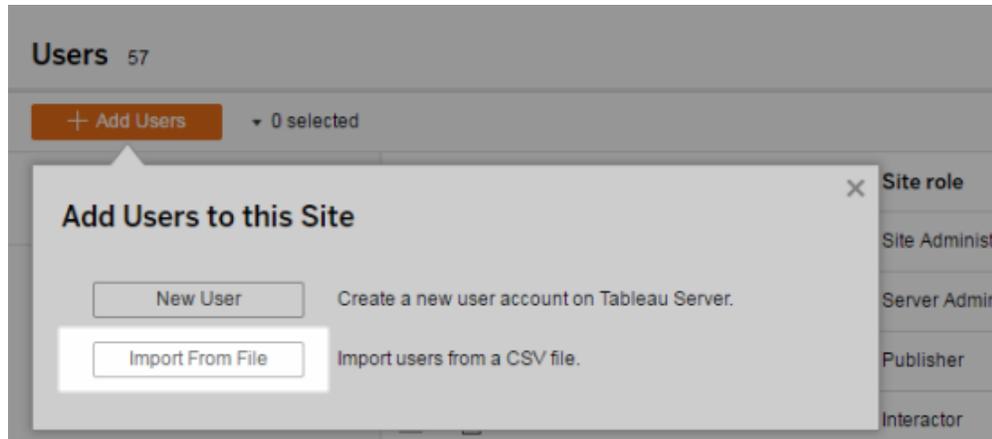
Site administrators can import users to a particular site; server administrators (Tableau Server only) can import users at the server level, to later add them to multiple sites.

Note: This topic contains the steps for importing, assuming that you have already created the CSV file. If you have not created the file yet, see [CSV Import File Guidelines](#) for a list of file format requirements and import options.

Add users from a CSV file

The following steps describe how to add users to a site or to the server. The images reflect adding users at the site level.

1. Do one of the following:
 - To add users at the site level, select **Users**, and then **Add Users**.



- To add users at the server level on a **single-site** server, select **Users**, and then **Add Users**.
 - To add users at the server level on a **multi-site** server, open the list of sites, and select **Manage All Sites**. Select **Users**, and then **Add Users**.
2. Click **Import From File**, click **Browse** and navigate to the file, and then click **Import Users**.

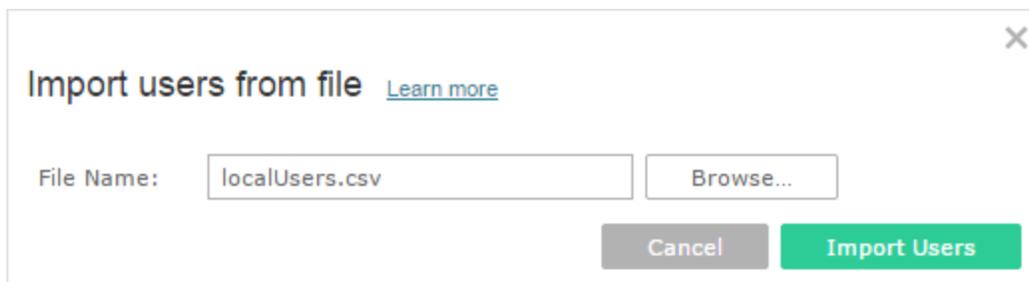
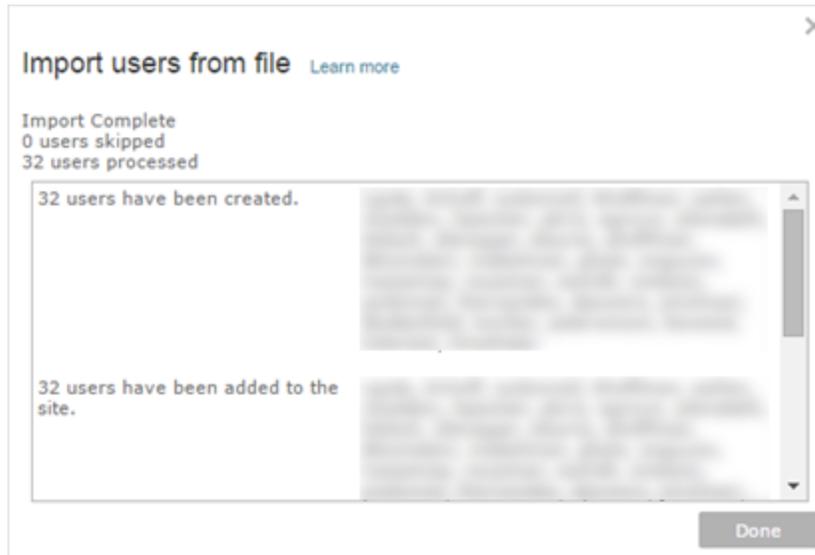


Tableau displays the results of the import process (names in this image are blurred).



3. Click **Done**.

How users' site roles are assigned or maintained

When you import at the site level or on a single-site server using `tabcmd`, you can specify the site role for all users in the CSV file. If a user already exists in the Tableau Server site, the site role assigned during the import process will be applied, even if it is more restrictive than users' existing site role. The exception is that you cannot affect a server administrator's site role.

For a multi-site server, when you import users in the Server Users page, you create server users with no site affiliation. Because these users do not belong to a site, they cannot have a site role. The only site role a user can have at the server level is **Unlicensed** or **Server Administrator**.

You can also specify the user's site role when you assign site membership to a user. For information, see [Manage users' site membership](#).

Importing at the server level in multi-site environments

If the server is running multiple sites and you are a server administrator, you can import a CSV file from two locations. Where existing user accounts are concerned, each location has different capabilities.

- The **Server Users** page appears in a multi-site environment. Only server administrators can access this page.

The screenshot shows the 'Server Users' page with 77 users. It includes a search bar, an 'Add Users' button, and a 'General Filters' section with a 'Max Site Role' dropdown set to 'Any site role'. The user list is as follows:

	† Display name	Username	Max site role	Sites	Last signed in
<input type="checkbox"/>	Adam Davis	adavis	Site Administrator	7	Jul 21, 2016, 5:12
<input type="checkbox"/>	Admin	Admin	Server Administrator	9	Aug 5, 2016, 9:19
<input type="checkbox"/>	Alejandro Grove	agrove	Interactor	1	

You can import the CSV file from here if you want to update existing user accounts in addition to adding new ones. For example, if you import a file that has a new password for each existing user, their passwords will be reset.

- The **Site Users** page.

The screenshot shows the 'Site Users' page with 65 users. It includes a search bar, an 'Add Users' button, and a 'General Filters' section with a 'Site role' dropdown set to 'Any site role'. The user list is as follows:

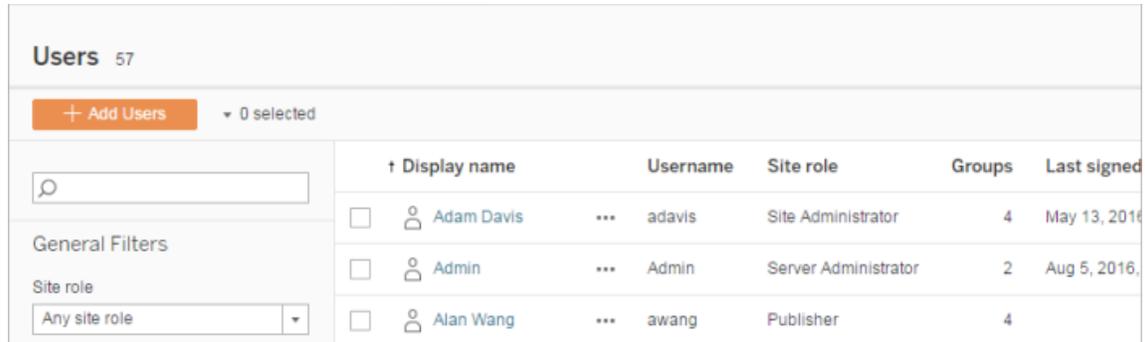
	† Display name	Username	Site role	Groups	Last signed in
<input type="checkbox"/>	Adam Davis	adavis	Site Administrator	2	Jul 21, 2016, 5:12
<input type="checkbox"/>	Admin	Admin	Server Administrator	3	Aug 5, 2016, 9:19
<input type="checkbox"/>	Andrew Allen	aallen	Publisher	2	Jul 21, 2016, 5:35

Server administrators can add new user accounts with CSV imports. If the CSV file includes existing users, the **Password** and **Display Name** fields must either match

the existing or be left blank. If new passwords or full names are used, the import will fail.

Importing to a single-site environment

Server and site administrators on a single-site server perform CSV user imports from the **Users** page in a site.



The screenshot shows the 'Users' page in Tableau, displaying a list of 57 users. The page includes a search bar, a '+ Add Users' button, and a '0 selected' indicator. A 'General Filters' section on the left allows filtering by 'Site role', currently set to 'Any site role'. The main table lists users with columns for 'Display name', 'Username', 'Site role', 'Groups', and 'Last signed in'.

	† Display name	Username	Site role	Groups	Last signed in
<input type="checkbox"/>	Adam Davis	adavis	Site Administrator	4	May 13, 2016
<input type="checkbox"/>	Admin	Admin	Server Administrator	2	Aug 5, 2016
<input type="checkbox"/>	Alan Wang	awang	Publisher	4	

Multi-site versus single-site import

Users can belong to more than one site on the same server, but they must use the same credentials for each site. This becomes important when you're adding users to a site and those users might already be members of a different site. If you try to import a user who already exists, and if the user's credentials in the CSV file don't match the existing credentials, the import fails for that user.

If you're importing users to a site and think the users might already exist on the server, you can try leaving the **Password** field in the CSV file blank (while including the delimiters for the field). If a user who is defined in the CSV already exists in another site, the user is added to the site where you're importing. However, if the user *doesn't* already exist on the server, the user is created, and the CSV import window alerts you that the new user doesn't have a password. You can then use the server administrator pages to assign a password to any user who doesn't have one.

Note: If the server is configured to use Active Directory for authentication, user passwords are managed by Active Directory, and you can leave the password field in the CSV file blank.

CSV Import File Guidelines

You can automate adding users by creating a comma-separated values (CSV) file with user information and then importing the file. You can include attributes in the CSV file, such as license level and the publishing access, to apply to the users at the same time you import them.

To import users, you can use the server or site administration pages or the `tabcmd` utility. Using `tabcmd` provides an option for assigning a site role to all users in the CSV file. For information, see `Import Users` or `createsiteusers filename.csv`.

You can import users at the site or server level. If you import users to the server (not to a specific site), the users aren't assigned to a site and are imported as Unlicensed.

CSV file format requirements

When you create the CSV file for importing users, make sure that the file meets the following formatting requirements:

- The file does not include column headings. Tableau Server assumes that every line in the file represents a user.
- The file is in UTF-8 format, and includes the byte-order mark (BOM).
- Character encodings such as BIG-5 have been converted to UTF-8. You can do this by opening the file in a text editor and using the **Save As** command.
- If a user name includes an @ character that represents anything other than a domain separator, you need to refer to the symbol using the hexadecimal format: `\0x40`

For example, `user@fremont@mycompany.com` should be `user-
\\0x40fremont@mycompany.com`

Required columns in the CSV file

The following values are required for each user:

- User name
- Password: If Tableau Server is configured to use Active Directory authentication, there must be a `password` column, but the column itself should be empty. If the server is using local authentication, you must provide passwords for new users.

Additional import file options

The CSV file can contain the following fields, in the order shown here:

- User name. The user name. If the server is configured to use Active Directory, this value must match a user defined in Active Directory. If the user name is not unique across domains, you must include the domain as part of the user name (for example, `example\Adam` or `adam@example`). This is the only required field.
- Password. A password for the user. If the server is configured to use Active Directory, this value is not used.
- Display name. The display name is part of the information used to identify a user on the server. If the user's display name is already in use, Tableau Server updates the existing user information with the settings in the CSV file. If the server is configured using Active Directory, this value is not used.
- License level. This can be **Creator**, **Explorer**, **Viewer**, or **Unlicensed**.
- Administrator level (**System**, **Site**, or **None**). This setting determines whether the user is imported as an administrator.

Tableau Server on Linux Administrator Guide

If you are using the web UI to import users, you can set the administrator site role to **System** only if you import the file at the server (All Sites) level. If you are signed in to a specific site, and if the administrator column for a user in the CSV file is set to **System**, Tableau Server imports the user as a site administrator.

- Publishing capability (**yes/true/1** or **no/false/0**). If you are using the web UI, the publisher setting is used only if you import while signed in to a specific site. If you import users at the server (All Sites) level, this value isn't used.
- Email address. The email address is part of the information used to identify a user on the server. If the email address is already in use, Tableau Server updates the existing user information with the settings in the CSV file.

The order of the columns is significant. The first column is treated as the user name, the second as the password, the third as display name, and so on, regardless of the content in the columns. If you omit values for a field, you must still include the field's comma delimiter.

Improve performance for large CSV files passed through `tabcmd`

A server administrator can enable server settings that help to improve performance for importing large CSV files through `tabcmd` commands. You can do this using the `tsm configuration set` command with the following options:

- `vizportal.csv_user_mgmt.index_site_users`
- `vizportal.csv_user_mgmt.bulk_index_users`
- `searchserver.index.bulk_query_user_groups`

Essentially, these options index users after the CSV file is processed, instead of one-by-one as they are added to the server's database. This reduces the number of calls to the database and the memory required to process the file. These `tsm configuration set` options apply to the `tabcmd createsiteusers`, `deletesiteusers`, `addusers`, and `removeusers` commands.

For descriptions for these settings, see `tsm configuration set` Options.

Notes

- If you are not signed in to a specific site and are importing users at the server level, you can assign only the Server Administrator and Unlicensed site roles.
- If you have a user-based server installation, and if adding users would exceed the number of users allowed by your license, the users are added as unlicensed users.
- If you use `tabcmd` and specify the license, but importing users would exceed your license limits, users are imported as Unlicensed.

CSV settings and site roles

The license level, administrator, and publishing settings for a user determine how the user's site role is set during the import process. The following table shows how the settings are converted to site roles.

CSV settings	Site role
License level=(any) Administrator=System Publisher=true	Server Administrator. This setting applies to Tableau Server only, and it is valid only if you are importing users while managing the server (that is, not signed in to a specific site). The Server Administrator site role always takes a Creator license if one is available. If no Creator license is available, see Troubleshoot Licensing to learn about the way Tableau Server handles this.
License level=Creator or Explorer Administrator=Site Publisher=true	Site Administrator Creator or Site Administrator Explorer. This setting is valid only if you are importing users while signed in to a specific site.
License level=Creator	Creator

Tableau Server on Linux Administrator Guide

CSV settings	Site role
Administrator=None Publisher=true	
License level=Explorer Administrator=None Publisher=true	Explorer (Can Publish)
License level=Explorer Administrator=None Publisher=false	Explorer
License level=Viewer Administrator=None Publisher=false	Viewer
License level=Unlicensed Administrator=None Publisher=false	Unlicensed

CSV import example for Tableau Server

The following example shows a CSV file that contains information for several users.

```

henryw,henrypassword,Henry Wilson,Creator,
None,yes,henryw@example.com
freds,fredpassword,Fred Suzuki,Viewer,
None,no,freds@example.com
alanw,alanpassword,Alan Wang,Explorer,
Site,yes,alanw@example.com
    
```

```
michellek,michellepassword,Michelle Kim,Creator,System,yes,michellek@example.com
```

If you import this file while managing a site, four users are added to that site. The `Administrator` setting for user Michelle is `System`. However, because you are importing the users into a site, Tableau Server give Michelle the Site Administrator Creator site role.

Three of the users are allowed to publish.

If you import this file while managing the server, four users are added to the server, but they are not added to any site. Only one user is imported as a server administrator; the rest are set to Unlicensed.

Manage Site User Visibility

By default, all site users can see aliases, project ownership and comments by other users when permissions allow. The User Visibility setting lets administrators manage if users with Viewer and Explorer site roles see other users and groups on the site, which can be important for sites that are used by multiple clients. To learn more about site roles, see [Set Users' Site Roles](#).

Limit user visibility

Setting User Visibility to **Limited** impacts certain collaboration tools and hides user information in Tableau Online and Tableau Server. Limited User Visibility either disables the feature for Viewers and Explorers (excluding Site Administrator Explorers), or removes user information from other areas. Note that Creators and administrators will still see user information when User Visibility is set to Limited.

To limit user visibility for Explorers and Viewers (excluding Site Administrator Explorers):

- Navigate to the site's **Settings** page
- Select **Limited** in the **User Visibility** setting

Tableau Server on Linux Administrator Guide

The following is a list of site areas impacted when User Visibility is set to Limited. Unless noted that the feature is disabled for all users, only non-administrator Explorers or Viewers are impacted.

Area	Impact
Search	User information not displayed
Content owners	User information not displayed (Explorers and Viewers can't see themselves, but can see their content in My Content)
Profile pictures	User information not displayed
Subscriptions	User information not displayed
Recommendations	Similar users not displayed (all users)
Add/Edit Tags	Explorers and Viewers can see tags but cannot delete or modify them
"Who has seen this view?"	Disabled
Ask Data usage analytics	Disabled
Permissions dialogs	Disabled
Named sharing	Disabled (all users)
Alerts	Disabled (all users) Existing alerts paused
Comments	Disabled (all users)
Public Custom Views	Disabled (all users) Existing public custom views appear as private

Request Access	Disabled (all users)
Tableau Desktop	Publishing workbooks disabled from Desktop User information not displayed on user filters
Tableau Catalog (with Data Management Add-on)	User information not displayed

When User Visibility is set to Limited, Tableau Server REST API and Metadata API calls behave as described in the table above.

Users on a site can interact with views and modify them, such as applying filters. If that user shares their modified view with others, or if the user creates something from that modified view (like a metric or a private custom view), then that user's name appears in the URL. Make sure that the URL for this modified view is only distributed to users who are permitted to see that person's name.

Note: If a user is a member of multiple sites, entering an email on the sign in page for Tableau Online will return the names of all sites the user is a member of.

Best practices for limiting user visibility

Administrators can also check that user and group information is not visible in these ways:

- Configure permissions to only provide content to appropriate parties. For more information, see [Permissions](#).
 - Limited User Visibility hides user identification information from search, but might return content that the user published, including when searching by owner name, if the person searching has viewing permission to that content.
 - A user publishing a workbook with a duplicate title in the same project might see a warning that a workbook with that title already exists.
- Apply row-level security when necessary.
- Check that metadata within dashboards does not contain user information.

- Check that calculations accessible to users don't contain user metadata (e.g., user filters).

Restore Full User Visibility

When administrators set User Visibility back to Full, features disabled for all users by Limited User Visibility (such as comments and alerts) remain off. Administrators can re-enable these features through the site's Settings page.

Any previous feature settings are not retained when User Visibility is set to Full, and affected features are not automatically turned on.

Guest User

Core-based licenses of Tableau Server include a Guest user option, which you can use to let people access Tableau views without an account on the server.

Guest user access is enabled by default when Tableau Server is installed with a core-based license. It is not available with user-based licensing. If you do not intend to use Guest user access, you should disable it.

Guest access allows users only to see and interact with Tableau views. The Guest user cannot browse the Tableau Server interface or see server interface elements in the view, such as user name, account settings, comments, and so on. For more information about licenses, see [Manage Licenses](#).

Tip: To share views with Guest users, either provide URL links or embed views into web pages. For more information, [see Tableau User Help](#).

Guest user permissions

A Guest user can have the following maximum capabilities:

- **Workbooks and views:** View, Export Image, Summary Data, View Comments, Filter, Full Data, Web Edit, Download (to save a local copy)
- **Data sources:** View and Download

If you add the Guest user to a group that has a higher level of access to a content resource, the Guest user's access does not exceed the capabilities listed above. However, the Guest user account will comply with more restrictive permissions settings.

Enable or disable Guest access

You must be a server administrator to change Guest account settings at either the server or the site level.

Note: Enabling the Guest user for a site can increase the number of potential simultaneous viewers beyond the user list you might be expecting. The administrative view **Status > Traffic to Views** can help you gauge the activity.

1. In the site menu, click **Manage All Sites** and then click **Settings > General**.
2. For **Guest Access**, select or clear **Enable guest access**.
3. Click **Save**.

This enables the Guest user on all sites. You can then go to the same setting for a specific site. To disallow Guest access for a site:

1. In the site menu, select a site.
2. Click **Settings**, and on the General tab, clear the **Enable guest access for this site** check box.

If the Guest account is enabled on some or all sites, and you turn off Guest access at the server level, it is turned off for all sites as well.

Note: You can turn off Guest user access at the server or site level; however, you aren't able to remove the user. So, although no one can access data or views without signing in to the server, the Guest user still appears in the Site Users list and group lists for groups you've added the Guest user to.

Additional Guest account characteristics

The Guest user is unique in the following additional ways:

- As a single user account, it represents all unauthenticated users accessing Tableau views.
- When enabled, it is a member of the All Users group.
- You can add it as a member of other groups on a site.
- You cannot edit it or select it as the owner of a content resource.
- If the Guest user needs to access a workbook with an extract connection, the Guest must also have the **View** capability on the published data source. (The Guest user is not allowed to connect to published data sources.)
- The account is not allowed to save custom views.
- Guest cannot be used in a user filter.
- You cannot delete the account; however, you can turn off access to it by clearing the check box described in the steps above.

Groups

You can create and delete user groups, add users to a group, and synchronize groups with Active Directory.

Add Users to a Group

You can organize Tableau Server users into groups to make it easier to manage multiple users. You can create groups on the server or import groups from Active Directory.

If you're managing users with an External identity store, such as Active Directory, add users to a group through the external identity store itself. Once users are added to a group in the external identity store, Tableau Server is able to update those users by synchronizing the group of users in the external identity store with the group of users on Tableau Server.

For example, to keep Active Directory group membership up to date, we recommend you review the following:

- Site administrators can synchronize selected groups on demand in a site. For more information, see [Synchronize Active Directory Groups on a Site](#).
- Server administrators can synchronize all Active Directory groups on the server based on a schedule or on-demand. For more information, see [Synchronize All Active Directory Groups on the Server](#).

Note: In the context of user and group synchronization, Tableau Server configured with LDAP identity store is equivalent to Active Directory. Active Directory synchronization features in Tableau Server function seamlessly with properly configured LDAP directory solutions.

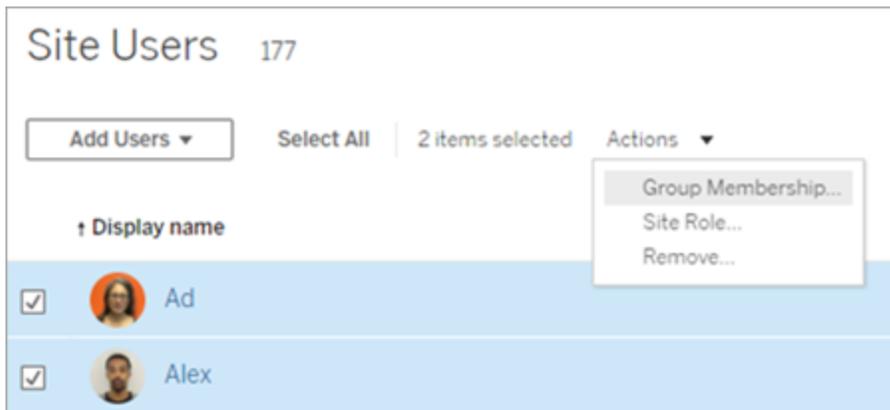
If you're managing users with a Local identity store, use the procedures described below to add users to a group.

To add a user to a group, the group must already exist.

Add users to a group (Users page)

1. In a site, click **Users**.
2. Select the users you want to add to a group, and then click **Actions > Group**

Membership.

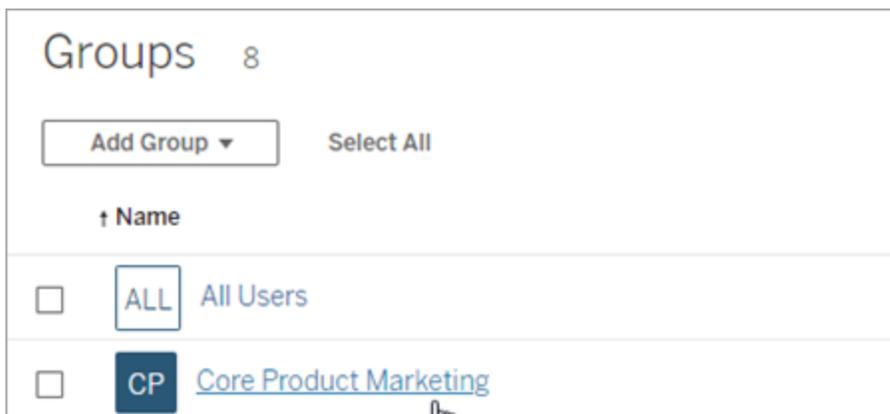


3. In the Group Membership dialog box, select the groups and then click **Save**.

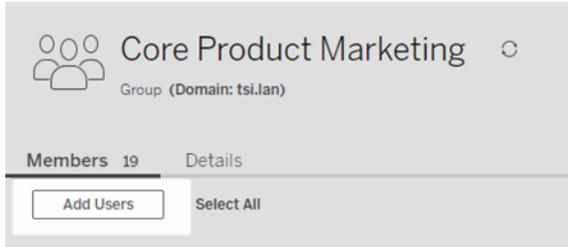
Add users to a group (Groups page)

Note: This is only possible for site administrators if the server administrator has configured the site so that both Server and Site administrators can manage users. For more information, see *Managing Users (Tableau Server Administrators only)*.

1. In a site, click **Groups**, and then click the name of the group.



2. In the group's page, click **Add Users**.

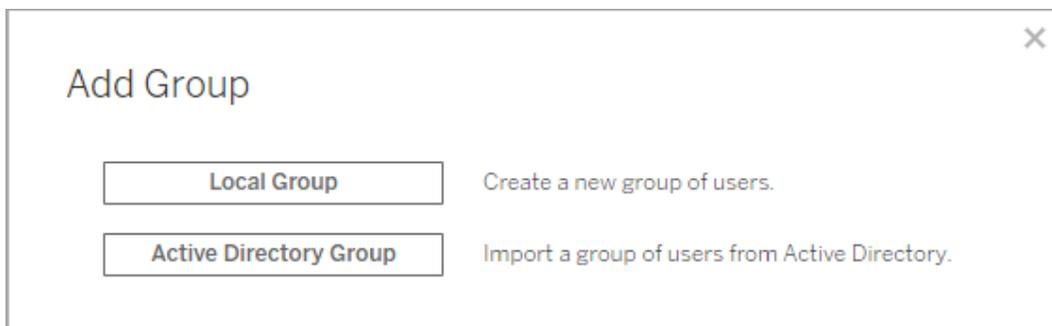


3. In the Add Users dialog box, select the users to be added, and then click **Add Users**.

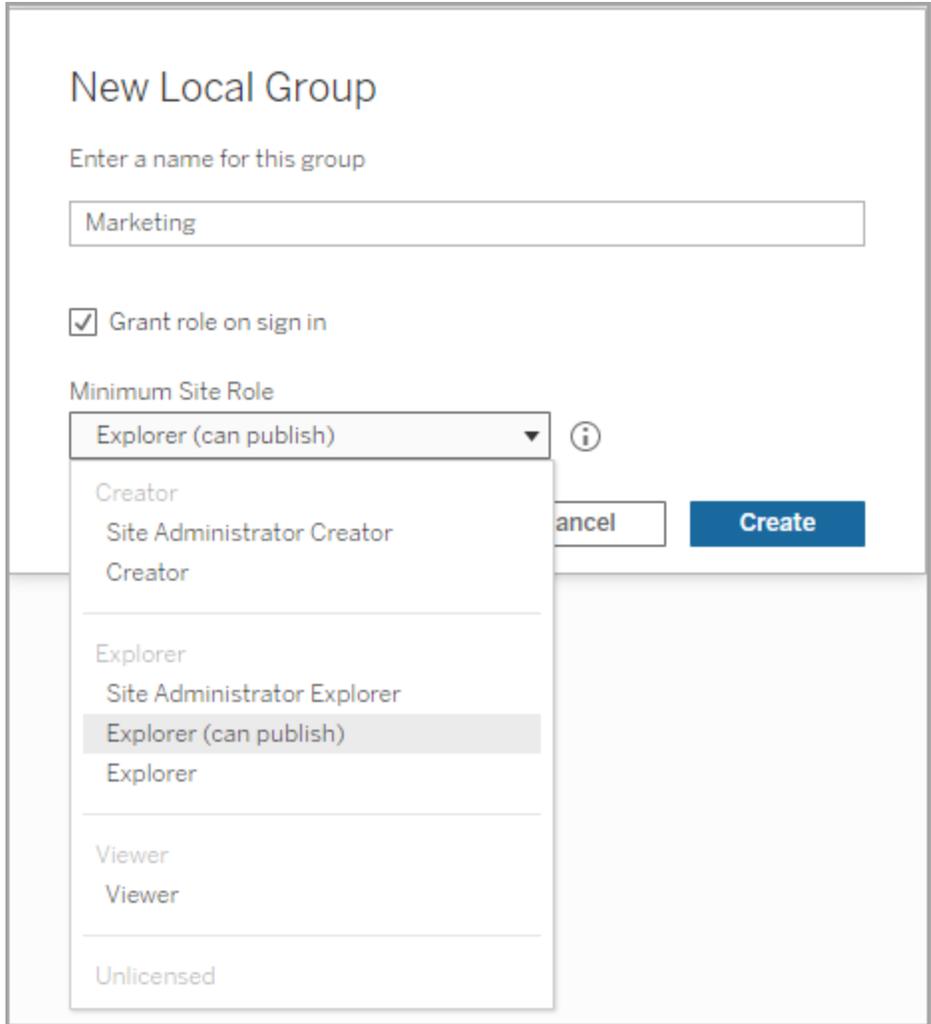
Create a Local Group

Local groups are created using the Tableau Server internal user management system. After you create a group you can add and remove users, as well as set a minimum site role to grant to users in the group when they sign in.

1. In a site, click **Groups**, and then click **Local Group**.



2. Type a name for the group.
3. To set a minimum site role for the group, select **Grant site role on sign in** and select a minimum site role from the drop-down list.



4. Click **Create**.

Create Groups via Active Directory

You can import Active Directory groups to create matching groups on Tableau Server, as well as a user for each member of an Active Directory group that is not already on the server.

Note: In the context of user and group synchronization, Tableau Server configured with LDAP identity store is equivalent to Active Directory. Active Directory synchronization

features in Tableau Server function seamlessly with properly configured LDAP directory solutions.

Each user is assigned a site role as part of the import process. If any of the users to be imported exist in Tableau Server, the site role assigned during the import process is applied only if it gives the user more access to the server. Importing users does not demote site roles.

Before importing groups, review User Management in Deployments with External Identity Stores to understand how multiple domains, domain naming, NetBIOS, and Active Directory user name format influence Tableau user management.

1. In a site, click **Groups**, and then click **Add Groups**
2. Type the name of the Active Directory group you want to import, and then select the group name in the resulting list.

Import a Group from Active Directory

Import a group of users from Active Directory.

marketing

Marketing

Site role: Explorer (can publish) ⓘ

Grant role on sign in

Cancel Import

3. Select the minimum site role for the users.

Import a Group from Active Directory

Import a group of users from Active Directory.

marketing

Marketing

Site role: Explorer (can publish) ⓘ

Grant

Creator
Site Administrator Creator
Creator

Explorer
Site Administrator Explorer
Explorer (can publish)
Explorer

Viewer
Viewer

Unlicensed

Cancel Import

- Optional: Select **Grant role on sign in** to provision new site roles and licenses when group users sign in. For more information, see [Grant License on Sign In](#).
- Click **Import**.

Note: You cannot change the name of groups imported from Active Directory. The

group name can only be changed in Active Directory.

Synchronize External Directory Groups in a Site

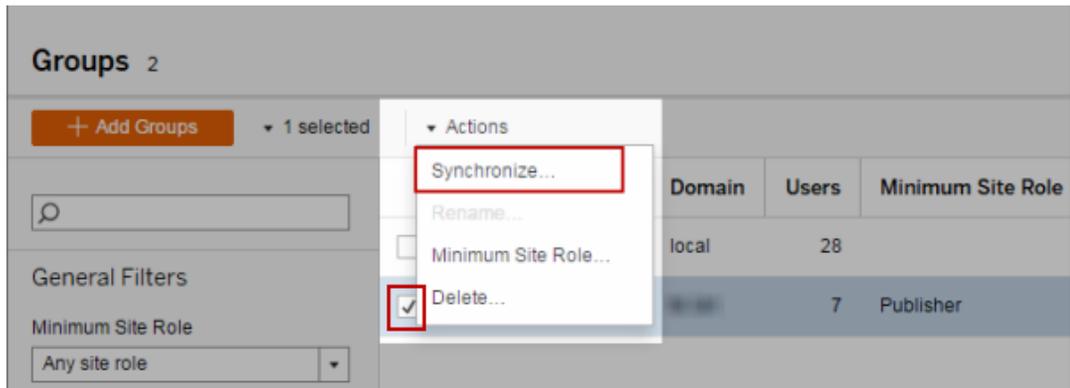
At any time, you can synchronize an external directory (such as Active Directory) group with Tableau Server to ensure new users in the external directory are also added in Tableau Server. You can synchronize individual groups or multiple groups at once.

Note: In the context of user and group synchronization, Tableau Server configured with LDAP identity store is equivalent to Active Directory. Active Directory synchronization features in Tableau Server function seamlessly with properly configured LDAP directory solutions.

1. In a site, click **Groups**.

On the Groups page, select one or more groups.

2. Click **Actions > Synchronize**.



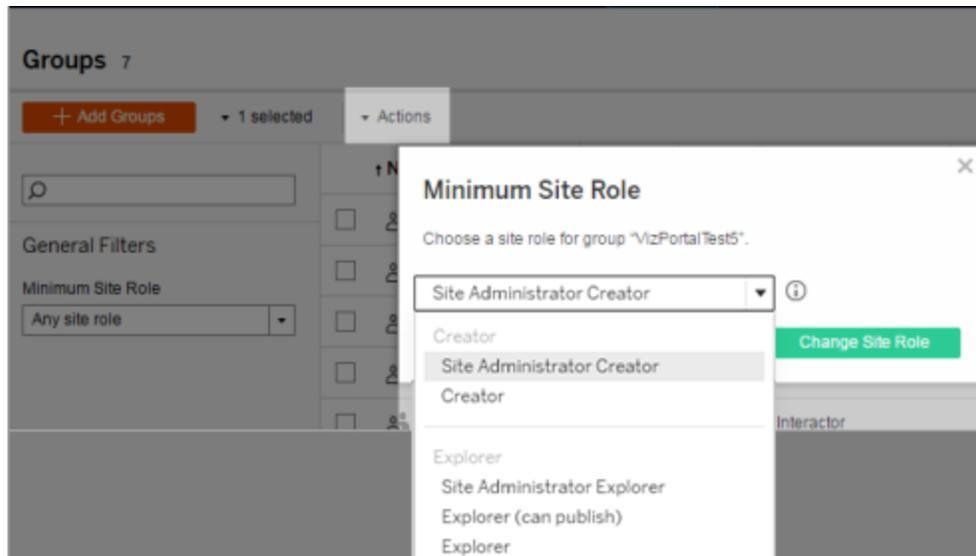
Set the minimum site role for users in an external directory group

In the **Groups - Details** page, administrators can set the minimum site role for group users to apply during synchronization.

This setting does not run synchronization; it sets the minimum site role to applied to the group every time synchronization runs. When you synchronize external directory groups, new users are added to the site with the minimum site role. If a user already exists, the minimum site role will be applied if it gives the user more access in a site. If you don't set a minimum site role, new users are added as **Unlicensed** by default.

Note: A user's site role can be promoted but never demoted based on the minimum site role setting. If a user already has the ability to publish, that ability will always be maintained. For more information on minimum site role, see Site roles and Active Directory import and synchronization.

1. In a site, click **Groups**.
2. On the Groups page, select a group, and then select **Actions > Minimum Site Role**.
3. Select the minimum site role, and then click **Change Site Role**.



What happens when users are removed in the source external directory?

Users cannot be automatically removed from the Tableau Server through an external directory sync operation. Users that are disabled, deleted, or removed from groups in the external directory remain on Tableau Server so that administrators can audit and reassign the user's content before removing the user's account completely. For more information, see [Sync behavior when removing users from Active Directory](#).

What happens when a user name changes in the source external directory?

By default, Tableau Server will not synchronize changes to the user display name or email address after the initial synchronization when the corresponding account is created in Tableau Server. For example, if the user name `jsmith` is used for the display name John Smith, changing the display name in external directory to Joe Smith will not synchronize to the corresponding `jsmith` user in Tableau Server. Similarly, if the user's email changes in the external directory, Tableau Server will not synchronize changes.

You can configure Tableau Server to update the name and email properties when they change in the source external directory by setting `vizportal.adsync.update_system_user` to `true`.

To change this behavior run the following `tsm` commands:

```
tsm configuration set -k vizportal.adsync.update_system_user -v true
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see [tsm pending-changes apply](#).

What happens when an external directory group is removed from Tableau Server?

Many Tableau administrators use external directory groups to import and create users. After the users are imported into Tableau Server, administrators will then delete the group in Tableau Server. Deleting a group does not delete the users in it.

Synchronize External Directory Groups on the Server

As a server administrator, you can synchronize all external directory (such as Active Directory) groups (that have been configured on Tableau Server) on a regular schedule or on-demand on the **General** tab of the **Settings** page for the server.

Active Directory Synchronization

Manage the synchronization of all Active Directory groups. [Learn more](#)

Last synchronized: (Server time)
[View synchronization activity](#)

Synchronize Active Directory groups on a regular schedule

Frequency Hourly Daily Weekly Monthly

at 12 : 00 AM

Note: In the context of user and group synchronization, Tableau Server configured with LDAP identity store is equivalent to Active Directory. Active Directory synchronization features in Tableau Server function seamlessly with properly configured LDAP directory solutions.

Before you begin

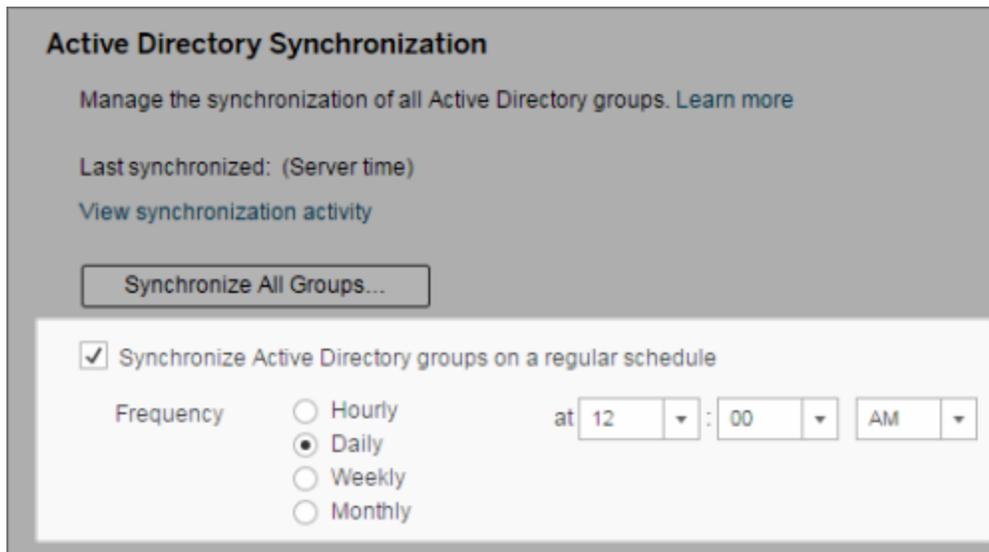
Before synchronizing groups as described in this topic, you must first import the external directory group into Tableau Server. See [Create Groups via Active Directory](#).

Synchronize external directory groups on a schedule

1. **Single-site:** Click **Settings> General**.

Multisite: In the site menu, click **Manage All Sites** and then click **Settings> General**.

2. Scroll down the page to **Active Directory Synchronization**, and then select **Synchronize Active Directory groups on a regular schedule**.



3. Select the frequency and time of synchronization.
4. Click **Save**.

Synchronize all external directory groups on demand

At any time, you can synchronize external directory (such as Active Directory) groups with Tableau Server to ensure that new users and changes in the external directory are reflected in all external directory groups on Tableau Server.

1. **Single-site:** Click **Settings> General**.

Multisite: In the site menu, click **Manage All Sites**, and then click **Settings> General**.

2. Under **Active Directory Synchronization**, click **Synchronize All Groups**.

Note: Beginning in versions 2021.1.23, 2021.2.21, 2021.3.20, 2021.4.15, 2022.1.11, 2022.3.3, a default time limit of 4 hours limits how long a scheduled group synchronization can take before it is canceled. You can change this time limit if your scheduled synchronization is of very large groups, or taking longer than this. For more information, see `Synchronize All Active Directory Groups on a Schedule and background.timeout.sync_ad_group`.

View synchronization activity

You can view the results of synchronization jobs in the **Background Tasks for Non Extracts** administrative view. **Queue Active Directory Groups Sync** is the task that queues and indicates the number of **Sync Active Directory Group** tasks to be run.

1. **Single-site:** Click **Status**.

Multisite: In the site menu, click **Manage All Sites** and then click **Status**.

2. Click the **Background Tasks for Non Extracts** link.
3. Set the **Task** filter to include **Queue Active Directory Groups Sync** and **Sync Active Directory Group**.

You can quickly navigate to this administrative view by clicking the **View synchronization activity** link in the **Settings** page for the server.

Set the minimum site role for users in an external directory group

In the **Groups - Details** page, you can set the minimum site role for group users to be applied during Active Directory synchronization.

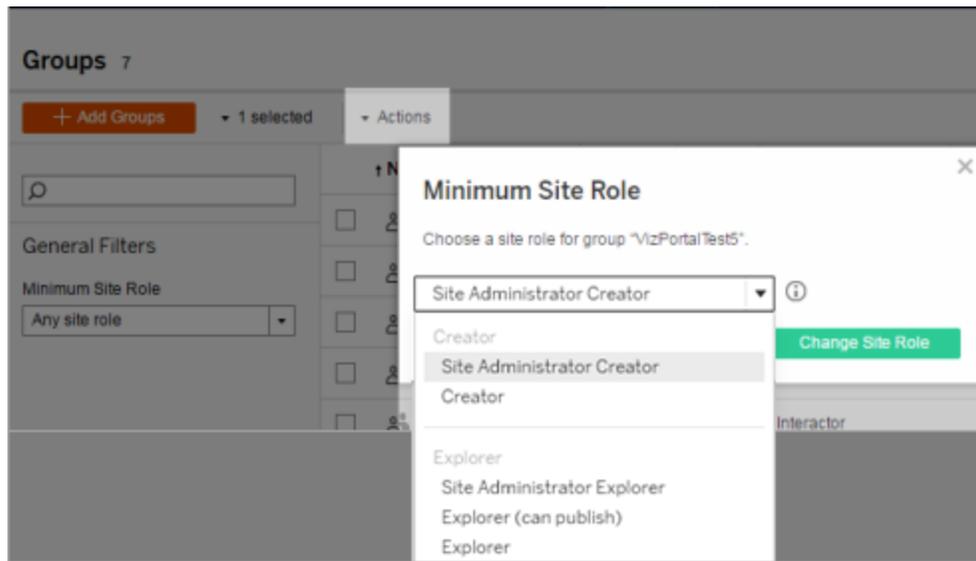
This setting does not run synchronization; instead, it sets the minimum site role to applied to the group every time synchronization runs. The result is that when you synchronize external directory groups, new users are added to the site with the minimum site role. If a user already exists, the minimum site role is applied if it gives the user more access in a site. If you don't set a minimum site role, new users are added as **Unlicensed** by default.

Note: A user's site role can be promoted but never demoted based on the minimum site role setting. If a user already has the ability to publish, that ability will always be maintained. For more information on minimum site role, see Site roles and Active Directory import and synchronization.

1. In a site, click **Groups**.
2. On the Groups page, select a group.

Click **Actions > Minimum Site Role**.

3. Select the minimum site role, and then click **Change Site Role**.



What happens when users are removed in the source external directory?

Users cannot be automatically removed from the Tableau Server through an external directory sync operation. Users that are disabled, deleted, or removed from groups in the external directory remain on Tableau Server so that administrators can audit and reassign the user's content before removing the user's account completely. For more information, see [Sync behavior when removing users from Active Directory](#).

Improving group synchronization performance

External directory synchronization is performed by the background process. The Background process is the same process that is used for managing and creating extracts, and is also used to generate subscription content. In large organizations with dynamic group membership and heavy extract usage, the external directory group synchronization process may be disruptive. We recommend running group synchronization during non-business hours.

By default, the Background process performs synchronization in a serial operation. This means that each group is synchronized, one after the other, in a single Background process. If you are running multiple instances of Background processes either on a single

Tableau Server or across a distributed deployment, consider enabling parallel processing for external directory synchronization. When parallel Backgrounder processing is enabled, the group synchronization is distributed across multiple Backgrounder processes for better performance.

To enable parallel backgrounder processing for group synchronization, open TSM CLI and enter the following commands:

```
tsm configuration set -k backgrounder.enable_parallel_adsync -v true  
  
tsm pending-changes apply
```

Synchronize All Active Directory Groups on a Schedule

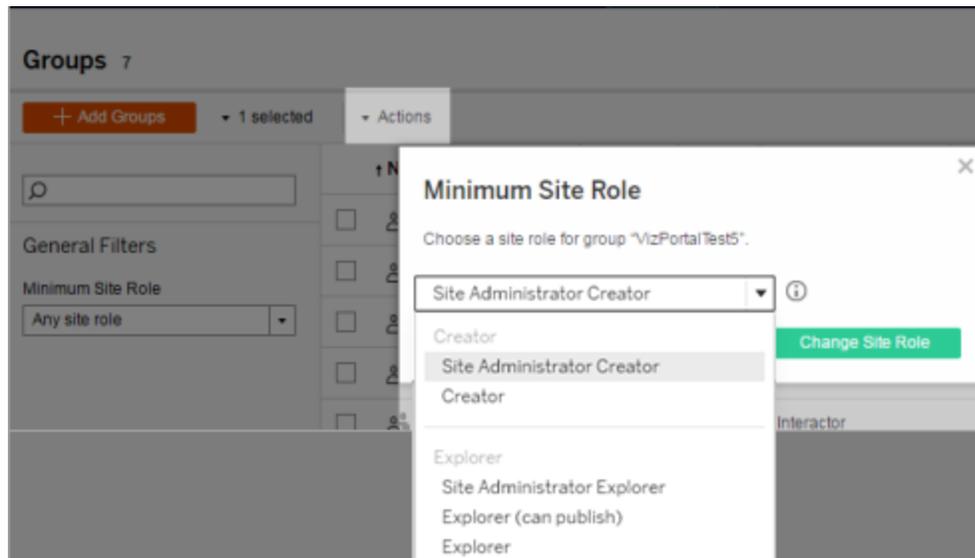
After you import Active Directory groups in Tableau Server, you can make sure they stay synchronized in Tableau Server by setting up a schedule. You can also synchronize all Active Directory groups on the server on-demand, at any time. The minimum site role setting for the group is applied when users are synchronized.

Note: In the context of user and group synchronization, Tableau Server configured with LDAP identity store is equivalent to Active Directory. Active Directory synchronization features in Tableau Server function seamlessly with properly configured LDAP directory solutions.

Beginning in versions 2021.1.23, 2021.2.21, 2021.3.20, 2021.4.15, 2022.1.11, 2022.3.3, you can set a maximum time limit for how long a scheduled group synchronization can take before it gets canceled. The default time limit is 4 hours. For more information, see `backgrounder.timeout.sync_ad_group`.

1 Set a minimum site role for synchronization

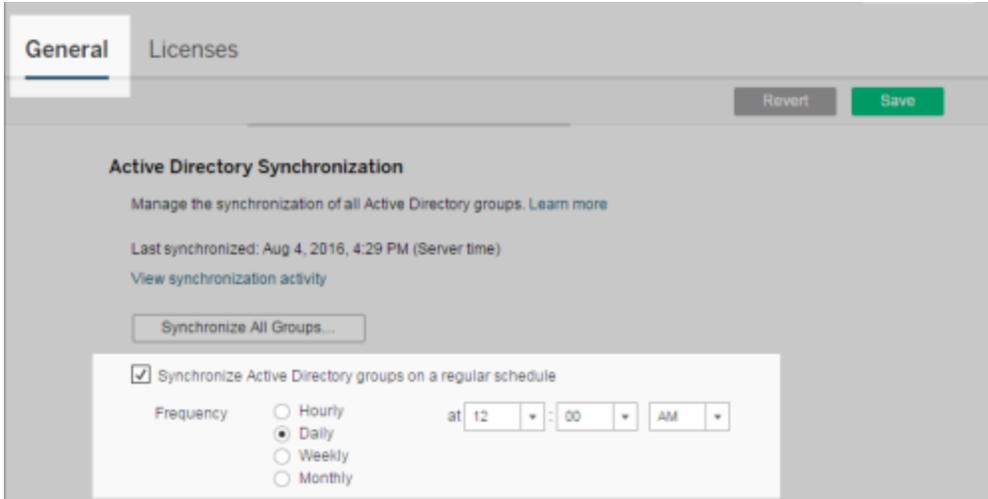
In a site, click **Groups**. Select a group, and then click **Actions**> **Minimum Site Role**. Select the minimum site role, and then click **Change Site Role**. Server and site administrators can set the minimum site role for group users to be applied during Active Directory synchronization. If you don't set a minimum site role, new users are added as **Unlicensed**.



Synchronizing can promote a user's site role, but will never demote a user's site role.

2 Set the schedule

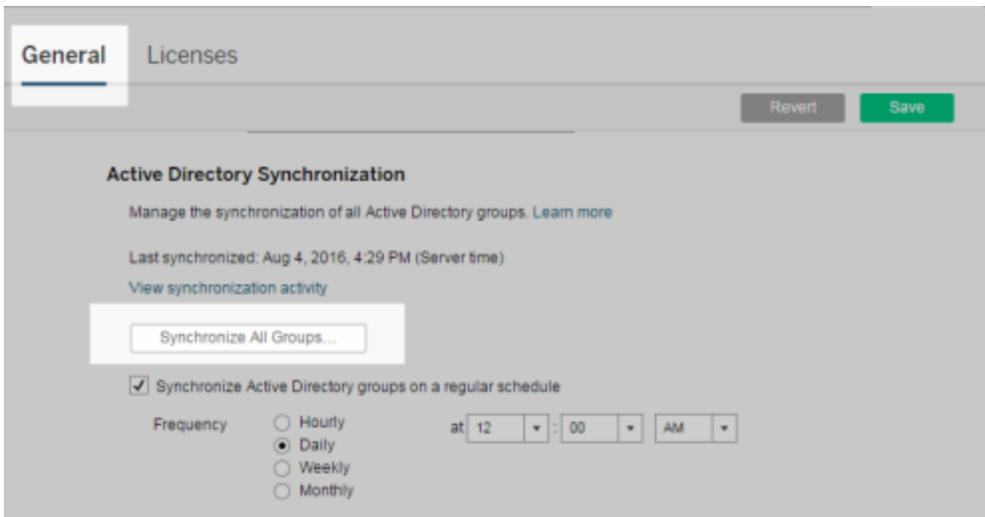
Server administrators can enable synchronization for all Active Directory groups on the **General** tab of the **Settings** page for the server. Enable synchronization, select the frequency settings, and then click **Save**.



All Active Directory groups on the server are synchronized according to the same schedule.

3 Run synchronization on-demand (optional)

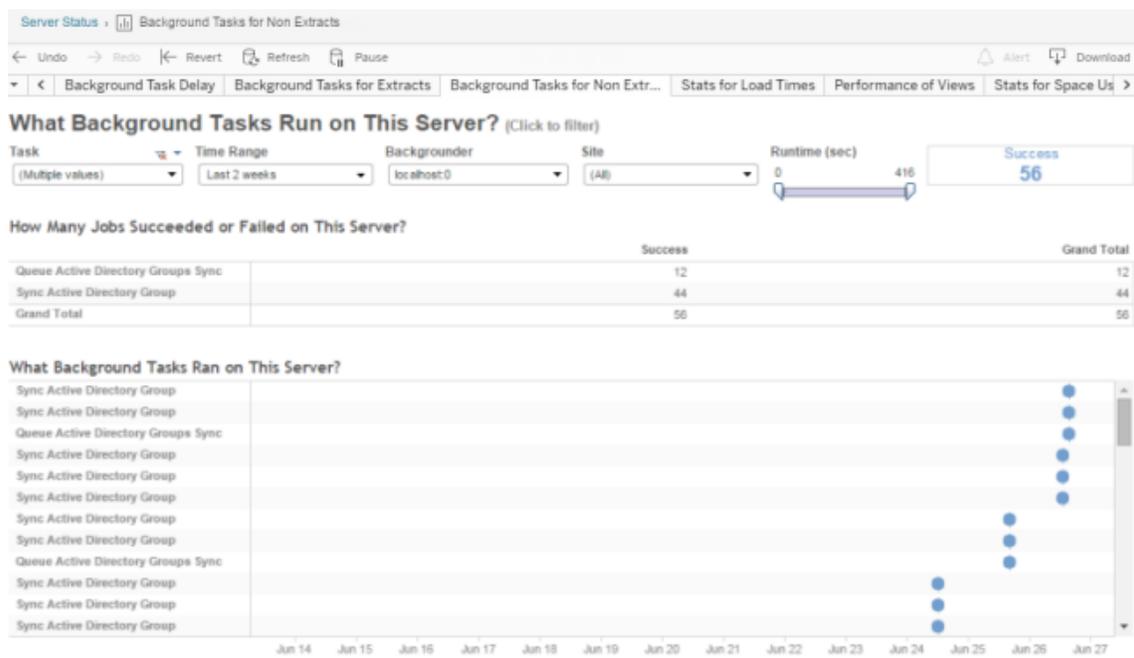
On the **General** tab of the **Settings** page, click **Synchronize All Groups** to synchronize all Active Directory groups on Tableau Server immediately. Click this button at any time to ensure new users and changes are reflected in all Active Directory groups on the server.



Click **Synchronize All Groups** to synchronize all Active Directory groups on the server outside of a schedule.

4 View the status of synchronization tasks

Server and site administrators can view the results of Active Directory synchronization jobs in the **Background Tasks for Non Extracts** administrative view. On the server or in a site, click **Status**. Under **Analysis**, click **Background Tasks for Non Extracts** and filter on the **Queue Active Directory Groups Sync** and **Sync Active Directory Group** tasks.



Queue Active Directory Groups Sync queues the **Sync Active Directory Group** tasks to be run.

Grant License on Sign In

Grant license on sign in (Grant role on sign in) lets unlicensed users in specific groups become licensed when they sign into a Tableau site. This streamlines license provisioning for administrators and removes the user's need to request a license before using Tableau.

For more information about site role capabilities and minimum site roles, see [Set Users' Site Roles](#).

For example, imagine an Active Directory group called Marketing with 100 users, but only 25 users need to access Tableau Server. A site or server administrator can import all users in the Marketing Active Directory group, set the group's minimum site role to Explorer, and select **Grant role on sign in**. When any of the Tableau users in Marketing sign into their Tableau site, they'll be granted Explorer licenses. Users who don't need Tableau Server remain unlicensed unless they sign in.

Note: For more information about benefits and best practices, see [Grant Role on Sign In](#) in Tableau Blueprint, Tableau's planning tool for data-driven organizations.

Activate Grant role on sign in

You can enable Grant role on sign in on new or existing groups. The following steps walk through how to use Grant role on sign in to add new users that are eligible for a license but may not consume one. This may be the case when your company has a lot of eligible users, but limited Tableau licenses.

1. In a site, click **Groups**, and then click **Add Group**.

Add new users by importing an Active Directory group. Type the name of the group you want to import, and then select the group name in the resulting list.

Import a Group from Active Directory

Import a group of users from Active Directory.

marketing

Marketing

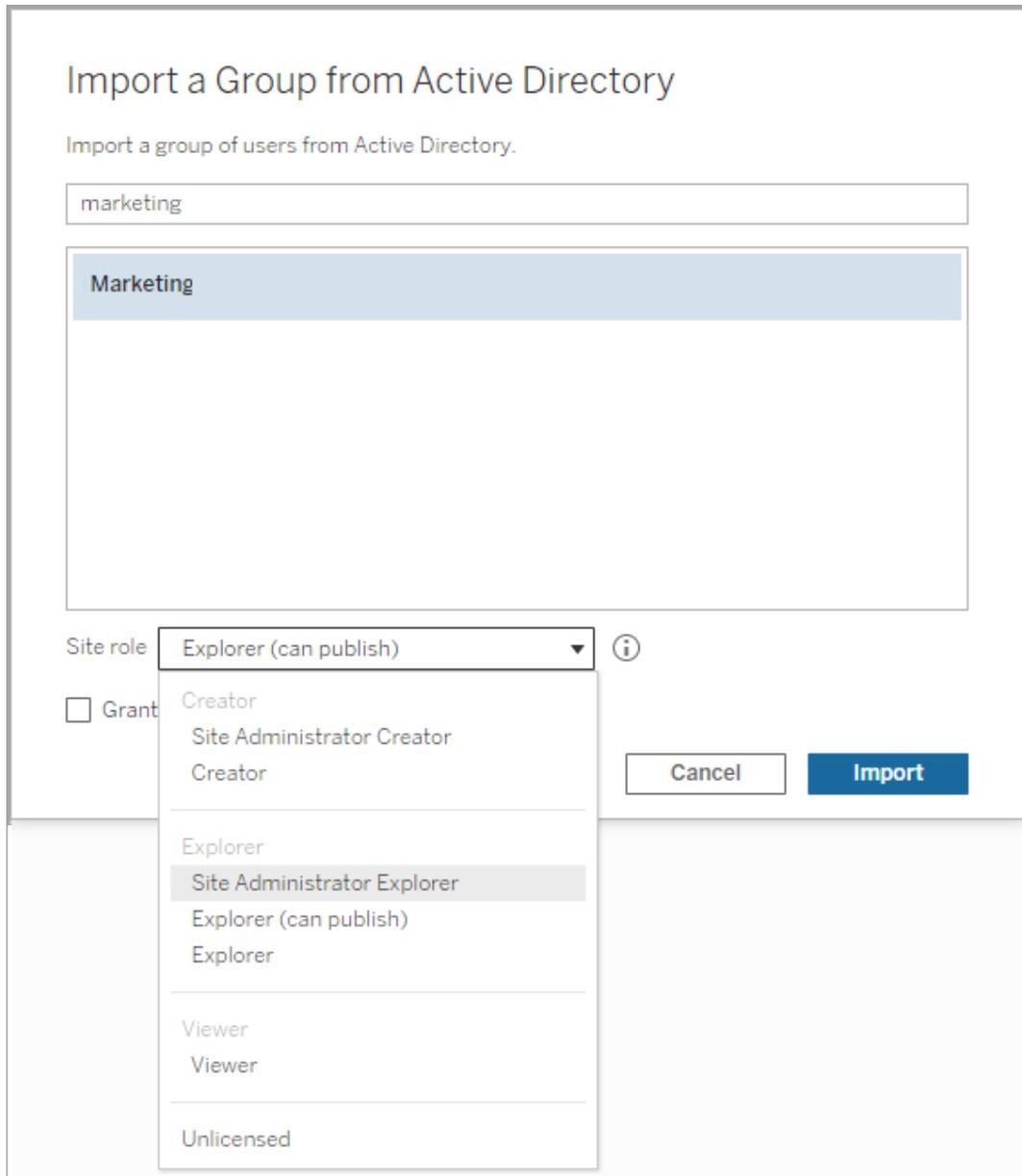
Site role Explorer (can publish) ⓘ

Grant role on sign in

Cancel Import

2. Select the minimum site role for the users, and select **Grant role on sign in**.

All users in the selected Active Directory group will be imported as unlicensed users. The minimum site role set for the group will only be provisioned to group users who sign into Tableau Server.



3. Click **Import**.

Note: Grant Site Role on Sign In can also be activated in local groups to provision minimum site roles to group members when they sign in to Tableau Server. For more information, see [Create a Local Group](#).

Modifying user roles with Grant role on sign in

If a user is part of a group using Grant role on sign in, then that user role can't be set to unlicensed or downgraded to a role lower than the minimum site role set for the group, whether or not they sign in. Administrators can upgrade a user's site role manually, however.

To downgrade a user's site role, or unlicense the user from the site, remove the user from the group(s) that have Grant role on sign in enabled.

In accordance with the terms of the [End User License Agreement](#), licenses granted on an Authorized User basis may be permanently reassigned to new users. Users may only be downgraded to a lower site role (including Unlicensed) when they will permanently discontinue access to Server Software at the higher role.

Removing users affected by Grant role on sign in

You can remove a user from a site only if the user does not own content. If you attempt to remove a user who owns content, the user site role will be set to Unlicensed and removed from all groups, but the user will not be removed from the site. To remove content owners, remove owners from group with Grant site role enabled or reassign content ownership to another user. For more information, see [Remove users from a site in the View, Manage, or Remove Users help topic](#).

If the default All Users group has Grant site role enabled, users who own content can't be removed from the site or unlicensed. To remove or unlicense these users, reassign content ownership to another user, then remove or unlicense the user.

REST API can be used to reassign content ownership of a workbook. For more information, see [Update Workbook](#) in the REST API documentation. REST API can also be used

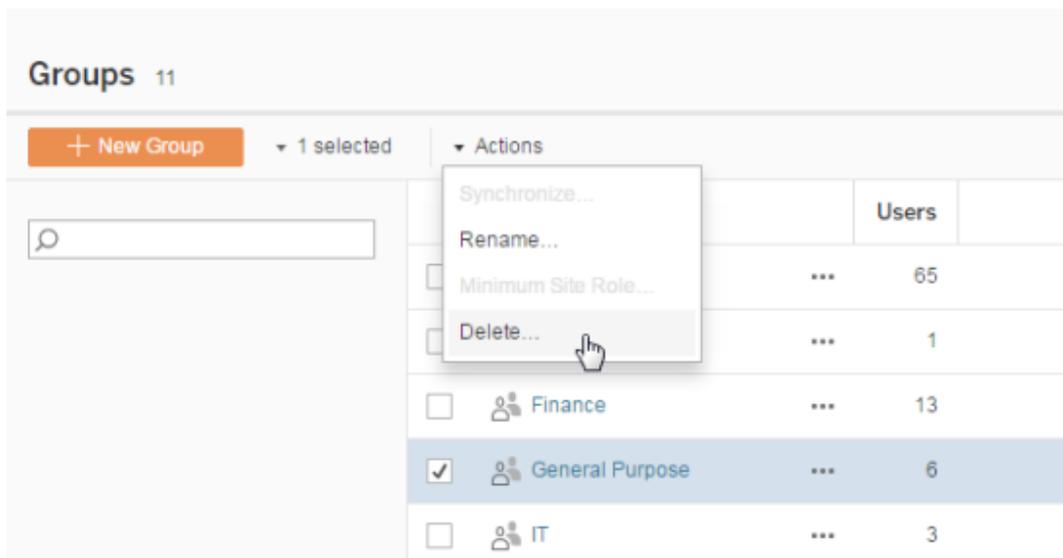
to remove users from the site and transfer content ownership to another user. For more information, see [Remove User from Site](#) in the REST API documentation.

For more information on changing content ownership in Tableau Server, see [Manage Content Ownership](#).

Delete Groups

You can delete any group from the server (with the exception of the All Users group). When you delete a group, the users are removed from the group but they are not deleted from the server.

1. In a site, click **Groups**.
2. On the Groups page, select one or more groups to delete.
3. Select **Actions > Delete**.



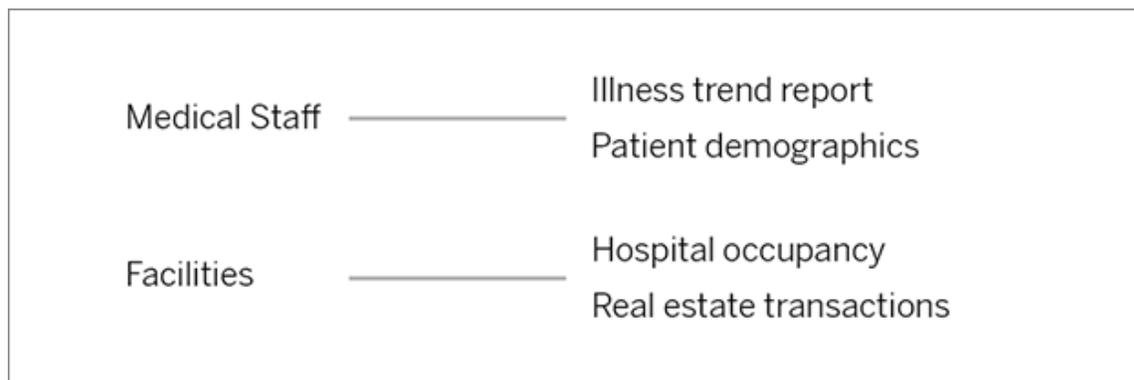
Dashboard-based Custom Portals

Note: This overview was inspired by the work of Tableau Zen Master Mark Jackson. For more details about the process, [check out Mark's blog](#).

The [standard Tableau Online or Tableau Server interface](#) works great for many organizations, but if you want to create a company-branded experience and don't have API developer skills, consider creating a custom portal based on a Tableau dashboard. A custom portal lets you organize content around specific departments or job roles, and you can even incorporate training that helps people interact with data more fully. As the volume of Tableau content grows, a custom portal guides your users directly to the data they need, while providing ready access to related views and Tableau Server search so they can easily explore further.

Sketch out a portal design

Begin outside of Tableau, on paper or in a wireframing application. Consider the structure of your organization and the number of Tableau views that apply to each group of users. Do you simply need one level of navigation that points immediately to content? Or might you need to start with a main page that provides navigational links for separate groups of users or report types, followed by a second level with related dashboards and views?



After you get buy-in on your design from key stakeholders and data consumers, you're ready to move on to the next phase.

Gather images for logos and navigation elements

Start thinking about images while refining your initial mockup, and then gather them from libraries of clip-art or approved brand graphics, or create them from scratch in an application like Photoshop or SnagIt. PowerPoint can also be a good source if you plan to create navigation thumbnails of common chart types.



Lay out text, images, and selected sheets on a dashboard

Create a workbook with a dashboard for the portal, using a [tiled layout](#) for more predictable positioning and scaling of elements on different devices. Then start adding Text and Image objects, as well as any sheets for data views you want to highlight right up front in your portal. To refine spacing between these dashboard elements, insert Blank objects and adjust their size. For more information, see [Create a Dashboard](#).

Tip: For dynamic elements that show popular or recently created content, include sheets based on [custom administrative views](#).

Link dashboard elements to content

If you have a second level of navigation in your portal, use [filter actions](#) to point to a secondary dashboard from the main one. To create links that directly open data views, right-click Image objects and choose Set URL. (In our example below, each colored arrow and accompanying text is an image that links to a view URL.) You can even point to empty views with

preloaded data sources, encouraging users to create new Tableau content in the web-authoring workspace.

Tip: To add hyperlinks to text objects, include the full URL (for example, <http://www.tableau.com>).

Health Group Data Portal

- Medical
 - Illness trend report
 - Patient demographics
- Facilities
 - Hospital occupancy
 - Real estate transactions
- Accounting
 - Billing by department
 - Year over year taxes

Learn how to explore geo data

Create your own view of geo data

Publish, test, and refine your portal

Publish the workbook to your server and distribute the dashboard URL to your users. Portal design, like data analysis itself, is a cyclical process. Now that your portal is out in the wild, start gathering user feedback so you can continuously improve the experience

Tip: As a finishing touch, hide the Tableau toolbar to give your portal a custom feel. After the question mark at the end of the dashboard URL, add `:embed-d=y&;:toolbar=n`

Manage Content Access

You can manage who can access content on your site and set the permissions that govern content ownership.

Set a Site's Web Authoring Access and Functions

Tableau Server administrators can specify at the site level whether to allow users to edit published views in the web environment and configure other web authoring functionality.

By default web authoring functionality is enabled for all sites. Users with the **Web Edit** capability can create and edit workbooks directly on the server. Turn off web authoring if you want users to be able to view and interact with published workbooks but not make any changes to the core information.

The steps below describe how to set web authoring and other associated functionality for an entire site. For more granular control over which users can use web editing, you can use projects, groups, and permissions. See [Set Web Edit, Save, and Download Access on Content](#).

For information about how to enable authoring for flows on the web, see [Create and Interact with Flows on the Web](#).

Turn web authoring on or off for a site

1. In a web browser, sign in to the server as an administrator and go to the site in which you want web authoring to be enabled. In that site, click **Settings**.
2. In the **Web Authoring** section, select **Workbooks. Let users edit workbooks in their browser.** to enable the functionality.

Clear the check box to turn off web authoring for that site.

Web Authoring

Users with the appropriate permissions can edit content in their browser.

- Workbooks. Let users edit workbooks in their browser.
- Flows. Let users edit flows in their browser.

3. If your site is already in production, and you want the change to take effect immediately, restart the server.

Otherwise, the change takes effect after server session caching expires or the next time users sign in after signing out.

Notes

- When you enable web authoring, make sure that, on the appropriate workbooks or views, the permission rule for a user or group allows the **Web Edit** capability.
- If you turn off web authoring on a production site and do not complete the last step to restart the server, users might continue to have authoring access until their session caches expire or they sign out.

See which sites allow web authoring

To confirm which sites allow web authoring, on the site-selection menu at the top, select **Manage All Sites**, and then go to the **Sites** page.

Sites 9											
+ New Site 0 selected											
Name	Users	Site administrators	Max users	Storage used	Max storage	Status	Metrics	Web authoring			
<input type="checkbox"/> Customer Support	...	4	2	Server limit	0 B	Server limit	Active	✓			
<input type="checkbox"/> Default	...	63	8	Server limit	25.6 MB	Server limit	Active	✓	✓		
<input type="checkbox"/> Development	...	4	2	Server limit	0 B	Server limit	Active	✓	✓		
<input type="checkbox"/> Documentation - 20 User Limit	...	5	1	20	3.2 MB	Server limit	Active	✓			
<input type="checkbox"/> Finance	...	13	2	Server limit	9.8 MB	Server limit	Active	✓	✓		

Configure cross-database join options

To improve performance for cross-database joins, users can allow Tableau to perform the join using the live database they are connected to instead of using Hyper. While this option is faster, if Tableau uses the connected database to perform the join, data from the file data source that the user is connected to is temporarily moved into temp tables in the database. Because this moves data outside of Tableau, as an administrator you may want to restrict access to this feature for users with web authoring permissions.

1. In a web browser, sign in to the server as an administrator and go to the site in which you want web authoring to be enabled. In that site, click **Settings**.
2. In the **Cross-Database Joins** setting, select one of the following options:
 - **Use Tableau or existing databases** - Select this option if you want to allow users to choose whether they want to allow Tableau to use the live database to perform cross database joins. Published data sources with this option enabled will continue to use the user's database for cross-database joins.
 - **Use Tableau only** - Select this option to restrict users to use only Hyper to perform cross data-base joins.

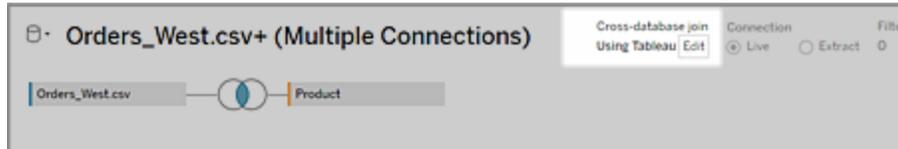
Cross-Database Joins

Choose where the join process happens when joining data from multiple sources. Allowing Tableau to use an existing database connection can be faster, but this means that data from the file connection may be temporarily moved to the live database connection. [Learn more](#)

Use Tableau or existing databases

Use Tableau only

If you select **Use Tableau only**, the option to choose how Tableau performs the cross-database join won't display in the canvas when the user connects to a supported data source and supported database. For more information about this feature, see [Improve performance for cross-database joins](#).



Set Web Edit, Save, and Download Access on Content

If you're enabling web authoring functionality on your site, you can configure more precisely which users on the site have access to this functionality. Using site roles and permissions rules at the content level, you can grant or deny **Web edit**, **Save**, or **Download** capabilities on projects, workbooks, and data sources.

Note: This document strives to use the phrase *Web edit* to specify the name of the capability in permissions rules, and *web authoring* to refer to the general functionality of creating and modifying workbooks on the server. However, you might otherwise see these two phrases used interchangeably.

Why allow users to work on the site directly

As an administrator, your initial thought about allowing people to populate a site with content, seemingly indiscriminately, might be one of skepticism. However, with a few controls, you can limit where this is done, while providing important benefits that centralized content management offers both you and your users.

Web authoring pros and cons

For publishers and business users, some benefits of web authoring include the following:

- It provides analyst teams who work collaboratively with a central location in which to provide input.

Tableau Server on Linux Administrator Guide

- It enables people who do not have Tableau Desktop to connect to data sources and create workbooks.
- It enables people to access content when they are away from their Tableau Desktop computer or VPN, whether on a computer or a hand-held device.
- It can provide a framework for enabling consistency across Tableau reports. (By making template workbooks available on the site, analysts can download or create new workbooks with data connections, branding, and formatting already in place.)

For administrators, benefits can include the following:

- Fewer Tableau Desktop deployments to manage and support.
- Fewer computers that need to have database drivers installed.
- Capacity to govern content.
- More accurate monitoring of what people are doing with Tableau.

Some disadvantages to web editing include the following:

- For analysts, web editing functionality is not as extensive as in Tableau Desktop (although it continues to evolve toward that parity).
- For administrators, more people working on the server might mean upgrading systems.
- Without publishing guidelines, content proliferation on the site is expected. This can confuse the people who rely on published Tableau dashboards and data sources, degrade server performance and data quality, and potentially affect data security.

Managing permissions to help users avoid content proliferation

To help users to avoid content proliferation on the site, many Tableau administrators use projects to allow varying levels of access to content. For example, one project can be configured to allow all users to edit and save workbooks; another can allow only approved publishers to save new content.

To get a better idea how this works, see the following resources:

- [Configure Projects, Groups, and Permissions for Managed Self-Service](#)
- [Projects and Content Permissions](#) in *Everybody's Install Guide*

- [Governed Self-Service at Scale](#), a Tableau whitepaper by Rupali Jain.
To view the PDF, you might need to provide your Tableau website credentials. These are the same ones you use for the community forums or to submit support cases.

Coordinate edit and save capabilities with site roles for the appropriate level of access

To edit, save, and download workbooks, users must have a site role that allows those actions, along with the capabilities—defined in permissions rules—that grant or deny editing-related access.

Site role access

- When the appropriate permissions are set at the content level, the **Creator** or **Explorer (can publish)** site role allows both **Save** (overwrite) and **Save As/Download**.

Note that **File > Save** is only available to the workbook owner. When the **Save** permission capability has been granted at the project and workbook level, a non-owner user can overwrite the existing workbook in web authoring by selecting **File > Save As** and using the same workbook name. This overwrites the existing content and they become the owner and gain full access to the content.

- The **Explorer** site role can be granted the **Web Edit** and **Save As/Download** capabilities, but they will not be able to save (neither overwriting existing nor saving changes to a new workbook).

For more information, see [Web Editing and Web Authoring](#).

Configure Projects, Groups, and Permissions for Managed Self-Service

Tableau Online and Tableau Server each provide an environment for easy open publishing and collaborative analysis of visualizations created in Tableau Desktop or web authoring. With that flexibility comes the challenge of making sure the *right* content is easy to find for

the people who rely on it for their work. Likewise, making sure the access you allow doesn't create performance or management nightmares on the site.

To address these challenges, many administrators set up their Tableau sites for what we'll refer to as *managed self-service*. This is just a way of saying that the site allows areas of open collaboration and web editing, alongside areas in which access to data and reports is more controlled. As the site administrator, you put guidelines in place to help users figure out where to go for the type of work they need to do.

To get started with a managed self-service approach, the following sections discuss how you as the site administrator can meet the following objectives:

- Create projects on the Tableau Server or Tableau Online site to match the ways people need to work with content.
 - For example, some projects are open to all for collaboration; others are visible only to authorized publishers.
- Create user groups based on the type of access users need to the content.
- Create a clear and scalable permissions strategy.

Note: The information provided here is adapted and simplified from practices of existing Tableau Zen Masters and customers who have shared their experiences. Links to their talks are available at the [bottom of this page](#).

Create a project team and adopt a permissions strategy

Although changing the project structure on your site after your users are publishing to it is not impossible, it's difficult and can be daunting. So before you make any lasting decisions or take definitive actions on your Tableau site, we recommend that you recruit users from various segments of your Tableau population, to create a project team of people who have differing uses for Tableau content.

Your permissions strategy will help your environment scale as you add new Tableau users. Make sure it incorporates two important practices: manage permissions only for groups, and set permissions only at the project level. Setting permissions at the individual user level and

on individual content resources becomes unmanageable quickly. If you need to deviate from this practice, make sure you document and communicate your strategy to other administrators and project leaders.

Important: We strongly recommend familiarizing yourself with Tableau's Permissions before proceeding.

Steps to coordinate projects and groups

To get projects and permissions (content) to work together with groups (people) in a managed self-service environment, you generally take the following steps:

1. 1. Plan your permissions: Find common themes in the type of access users need. This helps determine projects and groups.
2. 2. Remove permissions that will cause ambiguities
3. 3. Create groups
4. 4. Assign permissions to the groups
5. 5. Create projects and adjust permissions
6. 6. Lock permissions in each project

If you decide to follow the guidelines described here, you might want to Automate working with groups and projects.

1. Plan your permissions

Before you create groups and start assigning permissions, create a list of people who need access to content, and arrange them in groups according to what they'll want to do.

For example, someone who publishes or moves a data source to a *certified* content project would need different level of access than someone who only consumes published reports. (We use the term "certified" to mean "trusted" — these are the data sources or reports that your Tableau community can trust to be a source of truth for your organization.)

Keep in mind also that you can set permissions differently for each project. So someone who is a data steward for the Ops department might not get the equivalent access to the Marketing content.

This exercise, done outside of the Tableau environment, can be the most challenging part of setting up a site.

Use a closed permissions model for managed content

General models for setting permissions are *open* or *closed*. In an open model, users get a high level of access, and you explicitly deny capabilities. This model can work when your organization is very small, and everyone has a similar level of responsibility.

In a closed model, users get only the access they need to do their jobs. This is the model security professionals advocate, and the examples in this article will attempt to show.

2. Remove permissions that will cause ambiguities

Every site has a **Default** project and an **All Users** group. Any user added to the site becomes a member of the All Users group automatically. The Default project works as a template for new projects in the site and cannot be deleted, but you can change the permissions. Creating groups and setting baseline permissions here helps you to know and manage exactly who gets what level of access for each new project.

In the managed self-service context, setting baseline permissions means *removing* the permissions from the **All Users** group, so that the permissions are enabled only on groups you create and have control over.

1. Select the **Content** tab to open the top-level projects on the site.
2. On the **Default** project's **Action (...)** menu, select **Permissions**.
3. Next to the **All Users** group name, select **...**, and then select **Edit**.
4. For the tabs for **Project**, **Workbooks**, and **Data Sources**, use the template drop down and select **None**.
5. Select **Delete** to apply the changes.

3. Create groups

You create groups to match what people need to do with a set of content. In this case “a set of content” refers to the workbooks and data sources in a project.

When you create your groups, use descriptive names that make sense for your organization. For example, one possible set of groups might be as follows:

- **Project leaders.** You might also think of these as project-level administrators. Users who can perform all available capabilities on data sources, with the possible exception of setting permissions on them. People in this group can be site administrators, or users whose job it is to approve or certify data models or reports. To grant administrator capabilities at the project level, you can assign the **Project Leader** setting to users with the appropriate site roles. For more information, see [Permissions](#).
- **Analysts/Publishers.** This group is for users who can publish workbooks to production and other open projects, use web editing on some projects, and connect to data sources certified by the data stewards. This group is not allowed to set permissions on content or move it between projects.
- **Business Users.** This group is the most likely to include people who do not use Tableau Desktop, but use data to answer questions and make business decisions. They can view and interact with workbooks only in specific projects, and they can't publish, edit, save, or delete anything.
- **Administrators.** Depending on the size of your deployment, managing site or server administrators as a group helps you keep track of who has that level of access.

Note: Users with the Server Administrator or Site Administrator Creator site role have access to everything on the site, regardless of the groups you add them to.

If you have multiple Tableau roles per department, creating corresponding groups manually can be labor intensive. For alternatives, see [Automate working with groups and projects](#) later in this article.

Learn more: Add Users to a Group

4. Assign permissions to the groups

After you create groups, you can assign permissions in one of the following ways:

- In the Default project, apply a core set of permissions on each group that will stay more or less the same for all projects. You can then make minor adjustments in specific projects.
- Or
- Keep the Default project clean, and apply permissions only on projects you create.

For more information, see Permissions.

For the example we're using, it makes more sense to set permissions templates in the Default project. You will want to explicitly deny some capabilities across the board, and then allow them on only a few projects where you want to allow more open access.

Create permission rules

1. While you have the Default project open, on the **Actions** menu (...), select **Permissions**.
2. Create a permission rule for each group as follows:
 - a. Click **+ Add Group/User Rule** and start typing to search for a group or user.
 - b. For each tab, choose an existing template from the drop-down or create a custom rule by clicking the capabilities.
 - i. Templates are predefined sets of capabilities that make setup easier.
 - ii. One click sets the capability to **Allowed**, two clicks sets it to **Denied**, and a third click clears the selection (**Unspecified**).
 - c. When finished, click Save.
3. **Lock permissions to the project.**

Remember, a capability is only granted to a user if they are expressly allowed it. Leaving a capability as Unspecified will result in it being denied. For more information, see Permissions.

Example

For the groups defined above, here is one way you might set default permissions.

	Project tab	Workbooks tab	Data Sources tab
Project Leader group			
Analyst Publisher group	Publish template	Publish template	Publish template
Business User group	View template	Explore template	Explore template
		Set Web Edit and Download Full Data to Unspecified*	

* This assumes you want to allow web editing and downloading data only on select projects. You can allow those capabilities on specific projects or workbooks.

5. Create projects and adjust permissions

After the Default project is set with your custom permissions template, you can create projects that allow the content use cases you identified. For each project, you can adjust the default permissions as appropriate.

Example project structure

One way to structure projects could be to reflect the following use cases:

Workbooks shared for open collaboration on the server

Anyone in the department can publish to the open-collaboration project while their content is in development. Colleagues can collaborate using web editing on the server. Some

people call this a sandbox, some call it staging, and so on. On this project you can allow web editing, saving, downloading, and so on.

Here you want not only to enable collaboration, but also to enable people who don't have Tableau Desktop to contribute and provide feedback.

Shared reports that cannot be edited

This could be a project that people who create workbooks and data sources (Analysts and Data Stewards) could publish to when they want to make content available to business users for viewing, with confidence that their work cannot be “borrowed” or modified.

For this type of project, you would deny all capabilities that allow editing or getting the data off of the server for reuse. You would allow viewing capabilities.

Vetted data sources for Analysts to connect to

This would be where Data Stewards publish the data sources that are meet all of your data requirements and become the “source of truth” for your organization. Project leaders on this project can certify these data sources, so that they rank higher in search results and are included in recommended data sources.

You would allow authorized Analysts (that is, the Publishers group described earlier) to connect their workbooks to data sources in this project, but not download or edit them. You would deny the view capability to the Business Users group for this project, so those users would not even see this project.

Inactive content

Another possibility is to segregate workbooks and data sources that the site's administrative views show haven't been used for a period of time. You could give content owners a time limit before their content is removed from the server.

Whether you do this or delete directly from the working projects is up to your organization. In an active environment, don't be afraid to be intentional about removing content that is not being used.

Source for workbook templates

This is a project that people can download from but not publish or save to, where authorized publishers or project leaders make template workbooks available. Templates that have your organization's approved fonts, colors, images, and even data connections built in can save authors a lot of time and keep your reports looking consistent.

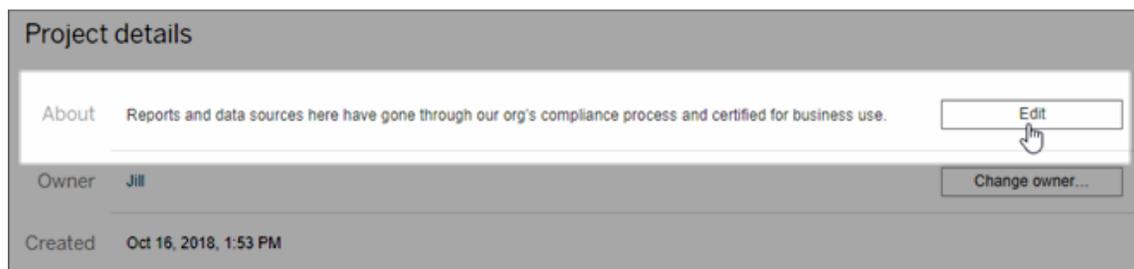
Help project leaders manage content and users find it

- Devise a scalable project-naming scheme that makes sense in your organization.

For example, basic structure might be **<Department> - <ContentUse>**; such as **Ops - Production**.

- Use the project's **Description** field.

The description you enter when you create a project appears when you hover the pointer over the project thumbnail, as well as on the **Project details** page.



6. Lock permissions in each project

After you refine the capabilities for each group in a project, you can lock the project's permissions, either for the project itself or all projects in the hierarchy. Do this on the Default project, too.

To configure the **Content Permissions**:

1. You must be logged into the site as an administrator, project owner, or project leader
2. Open the permissions dialog box for a project
3. Click the Content Permissions **Edit** link in the upper left and select the desired option in the Content Permissions dialog box

Locking permissions prevents publishers from setting permissions explicitly as part of the publishing process in Tableau Desktop. Instead, content inherits permissions set on the project it's published to, and only administrators and project leaders can set permissions.

For more information, see [Permissions](#).

Automate working with groups and projects

Creating multiple groups and projects and setting permissions manually can get a little tedious. To automate these processes, as well as make them repeatable for future updates, you can perform these tasks using [REST API](#) commands.

You can use `tabcmd` commands for tasks such as adding or deleting a single project or group and adding users, but not for setting permissions.

Next steps

Besides projects, groups, and permissions, other data governance themes include:

User education

Help *all* of your Tableau users become good data stewards. The most successful Tableau organizations create Tableau user groups, have regular training sessions, and so on.

For a common approach to orienting users to the site, see [Dashboard-based Custom Portals](#).

For publishing and data certification tips, see the following topics:

- [Use Certification to Help Users Find Trusted Data](#)
- [Prepare for Publishing a Workbook](#) (links to Tableau Help)
- [Best Practices for Published Data Sources](#) (links to Tableau Help)

Optimize extract refresh and subscription activity

If you use Tableau Server, create policies for extract refresh and subscription schedules, to avoid them dominating the site's resources. The TC customer presentations by Wells Fargo and Sprint address this subject in detail. In addition, see the topics under [Performance Tuning](#).

If you use Tableau Online, see the following topics to become familiar with the ways people can refresh extracts:

- [Keep Data Fresh](#)
- [Use Tableau Bridge to Expand Data Freshness Options](#)

Monitoring

Use administrative views to keep an eye on the site's performance and content use.

[Administrative Views](#)

Learn how Tableau and some of our customers address governance and self-service

The following list contains links to data governance and Center of Excellence (COE) presentations given at the Tableau Conference over recent years. Even if Tableau versions have evolved, the principles remain the same. You can explore the playlists for other videos related to COE, managing Tableau at scale.

[Creating a Centre of Excellence in Tableau](#) (TC Europe 2018)

[Server Admins: Don't Fear Web Authoring](#) (Sprint, TC16)

[The Past, Present, & Future at Charles Schwab](#) (TC 17)

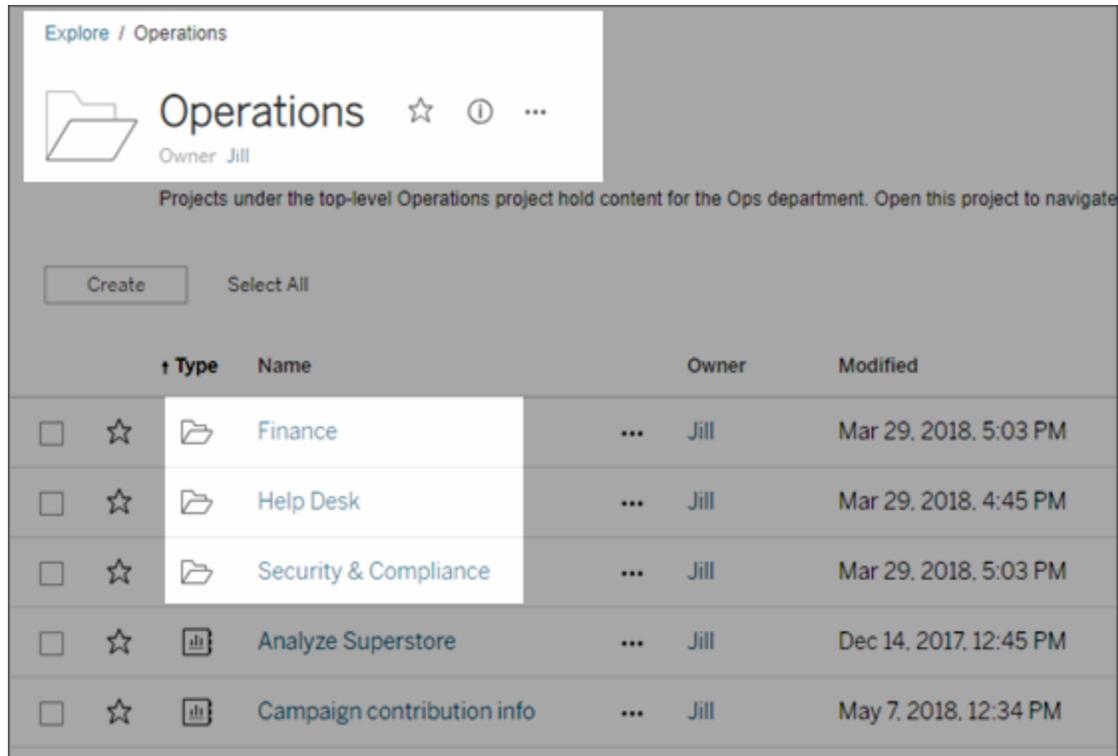
[Content Strategies in Tableau](#) (TC 17)

Use Projects to Manage Content Access

When Tableau Desktop users publish a workbook or data source to a site on Tableau Server, they can select a *project* to publish it to. If they don't specify a project, their content is published to the **Default** project.

As an administrator, you can create projects to hold and organize related *content*, or to delegate content management. Content on Tableau Server are workbooks, views, metrics, Ask Data lenses, data sources, and the projects that hold them. If the Data Management Add-on is present, content types also include flows and data roles.

The following image shows content within the top-level Operations project in the web authoring environment. The Operations project contains a few child projects (highlighted) and published workbooks. A project can also contain other content types, such as data sources and flows.



Why use projects

Projects help you to create a scalable process for managing access to the content published to Tableau Server. Advantages they have include:

- They enable administrators to delegate content management to project leaders who work with the content more closely, without having to give them administrator access to site or server settings.
 - Project leaders can create nested projects under their top-level project, enabling them to maintain their team's content within a single hierarchy.
 - **Note:** Project owners can delete top-level projects they own. Project leaders cannot delete top-level projects.
- They can make the site easier to navigate for self-service users.
 - They segment the Tableau Server site into areas that give users access based on how they use the data published to those areas, or on the Tableau user group they work with.

Tableau Server on Linux Administrator Guide

- You can hide projects from groups who don't need to use them, create a distinguishable project-naming scheme, and take advantage of project descriptions to clarify how to use the project.
- They enable you to track permissions effectively.
 - You can create groups based on the level of content access users in the group need, and set default permissions on projects. This enables you to know exactly which capabilities new users get by default, and likewise which capabilities all users get when a new project is created.

When to create project hierarchies (example)

Many organizations have several or more distinct groups of Tableau users, each with its own priorities and leaders. These groups might share some organization-wide content (or even draw from an org-wide pool of data sources), but primarily they use data and reports that are specific to their team. In this or similar scenario, an example for using project hierarchies might look as follows:

1. You, as a site or server administrator, can create top-level projects for each of your distinct Tableau teams.
2. On each top-level project, you assign the Project Leader status to team leads, and change project ownership. Project leaders effectively are the content administrators, so it's important that they understand how permissions work in Tableau, along with Tableau content management best practices.
3. Each project leader can manage their project, creating the structure within the project that works for their team. That is, they can create child projects they need, based on how their team members collaborate and share data and reports.

The benefit to you as the site administrator is that you can focus on system health. The benefit to your Tableau users is that people who know the best practices for working with Tableau and data can manage these things for their teams, without having to submit IT requests to change permissions or add projects.

Why not use sites?

If you manage your own Tableau Server deployment, you can create as many sites as you want. However, for managing data and reports across your company, projects allow the flexibility you need to administer shared data and reports, and users who might belong to

multiple groups. Many Tableau administrators configure projects as described in the previous section, to expose only what's necessary to users who need to work with it. Projects work better than sites for evolving content from development to staging to production.

Sites work well when content can remain completely separate during all phases, and there is little to no user overlap. A good (and common) example for using multiple sites is to create a site for each of multiple external clients, whose published content you manage as a consultant or vendor. Our own Tableau Online is an example of this on a large scale. Another example might be to use a separate site for sensitive content that you want only specific Human Resources or medical staff to use.

Project-level administration

As a server or site administrator, you can delegate administration of projects and their content, without exposing access to your site or server settings. You can do this by changing the ownership of a project or granting a group or user project leader status on a project.

The **project owner** is always one individual user. By default, the user who creates a project is its owner. The project owner has administrative access to the project and content in it—including making someone else the owner and assigning Project Leader permissions.

The **project leader** setting provides a way to allow multiple users administrative access to a project, its child projects, and all workbooks and data sources in those projects.

A project leader does not have to be a project owner or administrator. In addition to server and site administrators, the full scope of Project Leader permissions is available to users with a **Creator** or **Explorer (can publish)** site role.

Project ownership and project leader access in project hierarchies

In a multi-level project hierarchy, a user or group that is set as a project leader, at any level within the hierarchy, is implicitly given project leader access to all of that project's child projects and their content items.

To remove the project leader access, you must do so at the parent level in the hierarchy on which the ownership or setting was explicitly assigned.

Similarly, the owner of a project at any level has project leader access to all content in that project, as well as to any of its child projects, even if they do not own the child projects.

Only a project owner or administrator can change ownership of a content resource, and this can be done regardless of whether the project permissions are locked.

Actions project-level administrators can take on projects

Project leaders and owners can perform the tasks in the following list, as can server or site administrators.

- Create and delete projects as follows:
 - Server or site administrators can create or delete top-level or nested projects anywhere on the site.
 - Project owners and project leaders can create and delete child (nested) projects in projects they own, or on which they have project leader status.
- Project owners can change ownership of their projects. Both project owners and project leaders can assign the project leader status to groups or users.
 - As a project leader or owner, if you assign the project leader status to someone else, remember that full access to this role's capabilities depends on the user's site role, as specified earlier in this section.
- Set permissions for a project, as well as the child projects, workbooks, and data sources in it.
- Lock permissions to apply the project's default settings to all workbooks, data sources, and optionally child projects and their content. For information, see [Permissions](#).
 - Permissions can be modified only from the project they're locked on. Only admins, the owner, or project leaders can change permissions.
- Move workbooks and data sources to another project to which they have project leader or owner access. Moving the project can affect permissions. For information, see [Move content](#).
- Run, add, or remove extract refresh schedules.

See also [Permissions](#).

How the Default project acts as a permissions template

Tableau creates a **Default** project with every site.

The Default project serves as a template for new top-level projects you create on the site. When you create a new top-level project, settings and permissions from the Default template are applied to the new project, including permissions set on content within the project.

Note: Nested projects (projects you create within other projects) take the permissions set at their parent project, not the Default project.

Before you create other top-level projects, you can take steps to set up the Default project, to help you to know exactly:

- Which type of user gets what level of access for each new project.
- How you might need to modify permissions for each new project.

For more information, see Steps to coordinate projects and groups in the topic Configure Projects, Groups, and Permissions for Managed Self-Service.

Prevent publishers from changing permissions on content they own

Administrators and project leaders can prevent users from changing the permissions for workbooks and data sources in a project hierarchy. For example, you can disable the option to set permissions during the publishing process, and prevent publishers and content owners from changing them after publishing. To do this, you *lock* content permissions to the project. For more information, see Permissions.

Add Projects and Move Content Into Them

A content resource (workbooks and data sources) can live in only project. Server and site administrators can add or remove top-level projects on a site, and move published content from one project to another. Project leaders with appropriate site roles can add or remove child projects and move content between projects on which they have Project Leader access.

This article contains the steps for creating and moving projects. We recommend becoming familiar with the following related content as well:

Tableau Server on Linux Administrator Guide

- To learn about projects and when or why to use them, see [Use Projects to Manage Content Access](#).
- Before you create project hierarchies, become familiar with [Permissions](#).
- To see the specific site roles that allow full Project Leader access, see [Project-level administration](#).

Add a top-level or child (nested) project

1. While you're signed in to Tableau Server as an administrator or project leader, select the **Content** tab, and then do one of the following:
 - Select **New > Project** to create a new top-level project (only administrators can do this).
 - Navigate to and open the project in which you want to create a sub-project, and then select **New > Project**.

If you're not sure where to find a child project, display filters, and select **Show all projects**.

2. Enter a name and description for the project, and then click **Create**.

New Project

Enter a name for the new project:

CS Training - open collaboration

Description [Preview](#)

Use this project to fine tune your analysis with your CS colleagues. |

Show formatting hints ▾ 68 / 4,000

Cancel Create

You can include formatting and hyperlinks in the project description. Select **Show formatting hints** for syntax.

Note: To edit a project description later, select it to open it, select the information icon next to its name, and then click **Edit**.

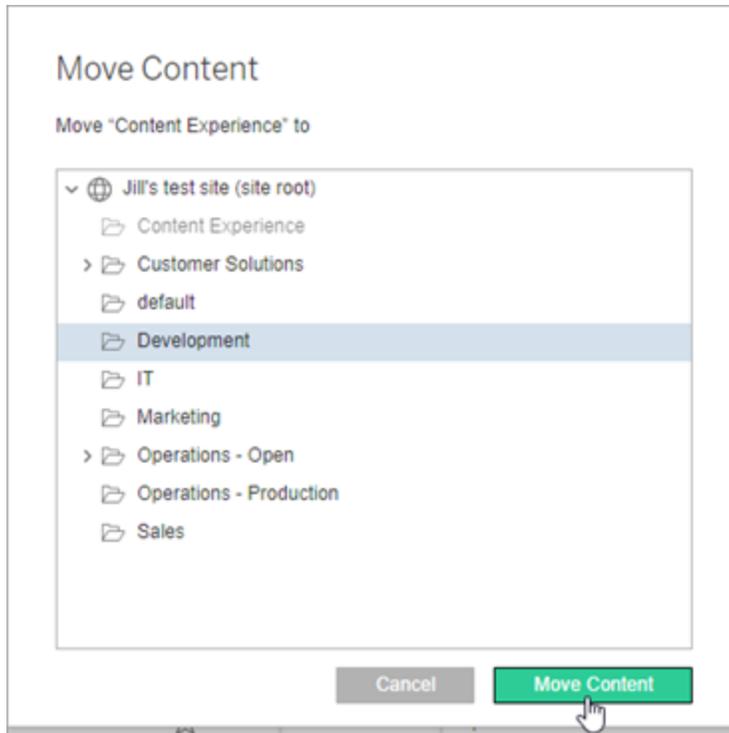
Move a content resource to another project

1. On the **Content** tab, find the content resource you want to move.

If you're not sure where to find a child project, display filters, and select **Show all projects**.

For other content types, you can navigate through its project hierarchy, or by selecting the content type on the **Explore** menu.

2. On the workbook's **Actions(...)** menu, select **Move**.
3. Select the new project for the workbook, and then click **Move Content**.



Moving a project includes moving everything in it, including child projects and their content.

How moving projects affect permissions

When you move a project, Project Leader permissions adapt to the new project environment.

- When the target project hierarchy is **locked**, previous Project Leader permissions are removed, and new Project Leader permissions are granted according to those set at the top-level of the target hierarchy.
- When the target project hierarchy is **unlocked** (managed by owner), previous implicitly granted Project Leader permissions are removed, explicitly set Project Leader permissions are retained, and new Project Leader permissions are granted according to those set at the top-level of the target hierarchy.

When you move project and content, permissions may be impacted. For more information, see [Permissions](#).

Delete a project

When you delete a project, all of the workbooks and data sources in the project are also deleted from the site. If you want to delete a project but not its content, move the content to another project, and then delete the project.

Important

- You cannot undo deleting a project.
- Deleting a project deletes all content in it, including child projects and their content.
- You cannot delete the **Default** project.

To delete a project:

1. On the **Content** tab, find the project you want to remove.

If you're not sure where to find a child project, display filters, and select **Show all projects**.

2. On the project's **Actions (...)** menu, select **Delete**.
3. Confirm that you want to delete the project.

Required access level for moving content

Moving content is effectively like removing it from one project and publishing it to another. For non-administrators, the permissions needed on the source project are different than those needed on the destination project.

Required site role

To move content, users must have one of the following site roles:

Tableau Server on Linux Administrator Guide

- Server Administrator (Tableau Server only)
- Site Administrator Creator or Site Administrator Explorer
- Creator or Explorer (Can Publish)

Users with a Server Administrator or Site Administrator site role do not need any additional capabilities.

Required permissions for the project that users move content *to*

Non-administrators must have the **Publish** permission capability for the project that is the move destination.

Required permissions for the project that users move content *from*

Non-administrator users must

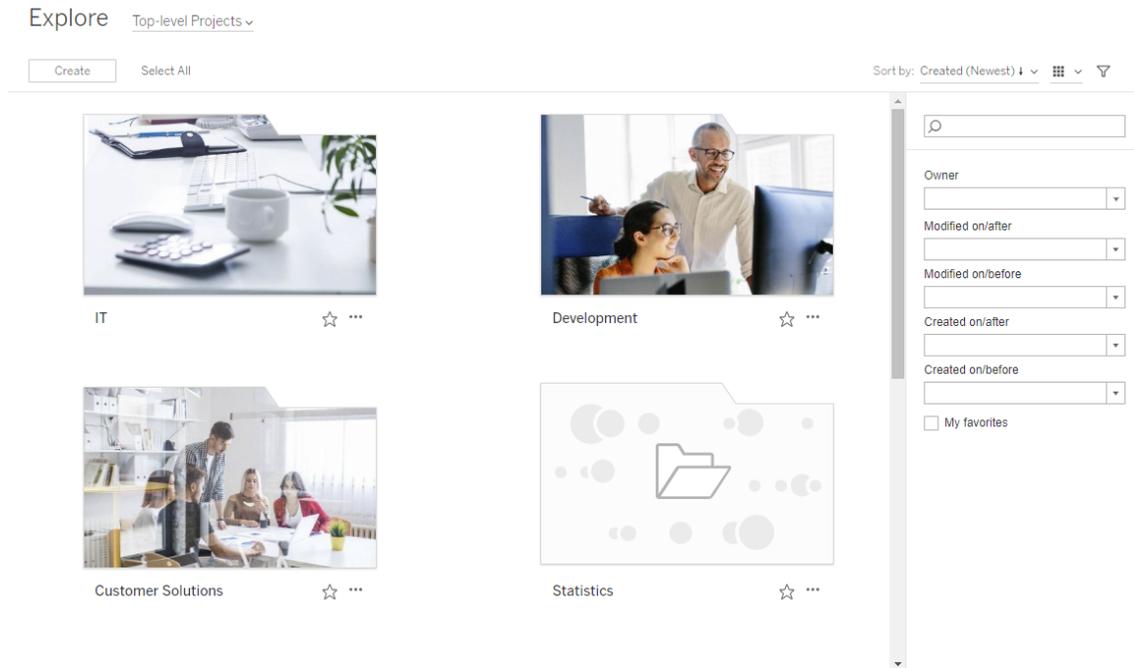
- Be the project owner, project leader, or content owner
- OR
- Have the **Move** permission capability for the content (or, for data sources, be the data source owner).

For more information, see [Move content](#).

Add a Project Image

To help distinguish projects you manage on Tableau Server (and help your users find them), you can add an image that appears in the thumbnail. Your image must meet the following requirements:

- The image must be accessible using HTTPS protocol. Shared network directory and related protocols (UNC, SMB, AFP, NFS, etc) are not supported. HTTP protocol for project images is not supported by Google Chrome.
- All users who access the project must have, at a minimum, "read-only" permission on the target image.
- The image must be common internet format: .jpg, png, or gif.

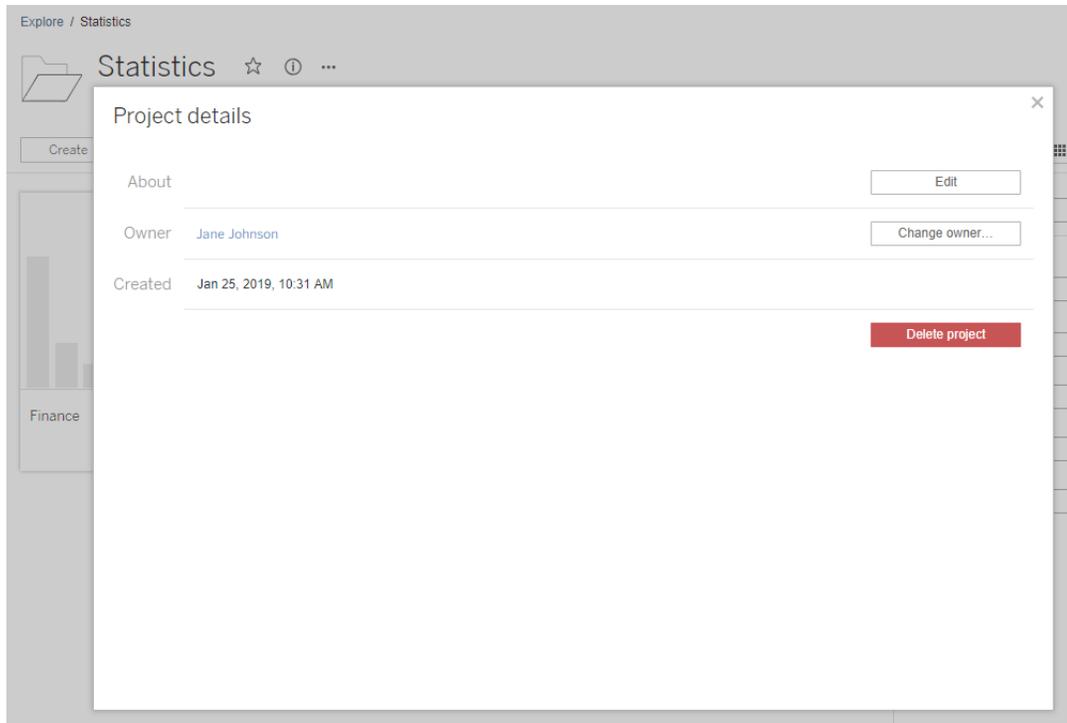


Set a project image

1. Sign in to a site on Tableau Server. In the list of **Top-level Projects** you have access to, select or navigate to the project you want to update. In this example, we'll add an image to the Statistics project folder.

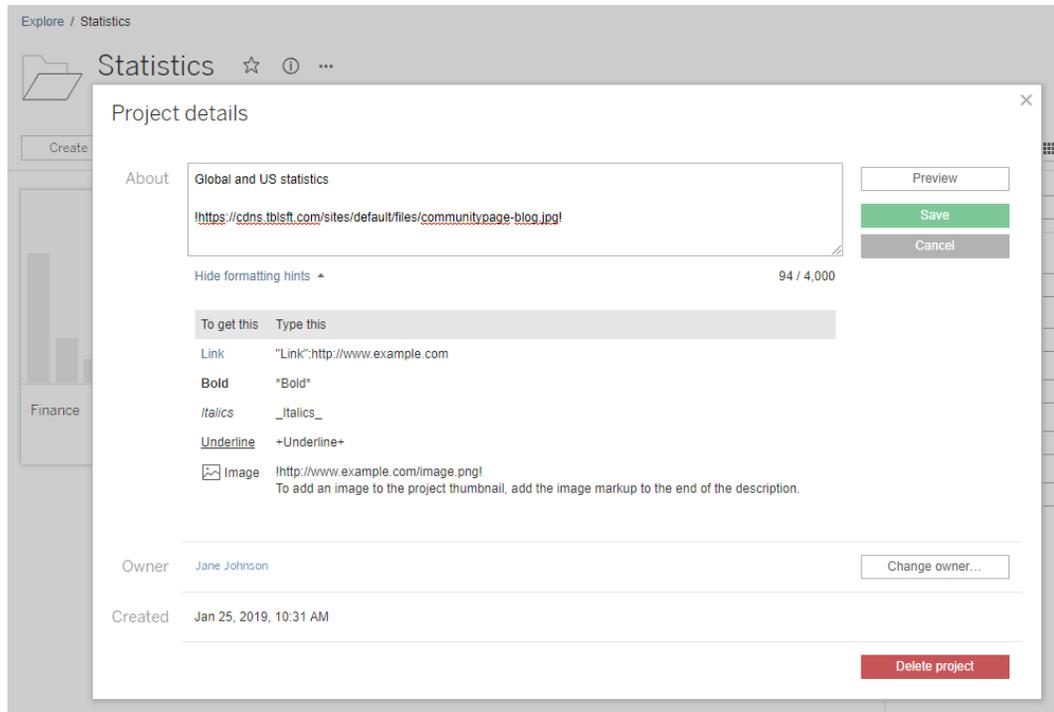
If you're not sure where to find a child project, use the **Explore** drop-down list and select **All Projects**.

2. Click the **Details** icon (i), to open the **Project details** dialog box, and then click **Edit**.



3. In the **About** field, you can enter a description for your project (optional), for example "Global and US statistics." At the end of the project description, add the URL for your image using the following syntax:

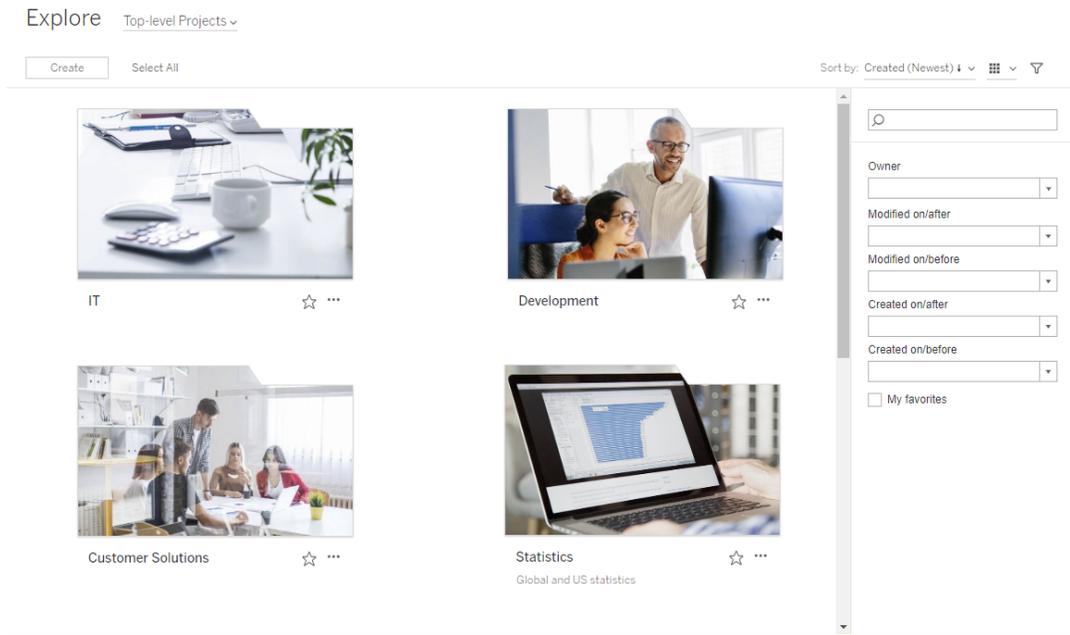
`!http://www.example.com/image.png!`



Select **Show formatting hints** to see how you can format description text.

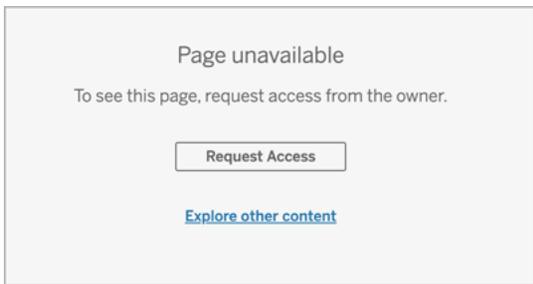
Note: Images embedded in project descriptions cannot be resized or positioned. Recommended size is (300 x 184 pixels). Images that are not 300 x 184 pixels may be stretched, shrunk, or cropped to fit the width of the thumbnail. In addition, they must be added at the end of the project description and be enclosed in ! (exclamation marks), otherwise they will not be displayed as the thumbnail.

4. Click **Save**.



Let Site Users Request Access to Content

Permissions determine if a user has viewing access to workbook, view, or content inside a project. If an existing site user clicks on content or a project they don't have access to, they can select Request Access to send a request to the owner controlling permissions for that piece of content.



When someone requests access, the owner who controls permissions for that content (either at the project or workbook level) receives an email with the name and email of the requester, the content or project requested, and a link to the project or content controlling permissions on the requested item.

For example, if a user requests access to a workbook and content permissions are locked to the project, then the project owner receives the request. Likewise, if a user requests access to a workbook and project permissions are managed by the workbook owner, then the workbook owner receives the request.

Once permission is granted, the owner can email the requester to let them know they have view capability to the project or workbook.

Default settings

The Request Access setting is enabled by default on a new site. To enable the setting if it's been disabled:

1. Go to the General tab of the Settings page for your site.
 - If you have a single site, on the side navigation, click **Settings** and **General**.
 - If you have multiple sites, select the site you want to configure and click **Settings** and **General**.
2. On the General tab, scroll down to Request Access and select **Let users request access to projects, workbooks, and views**.
3. Click **Save**.

Configure project permissions

You can control who will receive the access request by adjusting the project's content permissions. If content permissions are:

- Locked to the project: the project owner receives the request.
- Managed by the owner: The workbook owner receives the request.

To manage content access using projects, see [Use Projects to Manage Content Access and Permissions](#).

For more information about how permission rules are evaluated, see [Permissions: Evaluate permission rules](#).

Change project permissions

For administrators and project leaders

Tableau Server on Linux Administrator Guide

Permissions can be set at the project level for both the project itself and for any content in the project. For example, if workbook permissions are configured at the project level, all workbooks published into that project inherit those default permissions. However, the Creator can choose to change the permissions during publishing, or certain users can change the permissions on published content. To enforce the permissions established at the project level, **Content Permissions** can be locked to the project. For more information, see [Let Site Users Request Access to Content](#).

To set permissions at the project level:

1. Navigate to the project
2. Open the Actions menu (...) and click **Permissions**. The permissions dialog box opens.

This dialog box has two main areas: permission rules at the top and the effective permissions grid below. Each section (Project, Workbooks, Data Sources, Flows, Data Roles) can be expanded (▶) to reveal the capabilities for that type of content.

User / Group	Project	Workbooks	Data sources	Use	Edit	Flows
All Users (32)	None	None	None			None
Site Roles (8)	Publisher	Editor	Editor	✓	✓	Editor

User Permissions	Site Roles (8)
Creator	Publisher
Explorer	Viewer
Explorer (can publish)	Publisher
Server Administrator	Administrator
Site Administrator Creator	Administrator
Site Administrator Explorer	Administrator
Unclassified	Viewer
Viewer	Viewer

With a row selected at the top, the effective permissions grid populates. Use this to verify permissions. Hovering over a capability square provides information about why the capability is allowed or denied for that specific user.

3. To modify an existing permission rule, open the Actions menu (...) for that row and click **Edit**.

4. To create a new rule,
 - a. Select **+ Add a user or group rule**.
 - b. If necessary, use the drop-down box to the right to change between groups and users.
 - c. Select a group or user from the drop-down box. This creates a row where you can configure the permission rule.
5. In the row for the permission rule, choose an existing permission role template from the drop-down box for each section, or create a custom rule by expanding a section (») and clicking the capabilities.

One click sets the capability to **Allowed**, two clicks sets it to **Denied**, and a third click clears the selection (**Unspecified**).
6. When finished, click **Save**.

Change content permissions

For administrators, project leaders, and content owners

If project permissions are not locked, permissions for individual pieces of content can be modified.

Warning: Tableau recommends managing permissions at the project level within the Tableau site. These steps are relevant only for content in projects where permissions are managed by the owner.

Set permissions on content

1. Navigate to the content (workbook, data source, flow, data role)
2. Open the Actions menu (...) and click **Permissions**. The permissions dialog box opens.

This dialog box has two main areas: permission rules at the top and the effective permissions grid below.

Tableau Server on Linux Administrator Guide

The screenshot shows the Tableau Server permissions management interface. At the top, there is a search bar and a note: "Permissions for views are controlled independently". Below this is a table with columns: "User / Group", "Permissions", "View", "Interact/Edit", and "Edit".

User / Group	Permissions	View	Interact/Edit	Edit
All Users (39)	None	[Icons]	[Icons]	[Icons]
Site Roles (8)	Editor	[Green Checkmarks]	[Green Checkmarks]	[Green Checkmarks]

Below the main table is a section for "User Permissions Site Roles (8)". It contains a detailed grid of permissions for various roles:

User	Permissions	View	Interact/Edit	Edit
Creator	Editor	[Green Checkmarks]	[Green Checkmarks]	[Green Checkmarks]
Explorer	Custom	[Green Checkmarks]	[Green Checkmarks]	[Green Checkmarks]
Explorer (can publish)	Editor	[Green Checkmarks]	[Green Checkmarks]	[Green Checkmarks]
Server Administrator	Administrator	[Green Checkmarks]	[Green Checkmarks]	[Green Checkmarks]
Site Administrator Creator	Administrator	[Green Checkmarks]	[Green Checkmarks]	[Green Checkmarks]
Site Administrator Explorer	Administrator	[Green Checkmarks]	[Green Checkmarks]	[Green Checkmarks]
Unlicensed	Custom	[Green Checkmarks]	[Green Checkmarks]	[Green Checkmarks]
Viewer	Custom	[Green Checkmarks]	[Green Checkmarks]	[Green Checkmarks]

A tooltip for the "Explorer (can publish)" row indicates: "Save: Denied (by user's site role)".

With a row selected at the top, the effective permissions grid populates. Use this to verify permissions. Hovering over a capability square provides information about why the capability is allowed or denied for that specific user.

3. To modify an existing permission rule, open the Actions menu (...) for that row and click **Edit**.
4. To create a new rule,
 - a. Select **+ Add a user or group rule**.
 - b. If necessary, use the drop-down box on the right to change between groups and users.
 - c. Select a group or user from the drop-down box. This creates a row where you can configure the permission rule.
5. In the row for the permission rule, choose an existing permissions role template from the drop-down box or create a custom rule by clicking the capabilities.

One click sets the capability to **Allowed**, two clicks sets it to **Denied**, and a third click clears the selection (**Unspecified**).

6. When finished, click **Save**.

Set permissions on a view

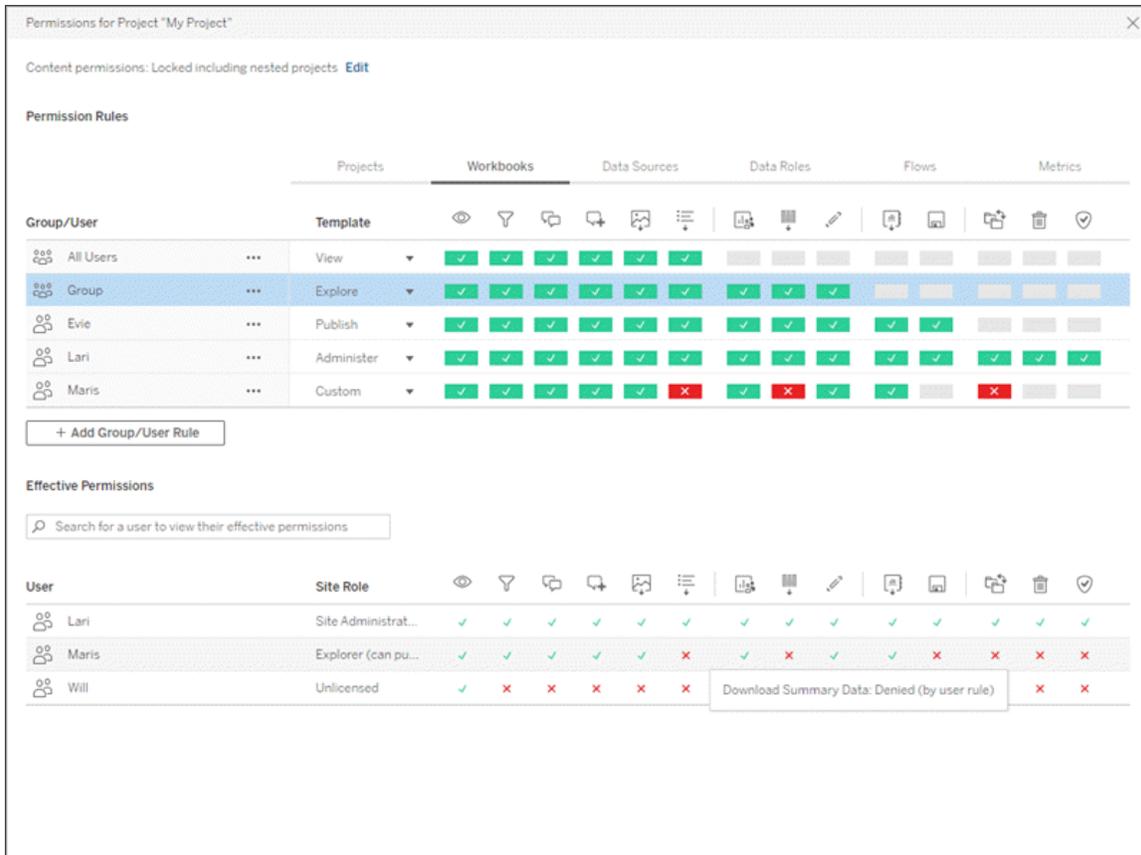
In some situations, it may be valuable to specify permissions on a view independently from the workbook that contains it. To set permissions on a published view, navigate to the view within a published workbook and follow steps above.

Warning: While it is possible to set view-level permissions within a workbook, we strongly recommend managing permissions at the project (or workbook) level as much as possible. For views to inherit permissions, the project must be locked or the workbook must be published with **Show Sheets as Tabs**. See [Let Site Users Request Access to Content](#) for more information.

Permissions

Permissions determine how users can interact with content such as workbooks and data sources. Permissions are set in the permission dialog or via the [REST API](#). At the top of the dialog, permission rules configure capabilities for groups or users. Below, the permissions grid displays the effective permissions for users.

Tableau Server on Linux Administrator Guide



There are several interrelated topics that discuss how to think about, set, and manage permissions. The main topics are:

- This topic, which covers the fundamentals, how to set permission rules for projects and other content, and permission considerations for specific scenarios.
- Permission Capabilities and Templates, which covers in detail the various capabilities that are used to build permission rules.
- Manage Permissions with Projects, which covers using projects to manage permissions and how nested and locked projects impact permissions.
- Effective permissions, which covers how permission rules are evaluated and how final permissions are determined.
- Permissions, Site Roles, and Licenses, which covers how permissions interact with site roles and licenses to determine what a user can do on a site.

Additionally, if the Data Management Add-on is present, permissions for external assets have additional considerations. For more information, see [Manage Permissions for External Assets](#).

Permissions fundamentals

Projects and groups

Tableau sites use *projects* to organize content and *groups* to organize users. Managing permissions is easier when permission rules are:

- Set at the project level instead of on individual pieces of content.
- Established for groups instead of individuals.

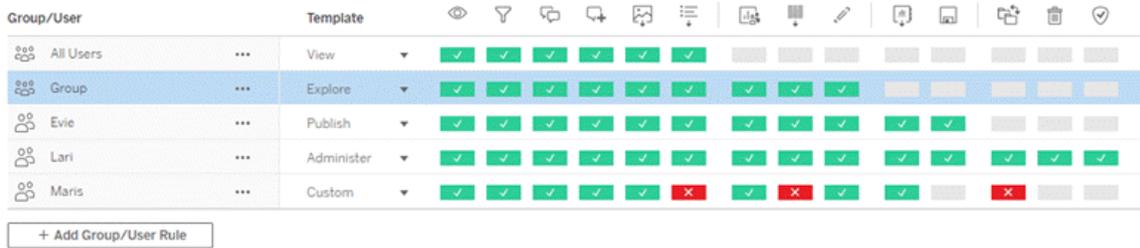
Permissions can only be established for users, groups, projects, or content that already exist. For more information about creating users and groups, creating projects, and publishing content, see [Manage Users and Groups](#), [Use Projects to Manage Content Access](#), and [Publish Data Sources and Workbooks](#).

Capabilities and permission rules

Permissions are made up of *capabilities*—the ability to perform actions like view content, web edit, download data sources, or delete content. *Permission rules* establish what capabilities are allowed or denied for a user or group on a piece of content.

Note: When talking about permissions in general, it's common to see a phrase like “a user must have the delete *permission*.” This is easy to understand in a broad context. However, when working with permissions at a technical level like in this article, it's more accurate to say “the delete *capability*.” In this topic we'll use the more precise term *capability*, but you should be aware that you might see *permission* in other places.

Tableau Server on Linux Administrator Guide



The screenshot shows a table with columns for Group/User, Template, and various permissions. The permissions are represented by green checkmarks (allowed) and red X marks (denied). The 'Group' row is highlighted in blue. Below the table is a button labeled '+ Add Group/User Rule'.

Group/User	Template	View	Download	Print	Export	Refresh	Interact													
All Users	View	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Group	Explore	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Evie	Publish	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Lari	Administer	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Maris	Custom	✓	✓	✓	✓	✓	✗	✓	✗	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓

The interplay between license level, site role, and potentially multiple permission rules factor into the final determination of what a user can or can't do. For each user this becomes their *effective permissions*. For more information, see [Effective permissions](#).

Some tasks such as creating new workbooks from a browser (web authoring) or moving content might require specific configurations of several capabilities rather than being captured in a single capability. For more information, see [Permission settings for specific scenarios](#).

Set permissions

Permission rules are set differently at the project level, at the content level, or when publishing content from Tableau Desktop.

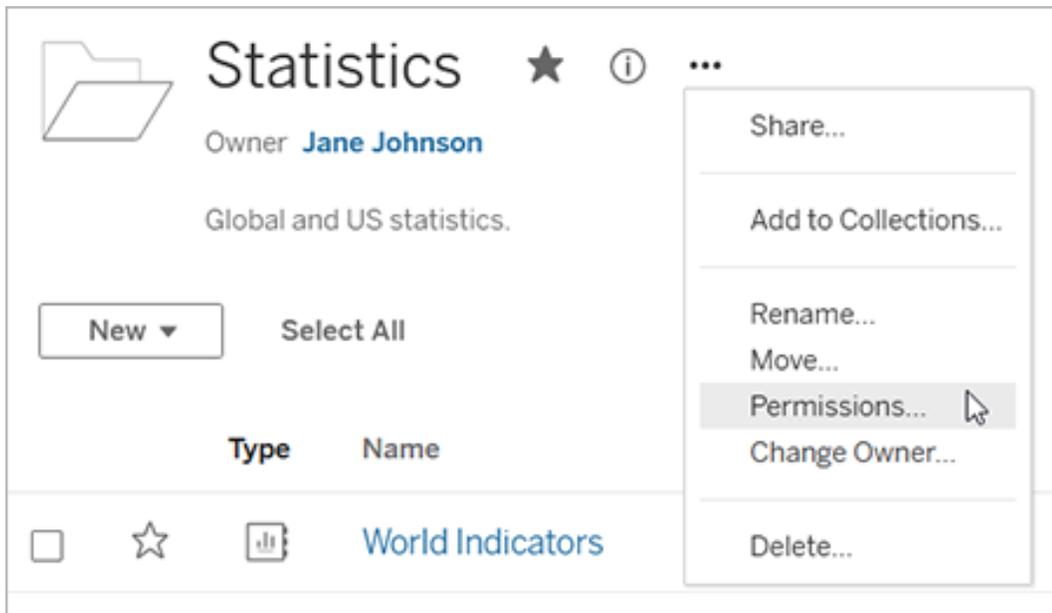
Note: The phrase “project permissions” can have two meanings. There are the permission capabilities for a project itself—View and Publish—that control how a user can interact with a project. There is also the concept of project-level permission rules for other content types. In this article “project-level permissions” means permission rules for workbooks, data sources, and the other content that are configured in the permission dialog for a project. This is in contrast to “content-level” permission rules that can be set on a specific workbook, data source, etc.

Project-level permissions

For administrators, project owners, and project leaders

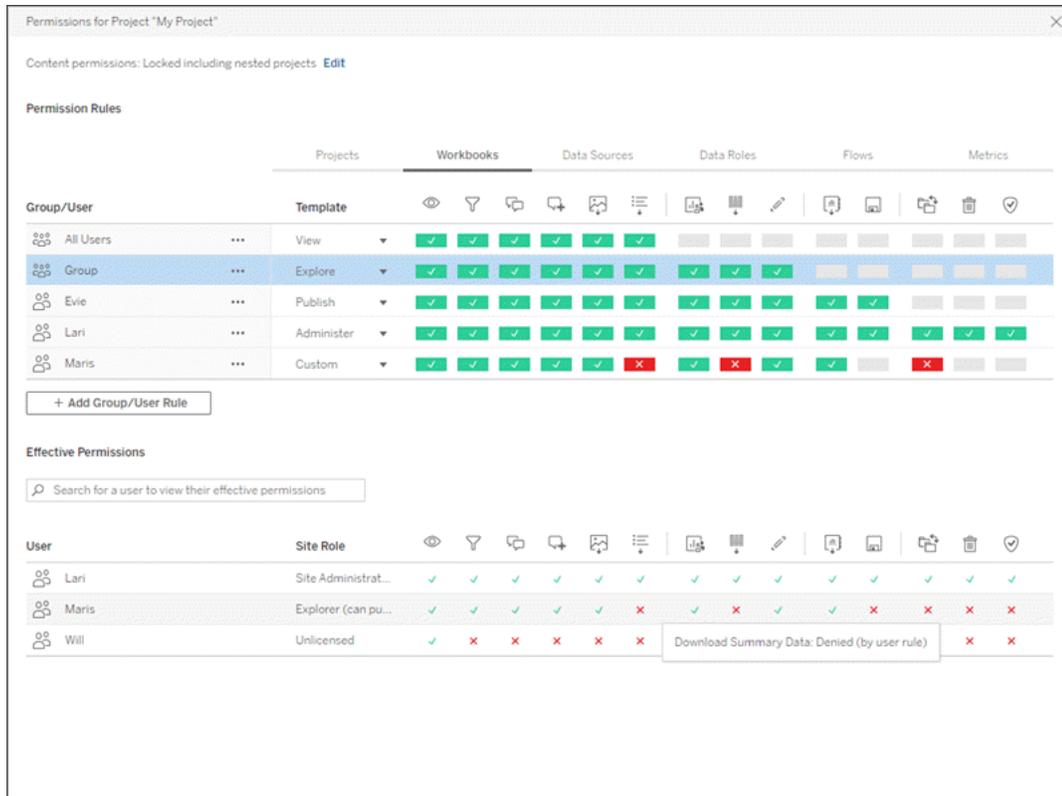
To set permissions at the project level:

1. Navigate to the project
2. Open the Actions menu (...) and click **Permissions**.



The permissions dialog opens. This dialog has two main areas: permission rules at the top and the effective permissions grid below. Each content type has a tab. The image below shows the Workbook tab.

Tableau Server on Linux Administrator Guide



With a row selected at the top, the effective permissions grid populates. Use this to verify permissions. Hovering provides information about why the capability is allowed or denied for that specific user.

3. To modify an existing permission rule, select the appropriate tab for that content type and click a capability.
4. To create a new rule, click **+ Add Group/User Rule** and start typing to search for a group or user. For each tab, choose an existing template from the drop-down box or create a custom rule by clicking the capabilities.

One click sets the capability to **Allowed**, two clicks sets it to **Denied**, and a third click clears the selection (**Unspecified**).

5. When finished, click **Save**.

Tip: Permission rules set at the project level act as a default for content saved in that project and any nested projects it contains. Whether those project-level default rules are enforced or only preliminary depends on the **content permission** setting. This setting can be configured

in two ways, either **Locked** or **Customizable**. For more information, see Lock content permissions.

Content-level permissions

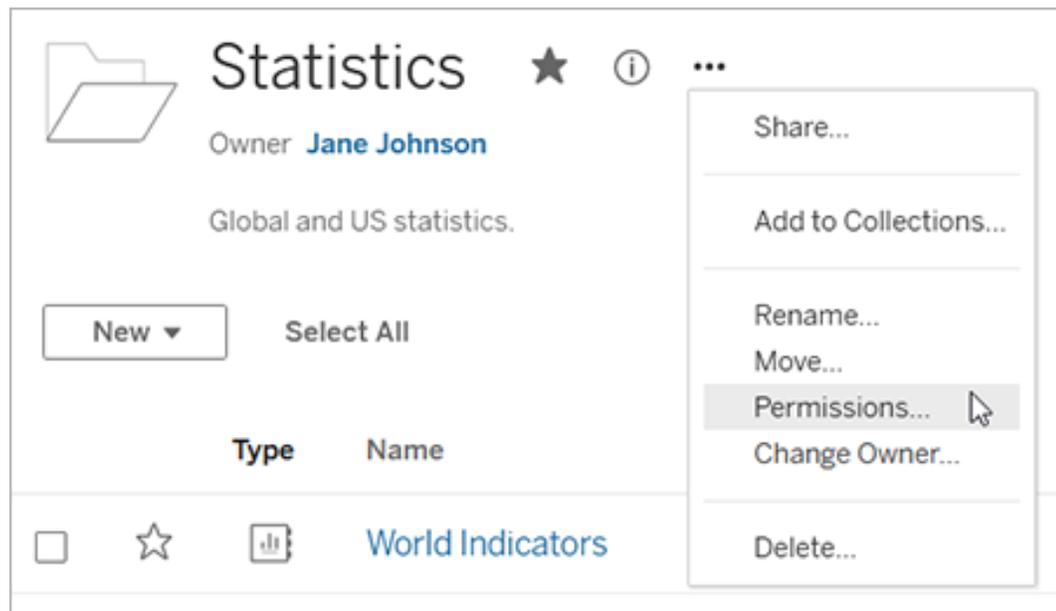
For administrators, project leaders, and content owners

If project content permissions are customizable, permissions for individual pieces of content can be modified. The information below is not relevant to content in locked projects. For more information, see Lock content permissions.

Tip: While it is possible to set permissions on individual content in customizable projects, we recommend managing permissions at the project level.

Set permissions on content

1. Navigate to the content (workbook, data source, flow, data role)
2. Open the Actions menu (...) and click **Permissions**.



Set permissions on a view

Tip: While it's possible to set view-level permissions within a workbook, we strongly recommend managing permissions at the project or workbook level.

If a workbook is published with **Show Sheets as Tabs** checked, the views in that workbook will inherit all permissions set for the workbook. The permission dialog for a view will be read-only.

In some situations, it may be valuable to specify permissions on a view independently from the workbook that contains it. If the workbook is published with **Show Sheets as Tabs** unchecked, the views will start with the workbook permissions but will be independent thereafter and can be set independently. Note that this means if the permission rules are modified for the workbook, those changes won't be applied to the views—each view's permissions will need to be managed individually.

See Show or Hide Sheet Tabs for more information.

Set permissions at publish

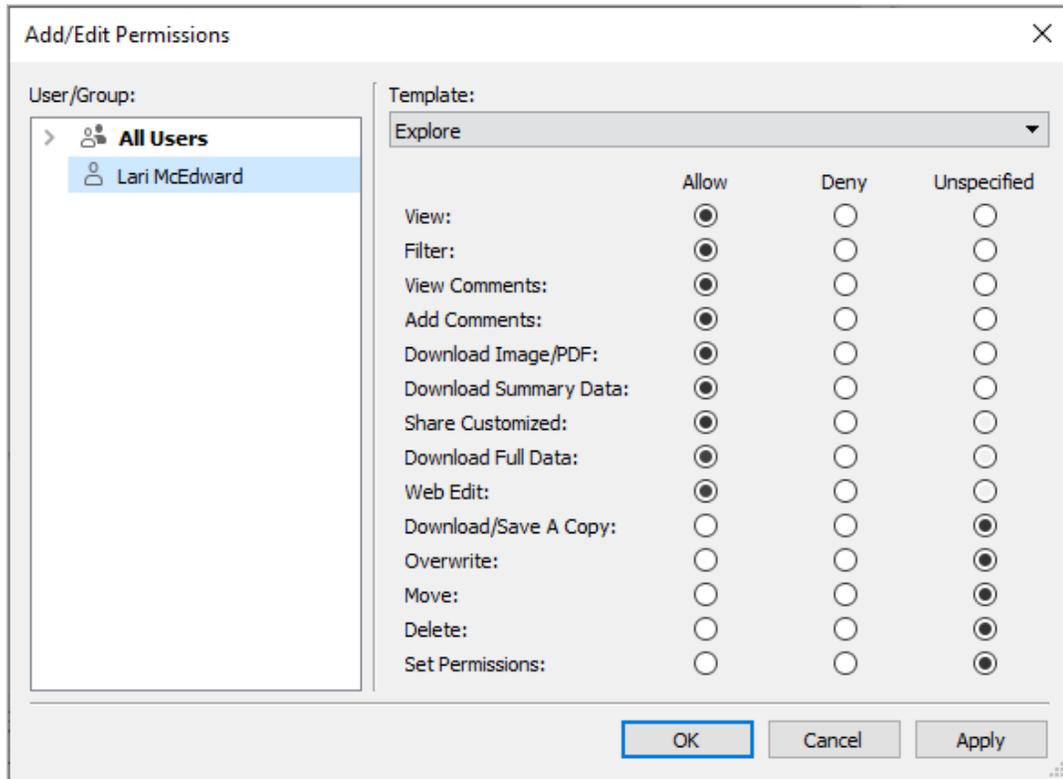
For content publishers

If project content permissions are customizable, permissions for individual content can be set when publishing from Tableau Desktop. The information below is not relevant for content in locked projects. For more information, see Lock content permissions.

Tip: While it's possible to set permissions on individual content in customizable projects, we recommend managing permissions at the project level.

1. From the publishing dialog, click the Edit link for **Permissions**.
If the Edit link is unavailable, permissions are locked to the project and can't be modified except by the project owner, project leader, or an administrator.
2. The Add/Edit Permissions dialog shows any existing permission rules. Click **Add** to add a new permission rule or **Edit** to modify an existing permission rule
 - a. Select the group or user from the left pane. You can expand a group to see which users it contains.

- b. Use the selector at the top of the right pane to choose an existing template, or use the radio buttons to create a custom rule.



Note that effective permissions can't be inspected from the publishing dialog.

- 3. When finished, click **OK** and resume publishing.

Note: Permissions can't be set while publishing flows from Tableau Prep Builder. To set permissions on a flow, refer to the steps for Project-level permissions or Content-level permissions.

Tip: By default, all users are added to an "All Users" group that has basic permissions for content. To start with a clean slate when building your own permission rules, we recommend that you delete the rule entirely or edit the rule for All Users to remove any permissions (set the permission role template to None). This will help prevent any ambiguity down the road by

reducing the number of rules that applies to any given user and therefore making effective permissions easier to understand.

Permission settings for specific scenarios

Certain actions require combinations of permission capabilities and possibly site roles. The following are some common scenarios and their necessary permission configurations

Saving, publishing, and overwriting

In the context of permissions, saving is essentially publishing. As such, the **Overwrite** and **Save a Copy** capabilities can only be given to users with a site role that allows publishing: Administrator, Creator, or Explorer (can publish). Explorer or Viewer site roles can't publish, overwrite, or save a copy.

(Prior to version 2020.1, the *Publish* and *Overwrite* capabilities were called *Save*, and the *Download Workbook/Save a Copy* capability was called *Download Workbook/Save As*.)

- The **Publish** capability for a project allows a user to publish content into that project.
- The **Overwrite** capability allows a user to save over an existing piece of content. By saving over the content, the user becomes the owner of that content. The **Overwrite** capability also allows users to edit minor aspects of existing pieces of content, such as the description for a metric or the synonyms for a data role. Editing the existing content in this way doesn't change the owner of the content.
- The **Save a Copy** capability allows a user to save a new copy of the content. This is usually done in conjunction with web authoring and means the user can save their modifications.

It's important to note that users aren't able to Save or Save As a piece of content unless they have the **Publish** capability for at least one project, because all content must be published into a project. Without the **Publish** capability at the project level, the content can't be published.

In web editing, the **Save** option in the File menu only appears to the content owner. If a user who is not the owner has the **Overwrite** capability (allowing them to save the content), they must use **File > Save As** and name the workbook the exact same name. This prompts a

warning that they are about to overwrite the existing content, which they can do. Conversely, a user with only the **Save a Copy** capability trying to use the same name gets an error stating they don't have permission to overwrite the existing content.

If a user who is not the content owner overwrites content, they become the owner, with all the permissions that entails. The original owner's access to the content is then determined by their permissions as a user rather than the owner.

Note: Download Workbook/Save a Copy is a joint capability for workbooks.

Explorers can be given this capability but they are only able to download the workbook, not save a copy. Giving the capability to Explorer (can publish), Creator, or Administrator site roles gives them both the ability to download workbooks *and* save a copy.

Web Editing and Web Authoring

Web editing and web authoring refer to the general ability for users to edit or create workbooks directly in the browser. The permission capability is called *Web Edit* and the site setting is called *Web Authoring*. This section will refer to any web-based editing or publishing action as *web authoring*.

To enable this functionality, there are several requirements.

- **Site setting:** Web authoring must be turned on for the entire Tableau site. See [Set a Site's Web Authoring Access](#). Without this setting enabled, no users can create new workbooks or edit existing workbooks from the browser, *even if they have the web edit capability*.
- **User site role:** The user must have the appropriate site role.
 - Viewers can never web edit.
 - Explorers can be given the web edit capability but can't publish. Essentially, they can use web editing to answer deeper questions based on existing content on the fly, but can't save their edits.
 - Explorers (can publish) or Site Administrator Explorers can publish, but they can only use data that is already published to the site.
 - Creators, Site Administrator Creators, and Server Administrators can publish and create new data sources.

- **Permission capabilities:** The user must have the necessary permission capabilities based on the desired functionality.

Required Permission Capability Settings

Desired functionality	Minimum Site Role	 Web Edit	 Download/ Save a Copy	 Overwrite (<i>workbook</i>)	 Publish (<i>project</i>)	 Connect (<i>data source</i>)
Web author without being able to save	<i>Explorer</i>	 Allow	 Deny	 Deny	Optional	 Allow
Web author and save as new content	<i>Explorer</i> (<i>can publish</i>)	 Allow	 Allow	 Deny	 Allow	 Allow
Web author and save (overwrite) content	<i>Explorer</i> (<i>can publish</i>)	 Allow	 Allow	 Allow	 Allow	 Allow
Web author with new data and save new content	<i>Creator</i>	 Allow	Optional	Optional	 Allow	Optional

Optional indicates this capability is not involved in the desired functionality

Data access for published Tableau data sources

Data sources published to a Tableau site can have native authentication as well as permissions within the Tableau environment.

When the data source is published to the Tableau site, the publisher can choose how to [Set Credentials for Accessing Your Published Data](#) which addresses how data source credentials are handled (such as requiring users to log into a database or enter their

credentials for Google Sheets). This authentication is controlled by whatever technology holds the data. This can be embedded when the data source is published, or the data source publisher can choose to prompt the user for their credentials to the data source. For more information, see [Publish a Data Source](#).

There are also data source capabilities that allow or deny users the ability to see (**View**) and connect to the published data source (**Connect**) in the context of Tableau. These capabilities are set like any other permissions in Tableau.

When a workbook is published that uses a published data source, the author can control how the Tableau authentication will behave for someone consuming the workbook. The author sets the workbook’s access to the published data source, either as **Embed password** (using the author’s Connect access to the data source) or **Prompt users** (using the Connect access of the person viewing the workbook), which may require data source authentication as well.

- When the workbook is set to **Embed password**, anyone who looks at the workbook will see the data based on the author’s access to the data source.
- If the workbook is set to **Prompt users**, the Tableau-controlled access is checked for the data source. The person consuming the workbook must have the Connect capability for the published data source to see the data. If the published data source is also set to Prompt user, the viewer must also enter their credentials for the data source itself.

Workbook authentication to the data source	Data source authentication to the data	How data access is evaluated for someone consuming the workbook
Embed password	Embed password	User sees the data as if they were the workbook author
Embed password	Prompt user	User sees the data as if they were the workbook author. (The author is prompted for data source authentication, not the user.)
Prompt user	Embed password	User must have their own Connect capability to the published data source

Prompt user	Prompt user	User must have their own Connect capability to the published data source and are prompted for their credentials to the underlying data
-------------	-------------	---

Note that this applies to consuming a workbook, not web editing. To web edit, the user must have their own Connect capability.

Move content

To move an item, open its Action menu (...) and click **Move**. Select the new project for the item, then click **Move Content**. If Move is unavailable or there are no available destination projects, verify the appropriate conditions are met:

- Administrators can always move content and projects to any location.
- Project leaders and project owners can move content and nested projects among their projects.
 - Note that non-administrators can't move projects to become top-level projects
- Other users can move content only if all three of the following requirements are met:
 - Creator or Explorer (Can Publish) site role.
 - Publishing rights (**View** and **Publish** capabilities) for the destination project
 - Owner of the content, or—for workbooks and flows—having the **Move** capability.

When a project is moved, the permissions for its content might change.

- Project leaders or project owners always gain permissions for items moved into their projects.
- When a project is moved into a **locked (including nested)** project, the permission templates for the locked project are enforced on the moved project and all its content and nested projects. (Note that this might strip the user moving the project of their ability to move it again if they don't have the correct permissions in the locked project.)
- When a project is moved into an **unlocked** project (customizable), the existing permissions are retained for the moved project and its content. If the project leader status has only implicitly been granted (from a higher-level project), that status is removed, though any explicitly set project leader status is retained.

Metrics

Metrics are created from views in published workbooks. A user can create metrics if they:

- Are a Creator or Explorer (can publish) site role
- Have the **Publish** capability on a project
- Have the **Download Full Data** capability for the relevant view or workbook

For more information, see [Create and Troubleshoot Metrics](#) and [Set Up for Metrics](#).

Permissions for metrics

Because metrics are independent content, it's important to note that the permissions for metrics are managed independently from the view they were created from. (This is unlike data-driven alerts and subscriptions, where the content of the alert or subscription can only be seen if the user has the correct permissions for the view itself.)

Although the capabilities for metrics are straightforward, the **View** capability should be considered carefully. It may be possible for a workbook with restricted permissions to be the basis for a metric with more open permissions. To protect sensitive data, you might want to prevent metric creation for specific workbooks.

Prevent metric creation

The ability to create a metric cannot be directly disabled on a per-workbook level (only per-site), but permissions can regulate access between metrics and workbooks.

To prevent metrics for a specific workbook, deny the **Download Full Data** capability on the workbook.

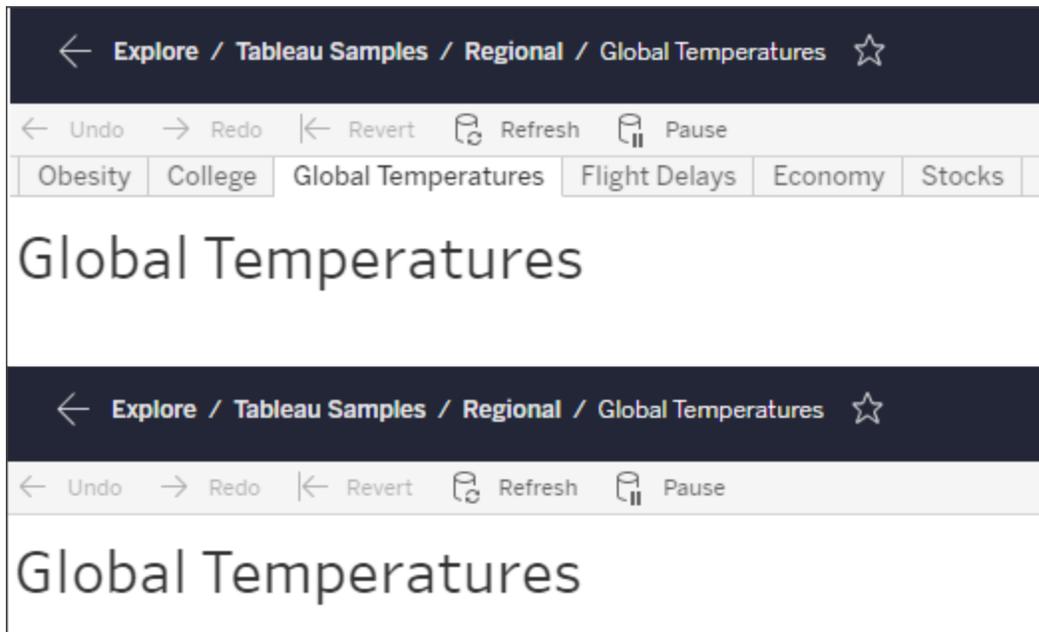
To ensure this capability cannot be changed, deny **Download Full Data** at the project level for all workbooks in the project, and lock the content permissions for the project.

Show or Hide Sheet Tabs

In the context of published content, sheet tabs (also referred to as tabbed views) is a distinct concept from sheet tabs in Tableau Desktop. Showing and hiding sheet tabs in Tableau

Desktop refers to hiding sheets in the authoring environment. For more information, see [Manage Sheets in Dashboards and Stories](#).

Showing and hiding sheet tabs (turning tabbed views on or off) for published content refers to navigation in a published workbook. When sheet tabs are shown, published content has navigational sheet tabs along the top of each view.



This setting also impacts how permissions function and may have security implications (see note).

Note: It is possible to have the **View** capability for a view without the **View** capability for the workbook or project that contain it. Normally if a user lacks the View capability for a project and workbook, they would not know those assets exist. If they have the View capability for a view, however, a user may be able to see the project and workbook name when looking at the view, such as in the navigational breadcrumb. This is expected and accepted behavior.

Turn off tabbed views to allow independent view permissions

Although it is not recommended as a general practice, there are times when it can be useful to set permissions on views independently of the workbook that contains them. To do so, three conditions must be met:

1. The workbook must be published—there is no way to set view permissions during publishing.
2. The workbook must be in a customizable project.
3. The workbook can't show sheets as tabs (tabbed views must be hidden).

When a workbook shows sheets as tabs, all views inherit the workbook permissions and any changes to the workbook permissions affect all of its views. **When a workbook in a customizable project does not show tabbed views, all views assume the workbook permissions upon publication, but any subsequent changes to the workbook's permission rules *will not be inherited by the views*.**

Changing the configuration of sheets as tabs on a published workbook will also impact the permission model. Show Tabs will override any existing view-level permissions and reinstate the workbook-level permissions for all views. Hide Tabs will break the relationship between the workbook and its views.

- To configure sheets as tabs on a published workbook, open the Actions menu (...) for the workbook and select **Tabbed Views**. Choose **Show Tabs** or **Hide Tabs** as desired.
- To configure sheets as tabs during publishing, refer to [Show sheets as tabs](#).
- To set view-level permissions, see Set permissions on content.

Important: In a customizable project, any modifications to the workbook-level permissions will *not* be applied if navigational sheet tabs are hidden (aka tabbed views are off). Changes to permissions must be made on individual views.

Collections

Unlike projects, which contain content, a collection can be thought of as a list of links to content. Project permissions can be inherited by the content in the project, but permissions for a collection have no affect on the content added to the collection. This means that different

users might see different numbers of items in a collection, depending on which items they have permission to view. To make sure that users can see all items in a collection, adjust the permissions for those items individually.

Permissions for a collection can be changed either by using the permissions dialog or by granting access upon sharing a collection, if you're an administrator or the collection owner. For more information, see [Manage Collection Permissions](#).

Private collections

When a collection is created, it is private by default. A private collection appears on the owner's My Collections page, but it doesn't appear in the list of all collections on a site. Private collections are simply collections with no permission rules added. Unlike other types of content, collections don't have the "All Users" group added by default. When you add permission rules to a collection, it is no longer flagged as private. To return a collection to a private state, remove the permission rules.

Private collections can be viewed by the collection owner as well as by administrators, whose site role gives them effective permissions to view all collections.

Permission Capabilities and Templates

Permissions are made up of capabilities, or the ability to perform a given action on a piece of content, such as view, filter, download, or delete. Each row in the Permission Rules area of the dialog is a *permission rule*. Permission rules are the setting for each capability (allowed, denied, or unspecified) for the group or user in that row. Permission rules have *templates* available that make it easier to assign capabilities quickly. Permission rules can also be copied and pasted.

Note: In the permission dialog for projects, there are tabs for each content type (Projects, Workbooks, Data Sources, Ask Data Lenses, Metrics and—if you have the Data Management—Data Roles and Flows). When a permission rule is added, the default for all capabilities across all content types is Unspecified. To allow or deny capabilities

for each content type, you must go to each tab in turn. In the permission dialog for a specific piece of content, there are no tabs and the permission rules only apply to that piece of content.

Templates

Templates group sets of capabilities that are often assigned together based on common user scenarios, **View**, **Explore**, **Publish**, and **Administer**. Assigning a template sets its included capabilities to **Allowed**, with the rest left as **Unspecified**. The templates are cumulative, so the Explore template includes everything from the View template plus additional capabilities. All content also has a template for **None** (which sets all capabilities to unspecified) and **Denied** (which sets all capabilities to denied).

Templates are meant to be a starting point and can be adjusted after they are applied. Capabilities can also be granted or denied without using a template at all. In both cases, the template column will then show **Custom**.

Copy and paste permissions

If there is a permission rule that needs to be assigned to multiple groups or users, you can copy and paste from one rule to another. You can't copy from or paste onto a rule that involves Project Leader status.

1. Open the action menu (...) for the existing rule you want to copy from and select **Copy Permissions**. This will only be available when the rule is not in edit mode.
2. Select an existing rule you want to paste over. You can also create a new rule by clicking **+ Add Group/User Rule** and selecting a group or user.
3. Open the action menu (...) and select **Paste Permissions**.

Capabilities

Each content type has specific capabilities:

Projects

Projects have only two capabilities and two templates. Prior to 2020.1, Project Leader was treated as a permission capability rather than a setting. For more information about project leaders and how to assign them in 2020.1 and later, see [Project administration](#).

View template

 **View** allows a user to see the project. If a user hasn't been granted the view capability, the project won't be visible to them. Granting the view capability for a project does not mean a user can see any content in the project, just the existence of the project itself.

Publish template

 **Publish** allows a user to publish content to the project from Tableau Desktop or Tableau Prep Builder. The publish capability is also required to move content into the project or save content to the project from web authoring. Prior to 2020.1, this capability was called Save.

Workbooks

View template

 **View** allows a user to see the workbook or view. If a user hasn't been granted the view capability, the workbook won't be visible to them.

 **Filter** allows a user to interact with filters in the view, including keep only and exclude filters. Users lacking this capability won't see filter controls in the view.

 **View Comments** allows a user to view the comments associated with the views in a workbook.

 **Add Comments** allows a user to add comments to views in a workbook.

 **Download Image/PDF** allows a user to download each view as a PNG, PDF, or PowerPoint.



Download Summary Data allows a user to view the aggregated data in a view, or in the marks they've selected, and download that data (as a CSV).

Explore template



Share Customized allows users to add their custom views to the list of “Other Views” visible on a workbook. When this capability is denied, users won't see the “Make visible to others” option when they create a custom view. For more information, see [Use Custom Views](#). This capability doesn't impact the ability to share a custom view with the share dialog or by copying the link.



Download Full Data allows a user to view the underlying data in a view, or in the marks they've selected, and download that data (as a CSV).



Web Edit allows a user to edit the view in a browser-based authoring environment.

- Note: Creating new content in the browser or saving views from the web edit interface requires a specific combination of capabilities. For more information, see [Web Editing and Web Authoring](#).
- The Web Editing feature must also be enabled for the entire site or even users with this capability allowed won't be able to web edit. For more information, see [Set a Site's Web Authoring Access](#).

Publish template



Download Workbook/Save a Copy allows a user to download a packaged workbook (as a TWBX). Allows a user to save (publish) a copy from the web edit interface as a new workbook. Prior to 2020.1, this capability was called Download Workbook/Save As.



Overwrite allows a user to overwrite (save) the content asset on the server. Prior to 2020.1, this capability was called Save.

- When allowed, the user can re-publish a workbook, data source, or flow, or save a workbook or flow in web authoring, thereby becoming the owner and gaining access to

all permissions. Subsequently, the original owner's access to the workbook is determined by their permissions just like any other user.

Administer template

 **Move** allows a user to move workbooks between projects. For more information, see [Move content](#).

 **Delete** allows a user to delete the workbook.

 **Set Permissions** allows a user to create permission rules for the workbook.

Views

In a workbook that is not in a locked project and does not show sheets as tabs for navigation, views (sheets, dashboards, stories) inherit the workbook permissions at publication, but any changes to permission rules must be made on individual views. View capabilities are the same as those for workbooks, except for **Overwrite**, **Download Workbook/Save a Copy**, and **Move** which are only available at the workbook level. We recommend showing navigational sheet tabs whenever possible so views continue to inherit their permissions from the workbook.

Data Sources

View template

 **View** allows a user to see the data source on the server.

 **Connect** allows a user to connect to a data source in Tableau Desktop, Tableau Prep Builder, Ask Data, or web editing.

- If a workbook author embeds their credentials to a published data source in a published workbook, they are essentially embedding their **Connect** capability. Therefore, users can see the data in the workbook regardless of their own **Connect** capability for that data source. If the workbook author doesn't embed their credentials to the published data source, the user needs their own **Connect** capability to the data

source in order to consume the workbook. For more information, see [Data access for published Tableau data sources](#).

- A user must have the **Connect** capability for a data source in order to use Ask Data and to create Ask Data lenses. For more information, see [Enable Ask Data for Sites and Data Sources](#).

Explore template



Download Data Source allows a user to download the data source from the server (as a TDSX)

- Cube data sources, like those for Microsoft Analysis Services or Oracle Essbase connections, must be used locally. To download the published data source to Tableau Desktop, the user must have the Download capability. For more information, see [Cube Data Sources](#).

Publish template



Overwrite allows a user to publish a data source to the server and overwrite the data source on the server. Prior to 2020.1, this capability was called Save.

Administer template



Delete allows a user to delete the data source



Set Permissions allows a user to create and edit permission rules for the data source

Ask Data Lenses

View template



View allows a user to see the lens.

Publish template



Overwrite allows a user to edit the lens.

- By default, users with a site role of Explorer (can publish) and Creator have the Overwrite capability for lenses. This means that any user with the appropriate role can edit the name, description, fields, synonyms, and suggested questions for a lens.
- To limit who can edit a lens, deny the Overwrite capability for specific users or entire groups. To limit all lenses in a project, deny the Overwrite capability for lenses at the project level and lock the content permissions for the project.

Administer template

 **Move** allows a user to move the lens between projects.

 **Delete** allows a user to delete the lens.

 **Set Permissions** allows a user to create permission rules for the lens.

Other content types

	View template	Explore template	Publish template	Administer template
Flows	 View allows a user to view the flow.	 Download flow allows a user to download the flow (as a TFLX).	 Run allows a user to run the flow.  Overwrite* allows a user to publish a flow and overwrite the published flow.	 Move allows a user to move content between projects. For more information, see Move content.  Delete allows a user to delete the content.  Set Per-

Data Roles	 View allows a user to view data roles.	n/a	 Overwrite [*] allows a user to publish data roles, overwrite published data roles, and edit a published data roles' synonyms.	missions allows a user to create per-mission rules for the content.
Metrics	 View allows a user to view metrics.	n/a	 Overwrite [*] allows a user to overwrite a metric and edit a metric's details.	
Collections	 View allows a user to view collections.	n/a	n/a	n/a

*Prior to 2020.1, the Overwrite capability was called Save.

Manage Permissions with Projects

Using projects can simplify permission management through features such as nested project hierarchies, hiding projects from certain users or groups, authorizing project leaders, and locking permissions.

Tip: How permissions are set at the project level is very important, especially for the Default project. When a new top-level project is created it inherits its default permission rules (for all content types) from the Default project. When a new project is created nested inside another project, the child project inherits its default permission rules from the parent project.

Project administration

Projects are containers used to organize and manage access to content. By giving non-administrators privileges to manage projects, certain content administration tasks can be handled at the project level.

Project Leaders: Projects can have project leaders, users who have been set as a **project leader**. This setting automatically grants a user their maximum capabilities—depending on their site role—for that project and all content in that project. Project leaders with site role of Explorer (can publish) and above will therefore have all capabilities. Project leaders are essentially local admins for the project without access to site or server settings.

Hierarchy: Only administrators can create top-level projects. Project owners and project leaders can create nested projects inside their projects. Project owners and leaders have full administrative access to the project and its content, as well as any nested projects it contains. In a hierarchy, project leaders are implicitly given project leader access to all child content. To remove project leader access, you must do so at the level in the hierarchy where the role was explicitly assigned.

Ownership: A project can have multiple project leaders, but each project has exactly one owner. By default, a project is owned by the user who created it. A project's owner can be changed (by the existing owner or an administrator, but not a project leader) to any user with a site role of Explorer (can publish) or Creator, or an administrator site role. Project ownership can be changed regardless of whether the project permissions are locked. Note that this refers to project ownership. Content ownership can be changed by project owners, project leaders, and administrators.

Deleting: Content can only exist inside a project. Only administrators can create and delete top-level projects, but project leaders can create or delete nested projects. Deleting projects also deletes all the content and nested projects they contain. To delete a project without losing its content, move the content to another project first. Deleting projects can't be undone.

For a deeper dive into project administration, see [Use Projects to Manage Content Access and Add Projects and Move Content Into Them](#).

Set a project leader

Project leaders are users who have administrator-like access for a specific project or project hierarchy. Prior to 2020.1, **Project Leader** was a capability that could be set to allowed, denied, or unspecified like any other capability. Starting in 2020.1, project leaders are now assigned through the action menu and function as a setting rather than a capability.

To assign project leader status to a group or user

1. Open the permission dialog for the appropriate project.
2. Select an existing permission rule, or click **+ Add Group/User Rule** and chose the desired group or user.
3. Open the action menu (...) for that permission rule and select **Set Project Leader...**

Note: If the action menu includes an option for **Enable “Set Project Leader”**, this will need to be selected before the group or user can be set as a project leader. This option only appears when that group or user was denied the Project Leader capability (prior to 2020.1). That denied capability needs to be removed before they can be set as a project leader.

Once a permission rule has been used to establish a group or user as a project leader, the templates and capabilities are no longer editable because all capabilities are allowed for project leaders. If a project leader is established on a project that contains nested projects, they will have inherited project leader status on all nested projects and their content.

Project leader status is always applied downward through the entire project hierarchy and can only be removed from the level where it was set. To remove project leader status, follow the same steps as above but select **Remove as Project Leader** from the action menu.

Once a group or user has been removed as project leader, that permission rule will have all capabilities set to Unspecified. This may mean their access to and capabilities for that project will be removed if there is no other permission rule giving them permissions to the content.

To keep their access to the project and its content, they will need to have capabilities set like any other group or user.

Lock content permissions

Permission rules set at the project level act as a default for content saved in that project and any nested projects it contains. Whether those project-level default rules are enforced or only preliminary depends on the **content permission** setting. This setting can be configured in two ways, either **Locked** (recommended) or **Customizable**. Locking a project removes the ability for content owners to modify the permission rules on their content. Locking permissions can be applied to nested projects or just to the parent project itself.

- When the content permissions are **locked (including nested projects)**, permission rules set at the project level are enforced for all content in the project and all nested projects. (This was the default behavior for locking projects prior to 2020.1)
- When the content permissions are **locked** (not including nested projects), permission rules set at the project level are enforced for content in the project, but nested projects can be configured independently with their own permission rules and as locked or customizable. (This is new behavior for locking projects as of 2020.1)
- When the content permissions are **customizable**, permission rules set at the project level are applied to all content in the project by default. However, permission rules can be modified for individual pieces of content during or after publishing. (This was called **Managed by the owner** prior to 2020.1)

Note: Whether permission rules are locked or customizable, the permissions on content are always applied. *Locked* and *customizable* refer only to how project-level permissions are inherited by content in the project and who can change them. Even in a project with customizable permissions, only specific users can modify permissions (content or project owner, project leader, admins, or those with the Set Permission capability).

In a locked project:

- The project permission rules per content type are applied to all content.
- Only administrators, project owners, and project leaders can modify permissions.
- Content owners lose the Set Permission capability but retain all other capabilities on their content.
- Permissions are predictable for all content in the project.

In a customizable project:

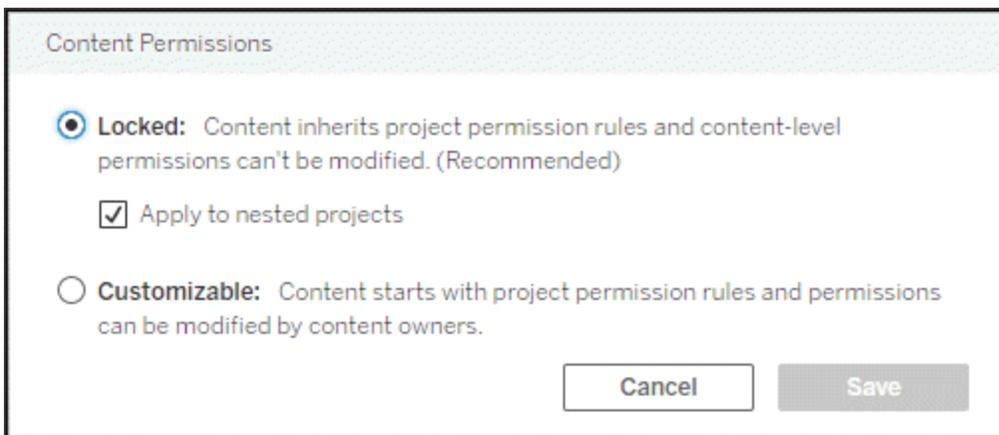
- The project permission rules are applied by default when content is published into the project or nested projects are created, but permissions can be modified during publication or after the content is created.
- Any user with the Set Permissions capability can modify permission rules for that content.
- Content owners have all capabilities on their content.
- Permissions can be different across content in the project.

Set content permissions (lock a project)

New top-level projects inherit all initial permission rules from the Default project but not the content permissions setting, which is set to **Customizable**. This can be changed to **Locked** if desired.

To configure the **Content Permissions**:

1. You must be logged into the site as an administrator, project owner, or project leader
2. Open the permissions dialog for a project
3. Click the Content Permissions **Edit** link in the upper left and select the desired option in the Content Permissions dialog



Note: If the upper left corner doesn't show an **Edit** link in step 3 above, you may be on the permissions dialog for (a) a nested project or a piece of content in a locked project, in

which case the link should bring you to the managing project, (b) a piece of content in a customizable project, which won't show anything, or (c) a view, which will indicate how the view permissions are tied to the workbook. For more information on the interplay of permissions for views and workbooks, see [Show or Hide Sheet Tabs](#).

Change content permissions

When the content permission setting for a project is changed, the outcome depends on the new setting. Changes to permission rules in a locked hierarchy must be done at the level of the managing project.

Changing from	Changing to	Outcome
Locked (including nested projects)	Locked	Doesn't modify existing permission rules. Any nested projects become customizable.
	Customizable	Doesn't modify existing permission rules, though they become customizable. Any nested projects become customizable.
Locked	Locked (including nested projects)	Overwrites existing custom permission rules for all nested projects and their content. This can't be undone.
	Customizable	Doesn't modify existing permission rules, though they become customizable. Any nested projects retain their content permission settings and permission rules.
Customizable	Locked (including nested projects)	Overwrites existing custom permission rules for content in the project, as well as all nested projects and

jects)	their content This can't be undone.
Locked	Overwrites existing custom permission rules for content in the project. This can't be undone. Any nested projects retain their permission rules and remain customizable.

Move projects and content

When a project is moved into another project, the permissions settings on the project being moved are maintained unless the destination project is scoped to include nested projects.

- If the destination project is set to **locked (including nested projects)**, the permissions for the project being moved are overwritten.
- If the destination project is set to **locked** (not including nested projects), the permissions for the project being moved are not overwritten. Whether or not the moved project is locked or customizable is preserved from its original setting.
- If the destination project is set to **customizable**, the permissions for the project being moved are not overwritten but they are now editable.
 - If the project being moved was previously nested under a parent that was *locked (including nested projects)*, when moved, the project takes on the setting of *locked (including nested projects)* and becomes the managing project for any projects it contains. Note: This is the same outcome if a project is moved to become a top-level project.

Effective permissions

A permission rule establishes who is impacted (a group or user) and what Capabilities they are **Allowed**, **Denied**, or **Unspecified**. While it seems straightforward to simply set a permission rule and have that be the whole story, whether a user has a capability may be unclear because of membership in multiple groups and the interplay of site roles and ownership with permission rules.

Multiple factors are evaluated in a specific order, yielding *effective permissions* on a piece of content.

Tip: To help keep things as straightforward as possible, we recommend (1) setting permission rules for groups instead of users, (2) managing permissions locked at the project level instead of setting permissions on individual content, and (3) deleting the All User group's permission rule or setting all capabilities to None.

A capability is allowed for a user if and only if the following three conditions are all met:

- The capability is within the scope of their site role.
- They have that capability:
 - based on a specific user scenario (such as being the content owner or a project leader, or they're an administrator site role),
 - OR
 - because they have been allowed the capability as a user,
 - OR
 - because they are both in a group that has been allowed the capability and no rules deny them the capability as a user or member of another group.
- There is no conflicting permissions settings at another content level that takes precedence.

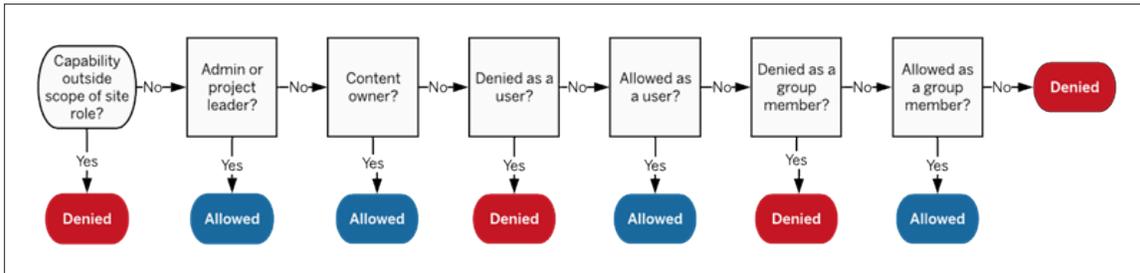
Any other situation denies the user the capability.

Hovering over a capability brings up a tooltip that explains the effective permission. Here are some common examples of why effective permissions—what the user can or can't do in actuality—might appear different than what a given permission rule states:

- A user might have a capability they are denied in a permission rule because their site role includes it (administrators).
- A user might have a capability they are denied in a permission rule because their user scenario allows it (because they own the content or are a project owner or leader).
- A user might lack a capability they are allowed in a permission rule because their site role doesn't allow it.
- A user might lack a capability they are allowed in a permission rule because a conflicting group or user rule denied it.
- A user might lack a capability they are allowed in a permission rule at one level of content (such as a workbook) because another level of content denied it (such as a view).

Evaluate permission rules

Permissions in Tableau are restrictive. Unless a capability is granted to a user, they are denied permission. The following logic evaluates if a capability is allowed or denied for an individual:



1. **Site role:** If a site role doesn't permit a capability, the user is denied. If the user's site role does permit the capability, then specific user scenarios are evaluated.
 - For example, a Viewer site role can't web edit. See [General capabilities allowed with each site role](#) for more information on what each site role can do.
2. **Specific user scenarios:**
 - If the user is an admin they have all capabilities on all content.
 - If the user is a project owner or project leader, they have all capabilities on all content in their projects.
 - If the user is the content owner, they have all capabilities* on their content.
 - If these scenarios do not apply to the user, then user rules are evaluated.

*Exception: Content owners won't have the **Set Permissions** capability in projects where permissions are locked. Only administrators, project owners, and project leaders can set permission rules in locked projects.

3. **User rules:** If the user is denied a capability, it is denied. If they are allowed a capability, it is allowed. If a capability is unspecified, then group rules are evaluated.
4. **Group rules:** If the user is in *any* group that is denied a capability, it is denied. If the user is in a group that is allowed a capability (and not in any groups that are denied that capability), it is allowed.
 - That is to say, if a user is a member in two groups, and one is allowed a capability and one is denied the same capability, the denial takes precedence for that user and they are denied.

5. If none of the above conditions apply, the user is denied that capability. In effect, this means that capabilities left as unspecified will result in denied.

A final effective permission of **Allowed** therefore occurs in three circumstances:

- Allowed by site role (Server Administrator, Site Administrator Creator, Site Administrator Explorer)
- Allowed because the user is the content owner, project owner, or project leader
- Allowed by a group or user rule (and not denied by a rule of higher precedence)

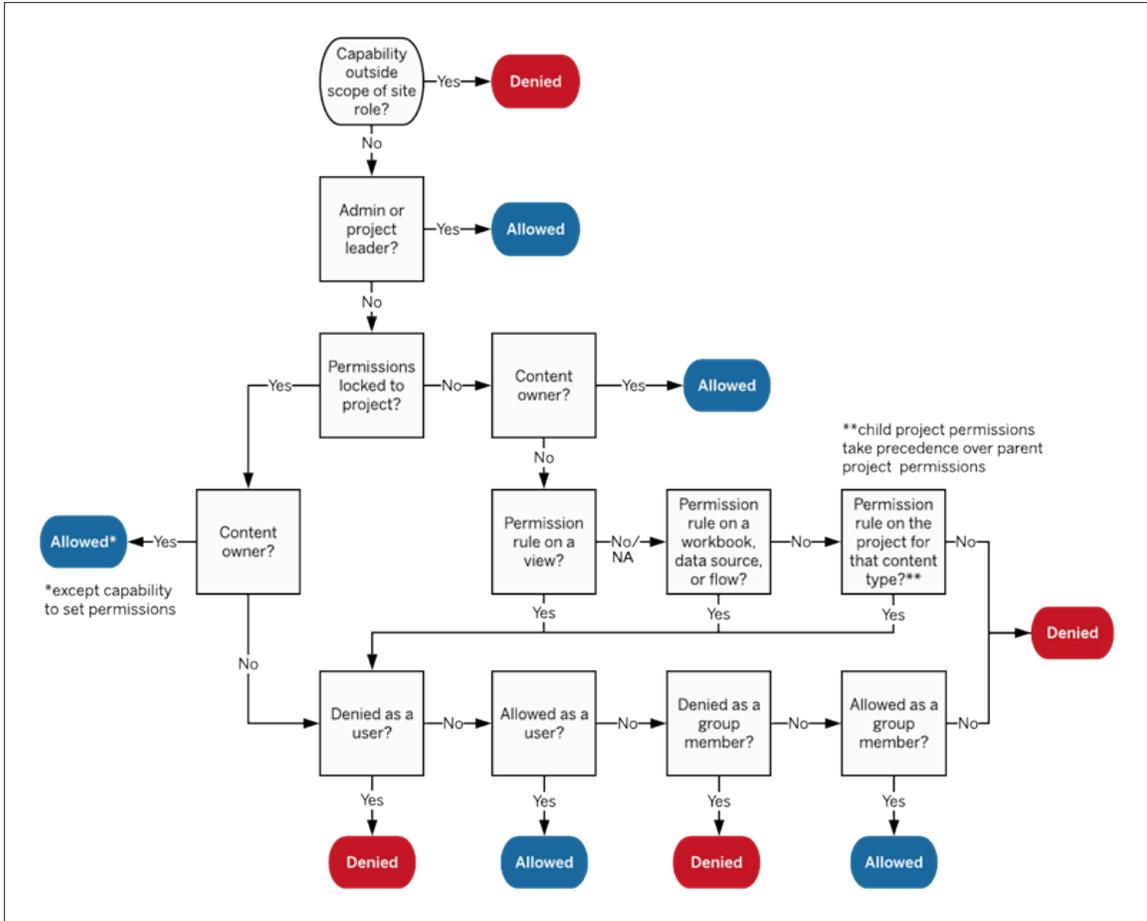
Denied occurs in three circumstances:

- Denied by site role
- Denied by a rule (and not allowed by a rule of higher precedence)
- Not granted by any rule

Evaluate permissions set at multiple levels

If project content permissions are *customizable*, it's possible to configure permission rules in multiple places. There are specific rules that determine what permissions are applied on the content.

- If there are nested projects, permissions set at the child level take precedence over permissions set at the parent level.
- Changes to permissions at the project level are not enforced for existing content.
- If there are permissions set on content (workbook, data source, or flow) during or after publication, these take precedence over rules set at the project level.
- If a workbook doesn't show navigational sheet tabs, any changes to the workbook-level permissions *won't* be inherited by the views and any changes to permissions must be done on the view.
- Configuring the workbook to show navigational sheet tabs will override existing view-level permissions and sync them with the workbook-level permissions. See [Show or Hide Sheet Tabs](#).



This image shows how capabilities are evaluated through multiple levels of content.

Permissions on views

In a workbook that is not in a locked project and does not show sheets as tabs for navigation, views (sheets, dashboards, stories) inherit the workbook permissions at publication, but any changes to permission rules must be made on individual views. View capabilities are the same as those for workbooks, except for **Overwrite**, **Download Workbook/Save a Copy**, and **Move** which are only available at the workbook level.

We recommend showing navigational sheet tabs whenever possible so views continue to inherit their permissions from the workbook. For more information, see Effective permissions.

Permissions, Site Roles, and Licenses

Adding a user to a Tableau Server requires a license. (Users can also be added as unlicensed and configured so they will consume a license only when they first sign in. For more information, see [Grant License on Sign In](#).) For each site the user belongs to they have exactly one site role, restricted by their license. A user has permissions for content on the site, restricted by what their site role allows.

Licenses and site roles apply to users. Permission capabilities apply to content.

Licenses are assigned to a user when they are created (or sign in for the first time) on the Tableau Server or Tableau Online site. Users are licensed as a **Creator**, **Explorer**, or **Viewer**.

- License levels are consumed based on the maximum *site role* a user can have on that server.
 - Server Administrator, Site Administrator Creator, and Creator site roles use a Creator license.
 - Site Administrator Explorer, Explorer (can publish), and Explorer site roles use at least an Explorer license.
 - Viewer site role uses at least a Viewer license.
 - An unlicensed user can exist on the server but they cannot log in unless they were added with grant site role on sign in.
- For Tableau Server, a user consumes only one license per server, even if they are a member of multiple sites. If a user is a member of multiple sites, their required license level is determined by their highest site role. (For example, if a user has a Creator site role in one site and a Viewer site role in two others, they consume a Creator license.)

Site roles are assigned to a user for each site they are a member of.

- Site roles determine the maximum capabilities a user can have in that site. (For example, a user with a site role of Viewer will never be able to download a data source even if that capability is explicitly granted to them on a specific data source.)
- Site roles do not inherently grant any capabilities in and of themselves—with the exception of the administrator site roles. Administrators always have all capabilities applicable to their license level.

Permissions consist of *capabilities*, like the ability to save to a project, web edit a workbook, connect to a data source, etc. They apply to group or user on a specific piece of content (project, data source, workbook, view, or flow).

- Permission capabilities are not given to a group or user in a vacuum but rather in the context of content. A user can have different capabilities for different content assets.
- Permissions are evaluated based on the interplay of a user’s site role and the permission rules for that user or any groups they are members of.
- Some actions such as web authoring might require combinations of capabilities. For more information, see Permission settings for specific scenarios.

Site roles and their maximum capabilities

These tables indicates what capabilities are available to each site role. There may be other ways for a user with a site role to perform a similar action. For example, although Viewers can’t be given the **Share Customized** capability to make their custom views visible to others on the workbook, they can share custom views by copying the view URL. See General capabilities allowed with each site role for more information on what each site role can do.

Projects

Capability	Creator	Explorer (can publish)	Explorer	Viewer
 View				
 Publish				

Workbooks

Capability	Creator	Explorer (can publish)	Explorer	Viewer
 View				
 Filter				

 View Comments				
 Add Comments				
 Download Image/PDF				
 Download Summary Data				
 Share Customized				
 Download Full Data				
 Web Edit				
 Download Workbook/Save a Copy				
 Overwrite				
 Move			*	
 Delete				
 Set Permissions				

Tableau Server on Linux Administrator Guide

Data Sources

Capability	Creator	Explorer (can publish)	Explorer	Viewer
 View				
 Connect				
 Download				

Data Source

 Overwrite				
 Delete				
 Set Permissions				

Data Roles

Capability	Creator	Explorer (can publish)	Explorer	Viewer
 View				
 Overwrite				
 Move			*	
 Delete				
 Set Permissions				

Flows

Note that Flows are part of the Data Management.

Capability	Creator	Explorer (can publish)	Explorer	Viewer
 View				
 Download				
Flow				
 Run Flow				
 Overwrite				
 Move			*	
 Delete				
 Set Permissions				

Ask Data Lenses

Capability	Creator	Explorer (can publish)	Explorer	Viewer
 View				
 Overwrite				
 Move			*	
 Delete				

Tableau Server on Linux Administrator Guide

 Set Per- missions				
--	---	---	--	---

Metrics

Capability	Creator	Explorer (can publish)	Explorer	Viewer
 View				
 Overwrite				
 Move			*	
 Delete				
 Set Per- missions				

Collections

Capability	Creator	Explorer (can publish)	Explorer	Viewer
 View				

*Although the Explorer role can be given the **Move** capability, they can't have the **Publish** capability on a project and therefore there is no place for them to move content to. The **Move** capability should therefore be considered not possible for Explorer site roles.

Quick Start: Permissions

You can use permission rules to control access to content on a site. A permission rule is a set capabilities that defines the level of access a group or user has on a content item. Content items are projects you create, and the workbooks and data sources published to them.

The most efficient way to manage permissions is to remove permissions from the **All Users** group before you create new groups for your environment. Then assign the permissions to groups at the project level.

1 Add users to groups

Create groups for users based on who should have the same permissions, and then add users to those groups. Within a site, select **Groups**. Select a group name, and then select **Add Users**.

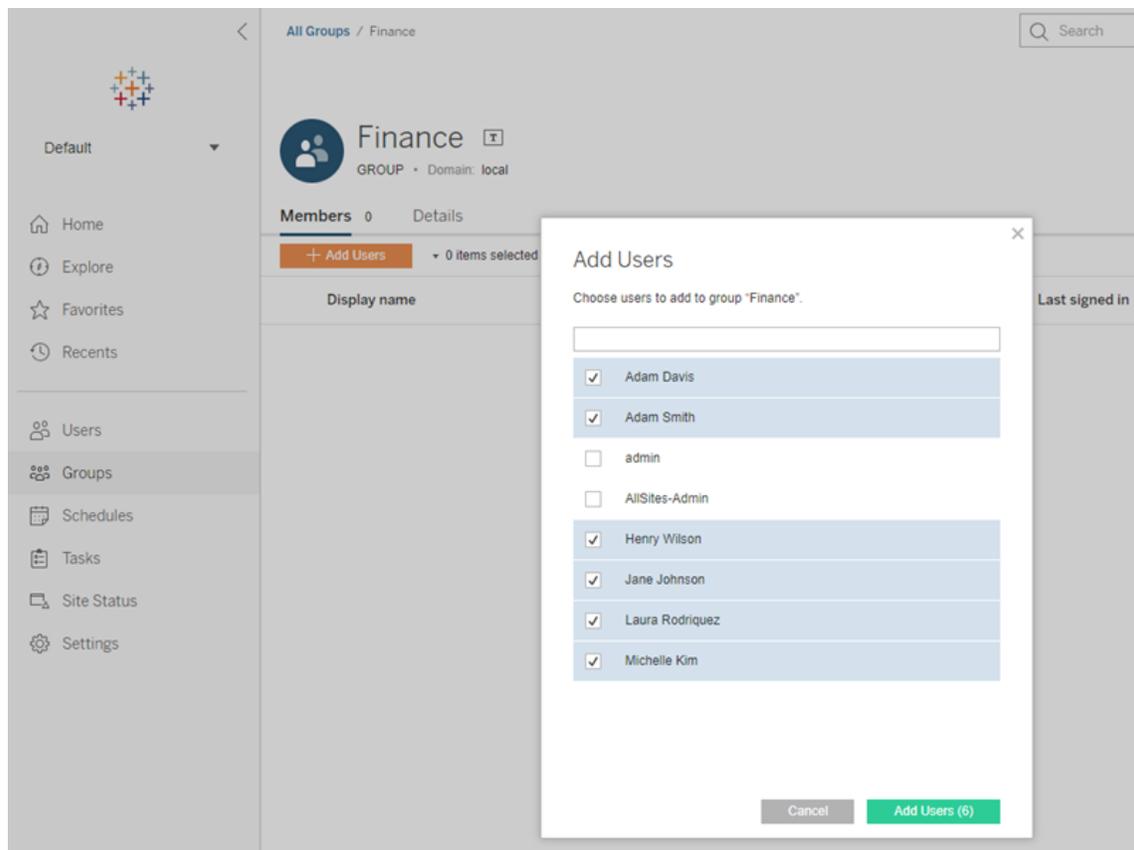
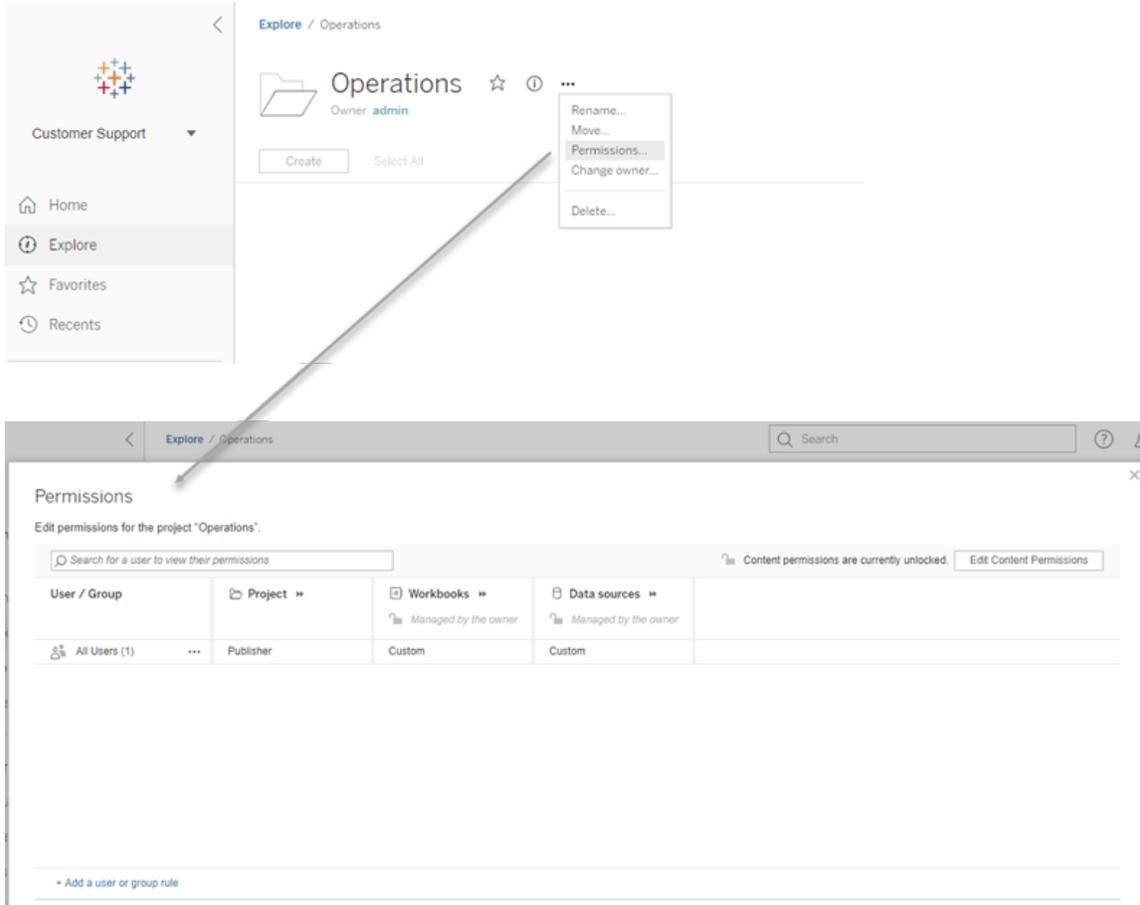


Tableau Server on Linux Administrator Guide

2 Open a project's permissions settings

The site's **Content** page shows the top-level projects. Navigate to the project you want to update, select its **Actions (...)** menu, and then select **Permissions**.

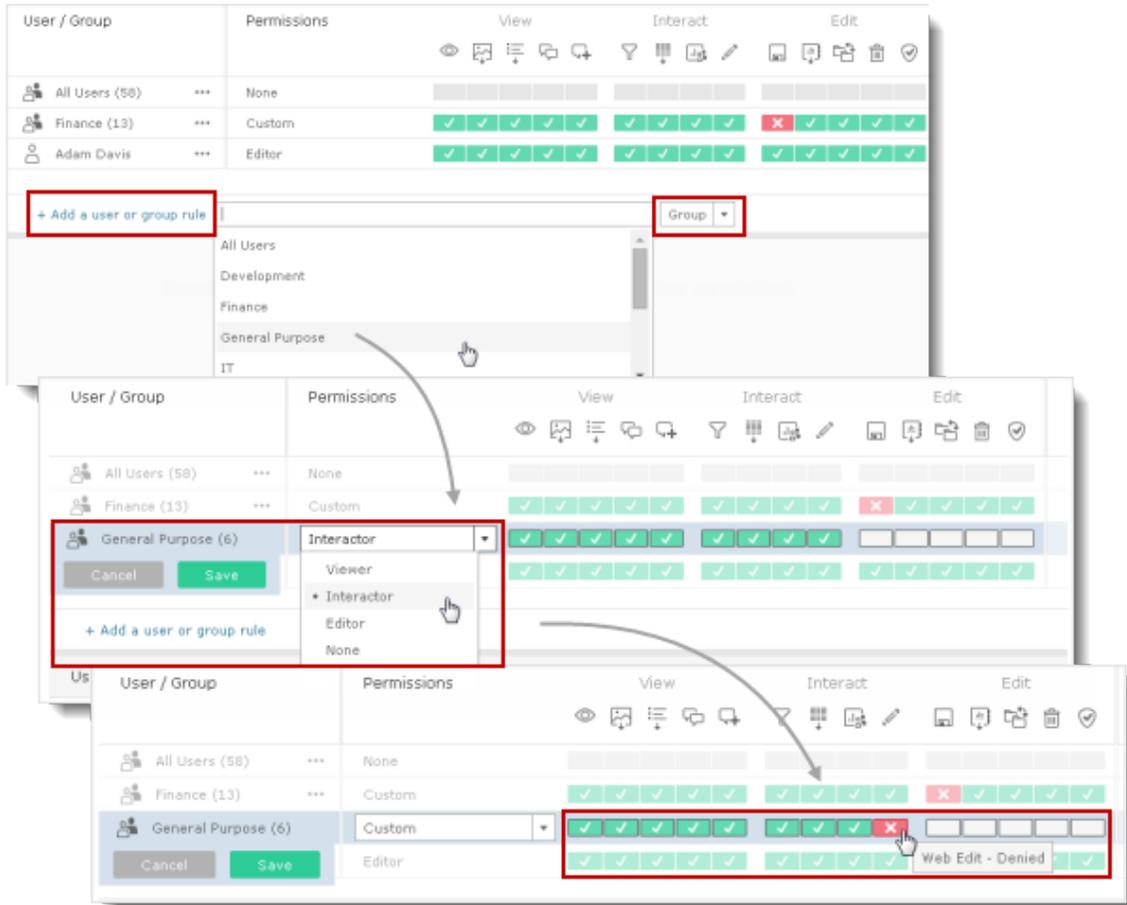


If you're not sure where to find a child project, display filters, and select **Show all projects**.

3 Create a permissions rule

Click **Add a user or group rule**, select **Group**, and then find and select the group.

Select a permissions role template to apply an initial set of capabilities for the group. Click a capability to set it to **Allowed** or **Denied**, or leave it **Unspecified**. Click **Save** when you are done.



Whether a user can set permissions is based on their site role and how their **Set Permissions** capability is set.

4 View a user's effective permissions

After you save the permissions rule for the group, you can view the effective permissions for that content.

Click a group name to see the group's users and their permissions. Hover over a capability box to see a tooltip with details on whether a capability is allowed or denied.

Tableau Server on Linux Administrator Guide

User / Group	Permissions	View	Interact	Edit
All Users (58)	None			
Finance (13)	Custom	✓	✓	✓
General Purpose (...)	Custom	✓	✓	✗
Adam Davis	Editor	✓	✓	✓

User Permissions <i>General Purpose (6)</i>	
Harold Pawlan	Custom
Henry MacAllister	Viewer
Henry Wilson	Administrator
Irene Maddox	Custom
Janet Molinari	Custom
Karen Daniels	Viewer

Custom indicates a user’s capabilities have been changed from the initial settings for their site role or content role.

For more information, see [Permissions](#).

Site roles

A user’s site role determines the maximum permissions allowed for that user.

- Server and site administrators can access all site content and take actions on it.
- Owners always get full access to the content they’ve published. When the parent project permissions are not locked, owners can change permissions for their published content.

For more information, see [Set Users’ Site Roles and Use Projects to Manage Content Access](#).

Permissions evaluation

- **Denied** takes precedence over **Allowed**.
- **Unspecified** results in **Denied** if no other permissions are specified.

- Specific user permissions on content take precedence over group permissions on content. In other words, user permissions trump group permissions.

For a couple of best-practice steps for how to implement permissions, see the following:

- Configure Projects, Groups, and Permissions for Managed Self-Service
- [Structure Content Projects, Groups, and Permissions](#) (links to *Everybody's Admin Guide*)

Manage Content Ownership

When you publish a data source or workbook on Tableau Server or when you create a project, you become its *owner*. A content owner, a project leader with an appropriate site role, or an administrator can change ownership of a content asset. After ownership is reassigned, the original owner has no special connection to the content item, and their ability to access it is determined by their permissions on the project or that specific item.

Who can change or be given ownership, by content type

Whether you can change or be given ownership depends on your permissions and your relationship to the content asset, as described in the following table.

Note: Full project leader access is available only with some site roles. For information, see Project-level administration.

Content asset type	Who can change ownership	Who can be given ownership
Top-level projects	Server administrator ¹ Site administrator	Server administrator Site administrator (Creator and Explorer) Creator Explorer (can publish)
Child (nested) projects	Server administrator Site administrator Project owner	Any administrator or owner, excluding Explorer and Viewer.
Workbooks and data	Server administrator	Any administrator or user of the

sources	<p>Site administrator</p> <p>Workbook or data source owner</p> <p>Project leader or owner of the project that contains the item</p>	<p>site, excluding Explorer and Viewer.</p>
Metrics	<p>Server administrator</p> <p>Site administrator</p> <p>Metric owner</p> <p>Project leader or owner of the project that contains the item</p>	<p>Any administrator or user of the site, excluding Explorer and Viewer.</p>
Ask Data lenses	<p>Server administrator</p> <p>Site administrator</p> <p>Lens owner</p> <p>Project leader or owner of the project that contains the item</p>	<p>Any administrator or user of the site, excluding Explorer and Viewer.</p>
Flows	<p>Server administrator</p> <p>Site administrator</p>	<p>Starting in version 2021.2, server and site administrators can only change the owner to themselves.</p>

Collections	Server administrator	Any administrator or user of the site, including Explorer and Viewer.
	Site administrator	
	Collection owner	

¹ The Server Administrator site role applies to Tableau Server only; not Tableau Online.

Considerations for changing content ownership

- Before you remove a user from Tableau Server, make sure they do not own any content assets.

If the user does own content, you must first reassign ownership of those assets before you can delete the user. Otherwise, their site role is set to **Unlicensed**, but they are not deleted, and only an administrator can take certain actions on that content.

- If you change the ownership of a workbook or data source that includes embedded credentials to connect to underlying data, the embedded credentials will be deleted. For flows, embedded credentials are preserved when changing ownership. Connections to published data sources are authenticated using the flow owner and authorized based on their permissions.

You can update the embedded credentials by editing the connection information on Tableau Server. For more information, see [Edit Connections](#). Alternatively, the new owner can download the flow, workbook, or data source and open the item in Tableau Desktop to update the embedded credentials and then re-publish the content.

- If you do not lock permissions to projects, make sure users you give content ownership to know your permissions guidelines, or you account for permissions as you change ownership. In unlocked projects, by default, content owners can set permissions on their content. For more information, see [Permissions](#).
- While it is possible to change the owner of a metric to a user with a site role of Viewer or Explorer, it is not recommended, because doing so will cause the metric refresh to be

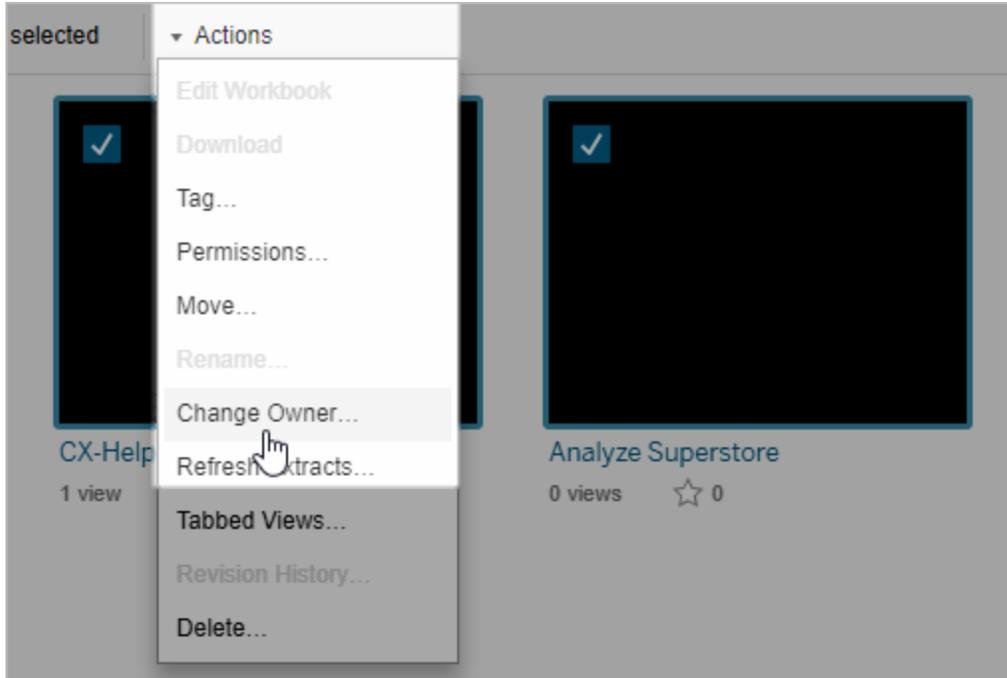
suspended. A site role of Creator or Explorer (can publish) is required to refresh, overwrite, or delete a metric.

Change the owner of a content resource

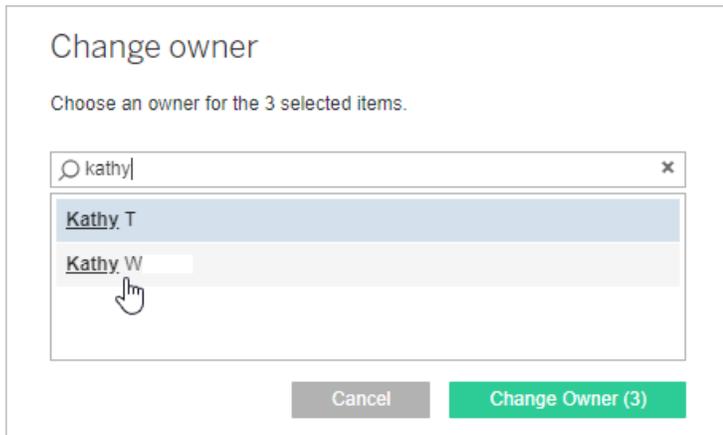
1. Sign in to the Tableau Server web environment, and from the navigation menu, select **Explore**.
2. Navigate to the content you want to assign to someone else.
 - If you want to reassign multiple of the same type of content, for example, multiple workbooks, select the content type from the drop-down menu.
 - If you want to reassign multiple items within the same project, navigate to the project.

If you're not sure where to find a child project, display filters, and select **Show all projects**.
 - If you want to reassign multiple content items with the same owner, find the user on the **Users** page.
3. Select the items you want to reassign, and then select **Actions > Change Owner**.

The other menu commands you see will depend on the content type.



4. Type the name of a user or select a user from the list.



5. Click **Change Owner**.

Manage Permissions for External Assets

Tableau Online and Tableau Server provide a space for accessing and managing published content. When Tableau Online or Tableau Server is licensed with the Data Management,

you have access to Tableau Catalog. Tableau Catalog adds a complementary space and a set of features across your site to track and manage metadata and lineage of external assets used by the content published to your site.

Tableau Catalog indexes content and assets

Catalog discovers, tracks, and stores metadata from the content that you publish to Tableau Online or Tableau Server.

Catalog indexes metadata for the following:

- **Tableau content:** workbooks, data sources, flows, projects, metrics, users, and sites
- **External assets:** databases and tables associated with Tableau content

Catalog classifies the metadata of any data that comes from outside the Tableau environment as external assets. The data that comes from outside the Tableau environment is stored in many different formats, such as a database server or a local .json file.

Catalog tracks only the metadata of the external data and does not track the underlying data in any form (raw or aggregated).

Catalog metadata includes the following:

- **Lineage information** or the relationship between items. For example, the Sales table has a relationship with both the Superstore data source and the Superstore Sample workbook.
- **Schema information.** Some examples include:
 - Table names, column names, and column types. For example, Table A contains Columns A, B, and C, which are types INT, VARCHAR, and VARCHAR.
 - Database name and server location. For example, Database_1 is a SQL Server database at <http://example.net>.

- Data source name, and the names and types of the fields the data source contains. For example, Superstore data source has fields AA, BB, and CC. Field CC is a calculated field that refers back to both field AA and field BB.
- **User curated, added, or managed information.** For example, item descriptions, certifications, user contacts, data quality warnings, and more.

How does Tableau Catalog work?

Tableau Catalog indexes all content published to Tableau Online or Tableau Server to track lineage and schema metadata. For example, the metadata comes from workbooks, packaged workbooks, data sources, and the Tableau Server or Tableau Online repository.

As part of the indexing process, lineage and schema metadata about external assets (databases and tables) used by the published content are also indexed.

Note: In addition to accessing Catalog from Tableau Online or Tableau Server, indexed metadata can also be accessed from the Tableau Metadata API and Tableau Server REST API. For more information about the Tableau Metadata API or metadata methods in the REST API, see [Tableau Metadata API](#) and [Metadata Methods](#) in the Tableau Server REST API, respectively.

Permissions on metadata

Permissions control who is allowed to see and manage external assets and what metadata (for both Tableau content and external assets) is shown through lineage.

Note: If Tableau Online or Tableau Server is not licensed with the Data Management, then by default, only admins can see database and table metadata through the Tableau Metadata API. This default can be changed to use "derived permissions," as described below.

Access metadata

The permissions used to access metadata through Catalog (or Metadata API) work similarly to permissions for accessing content through Tableau Online or Tableau Server, with some additional considerations for sensitive data that can be exposed through lineage and the capabilities granted on external assets.

Permissions on Tableau content

Catalog uses view and manage capabilities that are already used by existing Tableau content to control the metadata that you can see and manage on Tableau content. For more general information on these capabilities, see [Permissions](#).

Permissions on external assets using derived permissions

When Tableau Online or Tableau Server is licensed with the Data Management, by default Catalog uses *derived permissions* to automatically grant you capabilities to external assets in the following scenarios:

For **View** capability:

- If you are the owner of a workbook, data source, or flow, you can see the database and table metadata used *directly* by that workbook, data source, or flow. See Additional notes about lineage.
- If you are a project owner or project leader, you can see all the database and table metadata used by the content published to your project.
- Embedded files use the permissions of the source (such as the workbook, data source, or flow), rather than the derived permissions of the external asset (the database or table). For example, if you can see the workbook with an embedded file, you can see the embedded file and its metadata used by that workbook.

For both **Overwrite** and **Set Permissions** capabilities:

- If you are the owner of a flow, you can edit and manage permissions for the database and table metadata used by the flow output.

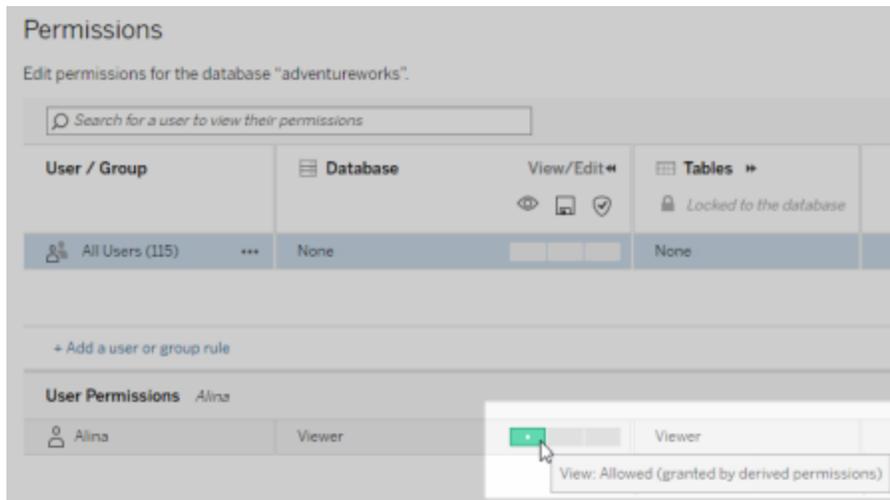
Note: For the flow cases above, the capabilities apply only after there has been at least one successful flow run under the current owner of the flow.

Check permissions

Tableau Server on Linux Administrator Guide

As an admin or someone who has been given the capability to set permissions for an asset, you can validate who has derived permissions by following the steps below.

1. Sign in to Tableau Online or Tableau Server.
2. From the left navigation pane, click **External Assets**.
3. From the drop-down menu, select **Databases and Files** or **Tables**.
Note: Local files, like .json or .csv files are grouped as external assets under **Data-bases**.
4. Select the check box next to the database or table whose permissions you want to modify, and then select **Actions > Permissions**.
5. In the Permissions dialog box, click **+ Add Group/User Rule** and start typing to search for a group or user.
6. Validate the permissions by clicking a group name or user name in the permission rules to see the effective permissions below.



Order of precedence in which Tableau evaluates derived permissions for external assets

When derived permissions are configured for your Tableau Online site or Tableau Server, each user's level of access to external assets depends on the associated Tableau content and the order of precedence of rules Tableau uses for its content.

Tableau follows the rules below, continuing on to the next rule, only if the current rule evaluates to "denied." If any rule evaluates to "allowed," the capability is allowed and Tableau stops evaluating. This rules list is based on the Permissions.

For **View** capability:

1. Admin role
2. License
3. Project leader (Tableau content)
4. Project owner (Tableau content)
5. Content owner (Tableau content)
6. *Derived permissions* (applies only to external assets and the View capability)
 - a. Admin role
 - b. License
 - c. Project leader (external assets)
 - d. Project owner (external assets)
 - e. Content owner (external assets)
7. Explicit permissions

For **Overwrite** and **Set Permissions** capabilities:

1. Admin role
2. License
3. Project leader (Tableau content)
4. Project owner (Tableau content)
5. Content owner (Tableau content)
6. Explicit permissions (Tableau content)
7. *Derived permissions* (applies only to external assets and the Overwrite and Set Permissions capabilities for flow outputs)

- a. Admin role
- b. License
- c. Project leader (external assets)
- d. Project owner (external assets)
- e. Content owner (external assets)

Turn off derived permissions

As an admin, you can turn off the derived permissions default setting for a site in favor of manually granting explicit permissions to databases and tables.

1. Sign in to Tableau Online or Tableau Server as an admin.
2. From the left navigation pane, click **Settings**.
3. On the General tab, under Automatic Access to Metadata about Databases and Tables, clear the **Automatically grant authorized users access to metadata about databases and tables** check box.

Note: Data quality warning messages on databases and tables that are visible to users through derived permissions remain visible to those users even when the check box is not selected.

Set permissions on individual external assets

In order to grant additional users permissions to view, edit (overwrite), and manage external assets, an admin can grant those capabilities explicitly on individual databases or tables for users or groups.

Database permissions act as a permissions template

Database permissions function like Permissions. In other words, when permissions are set at the database level, those permissions can serve as a template for any newly discovered and indexed child tables of that database. Furthermore, database permissions can also be locked so that the child tables will always use the permissions set at the database level.

Granting permission at the database level can help create a scalable process for enabling permissions to tables.

Summary of permissions capabilities

The following table shows the capabilities you can set for external assets (databases and tables):

Capability	Description	Template
 View	See the database or table asset.	View
 Overwrite	Add or edit data quality warnings and descriptions of the database or table asset. Prior to version 2020.1, the Overwrite capability was called Save.	Publish
 Set Permissions	Grant or deny permissions for the database or table asset.	Administer

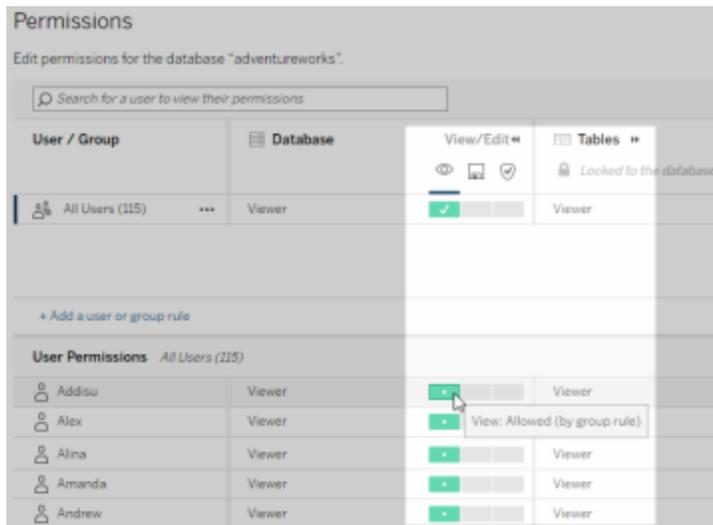
Set permissions on a database or table

To set permissions on databases or tables, use the following procedure.

1. Sign in to Tableau Online or Tableau Server as an admin or someone who has been granted the "Set Permissions" capability.
2. From the left navigation pane, click **External Assets**.
3. From the drop-down menu, select **Databases and Files** or **Tables**.
Note: Local files, like .json or .csv files are grouped as external assets under **Databases**.
4. Select the check box next to the database or table whose permissions you want to modify, and then select **Actions > Permissions**.

Tableau Server on Linux Administrator Guide

5. In the Permissions dialog box, click **+ Add Group/User Rule** and start typing to search for a group or user.
6. Select a permission role template to apply an initial set of capability for the group or user, and then click **Save**. Available templates are: View, Publish, Administer, None, and Denied.
7. To further customize the rule, click a capability in the rule to set it to Allowed or Denied, or leave it unspecified. Click save when you are done.
8. Configure any additional rules you want for other groups or users.
9. Validate the permissions clicking a group name or user name in the permission rules to see the effective permissions below.



Lock permissions to the database

To lock (or unlock) permissions to the database, use the following procedure.

1. Sign in to Tableau Online or Tableau Server as an admin or someone who has been granted the "Set Permissions" capability.
2. From the left navigation pane, click **External Assets**. By default, the External Assets

page shows a list of databases and files.

3. Select the check box next to the database whose permissions you want to lock, select **Actions > Permissions**, and then click the Table Permissions **Edit** link .
4. In the Table Permissions in Database dialog box, select **Locked** and click **Save**.
5. To unlock permissions, click **Edit** again, and select **Customized**.

Access lineage information

Catalog (and the Metadata API) can expose relationship and dependencies metadata, also referred to as *lineage*, among the content and assets on Tableau Online or Tableau Server.

Lineage can show three primary things:

- How items relate to each other, either directly or indirectly
- How many of those items relate to each other
- With the appropriate permissions, shows sensitive data about items in the lineage

Sensitive lineage data

In some cases, lineage can contain sensitive data, such as data quality warning messages, content or asset names, or related items and metadata.

By default, complete lineage information displays for all users while its sensitive data is blocked from specific users who don't have the appropriate View capabilities. The concept of blocking sensitive data is called obfuscation.

Obfuscation allows all metadata in the lineage to be visible while keeping its sensitive data blocked from specific users who don't have the appropriate View capabilities. This default enables workflows that rely on a complete impact analysis.

If obfuscating sensitive data in the lineage is not enough for your organization, certain parts of the lineage, including its sensitive data, can be filtered.

Filtering omits certain parts of the lineage (and lineage-related areas like data details) for specific users who don't have the appropriate View capabilities to its sensitive data.

Because filtering omits parts of lineage, it prevents workflows that rely on a complete impact analysis.

To change how sensitive data is handled, do the following:

1. Sign in to Tableau Online or Tableau Server as an admin.
2. From the left navigation pane, click **Settings**.
3. On the General tab, under **Sensitive Lineage Information**, select the radio button that best handles lineage information for all users on your Tableau Online site or Tableau Server.

Additional notes about lineage

- **If you have the View capability on related assets**, you can see when and what assets and content are related to each other, and their sensitive metadata.

For example, you can see 1) the names, data quality warnings, and total number of related upstream databases and tables and 2) the combined number of sheets (visible and hidden) in the lineage of the downstream workbook of the asset you are evaluating.

- **If you *don't* have the View capability on related assets**, you can always see when assets relate to each other.

For example, you can see 1) whether related upstream databases and tables exist in the lineage and 2) the total number of databases or total number of tables that are related to the asset you are evaluating.

However, you can't see the metadata associated with those assets when you don't have the view capability for them. When metadata is blocked because of limited permissions, you see **Permissions Required**.

External Assets / Opportunities

Opportunities

About

Hostname: login.salesforce.com Database Type: Salesforce

<input type="checkbox"/>	Type	Name	Workbooks
<input type="checkbox"/>	Table	Permissions Required	2
<input type="checkbox"/>	Table	Permissions Required	0
<input type="checkbox"/>	Table	Price	0
<input type="checkbox"/>	Table	Product	74
<input type="checkbox"/>	Table	User	109
<input type="checkbox"/>	Table	Permissions Required	1

- If you *don't* have the **View** capability on related assets, you can always see whether the assets are certified.

However, the level of detail that you can't see if you don't have View capability is the sensitive information related to the certification, like the names of the related databases and tables. When metadata is blocked because of limited permissions, you see **Permissions Required**.

Tables (6)				
	Type	Name	Workbooks	Data Sources
<input type="checkbox"/>	Table	Permissions Required	1	1
<input type="checkbox"/>	Table	Permissions Required	0	1
<input type="checkbox"/>	Table	Permissions Required	0	1
<input type="checkbox"/>	Table	REI	2	4
<input type="checkbox"/>	Table	Permissions Required	0	1
<input type="checkbox"/>	Table		0	5

On this asset

- Under maintenance

Permissions Required

Set by workgroupuser
Aug 7, 2019, 10:23 AM

For more information about lineage see Use Lineage for Impact Analysis.

Additional notes about tags discoverable through lineage data

In addition to Tableau content, external assets can also be tagged. Although tags are always visible, tagged items that you see through lineage data can either be obfuscated (default) or filtered as described earlier in this topic.

When tagged items are obfuscated:

- **If you have the View capability for tagged items**, you can see the tagged items and related tagged items, and all metadata.
- **If you don't have the View capability for tagged items:**
 - You can see the type of tagged and related tagged items but you can't see sensitive metadata about the items. For example, suppose you use a tag filter to see items with the tag "Noteworthy." Although you can see that there are database items tagged with "Noteworthy," you can't see the names of the tagged databases.
 - You can see how many related tagged items there are. For example, suppose

you do a tag query on “Noteworthy.” Your query returns five tagged databases.

When tagged items are filtered, the tagged and related tagged items you see are limited to only the items that you have the View capability for.

For more information about tags, see [Tagged Items](#) in the Tableau User Help.

Potential mismatch between asset results and content results

When Catalog shows lineage information, it provides information between content and assets. Catalog lineage always shows the true count or result of associated items. However, elsewhere in Tableau Online or Tableau Server, you might see fewer number of items. One reason for this is because of your View capabilities. Outside of Catalog, or elsewhere in Tableau Online or Tableau Server, you see a filtered count or result of the content that you have access to according to your content permissions.

For example, suppose you're looking at the Superstore data source. The lineage for the Superstore data source can show how many upstream underlying tables the data source connects to and how many downstream workbooks rely on the data source. However, because you might not have the View capability on all of those downstream workbooks, the total number of related workbooks might be different when you're looking at Catalog lineage information versus the total number of workbooks represented in the **Connected Workbooks** tab.

There might be other reasons why, which are not related to permissions, you might see a mismatch between asset counts and content counts. For more information, see [Use Lineage for Impact Analysis](#).

Who can do this

The following information summarizes the types of users who can do the tasks described in this topic.

Tableau Online site or Tableau Server admin

Tableau Server on Linux Administrator Guide

Data Management	Capability	Requirements
Licensed	See assets and their metadata	None

Data Management	Capability	Requirements
	Edit assets and their metadata	None
	Change permission on assets and their metadata	None
	Grant users ability to see assets and their metadata	<p>Default: When “derived permissions” is on, your users can see metadata on external assets for the content that they own, or for the content that is published to a project that they are a project leader or project owner of.</p> <p>Ad-hoc: You can configure explicit View permissions on a specified external asset.</p>
	Grant users ability to edit assets and their metadata	You can configure explicit "write" or Overwrite permissions on a specified external asset (if not automatically granted because the user is a flow owner) .
	Grant users ability to change permissions on assets and their metadata	You can configure explicit "edit" or Set Permissions on a specified external asset (if not automatically granted because the user is a flow owner) .

Data Management	Capability	Requirements
Not licensed	See all assets and their metadata	Applies to Metadata API only
	Edit assets and their metadata	Requires the Data Management
	Change permission on assets and their metadata	Requires the Data Management
	Grant users ability to see assets and their metadata	Applies to Metadata API only: You can turn on derived permissions as described above. If “derived permissions” is on, your users can see metadata on external assets for the content that they own, or for the content that is published to a project that they are a project leader or project owner of.
	Grant users ability to edit assets and their metadata	Requires the Data Management
	Grant users ability to change permissions on assets and their metadata	Requires the Data Management

User with Creator or Explorer license

Data Management	Capability	Requirements
Licensed	See assets and their metadata	<p>Default: When "derived permissions" is enabled by your Tableau Online site admin or Tableau Server admin, you can see metadata on external assets for the content that you own, or for the content that is published to a project that you are a project leader or project owner of.</p> <p>Ad-hoc: You can see metadata on external assets that you have been granted explicit View permissions to.</p>
	Edit assets and their metadata	You can edit metadata on an external asset that you have been granted explicit "write" or Overwrite permissions to (if not automatically granted because the user is a flow owner).
	Change permissions on assets and their metadata	You can change permissions on an external asset that you have been granted explicit "edit" or Set Permissions to ((if not automatically granted

Data Management	Capability	Requirements
		because the user is a flow owner).
	Grant other users permissions to see assets and their metadata	You can change permissions on an external asset that you have been granted explicit "edit" or Set Permissions to ((if not automatically granted because the user is a flow owner).
Not licensed	See assets and their metadata	<p>Applies to Metadata API only:</p> <p>If “derived permissions” is enabled by your Tableau Online site admin or Tableau Server admin, you can see metadata on external assets for the content that you own, or for the content that is published to a project that you area project leader or project owner of.</p>
	Edit assets and their metadata	Requires the Data Management
	Change permissions on assets and their metadata	
	Grant other users permissions to see assets and their metadata	

Manage Data

You can connect to and manage the data you that you use in Tableau.

Tableau Server Data Sources

When your Tableau users want to share data connections they've defined, they can publish data sources to Tableau Server. When a data source is published to the server, other users can connect to it from their own workbooks, as they do other types of data. When the data in the Tableau data source is updated, all workbooks that connect to it pick up the changes.

Looking for Tableau Server on Windows? See [Tableau Server Data Sources](#).

A Tableau Server data source consists of metadata that describes the following:

- **The connection information:** Defines whether the data is in a live database or an extract, and which of that data to bring in to Tableau.
- **Customization and cleanup:** Includes information that facilitates efficient use of the data. For example, calculations, sets, groups, bins, parameters, custom field formatting, and so on.
- **Data access and refresh instructions:** Includes the location of the underlying database server (whether on-premises or in the cloud), network paths for file-based data, security information such as credentials or access tokens, and related information.

In addition to helping your users create data consistency and reliability, using Tableau data sources offers advantages to you as the administrator. Because multiple workbooks can connect to one data source, you can minimize data source proliferation and save on storage space and processing time. When someone downloads a workbook that connects to a Tableau data source that in turn has an extract connection, the extract stays on the server, reducing network traffic. Finally, if a connection requires a database driver, you need to install and maintain the driver only on the server, instead of on each user's computer. If you

use Tableau Online, all supported drivers are available to data sources published to your site.

Managing data sources

You can perform some or all management tasks on a data source if you have one of the following levels of access:

- Site or server administrator
- Project leader or owner of the project the data source is published to

Full project leader access is available only with some site roles. For information, see [Project-level administration](#).

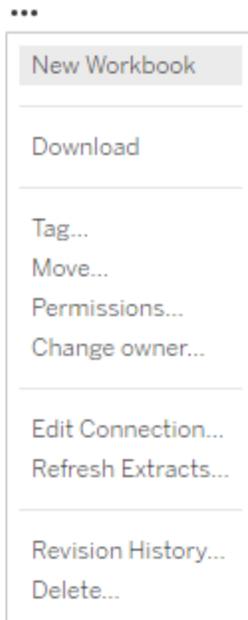
- Data source owner

Unless you're a site or server administrator, you might not have access to all management tasks. For example, if the Permissions menu is not available, chances are that the project permissions are locked, which denies setting permissions on individual workbooks and data sources.

Tip: A best practice is to designate a person or team to manage all data sources published to a project or to the whole site, so that all data sources can be maintained under the same set of guidelines.

To perform the management tasks that you have access to, do the following:

1. Sign in to the site, and on the **Content** tab, select **Explore > Data sources**.
2. On a data source, select the **Actions (...)** menu.



- **New workbook or Download:** Create a new workbook in the browser environment that connects to this data source. Or download the data source to use locally.
- **Tag:** Add or remove keyword tags. Tags can contain a single word or multiple words, delimited by a comma.
- **Move:** Move a data source from one project to another. This requires specific settings on each project. For information, see [Required access level for moving content](#).
- **Permissions:** View or update permissions that specify which users or groups can connect to, modify, or download the data source. As mentioned at the beginning of this section, if this action is not available, the project permissions might be locked, and only the project leader or administrator can change permissions.
- **Change owner:** Making someone an owner gives them complete access to it.
- **Refresh extracts:** If a data source includes an extract, you can assign the extract to a refresh schedule.

For information, see [Refresh Data on a Schedule](#).

- **View the data source's revision history**
- **Delete:** Deleting a data source affects workbooks that connect to the data source. Before you delete a data source, ensure that there are no workbooks that connect to the data source or edit the workbooks to use another data source.

In addition, for data sources that are proxy connections, administrators can stay aware of how users authenticate to the database, and whether the appropriate drivers are installed. For information, see [Database Drivers and Data Security](#).

Restrictions

Published data sources often function as curated and trustworthy data sources. As such, there are restrictions on how they can be modified and used.

Aliases and calculations can't be edited.

- New aliases also can't be created. Fields can be duplicated and the copy can be aliased.
- New calculations can be created. Existing calculation can also be copied and the copy can be edited.

Relationships and joins can't be edited.

Published data sources cannot be used in joins or relationships.

- Use blends if you need to combine published data sources.

Extract Upgrade to .hyper Format

Beginning in version 10.5, new extracts use the .hyper format instead of the .tde format. Extracts in the .hyper format take advantage of an improved data engine, which supports the same fast analytical and query performance as the data engine before it, but for even larger extracts.

This article covers what you and your users should be aware of as a result of the extract format change in version 2021.2 and how the change impacts your users.

Changes to extracts in version 10.5

With the introduction of Tableau's improved data engine technology, Hyper, comes a change to the extract format. As mentioned above, extracts created in version 2021.2 use the .hyper format. Though Tableau version 2021.2 can continue to read .tde extracts, it cannot create new .tde extracts. The impact of this format change means that when certain extract tasks are performed either by a user or by Tableau Server, such as an extract refresh or append data, the .tde extract is automatically upgraded and converted to a .hyper extract.

After an extract is upgraded to a .hyper extract, your users can interact with the .hyper extract as they would a .tde extract. However, your users should be aware of the following backward limitations:

- The upgrade can't be reversed. An upgraded extract can't be converted back to a .tde extract.
- The upgraded extract can't be opened in Tableau Desktop 10.4 and earlier.
- Workbooks that contain upgraded extracts can't be published as an older version from Tableau Desktop to Tableau Server.
- Workbooks that contain upgraded extracts can't be downgraded as an older version in Tableau Desktop.
- Workbooks that contain an upgraded extract, which are downloaded from Tableau Server using the **Download Tableau Workbook** option, cannot be opened in Tableau Desktop 10.4 and earlier.

Tasks that cause extract upgrade

There are three distinct ways a .tde extract can get upgraded to a .hyper extract: 1.) during an extract refresh (full or incremental), 2.) when appending data to an extract, and 3.) when an extract is upgraded manually using Tableau Desktop 2021.2. After an extract has been upgraded, the original .tde extract is automatically removed from Tableau Server if it's not being reference by other workbooks.

The followings tasks on Tableau Server automatically upgrades and converts a .tde extract to a .hyper extract:

- Manual extract refresh
- Scheduled extract refresh
- Automated refresh task that's performed through tabcmd
- Automated refresh task that's performed through the Extract Command-Line Utility
- Automated append data to an extract using tabcmd
- Automated append data to an extract using the Extract Command-Line Utility
- Encryption of a .tde extract. For more information, see [Extract Encryption at Rest](#).

Impact of extract upgrade

Tableau recommends that the Tableau Desktop version in your environment be upgraded to match Tableau Server 2021.2. If they don't match, extract compatibility can be an issue depending on the task your users want to perform.

To get a better understanding of when extract upgrades can occur and potential compatibility issues your users might experience, review a detailed explanation of the [extract compatibility scenarios](#) on the Tableau Support page or a summary explanation below.

Note: In the tables below, "10.4" represents Tableau 10.4 and earlier and "10.5" represents Tableau 10.5 and later.

Performing tasks on Tableau Server 10.5

When working with extracts created in Tableau Desktop 10.4 and earlier, your users should be aware of the following extract-related compatibility scenarios around common tasks performed on Tableau Server 10.5.

Task	10.4 workbook	10.5 workbook	
	.tde extract	.tde extract	.hyper extract
Publish from Tableau Desktop 10.4	√	Not possible	Not possible
Publish from Tableau Desktop 10.5	√	√	√
Refresh, scheduled refresh, or append	Workbook version remains unchanged, but extract upgrades to .hyper format	Workbook version remains unchanged, but extract upgrades to .hyper format	√
Edit/save in web authoring	Workbook version changes to 10.5, extract remains in .tde format	√	√
Edit/save in web authoring then refresh or append	Workbook version changes to 10.5, and extract upgrades to .hyper format	Workbook version remains unchanged, but extract upgrades to .hyper format	√

Tableau Server on Linux Administrator Guide

Downloading from Tableau Server 10.5

When working with extracts created in Tableau Desktop 10.4 and earlier, your users should be aware of the following extract-related compatibility scenarios when downloading from Tableau Server 10.5.

Task		10.4 work-book	10.5 workbook	
		.tde extract	.tde extract	.hyper extract
Download and open	in Tableau Desktop 10.4	√	Can't open workbook; you see a "this workbook uses a .hyper extract and is not compatible with this version; open the workbook in version 10.5 or later" error message, and then asked to locate the extract	Can't open workbook; you see a "this file was created by a newer version; upgrade Tableau" error message
Download and open	in Tableau Desktop 10.5	√	√	√
Export As Version	from Tableau Desktop 10.5	√	√	Option is greyed out

Automating refresh and append tasks

When working with extracts created in Tableau Desktop 10.4 and earlier, your users should be aware of the following extract-related compatibility scenarios around automating refresh and append tasks using `tabcmd` or the Tableau Command-Line Utility.

Task		10.4 workbook	10.5 workbook	
		.tde extract	.tde extract	.hyper extract
Refresh or append	using 10.4 <code>tabcmd</code>	Workbook version remains unchanged, but extract upgrades to .hyper format	Workbook version remains unchanged, but extract upgrades to .hyper format	√
	using 10.5 <code>tabcmd</code>	Workbook version remains unchanged, but extract upgrades to .hyper format	Workbook version remains unchanged, but extract upgrades to .hyper format	√
Publish	using 10.4 <code>tabcmd</code>	√	√	Not possible
	using 10.5 <code>tabcmd</code>	Not possible	Not possible	√
Refresh or append	using 10.4 Tableau Command-Line Utility	√	Not possible	Not possible
	using 10.5 Tableau Command-Line Utility	Workbook version remains unchanged, but extract upgrades to .hyper format	Workbook version remains unchanged, but extract upgrades to .hyper format	√

Why keep an extract in .tde format?

If the version of Tableau Desktop cannot be upgraded to match Tableau Server 2021.2, your users will need to keep their extracts in .tde format.

How to keep an extract in .tde format

To keep an extract in the .tde format, the extract should not be upgraded. To help your users keep their extracts from upgrading, advise them against performing any of the tasks listed above in the Extract Upgrade to .hyper Format section. Then, consider the following suggestions to maintain a .tde version of an extract:

- Disable existing extract refresh schedules on Tableau Server until you can identify which extracts should and shouldn't be upgraded.

Note: To access an extract's refresh schedule, you must be a data source owner, or be a server administrator, site administrator, or Project Leader. For more information about which site roles allow full Project Leader capabilities, see Project-level administration.

- Continue to perform extract refresh and append data tasks using Tableau Server 2021.2, but maintain an earlier version of Tableau Desktop and the ability to connect to the original data in case you need to recreate the .tde extract.
- Use an earlier version of Tableau Desktop to perform extract tasks, such as extract refresh or append data.

What to expect after extract upgrade

After an extract upgrade has taken place, your users should expect some additional changes when working with extracts in version 2021.2. For more information, see [After an extract upgrade](#) section in Tableau Help.

Set the Site Time Zone for Extracts

The default time zone for extract-based data sources in a site is Coordinated Universal Time (UTC). Site administrators can set a different time zone.

To set the site time zone for extracts:

1. Sign in as a site administrator.
2. On the site you want to configure, click `Settings`.
3. In the `Site Time Zone for Extracts` section, select a time zone and then click `Save`.

In calculated fields, functions such as `NOW()` or `TODAY()` look at the time zone. For more information about Extracts, see [Extract Your Data](#) in the Tableau Desktop help.

The timezone setting, in addition to being used for extract-based data sources, also affects internal extracts. One example is when Tableau connects to file-based data sources like text files, it automatically creates an extract in the backend. Another example is data integration, where Tableau uses an extract in the backend to integrate data from different sources.

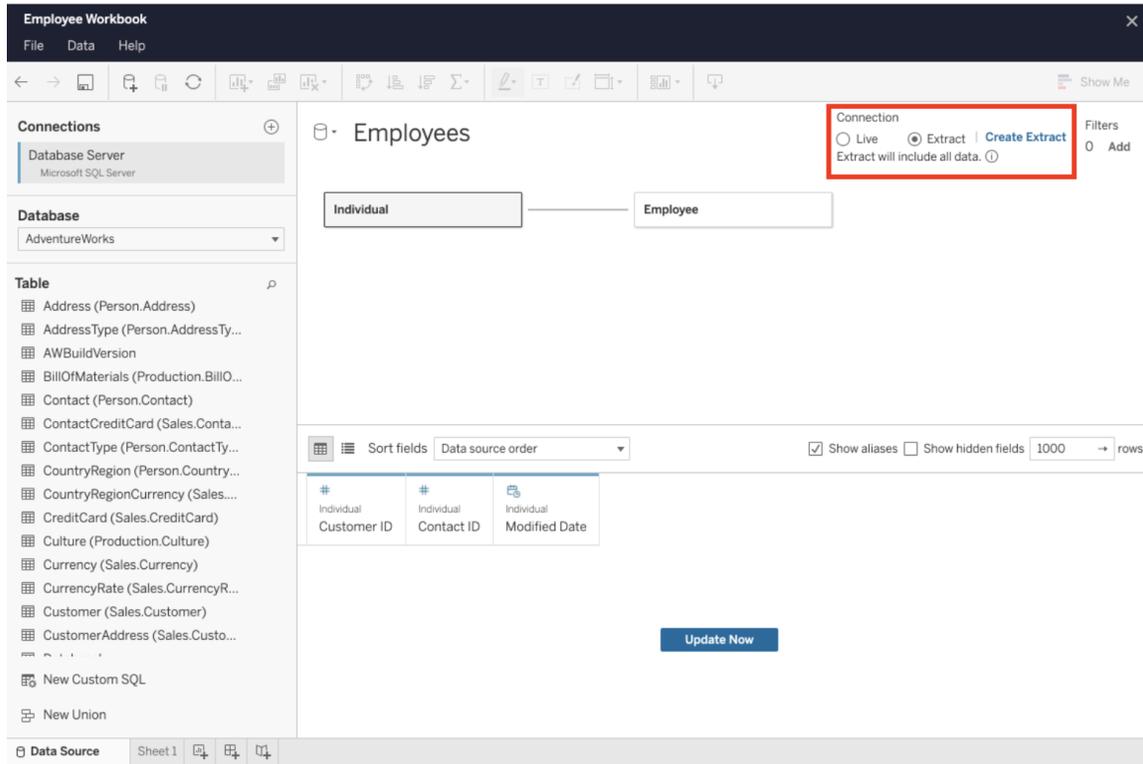
Create Extracts on the Web

You can extract your data sources in the web (without using Tableau Desktop) to improve data source performance and support additional analytical functions. When you extract your data source, Tableau will copy the data from your remote data store to Tableau Server or Online. To learn more about the benefits of extracting your data, see [Extract Your Data](#). In the web, you can extract while in Web Authoring or while in Content Server.

Create extracts in Web Authoring

You can create extracts directly in web authoring with default extract settings.

Extract an Embedded Data Source in Web Authoring



To create an extract in web authoring:

Tip: It is recommended to finalize your data model before you create the extract. Extract creation may take a long time and any changes to your data model, such as adding new logical tables, will invalidate the extract.

1. Click the **Data Source** tab in the bottom left corner of the web authoring pane. For new workbooks, you will start in the **Data Source** tab.
2. In the top-right corner, change the connection type from **Live** to **Extract**.
3. Click **Create Extract**. You will see the **Creating Extract** dialog box.

Extract creation might take a long time and you can close your authoring session while the extract is being created. To ensure your extract creation is not lost, in the dialog box, click **Notify Me When Complete** to specify a location for the extracted workbook to be saved. If your extract succeeds, your workbook will be saved to the specified location and you will be

notified that you can continue your web authoring session. If your extract creation fails, you will be notified that the extract could not be created and you can restore your unsaved changes by reopening the original workbook in web authoring.

Define your Extract Settings

Extract Data [Close]

Specify how to store data in the extract

Logical Tables Physical Tables

Store data using one table for each logical table. [Learn more](#)
Use this option if you need to use extract filters, aggregation, or other extract settings.

Specify how much data to extract

Filters
0 [Edit](#)

Aggregation
No [Edit](#)

Number of Rows
All rows [Edit](#)

Optionally, configure one or more of the following options to tell Tableau how to store, define filters for, and limit the amount of data in your extract:

- **Decide how the extract data should be stored**

You can choose to have Tableau store the data in your extract using one of two structures (schemas): logical tables (denormalized schema) or physical tables (normalized schema). For more information about logical and physical tables, see [The Tableau Data Model](#).

The option you choose depends on what you need.

- **Logical Tables**

Stores data using one extract table for each logical table in the data source. Physical tables that define a logical table are merged and stored with that logical table. For example, if a data source was made of a single logical table, the data would be stored in a single table. If a data source was made of three logical tables (each containing multiple physical tables), the extract data would be stored in three tables—one for each logical table.

Select **Logical Tables** when you want to limit the amount of data in your extract with additional extract properties like extract filters, aggregation, Top N, or other features that require denormalized data. Also use when your data uses pass-through functions (RAWSQL). This is the default structure Tableau uses to store extract data. If you use this option when your extract contains joins, the joins are applied when the extract is created.

- **Physical Tables**

Stores data using one extract table for each physical table in the data source.

Select **Physical Tables** if your extract is comprised of tables combined with one or more equality joins and meets the conditions for using the Physical Tables option listed below. If you use this option, joins are performed at query time.

This option can potentially improve performance and help reduce the size of the extract file. For more information about how Tableau recommends you use the Physical Tables option, see [Tips for using the Physical Tables option](#) in the Tableau Desktop help. In some cases, you can also use this option as a work-around for row-level security. For more information about row-level security using Tableau, see [Restrict Access at the Data Row Level](#) in the Tableau Desktop help.

Conditions for using the Physical Tables option

To store your extract using the Physical Tables option, the data in your extract must meet all of the conditions listed below.

- All joins between physical tables are equality (=) joins
- Data types of the columns used for relationships or joins are identical
- No pass-through functions (RAWSQL) used
- No incremental refresh configured
- No extract filters configured
- No Top N or sampling configured

When the extract is stored as physical tables, you cannot append data to it. For logical tables, you can't append data to extracts that have more than one logical table.

Note: Both the Logical Tables and Physical Tables options only affect how the data in your extract is stored. The options do not affect how tables in your extract are displayed on the Data Source page.

- **Determine how much data to extract**

Click **Add** to define one or more filters to limit how much data gets extracted based on fields and their values.

- **Aggregate the data in the extract**

Select **Aggregate data for visible dimensions** to aggregate the measures using their default aggregation. Aggregating the data consolidates rows, can minimize the size of the extract file, and increase performance.

When you choose to aggregate the data, you can also select **Roll up dates** to a specified date level such as Year, Month, etc. The examples below show how the data will be extracted for each aggregation option you can choose.

<p>Original data</p>	<p>Each record is shown as a separate row in your data.</p> <table border="1"> <thead> <tr> <th></th> <th>Date</th> <th>Region</th> <th>Sales</th> </tr> </thead> <tbody> <tr><td>2</td><td>1/1/2009</td><td>South</td><td>\$500</td></tr> <tr><td>3</td><td>1/1/2009</td><td>West</td><td>\$200</td></tr> <tr><td>4</td><td>1/1/2009</td><td>West</td><td>\$100</td></tr> <tr><td>5</td><td>1/1/2009</td><td>East</td><td>\$300</td></tr> <tr><td>6</td><td>1/2/2009</td><td>South</td><td>\$600</td></tr> <tr><td>7</td><td>1/2/2009</td><td>South</td><td>\$400</td></tr> <tr><td>8</td><td>1/2/2009</td><td>East</td><td>\$100</td></tr> <tr><td>9</td><td></td><td></td><td></td></tr> </tbody> </table>		Date	Region	Sales	2	1/1/2009	South	\$500	3	1/1/2009	West	\$200	4	1/1/2009	West	\$100	5	1/1/2009	East	\$300	6	1/2/2009	South	\$600	7	1/2/2009	South	\$400	8	1/2/2009	East	\$100	9				<p>There are seven rows in the extract.</p>
	Date	Region	Sales																																			
2	1/1/2009	South	\$500																																			
3	1/1/2009	West	\$200																																			
4	1/1/2009	West	\$100																																			
5	1/1/2009	East	\$300																																			
6	1/2/2009	South	\$600																																			
7	1/2/2009	South	\$400																																			
8	1/2/2009	East	\$100																																			
9																																						
<p>Aggregate data for visible dimensions <i>(no roll up)</i></p>	<p>Records with the same date and region have been aggregated into a single row. There are five rows in the extract.</p> <table border="1"> <thead> <tr> <th></th> <th>Date</th> <th>Region</th> <th>Sales</th> </tr> </thead> <tbody> <tr><td>2</td><td>1/1/2009</td><td>East</td><td>\$300</td></tr> <tr><td>3</td><td>1/1/2009</td><td>South</td><td>\$500</td></tr> <tr><td>4</td><td>1/1/2009</td><td>West</td><td>\$300</td></tr> <tr><td>5</td><td>1/2/2009</td><td>East</td><td>\$100</td></tr> <tr><td>6</td><td>1/2/2009</td><td>South</td><td>\$1,000</td></tr> <tr><td>7</td><td></td><td></td><td></td></tr> </tbody> </table>		Date	Region	Sales	2	1/1/2009	East	\$300	3	1/1/2009	South	\$500	4	1/1/2009	West	\$300	5	1/2/2009	East	\$100	6	1/2/2009	South	\$1,000	7				<p>There are five rows in the extract.</p>								
	Date	Region	Sales																																			
2	1/1/2009	East	\$300																																			
3	1/1/2009	South	\$500																																			
4	1/1/2009	West	\$300																																			
5	1/2/2009	East	\$100																																			
6	1/2/2009	South	\$1,000																																			
7																																						
<p>Aggregate data for visible dimensions <i>(roll up dates to Month)</i></p>	<p>Dates have been rolled up to the Month level and records with the same region have been aggregated into a single row. There are three rows in the extract.</p> <table border="1"> <thead> <tr> <th></th> <th>Date</th> <th>Region</th> <th>Sales</th> </tr> </thead> <tbody> <tr><td>2</td><td>1/1/2009</td><td>East</td><td>\$400</td></tr> <tr><td>3</td><td>1/1/2009</td><td>South</td><td>\$1,500</td></tr> <tr><td>4</td><td>1/1/2009</td><td>West</td><td>\$300</td></tr> <tr><td>5</td><td></td><td></td><td></td></tr> </tbody> </table>		Date	Region	Sales	2	1/1/2009	East	\$400	3	1/1/2009	South	\$1,500	4	1/1/2009	West	\$300	5				<p>There are three rows in the extract.</p>																
	Date	Region	Sales																																			
2	1/1/2009	East	\$400																																			
3	1/1/2009	South	\$1,500																																			
4	1/1/2009	West	\$300																																			
5																																						

- **Choose the rows to extract**

Select the number of rows you want to extract.

You can extract **All rows** or the **Top *N* rows**. Tableau first applies any filters and aggregation and then extracts the number of rows from the filtered and aggregated results. The number of rows options depend on the type of data source you are extracting from.

Notes:

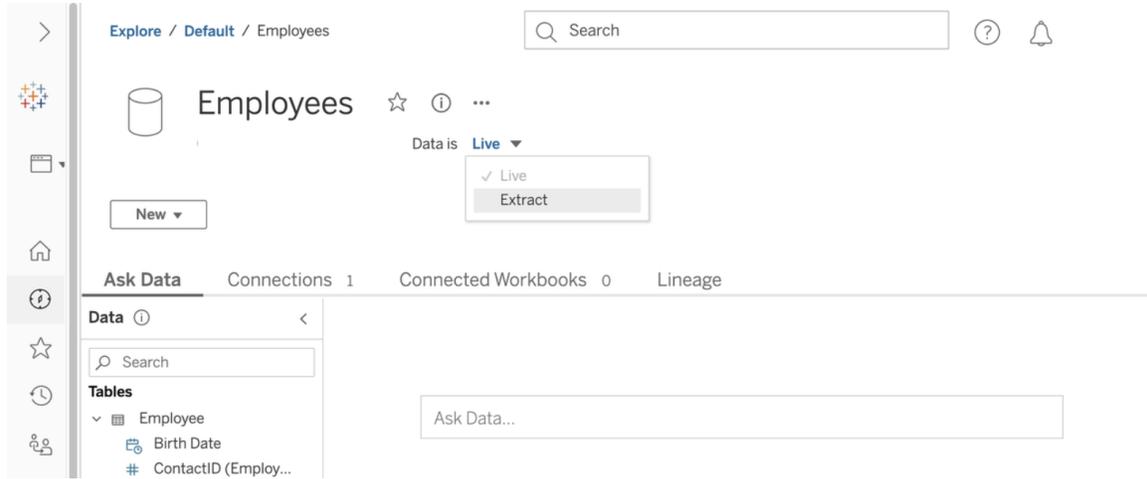
- Not all data sources support sampling. Therefore, you might not see the **Sampling** option in the Extract Data dialog box.
- Any fields that you hide first in the Data Source page or on the sheet tab will be excluded from the extract.

Limitations

- You can't create extracts for embedded data sources that reference published data sources. As a workaround, create the extract directly on the published data source. For more information, see [Extract a Published Data Source on Content Server](#).
- You can't create extracts for file-based data sources. File-based data sources already have special performance features and adding extraction will have no performance benefit.
- This feature does not apply to bridge-based data sources in Tableau Online.

Create extracts in Content Server

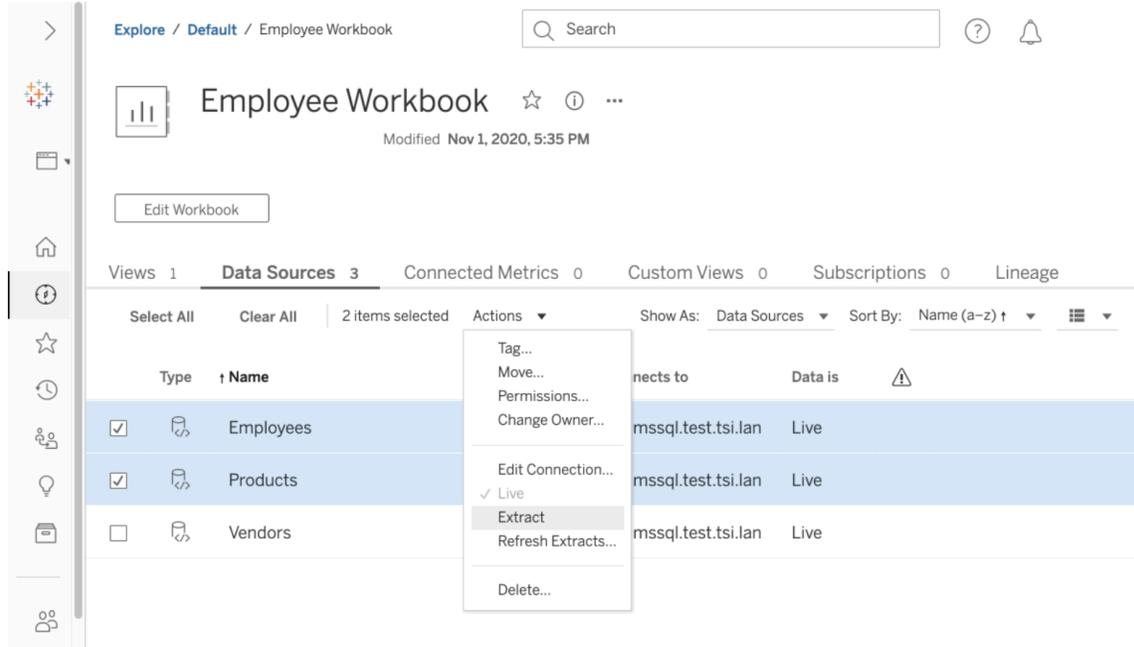
Extract a Published Data Source on Content Server



To extract a published data source:

1. Sign in as an administrator or as the owner of the data source.
2. On the Content tab, select **Explore > Data sources**.
3. Select a data source by clicking on the Data Source name.
4. At the top of the screen, under the Data Source name, select the drop-down menu that says **Live**.
5. Change the connection type from **Live** to **Extract**. If the extract encryption at rest feature is enabled on the site, select either **Encrypted** or **Unencrypted**.
6. If you see an error message about embedded credentials, embed your credentials in the data source. To do this, click **Edit Connection**. Select "Embedded password in connection" and then click **Save**.

Extract an Embedded Data Source on Content Server



To extract one or more data sources that are embedded in a published workbook:

1. Sign in as an administrator or as the owner of the data source.
2. Navigate to the published workbook.
3. Navigate to the Data Sources tab
4. Select one or more of the data sources.
5. Click the **Action** button.
6. Click **Extract**. If the extract encryption at rest feature is enabled on the site, select either **Encrypted** or **Unencrypted**.

Limitations

- Your connection credentials must be embedded in the data source.
- In the web, you can't specify extract settings like incremental refresh and extract filters.
- You can't create extracts for embedded data sources that reference published data sources. As a workaround, create the extract directly on the published data source.

- You can't create extracts for file-based data sources. File-based data sources already have special performance features and adding extraction will have no performance benefit.
- This feature does not apply to bridge-based data sources in Tableau Online.

Keep Extracted Data Fresh

After data is extracted, you can optionally set up an extract refresh schedule to keep the data fresh. For more information, see [Refresh Data on a Schedule](#).

Monitor and Manage Extracts

Server administrators can monitor extract creation on the **Background Tasks for Extracts** admin view. For more information, see [Background Tasks for Extracts](#).

Server administrators can manage extracts on the Jobs page. For more information, see [Managing Background Jobs in Tableau Server](#).

Extract creation jobs, like extract refresh jobs, have a maximum query limit before they timeout. This is to prevent jobs from running forever and using an unbounded amount of server resources. The extract query limit timeout can be configured by server admins using the TSM command line interface configuration setting `backgrounder.querylimit`. For more information, see [tsm configuration set Options](#).

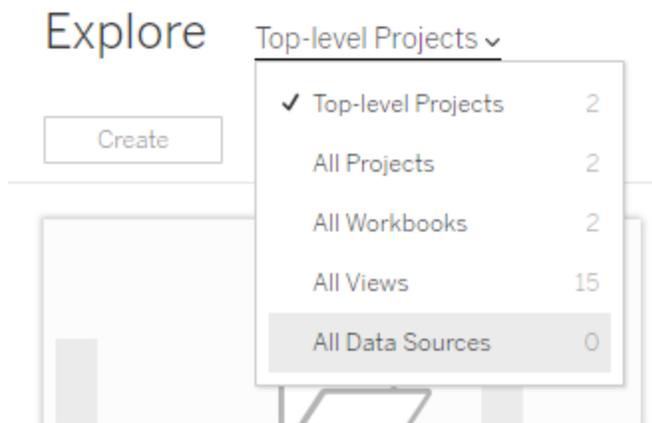
Server administrators can manage web authoring. For more information, see [Set a Site's Web Authoring Access and Functions](#).

View Data Source Attributes

In the **Content** area of the Tableau Server web authoring environment, you can filter the view to show only data sources or connections and their attributes.

View data sources by name

To filter by data source name, under **Explore**, select **Data sources**.

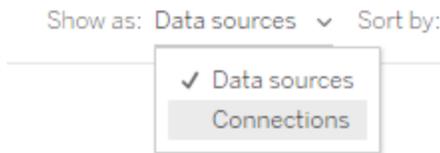


Attributes in the data source name list include the following:

- **Type**—The icon next to the data source name indicates whether the data source is published as a live connection (📄), an extract connection (📄), or is embedded in a workbook (📄).
 - Select the name of a published data source to open its data source page, with tabs for viewing connections and workbooks that connect to it.
 - Select the name of an embedded data source to open the workbook associated with it, as well as tabs for viewing other data it connects to.
- **Connects To**—Indicates the name of the database server or data file. This could be a database outside of Tableau Server, an extract, or a published data source.
- **Live or Last Extract**—This column tells you whether the connection to the data is live, or, if it is a connection to an extract, when the extract was last updated.

View a list of connections

To filter by connection type, in the **View** list, select **Connections**.



Connection attributes include:

- **Connects to**—Indicates the name of the database server or data file. This could be a database outside of Tableau Server, an extract, or a published data source.
- **Connection type**—Shows the type of data. **Tableau Server** indicates that the connection is to a data source published on the site. **Tableau Data Engine** means the data source has an extract stored on the Tableau data server.

Use Certification to Help Users Find Trusted Data

In a self-service environment with multiple publishers, it's common for a project on Tableau Server to contain a variety of content that is named similarly, or is based on the same or similar underlying data, or is published without any descriptive information about it. When this is the case, analysts might lack confidence about the data they should use.

To help your users find the data that's trusted and recommended for their type of analysis, you can *certify* the data that complies with your organization's data standards.

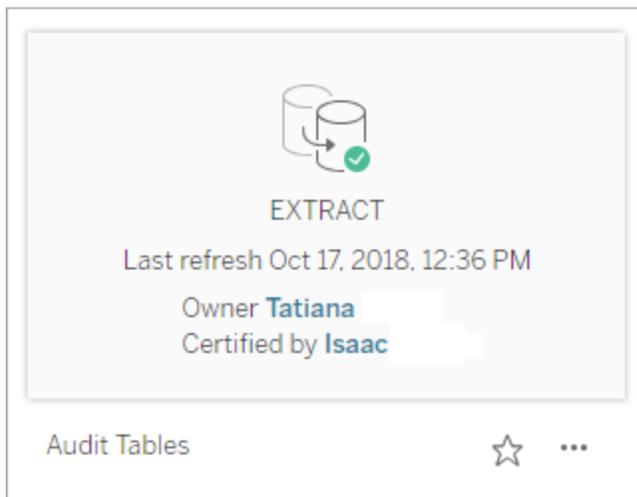
Certification complements the Recommendations Training Schedule feature by offering a way to promote data through curation.

Starting in 2019.3, Tableau Catalog is available in the Data Management Add-on to Tableau Server and Tableau Online. When Tableau Catalog is enabled in your environment, in addition to certifying published data sources, you can also certify the databases and tables that are associated with your workbook, flow, or data source content. For more information about Tableau Catalog, see "About Tableau Catalog" in the [Tableau Server](#) or [Tableau Online](#) Help.

How certification helps users find trusted data

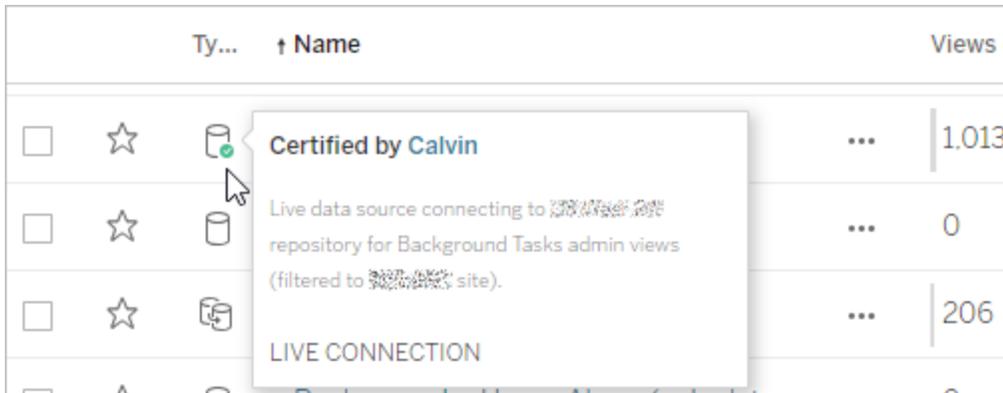
When you certify a data source (or database or table, if you have Tableau Catalog in your environment), it gets a green check mark on its icon in the list view or thumbnail in the grid view. This appears anywhere the data source (or database or table) is shown, including when connecting to data in web authoring or from Tableau Desktop.

	Type	Name	
<input type="checkbox"/>	☆	 Audit Tables	...
<input type="checkbox"/>	☆	 Background Tasks	...
<input type="checkbox"/>	☆	 Resources and Extracts	...



Certified data sources rank higher in search results and are added to recommended data sources.

In addition, you can provide notes about the certification status, which appear in a tooltip when users hover over the data source icon or card. The tooltip in list view also shows who certified the data source and the data source's connection type.



For more information, see the How to certify data steps below.

Create guidelines for selecting data to certify

As with most Tableau functionality, certification is flexible. You can define for your organization the criteria you use to determine when to certify a data source (or database or table). As you do this, you might want to document and share your guidelines. As new data sources are published, the guidelines can help you and other administrators or project leaders to be consistent with your certification choices. They can also help users understand what certification means.

Whether you use the same certification criteria across all projects, or define unique criteria for each project, the important thing is to be clear about what certification means in your environment.

Who can certify data

To certify a data source, you must

- be an administrator, or
- have a site role of Explorer (Can Publish) or Creator and have the Project Leader capability on the project containing the data you want to certify.

To certify databases or tables, you must have Tableau Catalog enabled in your environment and either of the following permissions levels:

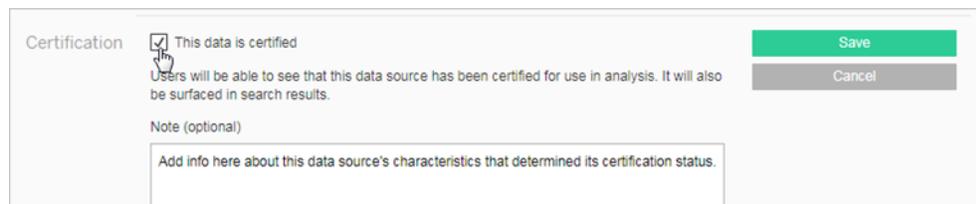
- Site Administrator site role.
- "Manage permissions" capability on a database to certify that database or any tables within that database.

How to certify data

The data you can certify depends on whether you have Tableau Catalog enabled in your environment. All users with the right permissions can certify data sources. Tableau Catalog users with the right permissions can also certify databases, tables, and files.

Certify data sources

1. Sign in to Tableau Server.
2. To certify a data source, on the **Explore** page, select **Data Sources**.
3. On the page, select the data source, and then select the **Details** icon .
4. Next to **Certification**, select **Edit Certification Status**, and then do the following:
 - a. Select the **This data is certified** check box.
 - b. Add a note that gives users context for the certification status, intended use for the data, or other helpful information.



Certification This data is certified

Users will be able to see that this data source has been certified for use in analysis. It will also be surfaced in search results.

Note (optional)

Add info here about this data source's characteristics that determined its certification status.

Save

Cancel

Information you add to the **Note** section appears in the certification badge tooltip, mentioned earlier in How certification helps users find trusted data.

Certify databases and tables

When Tableau Catalog is enabled in your environment, you can certify databases and tables, if you have the right permissions.

1. Sign in to Tableau Server.
2. To certify a database or table, on the **External Assets** page, select **Databases and Files** or **Tables**.

3. On the page, select the More actions menu (...) next to the asset name you want to certify.
4. Select **Edit Certification** and then do the following:
 - a. Select the **This data is certified** check box.
 - b. Add a note that gives users context for the certification status, intended use for the data, or other helpful information, and then click **Save**.

Certification

This data is certified

When enabled, you certify this data for use in analysis. Certification is visible to all users.

Note (optional)

Add information here about how this data source meets your certification standards.

Cancel Save

Information you add to the **Note** section appears in the certification badge tooltip, mentioned earlier in How certification helps users find trusted data.

Keep Data Fresh

You can manually refresh data, as well as schedule data refreshes.

Manage Refresh Tasks

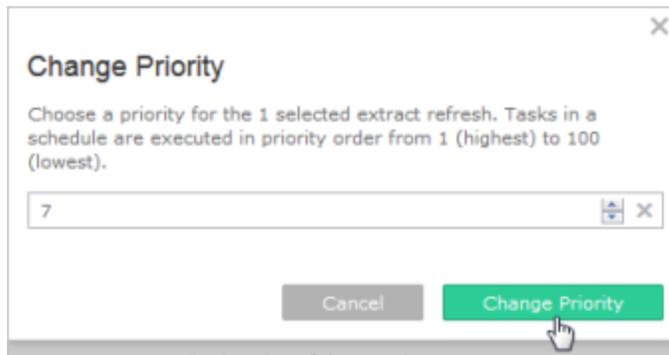
Administrators can change the priority of scheduled extract refreshes relative to other server tasks, manually refresh extracts, or delete their schedules.

1. Sign in to the site that has the schedules you want to manage, and then click **Tasks**.
2. Select one or more scheduled extract refreshes.
3. From the **Actions** menu, do any of the following:
 - Select **Change Schedule**, and choose a new schedule from the list.

- Select **Run Now** to refresh manually.

Note: If an extract does not have a scheduled refresh, you can refresh it on demand from the Data Connections page.

- Select **Change Priority**, and enter a number between 1 and 100 to move the extract up or down in the priority list.



- Select **Delete** to completely remove the schedule for the selected data sources.

See also

Enable Extract Refresh Scheduling and Failure Notification

Refresh Data on a Schedule

You can schedule refresh tasks for published extract data sources and published workbooks that connect to extracts. New schedules can be created by Tableau Server Administrators on the **Schedules** page. For more information, see [Create or Modify a Schedule](#).

Note: When a refresh is performed on extracts created in Tableau 10.4 and earlier (that is, a .tde extract), the extract is upgraded to a .hyper extract automatically. While there are many benefits of upgrading to a .hyper extract, your users won't be able to open the extract with earlier versions of Tableau Desktop. For more information, see [Extract Upgrade to .hyper Format](#).

For information on how to refresh flow outputs, see [Schedule a Flow Task](#).

1. When you're signed in to Tableau Server, select **Explore** from the left navigation pane, and then, depending on the type of content you want to refresh, select **All Workbooks** or **All Data Sources** from the drop-down menu.
2. Select the check box for the workbook or data source you want to refresh, and then select **Actions > Refresh Extracts**.
3. In the Refresh Extracts dialog, select **Schedule a Refresh**, and complete the following steps:
 - Select the schedule you want.
 - If available, specify whether you want a full or incremental refresh.

A full refresh is performed by default. Incremental refresh is available only if you configured for it in Tableau Desktop before publishing the extract. For more information, see [Refreshing Extracts](#) in the Tableau Help.

- Click **Schedule Refresh**.

Refresh Extracts

Refresh Now **Schedule a Refresh**

Choose a refresh schedule for workbook "Books".

Search

- CacheWarming - Cache
- CacheWarming - daily
- CacheWarming - monthly
- CacheWarming - weekly
- End of the month
- Every 15 minutes
- ExtractDaily0
- ExtractHourly0
- ExtractMonthly0
- ExtractWeekly0

Cancel Schedule Refresh

Quick Start: Refresh Extracts on a Schedule

For published workbooks that connect to data extracts, you can set up the server to refresh the extracts on a recurring schedule, so all workbooks connected to them always show the most up-to-date data.

To schedule refreshes you need to have administrator or data owner permissions.

Note: This topic applies to extracts published to Tableau Server. For Tableau Online, how you refresh extracts depends on the underlying data they connect to. For more information, see [Keep Data Fresh](#).

1 Set up a schedule on the server

Sign in to the server, go to the **Schedules** page, and click **New Schedule**.

Schedules 4							
+ New Schedule		▼ 1 selected		▼ Actions			
↑ Name		Frequency	Task type	Tasks	Execution	Next run at	
<input checked="" type="checkbox"/>	📅 End of the month	...	Monthly	Extract Refresh	0	Parallel	Jul 31, 2016, 11:00 PM
<input type="checkbox"/>	📅 Saturday night	...	Weekly	Extract Refresh	1	Parallel	Jul 16, 2016, 11:00 PM
<input type="checkbox"/>	📅 Weekday early mornings	...	Weekly	Extract Refresh	1	Parallel	Jul 18, 2016, 4:00 AM
<input type="checkbox"/>	📅 Weekday mornings	...	Weekly	Subscription		Parallel	Jul 18, 2016, 6:00 AM

Tableau provides a few refresh schedules. You create additional schedules you need.

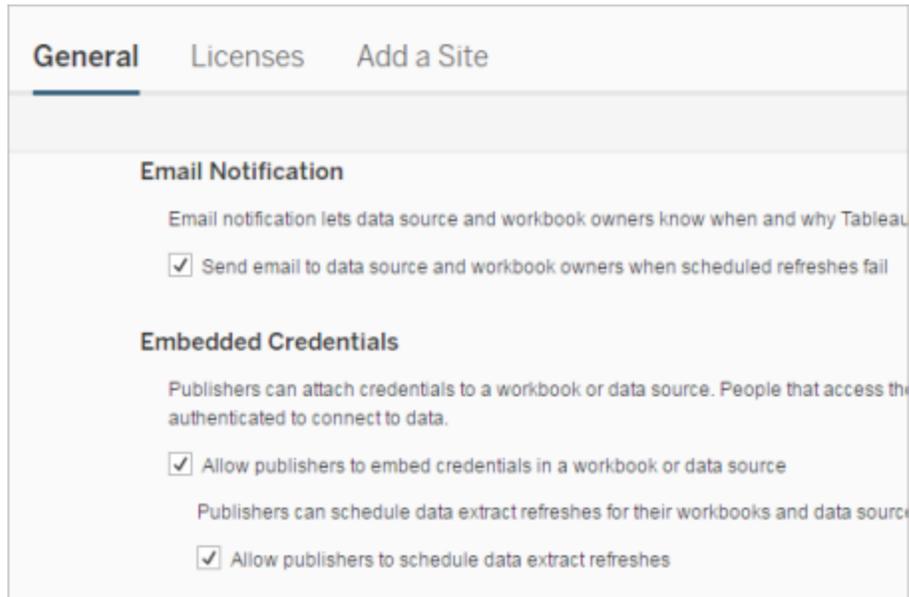
2 Enable scheduled extract refreshes and failure emails

As a server or site administrator, you can enable schedules, as well as email notification when extract refreshes fail.

Select **Settings**, and then go to the **General** page.

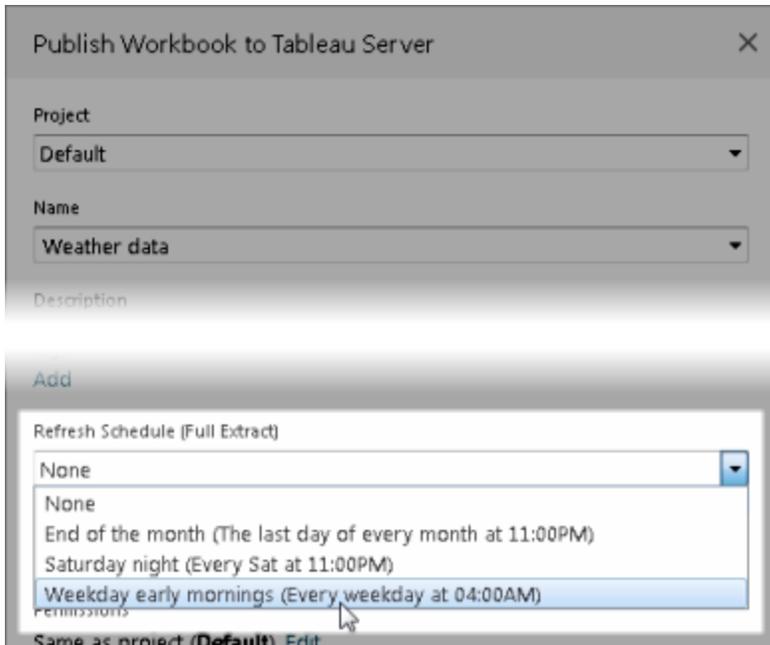
- Under Email Notification, select **Send email to data source and workbook owners when scheduled refreshes fail**.
- Under **Embedded Credentials**, select both check boxes to allow publishers to

embed credentials and schedule extract refreshes.



3 Publish a workbook with an extract

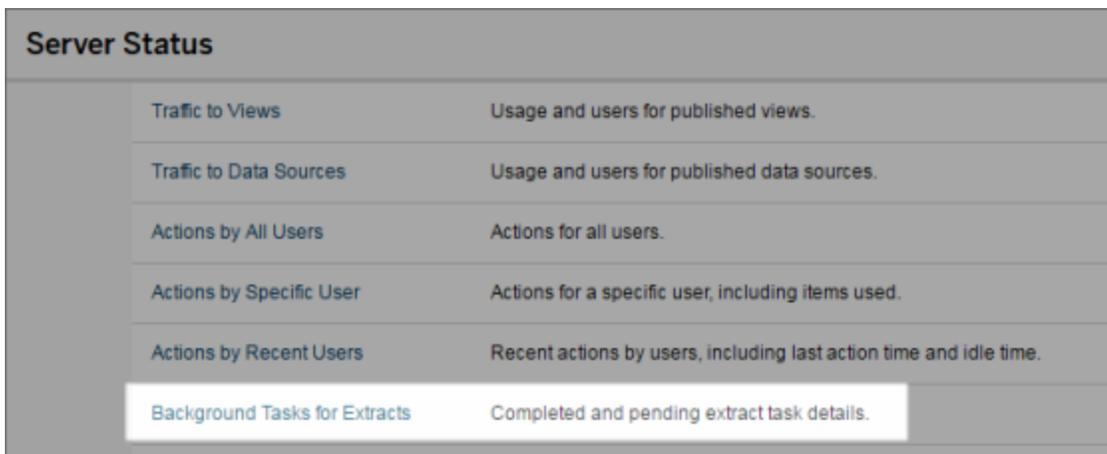
In Tableau Desktop, select **Server > Publish Workbook**. Sign in to the server if you're not already. In the **Publish Workbook to Tableau Server** dialog box, click **Schedules & Authentication**. Under **Extract Schedule**, select the schedule from the list.



If the original data requires authentication, you will also need to select how you want people to access it.

4 Monitor refresh performance

You can monitor scheduled tasks by viewing **Background Tasks for Extracts** on the **Status** page.



Automate Refresh Tasks

You can associate extract refresh tasks with schedules in Tableau Server to automate refreshing extracts. You can also automate extract refreshes using `tabcmd`, a command line utility that you can download for use with Tableau Server. In particular, you can use the `refreshextracts` command in combination with other commands in your own script.

For example:

```
tabcmd login - http://mytabserver -u jsmith -p P@ssw0rd!  
refreshextracts --datasource salesq4
```

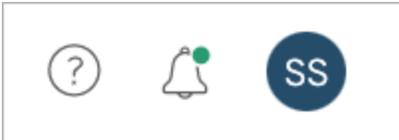
Note: When an extract refresh is performed on extracts created in Tableau 10.4 and earlier (that is, a `.tde` extract), the extract is upgraded to `.hyper` extract automatically. While there are many benefits of upgrading to a `.hyper` extract, your users won't be able open the extract with earlier versions of Tableau Desktop. For more information, see [Extract Upgrade to .hyper Format](#).

For information about downloading the `tabcmd` utility, see [tabcmd](#).

Handle Extract Refresh Alerts

When Tableau Server cannot complete a scheduled refresh, an alert appears to indicate that the refresh has failed. If a scheduled refresh fails five consecutive times, Tableau Server suspends the refresh. When a refresh is suspended, Tableau Server does not try to run it again until someone takes an action that attempts to correct the cause of the failure.

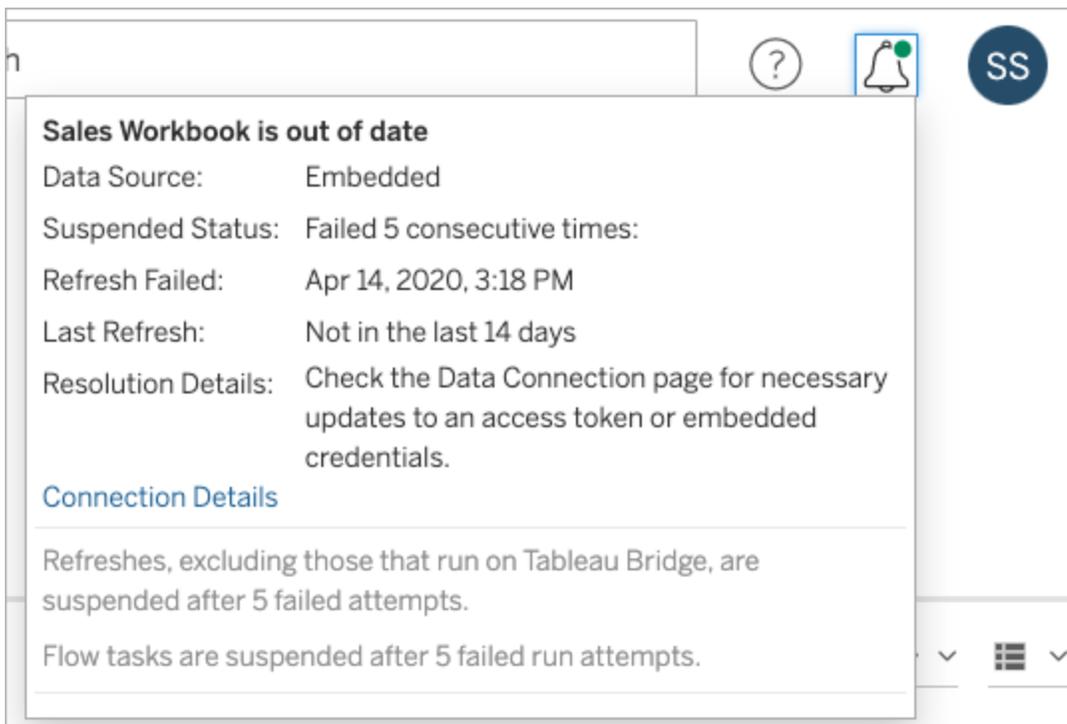
Note: The number of consecutive failures for a refresh is set to five by default, but can be changed by a Tableau Server administrator, using the `back-grounder.failure_threshold_for_run_prevention` option. For more information, see [tsm configuration set Options](#).



You will see the Alerts menu only if an extract refresh failed and you are:

- A system or site administrator
- The author of the workbook or data source that couldn't be refreshed
- The author of a workbook that connects to a data source that couldn't be refreshed

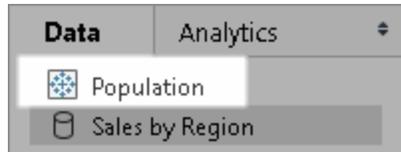
When you open the Alerts menu you can see more information about the refresh failure(s):



When a **Data source** is listed as **Embedded** it means that the data source definition (which includes things like the data source credentials or the database name) is embedded, or resides, within the workbook itself, originally created in Tableau Desktop.

When a data source name or workbook name is listed as the **Data source** (for example, **Data source: sales_data**), it means that the data source is a Tableau Server data source. The data source definition resides on Tableau Server.

In the Data pane on Tableau Desktop, you can determine whether the data source is on Tableau Server or is local. If the data source is on the server, a Tableau icon is displayed next to the data source name instead of a database icon :



Resolving Extract Refresh Problems

To resolve refresh issues, you can take any of these actions, based on the cause indicated in the alert:

- **Errors related to access token validation or user credentials**

You can resolve some extract refresh problems by clicking the **Connection Details** in the alert. Select the check box next to the problematic data source, click **Actions > Edit Connection**, and then enter the missing information. Click **Save** when you're done. After you update the connection information, Tableau Server restarts the refresh schedule.

If you originally embedded the credentials or other data connection information when you published the workbook or data source from Tableau Desktop, you can also republish the workbook or data source. As part of the publishing process, you can choose to set a new refresh schedule. If you don't choose a new schedule, Tableau Server restarts the existing schedule.

- **Errors that indicate the database was unreachable**

Confirm that the database is online and that you can sign in to access the data. You can use the **Try again** link in the alert to restart the refresh schedule.

- **Errors when using user filters or impersonation**

See the [Tableau Knowledge Base](#).

If the problem cannot be corrected by editing the data connection, you will need to resolve it in Tableau Desktop and republish the workbook.

Tip: Administrators can edit data connections at any time on the **Data Connections** page, accessible from each site by clicking the **Content** tab and Data Connections

Automatically Suspend Extract Refreshes for Inactive Workbooks

To save resources, Tableau can automatically suspend extract refresh tasks for inactive workbooks. This feature applies to full extract refreshes only, not to incremental extract refreshes. This feature only applies to refresh tasks that run more frequently than weekly, not to weekly or less frequent refresh tasks. This feature only applies to extracts that are embedded in workbooks and not to published data sources.

If any of these events occur, the workbook inactivity countdown timer is reset.

- Viewing the workbook sheets
- Setting up a data-driven alert or subscription on the workbook
- Downloading the workbook
- Moving the workbook's location or changing the owner

Configure the feature

1. Sign in to Tableau Server as a server administrator.
2. Go to the General tab of the Settings page for the site:
 - If you have a single site, at the top the browser window, click **Settings** and **General**.

- If you have multiple sites, select the site you want to configure and click **Settings** and **General**.
3. On the **General** page, do the following:
- Under **Automatically Suspend Extract Refresh Tasks**, select the **Automatically suspend extract refresh tasks for inactive workbooks** check box.
 - Specify the number of days, from 7 through 100, that a workbook should be inactive before extract refresh tasks are suspended. The default is 32 days.
 - Click **Save**.

Note: This feature is only configurable on Tableau Server. It is not configurable on Tableau Online.

Notifications

An email notification is sent three days before the extract refresh schedule is suspended.

Another email notification is sent when the extract refresh schedule is suspended.

Resume suspended extract refreshes

Suspended extract refreshes won't resume automatically if someone uses the workbook. It must be done manually.

To view and resume extract refreshes that were suspended:

1. Sign into the site and click **Tasks**.
2. Click the **Extract Refreshes** tab.
3. Select one or more items.
4. From the **Actions** menu, select **Resume**.

Edit Connections on Tableau Server

Administrators and data source owners can manage the information that describes how a published data source connects to the underlying data. This information includes the data-

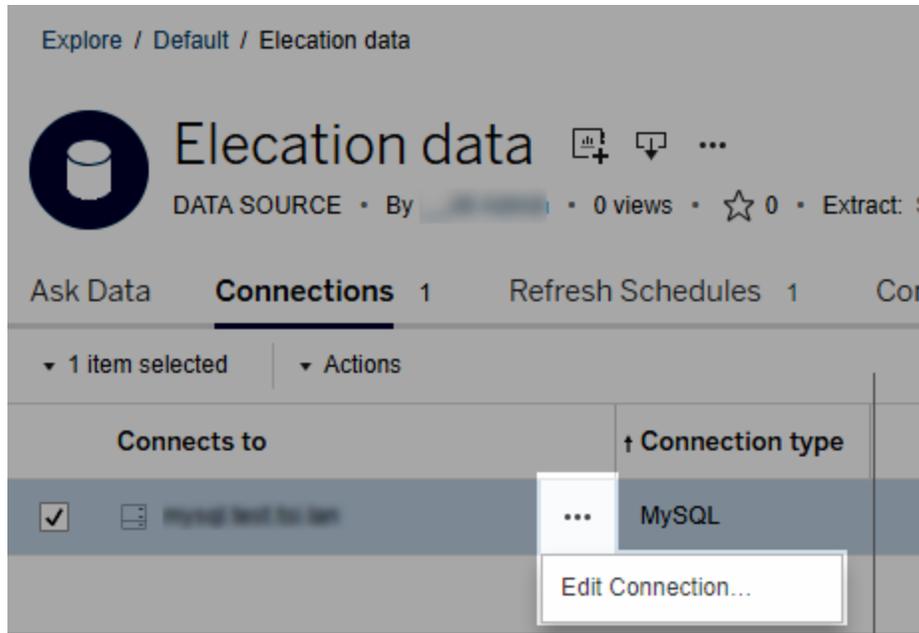
base server name or address, the server port, the database user name, and whether or not the password is embedded in the connection.

Note: Ability to edit connections is determined primarily by your site role, rather than by your permissions on the data source. To edit connections, your site role must be **Server Administrator**, **Site Administrator**, or **Creator**. If your site role is **Creator**, you also must be the data source owner.

1. Sign in to the site that has the data sources you want to modify, and on the **Content** tab, select **Explore > Data sources**.
2. Select the name of the data source with the connection you want to update.

Display filters to search for the data source or narrow the scope of the data source list. The values you type into the **Server name** and **Database username** fields are treated as regular expressions.

3. In the **Connections** view, select the **Actions (...)** menu for the data source, and then select **Edit Connection**.

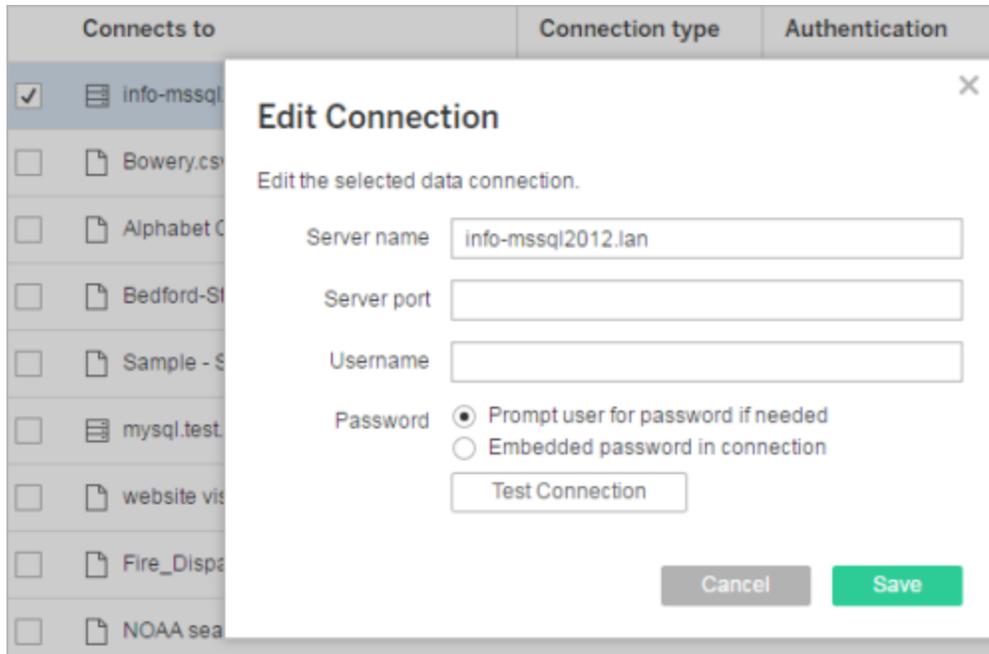


To edit multiple data sources, select the data sources you want to edit, then click the **Actions** menu and **Edit Connection**.

4. Update the connection information.

For **Server name**, if you want to use an IP address, make sure the database or its driver supports that type of connection. If it doesn't, enter the database server name.

See also Authentication types for Google, Salesforce, and WDC data later in this topic.



5. Click **Save**.
6. Refresh the browser page for your changes to take effect.

Authentication types for Google, Salesforce, and WDC data

Google BigQuery, Google Analytics, Salesforce.com, and many web data connector (WDC) connections use the OAuth authentication standard, which uses secure access tokens instead of “raw” user name and password credentials. With OAuth, database credentials do not need to be stored in Tableau, and all users connect through this access token, including Tableau Desktop users who want to create or edit workbooks that connect to the data source.

The following sections describe Google and Salesforce connection options. Web data connector options vary, but all involve signing in through the provider’s web-based sign-in form to establish the access token.

Google authentication options

When you edit Google BigQuery or Google Analytics connections, select either of the following options in the **Edit Connection** dialog box:

- Select **Embed Google BigQuery (or Google Analytics) credentials in the connection** to authenticate through a designated account, and then select an existing account from the list or select **authenticate account now...** to add a new one.

When you add a new account, the Google sign-in page appears. After you provide your database credentials, Google prompts you to confirm Tableau access to the data. When you click **Accept**, Google returns an access token to use for connecting to the data.

Note: If you create extracts of your Google data source, select this first option, so that you can schedule refresh tasks.

- Select **Prompt user for Google BigQuery/Analytics credentials** to require users to connect through their own individual access tokens or sign in each time they connect.

Salesforce.com authentication options

Note: This applies only if Tableau Server is configured to use saved credentials for Salesforce with OAuth. If the server is not configured for this, use the standard process above for modifying connections. For more information about configuring Tableau Server to use saved credentials with OAuth, see [Change Salesforce.com OAuth to Saved Credentials](#).

When you edit Salesforce.com connections, you can select any of the following options in the Edit Connection dialog box:

- Select **Embedded Salesforce username and password in the connection** to use a traditional authentication method.
- Select **Embedded Salesforce credentials in the connection** to use an OAuth connection and schedule refresh tasks, and then select an existing account from the list or click **Add a Salesforce Account** to add a new one.

When you add a new account, the Salesforce.com sign-in page appears. After you provide your database credentials, Salesforce.com prompts you to confirm Tableau access to the data. When you allow Tableau access, Salesforce.com creates an access token through which it connects to the data.

Edit Connection

Edit the selected data connection.

Authentication

Embedded Salesforce username and password in the connection

Username

Password

Embedded Salesforce credentials in the connection

No Salesforce authentication

Use this option if you do not need to schedule data extract refreshes

Test Connection

Cancel

Save

- Select **No Salesforce authentication** to require users to sign in to Salesforce.com each time they connect. (This option does not allow scheduled extract refreshes.)

Monitor progress

When you save your changes in the **Edit Connection** dialog box, the dialog displays the progress. If you close the dialog box, the modifications continue to run in the background until completed. Tableau Server will make as many changes as possible. Any failures will be skipped, but they will not impede other changes. For example, if you try to change the server name and add a password to several connections, the server names will be

changed, and the passwords on workbooks will be changed. However, because you cannot add a password to a data source, the passwords for the data sources will not be changed.

For information about checking the progress of these tasks, see [Background Tasks for Extracts](#).

Cube Data Sources

Cube (multidimensional) data sources have certain characteristics that make them unique in Tableau.

Cube data sources do not support pass-through connections. This means that when a cube data source is published, you cannot make a connection from Tableau Server using the data source. It also means you cannot create a workbook using the data source in Tableau Server.

Publishing a cube data source to Tableau Server gives you the ability to store the data source on the server. However, to use the data source, you must download the data source to Tableau Desktop and use it locally. To download a published data source you need:

- The **Download/Save As** permission for the data source. For more information, see [Permissions](#).
- Correct drivers installed and ports opened on computer running Tableau Desktop.

For information about using cube data sources with Tableau Desktop, see [Cube Data Sources](#).

Web Data Connectors in Tableau Server

Web data connectors (WDCs) are web pages that provide a data connection that is accessible over HTTP for data sources that don't already have a connector in Tableau. WDCs allow users to connect to almost any data that is accessible over the web and to create extracts for their workbooks. Data sources for a WDC can include internal web services, JSON data, REST APIs, and other sources that are available over HTTP or HTTPS. Users can create

their own WDC or use connectors that were created by others.

For information about how to use a WDC in Tableau Desktop, see [Web Data Connector](#) in the Tableau Desktop documentation.

For information about how to create a WDC, see the [Web Data Connector documentation](#) on Github.

Before you run connectors on Tableau Server

As a security measure, Tableau Server won't run WDCs unless you approve the connector, as explained in this topic.

Note: You must be a server administrator to approve WDCs for use on Tableau Server.

WDCs require your approval because they contain executable code and typically make requests to third-party websites. Before a user can use a WDC with Tableau Server, you must add the domain and port used by the connector to a safe list and also include the domains that a connector can send requests to and receive requests from on a secondary safe list. Before you do this, we recommend that you vet and test the connector so that you know what the connector does and what sites it connects to. For more information, see [Testing and Vetting Web Data Connectors](#).

When you add a connector to the safe lists, you configure Tableau Server to allow connections to a particular URL where the connector is hosted and from a URL which the connector can query. This is the only way to allow Tableau Server to run WDCs. The connectors can then be hosted on a server inside your organization's firewall or on an external domain. Importing WDCs is not supported for Tableau Server.

Manage connectors in a safe list

To add a WDC to the safe list, use the `tsm data-access web-data-connectors add` command. This command and the related commands described below let you perform the following tasks:

- Add WDCs to the safe list and secondary safe list.
- Allow or disallow all WDCs, or WDC refreshes.
- Remove one or more connectors from the safe list.
- List all WDCs on the safe list and secondary safe list.

Updating WDC safe lists requires a server restart

After running any commands that make changes to WDCs, you need to apply your pending changes using the `tsm pending-changes apply` command.

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

Add connectors to the safe list and secondary safe list

To add a connector to the safe list and secondary safe list, use the `tsm data-access web-data-connectors add` command, providing the name, safe list URL, and secondary safe list URLs. You can also manage WDCs using the `web-data-connector-settings` Entity. After running this command, you need to apply your pending changes using the `tsm pending-changes apply` command. A prompt warns you that the command will restart Tableau Server. If Tableau Server is currently running it is restarted, and if it is stopped it is left in a stopped state with no restart.

```

tsm data-access web-data-connectors add --name "USGS Earth-
quakes" --url https://tableau-
ableau-
.git-
hub.io:443/webdataconnector/Examples/html/earthquakeUSGS.html -
-secondary https://tableau-
ableau-
.git-
hub.io/.*,https://earth-
quake.usgs.gov-
/.*,https://max-
cdn-
boot-
strap-
cdn-
com/.*,https://ajax.googleapis.com/.*,https://connectors.tableau.com/.*

```

Notes on formatting:

- Be sure to use straight quotes (" and '), not curly or "smart" quotes, around the name of the WDC. Use double quotes (") if the name contains a space.
- For many WDCs, the port specified for the `--url` option is 443 or 80, but you can check the value for your connector by looking at the data source details on Tableau Server. You must specify the port number as part of the URL if the WDC is using SSL (HTTPS). For example, to use the default port for HTTPS, the URL might look like the following: `https://example.com:443/WDC/`.
- If a WDCs send requests to and receive data from multiple domains, use a comma-delimited list of URL for the `--secondary` option. Note that if you do not add a secondary list, all requests are allowed. For increased security, be sure to include a

Tableau Server on Linux Administrator Guide

secondary list that specifies the domains that provide the libraries and sources the WDC needs access to.

- To add an entire domain to the secondary safe list, end the domain URL with a wildcard expression. Use `.*` as the wildcard to indicate the entire domain, as in the following example: `https://example.com/.*`.

Allow or disallow WDCs or WDC extract refreshes

To allow or disallow WDCs, or WDC extract refreshes, use the `tsm data-access web-data-connectors allow` command with the `-t` or `-r` options. By default, WDCs and WDC extract refreshes are allowed. If you change this setting, you need to run the `tsm pending-changes apply` command. For more information about the command and the command options, see [tsm data-access web-data-connectors allow](#).

Remove one or more WDCs from the safe list

To remove one or more WDCs from the safe list, use the `tsm data-access web-data-connectors delete` command. If you change this setting, you need to run the `tsm pending-changes apply` command. For more information, see [tsm data-access web-data-connectors delete](#).

List all WDCs on the safe list

To list all WDCs on the safe list, use the `tsm data-access web-data-connectors list` command. For more information, see: [tsm data-access web-data-connectors list](#).

Refresh the extract for a connector

When a user creates a workbook that uses a WDC, Tableau Server creates an extract from the data returned by the connector. If the user then publishes the workbook, the publish process sends the workbook and the data extract to the server.

Tableau can refresh an extract that was created by a WDC, the same as it can refresh any extract. If the connector requires credentials to sign in to the web-based data source, you need to ensure that the credentials are embedded with the data source, and that the WDC is

on the safe list for the server. Tableau Server cannot refresh the extract if the connector requires credentials and they are not embedded with the data source. This is because the refresh can occur on a schedule or in some other background context, and the server cannot prompt for credentials.

Currently, there is no way to re-authenticate a data source from Tableau Server directly. If the data source has credentials that expire, or was published without embedding the credentials, the workbook and data extract need to be published again with the new embedded credentials.

If the background process that performs the refresh operation fails, it creates an alert and a log entry that indicates this issue. Users will be able to see that the timestamp on the extract does not change.

To disable refresh for all WDCs, use the `tsm data-access web-data-connectors allow -r false` command.

Troubleshooting

If the server experiences problems with adding connectors to the safe list, you can examine the log files. Be sure to check the log files on both the initial server node and on the other nodes that are running the gateway process. For more information about log files, Tableau Server Logs and Log File Locations.

If the issue is that Tableau Server will not refresh an extract that was created by a WDC, make sure that the `webdataconnector.refresh.enabled` configuration setting has been set to `true`. If it is set to `false`, run the following command to allow extract refreshes for all WDCs on the server:

```
tsm data-access web-data-connectors allow -r true
```

Note: The safe list is the only way of allowing Tableau Server to run web data connectors. Importing web data connectors was deprecated starting with version 10.5.

Testing and Vetting Web Data Connectors

Web Data Connectors (WDCs) contain JavaScript that typically connects to data on another site. Because of this, you should test and vet web data connectors before users use them as data sources for a workbook, and before you use them with Tableau Server.

This topic includes some suggestions for testing and vetting web data connectors.

Examine the source

The code in a web data connector is in JavaScript, so you can open the file (and any external files that the connector uses) and examine the source code.

Many connectors reference external JavaScript libraries, such as the jQuery library or API libraries for third parties. Validate that the URL for external libraries points to a trusted location for the library. For example, if the connector references the jQuery library, make sure that the library is on a site that is considered standard and safe. If it is practical for you to change the source code of the connector, use HTTPS protocol (`https://`) to reference external libraries (if the source site supports HTTPS) to help verify the site's authenticity.

To the extent possible, make sure you understand what the code is doing. In particular, try to understand how the code is constructing requests to external sites, and what information is being sent in the request.

Note: Experienced JavaScript programmers often compress (minify) their code to reduce the size of the code for download. Dense blocks of code that use cryptic function and variable names are not uncommon. While this can make it more difficult to examine the code, it is not a sign that the code was written to be deliberately difficult to understand.

Test the web data connector in an isolated environment

If possible, test the web data connector in an environment that is isolated from your production environment and from user computers. For example, add a web data connector to a

safe list on a test computer or virtual machine that's running a version of Tableau Server that is not used for production.

Monitor the traffic created by the web data connector

When you test a web data connector, use a tool like [Fiddler](#), [Charles HTTP proxy](#), or [Wireshark](#) to examine the requests and responses that the connector makes. Make sure that you understand what sites the connector makes requests to and what content the connector is requesting. Similarly, examine the responses and their content to be sure that the connector is not reading data or code that is not directly related to the connector's purpose.

Test the performance and resource usage of the web data connector

When you test a web data connector, use tools to monitor its CPU and memory usage. Remember that the web data connector will run on Tableau Server, which is an environment in which many processes are already running. You want to make sure that when the connector fetches data, the connector does not have an undue impact on server performance.

Check whether the connector writes to disk. If it does, check how much disk space it occupies, and examine the output to make sure you understand what it's writing and why.

Enable Tableau Catalog

[Tableau Catalog](#) discovers and indexes all of the content on your Tableau Online site or Tableau Server, including workbooks, data sources, sheets, metrics, and flows. Indexing is used to gather information about the content, or metadata, about the schema and lineage of the content. Then from the metadata, Catalog identifies all of the databases, files, and tables used by the content on your Tableau Online site or Tableau Server.

Catalog is available with the Data Management. For more information, see [About Data Management Add-on](#).

In addition to Catalog, metadata about your content can also be accessed from both the [Tableau Metadata API](#) and the Tableau REST API using [Metadata Methods](#).

Before enabling Catalog

As a Tableau Server admin, there are a few things that you need to consider before and while enabling Catalog to ensure optimal performance of Catalog in your Tableau Server environment.

Required versions

Before enabling Catalog, make sure you're running *one* of the following versions of Tableau Server:

- At least Tableau Server 2019.3.4 or later
- At least Tableau Server 2019.4.2 or later
- At least Tableau Server 2020.1.0 or later
- At least Tableau Server 2020.2.15 or later
- Tableau Server 2020.3 and later

For more information about why these versions are required, see the [Tableau Knowledge Base](#).

What to expect when enabling Catalog

When Catalog is enabled, the content that already exists on your Tableau Server is immediately indexed.

Initial ingestion

The indexing process is comprised of two primary components, one of which is called initial ingestion. Initial ingestion can be broken down into two additional components:

- Content backfill
- Lineage backfill

The status of content backfill and lineage backfill are important to note later on when monitoring progress and validating that Catalog has been successfully turned on and is running in your Tableau Server environment.

Initial ingestion speed

The time it takes Catalog to index the content for the first time depends on a couple of factors:

- **Amount of content on Tableau Server:** The amount of content is measured by the total number of workbooks, metrics, published data sources, and flows published to Tableau Server. For more information, see [Disk space to store metadata](#).
- **Number of non-interactive microservices containers:** Catalog uses the non-interactive microservices container to index all the content on Tableau Server. For more information, see [Memory for non-interactive microservices containers](#).

Understanding the factors that impact initial ingestion can help you gauge how long it might take to enable and run Catalog in your environment.

Disk space to store metadata

During initial ingestion, metadata is generated and stored in the Tableau Server repository ("relationship" PostgreSQL database). The amount of disk space needed to store the metadata is roughly up to half of the disk space currently used by the repository ("work-group" PostgreSQL database).

For example, suppose the repository uses 50 GB of disk space prior to enabling Catalog, the repository can use up to 75 GB of disk space after enabling Catalog.

Memory for non-interactive microservices containers

Initial ingestion runs inside of the non-interactive microservices container. The non-interactive microservices container is one of two [Tableau Server microservices containers](#) processes. By default, one instance of the non-interactive microservices container is added to every node that has a backgrounder process installed.

By default, initial ingestion on a single instance of the non-interactive microservices container can use up to 4 GB of memory on the backgrounder node. If the amount of content on Tableau Server exceeds 10,000, a non-interactive microservices container may require up

to 16 GB of memory on the backgrounder node. Therefore, when enabling Catalog, ensure that every backgrounder node has the available capacity to support each non-interactive microservices container during the initial ingestion process. If capacity needs to be increased, you must update the JVM heap size for non-interactive containers to allocate up to 16 GB of memory on the backgrounder nodes. For more information, see `non-interactivecontainer.vmopts`.

If you are planning to add more non-interactive microservices containers to decrease the time of initial ingestion, first determine how many total containers are needed (using [Step 2: Estimate how long initial ingestion will take](#), below) and then verify if your Tableau Server environment is configured with enough capacity to support *all* non-interactive microservices containers. Depending on how your Tableau Server environment is already configured, you might not be able to add all the additional non-microservices containers that you need to decrease initial ingestion time.

Best practices for enabling Catalog

Because the speed of initial ingestion and requirements are unique to each Tableau Server environment, Tableau recommends that when you enable Catalog you do one or more of the following:

- Make sure there is enough disk space that the Tableau Server repository can use to support the additional metadata that initial ingestion will generate and store. As a general rule, the repository will need an additional 50% of disk space currently used by the repository. For more information about Tableau Server disk usage, see [Server Disk Space](#).
- Depending on the amount of content on Tableau Server, make sure each backgrounder node has at least 4-16 GB of available memory for each instance of a non-interactive microservices container during initial ingestion.
- Perform the process over the weekend to allow initial ingestion to complete before your users start using Catalog capabilities.

- Perform the process in a test environment with production content first. This is because the type of content that needs to be ingested can play a significant role on ingestion speed.

Summary of steps to enable Catalog

The following steps summarize the process to turn on and run Catalog on Tableau Server. The steps must be performed sequentially.

1. Determine the amount of content on Tableau Server
2. Estimate how long initial ingestion will take
3. Decrease the time of initial ingestion
4. Activate the Data Management Add-on
5. Turn off Catalog capabilities
6. Run the `tsm maintenance metadata-services` command
7. Monitor initial ingestion progress and validate its status
8. Configure SMTP
9. Turn on Catalog capabilities

Note: Because indexing metadata about Tableau content on Tableau Server is powered by the Metadata API, enabling the Metadata API is required to run and use Catalog.

Enable Catalog

Step 1: Determine the amount of content on Tableau Server

To determine the amount of content on Tableau Server, do the following:

1. Sign in to Tableau Server using your admin credentials.
2. Go to the **Explore** page.
3. Click the Top-Level Project drop-down menu and add the numbers next to **All Workbooks**, **All Metrics**, **All Data Sources**, and **All Flows** together. This is the total amount of content on Tableau Server.

Tableau Server on Linux Administrator Guide

Step 2: Estimate how long initial ingestion will take

To estimate the time it will take Catalog to ingest content on Tableau Server for the first time (initial ingestion), compare *your* Tableau Server setup to a *baseline* Tableau Server setup.

For a Tableau Server with the following setup, initial ingestion could take about 6 hours to complete.

Components	Baseline values
Content	17,000 workbooks, metrics, published data sources, and flows
Non-interactive microservices containers	10
Ingestion	~6 hours

If you have roughly *half* the content in your Tableau Server environment, initial ingestion might take half the time to complete.

For example: 8,500 (workbooks, metrics, published data sources, and flows) + 10 non-interactive microservices containers = ~3 hours (initial ingestion)

If you have roughly *double* the content in your Tableau Server environment, initial ingestion might take double the time to complete.

For example: 34,000 (workbooks, metrics, published data sources, and flows) + 10 non-interactive microservices containers = ~12 hours (initial ingestion)

Step 3: Decrease the time of initial ingestion

As a general rule, the time it takes for Catalog to perform initial ingestion is correlated to the number of non-interactive microservices containers. To help decrease the time of initial ingestion, you can increase the number of non-interactive microservices containers.

Increase the number of non-interactive microservices containers

By default, one non-interactive microservices container is added to every node that has a backgrounder. To help decrease the time of initial ingestion, Tableau recommends that you increase the number of non-interactive microservices container using the `tsm topology set-process` command.

1. Open a command prompt as an admin on the initial node (where TSM is installed) in the cluster.
2. Run the command: `tsm topology set-process --count <process_count> --node <node_ID> --process <process_name>`

For example, to increase the non-interactive microservices container on the initial node to 4 containers, run the following command:

```
tsm topology set-process --count 4 --node node1 --process noninteractive
```

For more information about running the command and its global options, see [tsm topology](#).

Important: Before increasing the number of non-interactive microservices containers, review the following:

- The recommendation for increasing non-interactive microservices containers is for total number of non-interactive microservices containers, not total non-interactive microservices containers per node. For example, suppose you have 4 nodes but you want to increase the number of non-interactive containers to 8. The `--count` value you use in the `tsm` command is 2.
- For each non-interactive microservices container added, 4 GB of additional memory will be used on the node and load will be added to the Tableau Server repository (PostgreSQL database).

Tableau Server on Linux Administrator Guide

- Tableau recommends that you incrementally increase non-interactive microservices containers by no more than 2 at a time while closely monitoring your Tableau Server environment to avoid issues with CPU utilization of the Tableau Server repository (PostgreSQL database).
- Be aware that when too many non-interactive microservices are added, CPU utilization of the PostgreSQL database might spike and failover. Symptoms to watch for include SQLException errors in the vizportal logs. For more information, see [Repository Failover](#) topic.

Step 4: Activate the Data Management

(Requires the Data Management)

If not already done, you can activate the Data Management. For more information, see [License the Data Management Add-on](#).

Step 5 (optional): Turn off Catalog capabilities for each site

(Requires the Data Management)

As part of the Data Management activation, Catalog capabilities are turned on by default. Because of the indexing process and the estimated time it takes to complete, consider temporarily turning off Catalog capabilities for each site so that Tableau Server users can't access Catalog capabilities until Catalog is ready and able to provide complete and accurate results.

1. Sign in to Tableau Server using your admin credentials.
2. From the left navigation pane, click **Settings**.
3. On the General tab, under Tableau Catalog, clear the **Turn on Tableau Catalog** check box.
4. Repeat steps 2-3 for each site on your Tableau Server.

Step 6: Run the `tsm maintenance metadata-services` command

Run the `tsm maintenance metadata-services` command to enable the Tableau Metadata API. Running the command begins initial ingestion. If your Tableau Server is licensed with the Data Management Add-on, running the command also turns on Catalog capabilities (if it wasn't turned off earlier).

1. Open a command prompt as an admin on the initial node (where TSM is installed) in the cluster.
2. Run the command: `tsm maintenance metadata-services enable`

For more information about running the `tsm` command, see [tsm maintenance](#).

Notes: When running this command, keep the following points in mind:

- This command stops and starts some services used by Tableau Server, which causes certain functionality, such as the Recommendations capability, to be temporarily unavailable.
- A new index of metadata is created at this time. Running this command any subsequent times will create and replace the previous index.

Step 7: Monitor initial ingestion progress and validate its status

Running the `tsm` command above starts the initial ingestion process. To ensure that the initial ingestion process is going smoothly, you can monitor its progress using the Backfill API. For more information, see [Get Initial Ingestion Status](#).

Step 8: Configure SMTP Setup

If not already set up for Tableau Server, configure SMTP Setup. SMTP supports sending email to owners who need to be contacted about changes to data. For more information about configuring SMTP, see [Configure SMTP Setup](#).

Step 9 (optional): Turn on Catalog capabilities for each site

(Requires the Data Management)

If you turned off Catalog capabilities before enabling Catalog in one of the procedures above, you must turn on Catalog to make its capabilities accessible to your users.

1. Sign in to Tableau Server using your admin credentials.
2. From the left navigation pane, click **Settings**.
3. On the General tab, under Tableau Catalog, select the **Turn on Tableau Catalog** check box.
4. Repeat steps 2-3 for each site on your Tableau Server.

Troubleshoot Catalog

You or your users might encounter one of the following issues when using Catalog.

Timeout limit and node limit exceeded messages

To ensure that Catalog tasks that have to return a large number of results don't take up all Tableau Server system resources, Catalog implements both timeout and node limits.

- **Timeout limit**

When tasks in Catalog reach the timeout limit, you and your users see the following message:

"Showing partial results, Request time limit exceeded. Try again later." or TIME_LIMIT_EXCEEDED

To resolve this issue, as a Tableau Server admin, you can increase the timeout limit using the `tsm configuration set -k metadata.query.limits.time` command. For more information, see the [tsm configuration](#) and [tsm configuration set Options](#) topics.

Important: Increasing the timeout limit can utilize more CPU for longer, which can the performance of other processes on Tableau Server.

- **Node limit**

When tasks in Catalog reach the node limit, you and your users see the following message:

NODE_LIMIT_EXCEEDED

To resolve this issue, as a Tableau Server admin, you can increase the node limit using the `tsm configuration set -k metadata.query.limits.count` command. For more information, see the [tsm configuration](#) and [tsm configuration set Options](#) topics.

Important: Increasing the timeout limit can affect system memory.

Missing content

- If you suspect, after initial ingestion, content is missing from Catalog, you can use the Eventing API to help troubleshoot. Eventing handles indexing content on Tableau Server after initial ingestion. For more information, see [Get Eventing Status](#).
- When the connection between an embedded external asset and its downstream Tableau content is removed, it remains in Catalog (or the Tableau Metadata API) until it's automatically deleted by a background process that runs everyday at 22:00:00 UTC (coordinated universal time). For example, suppose a workbook, initially published with an embedded text file A, is republished with an embedded text file B. File A remains visible (or query-able) as an external asset until the background process is able to delete it.

You can disable this background process from running if you do not want to remove these types of external assets or if you believe that it takes up system resources that you don't want to dedicate to this process. Alternatively, you can adjust the number of external embedded assets that are deleted. For more information, see [features.DeleteOrphanedEmbeddedDatabaseAsset](#) and [databaseservice.max_database_deletes_per_run](#).

You can monitor this process in one of two ways:

- Filter on the **One-time job re-canonicalize existing database/table assets after a canonicalization logic change** task type in the [Background Tasks for Non Extracts](#) admin view.
- Refer to the **Finished removal of orphaned embedded databases or database_service_canonicalization_change** events in the [Tableau Server log files](#).

Performance after initial ingestion

In some Tableau Server environments where specific content that is updated very frequently (for example, through high-frequency schedules or command line or API requests), the indexing process might get over saturated. In these cases, as the server admin, you might consider enabling event throttling to better preserve Catalog performance. For more information, see `metadata.ingestor.pipeline.throttleEventsEnable`.

Note: When event throttling is enabled, users might notice an intended delay in content changes in Catalog.

Out of memory errors

In some cases, Tableau Server out of memory errors can occur as a result of problems with ingesting complex content. If you suspect ingestion is the cause of the out of memory errors on your Tableau Server, contact and work with Tableau Support to `metadata.ingestor.blocklist` from being ingested to help resolve the issue.

Disable Catalog

As a Tableau Server admin, you can disable Catalog in one of two ways.

Turn off Catalog capabilities for each site

(Requires the Data Management)

You can turn off Catalog capabilities at any time. When Catalog capabilities are turned off, the features of Catalog, such as adding data quality warnings or the ability to explicitly manage permissions to database and table assets, are not accessible through Tableau Server.

However, Catalog continues to index published content and the metadata is accessible from the Tableau Metadata API and metadata methods in the Tableau REST API.

1. Sign in to Tableau Server using your admin credentials.
2. From the left navigation pane, click **Settings**.
3. On the General tab, under Tableau Catalog, clear the **Turn on Tableau Catalog** check box.

Stop indexing metadata on Tableau Server

To stop indexing the published content on Tableau Server, you can disable the Tableau Metadata API. To disable the Metadata API, run the `tsm maintenance metadata-services disable` command. For more information, see [tsm maintenance](#).

Get Initial Ingestion Status

After enabling the Tableau Metadata API using the `tsm maintenance metadata-services` command, you can use the Backfill API to monitor progress of initial ingestion and get status information for content and lineage backfills.

The steps described below must be performed by a server admin and recommended to be used in conjunction with Step 7: Monitor initial ingestion progress and validate its status in the Enable Tableau Catalog topic.

Step 1: Authenticate using the REST API

To access the Backfill API, you must first authenticate against Tableau Server and get a token. You can do this using the Tableau REST API. For more information, see [Signing In and Signing Out \(Authentication\)](#) in the REST API Help.

Alternatively, you can sign in to Tableau Server using your admin credentials.

Step 2: Make a GET request

Make the following GET request or paste the URL into your browser:

Tableau Server on Linux Administrator Guide

`http://my_tableau_server/api/metadata/v1/control/backfill/status`

The request initially returns information about content backfill. When content backfill is complete, lineage backfill information is returned.

- For content backfill, the request returns a status summary and additional status information for each content type depending on what content is available on Tableau Server. Indexing for each content type happens concurrently.
- For lineage backfill, the request returns a status summary.

Status values from the response

The following values are returned by the Backfill API.

- **contentBackfillTotalDurationSeconds** and **lineageBackfillTotalDurationSeconds** is how much time has progressed, in seconds, on the respective backfill type. When **backfillComplete** is `true`, **contentBackfillTotalDurationSeconds** and **lineageBackfillTotalDurationSeconds** is the total time spent to complete the respective backfill type.

For content backfill:

- **contentType** can show the following content types: `PublishedDatasource`, `Database`, `DatabaseTable`, `Metric`, `Workbook`, and `Flow`.
- **contentId** is the identifier of the last indexed item.
- **successfullyIngestedCount** is the number of items successfully indexed.
- **failedIngestedCount** is the number of items that could not be indexed.
- **durationSeconds** is the time spent, in seconds, to index items for the content type.
- **totalCount** is the total number of items to index.
- **checkpointCreatedTime** is the last recorded time, in UTC, an item was indexed. The Backfill API checks for the last indexed item every five minutes.
- **backfillComplete** is `true` when indexing is complete for all items of the content type.

For lineage backfill:

- **totalCount** is the total number of lineage relationships to index.
- **processedCount** is the number of indexed lineage relationships.
- **lastLineageConnection** is the last indexed lineage relationship.
- **backfillComplete** is `true` when indexing is complete for all lineage relationships.

Example response

The request returns JSON text. To view the JSON in a more readable format, you can use a JSON viewer or browser add-on.

```
{
  "contentBackfillTotalDurationSeconds": 362,
  "lineageBackfillTotalDurationSeconds": 14,
  "contentBackfillStatuses": [
    {
      "contentType": "PublishedDatasource",
      "contentId": "sites/1/datasources/-631379806-1912815680",
      "successfullyIngestedCount": 20,
      "failedToIngestCount": 0,
      "durationSeconds": 312,
      "totalCount": 20,
      "checkpointCreatedTime": "2020-07-29T23:50:25.763Z",
      "backfillComplete": true
    },
    {
```

Tableau Server on Linux Administrator Guide

```
    "contentType": "Database",
    "contentId": "sites/1/databases/e1331f9d-4d73-ee04-9edf-
96fd1c37cb8e",
    "successfullyIngestedCount": 35,
    "failedToIngestCount": 0,
    "durationSeconds": 26,
    "totalCount": 35,
    "checkpointCreatedTime": "2020-04-29T23:50:25.769Z",
    "backfillComplete": true
  },
  {
    "contentType": "DatabaseTable",
    "contentId": "sites/1/tables/d946d084-53a8-09b6-2ad2-
93301e6b4b15",
    "successfullyIngestedCount": 64,
    "failedToIngestCount": 0,
    "durationSeconds": 49,
    "totalCount": 64,
    "checkpointCreatedTime": "2020-04-29T23:50:25.774Z",
    "backfillComplete": true
  },
```

```
{  
  "contentType": "Metric",  
  "contentId": "sites/1/metrics/metric1",  
  "successfullyIngestedCount": 2,  
  "failedToIngestCount": 0,  
  "durationSeconds": 254,  
  "totalCount": 2,  
  "checkpointCreatedTime": "2020-04-29T23:50:25.779Z",  
  "backfillComplete": true  
},  
  
{  
  "contentType": "Workbook",  
  "contentId": "sites/1/workbooks/6749399-1501801290",  
  "successfullyIngestedCount": 10,  
  "failedToIngestCount": 0,  
  "durationSeconds": 267,  
  "totalCount": 10,  
  "checkpointCreatedTime": "2020-04-29T23:50:25.784Z",  
  "backfillComplete": true  
},
```

Tableau Server on Linux Administrator Guide

```
{
  "contentType": "Flow",
  "contentId": "sites/1/flows/4",
  "successfullyIngestedCount": 4,
  "failedToIngestCount": 0,
  "durationSeconds": 195,
  "totalCount": 4,
  "checkpointCreatedTime": "2020-04-29T23:50:25.788Z",
  "backfillComplete": true
},
"lineageBackfillStatus": {
  "totalCount": 45,
  "processedCount": 18,
  "lastLineageConnection": "CloudFile downstreamWorkbooks Workbook",
  "backfillComplete": false
}
```

Get Eventing Status

After you have enabled Tableau Catalog (or the Tableau Metadata API) in your Tableau Server environment, you can use the Eventing API to gauge indexing performance.

The steps described below must be performed by a server admin.

Step 1: Authenticate using the REST API

To access the Eventing API, you must first authenticate against Tableau Server and get a token. You can do this using the Tableau REST API. For more information, see [Signing In and Signing Out \(Authentication\)](#) in the REST API Help.

Alternatively, you can sign in to Tableau Server using your admin credentials.

Step 2: Make a GET request

Make the following GET request or paste the URL into your browser:

```
http://my_tableau_server-  
/api/metadata/v1/control/eventing/status
```

Status values from the response

The following values are returned by the Eventing API.

- **contentType** is the content type that was most recently indexed.
- **queueSize** is the number of items in the indexing queue. The larger the queue size, the longer it can take for items to show in Catalog or Metadata API. If the queue size increases over time, you might need to adjust background capacity to support the non-interactive containers and indexing process. For more information, see [Memory for non-interactive microservices containers](#).
- **checkpointCreatedTime** is the last recorded time, in UTC, an item was indexed. The Eventing API checks for the last indexed item every five minutes.

Example response

The request returns JSON text. To view the JSON in a more readable format, you can use a JSON viewer or browser add-on.

```
{  
  
  "contentType": "PublishedDatasource",  
  
  "queueSize": 312,  
  
  "checkpointCreatedTime": "2020-07-29T23:50:25.763Z"  
  
}
```

Use Lineage for Impact Analysis

Knowing where your data comes from is key to trusting the data, and knowing who else uses it means you can analyze the impact of changes to data in your environment. The lineage feature in Tableau Catalog helps you do both these things.

Lineage requires the Data Management Add-on. Starting in 2019.3, Tableau Catalog is available in the Data Management Add-on to Tableau Online and Tableau Server. When Tableau Catalog is enabled in your environment, you have access to lineage for your data sources, metrics, flows, databases, and tables. For more information about Tableau Catalog, see "About Tableau Catalog" in the [Tableau Server](#) or [Tableau Online](#) Help.

Navigate lineage

How you navigate to the **Lineage** pane depends on what kind of asset you're working with.

To see the lineage for Tableau content such as data sources or flows, from **Explore**, navigate to and open the content asset, and then select the **Lineage** tab.

Sample - Superstore ☆ ⓘ ...
 Owner M. Kim Certified by A. Garcia

New ▾

Ask Data Connections 1 Connected Workbooks 69 **Lineage**

Fields (26)

Type	Name	Sheets	Description
Abc	Category	24	No description
Abc	City	6	No description
Abc	Country/Region	6	No description
Abc	Customer Name	14	No description
#	Discount	5	No description
👤	Location	0	No description
🔗	Manufacturer	0	No description
#	Migrated Data (Count)	0	No description
+#	Number of Records	4	No description

Lineage

- Databases 1
- Tables 1
- Sample - Superstore** Fields 26
- Workbooks 56
- Sheets 402
- Dashboards 16
- Owners 14

To see lineage for external assets, such as databases and tables, from **External Assets**, navigate to and select an asset from the list. When you select a table, for example, a page opens with information about that table, for example, the name, type, full name, description, columns, and details about each column. To the right of the table information is the Lineage pane, which shows the lineage for that table.

Tableau Server on Linux Administrator Guide

The screenshot displays the Tableau Server interface for a workbook named 'TestResult'. The interface includes a sidebar with navigation icons, a main content area, and a 'Lineage' panel on the right.

About

Database Name	Connection Type	Hostname	Full Name
TestV1	PostgreSQL	postgres.test.tsi.lan	[public].[xy]

Tags: results
Description: Table contains results from the latest test

Columns (3)

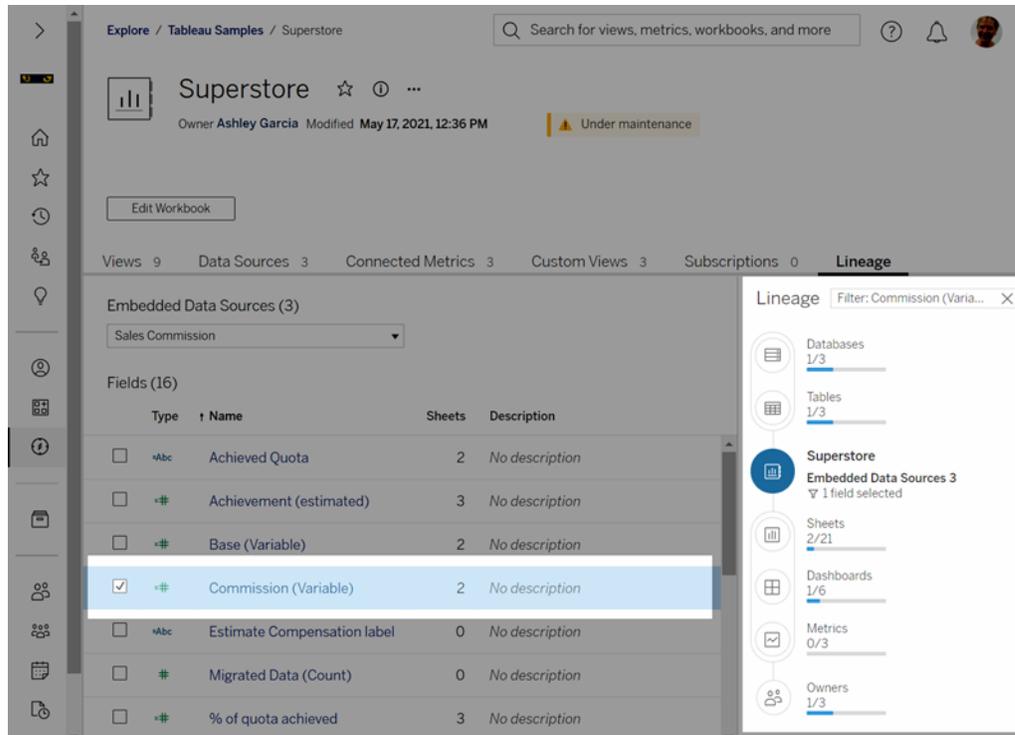
Type	Name	Sheets	Data sources	Description	Tags	
<input type="checkbox"/>	Abc	a	7	3	No description	tag, hello, PII (3 total)
<input type="checkbox"/>	#	x	6	3	No description	test (1 total)
<input type="checkbox"/>	#	y	5	3	No description	

Lineage

- Databases: 1
- TestResult: Columns 3
- Data Sources: 3
- Workbooks: 9
- Sheets: 10
- Owners: 3

Lineage shows dependencies in relationship to the lineage anchor, which is the asset selected. A lineage anchor can be a database, table, workbook, published data source, or a flow. (In the first example, the anchor is the Sample - Superstore data source and in the second example, it's the TestResult table). All the assets below the anchor depend, either directly or indirectly, on the anchor and are called outputs or downstream assets. The assets above the anchor are the assets the anchor is either directly or indirectly dependent on and are called inputs or upstream assets.

When you select a field in a data source or a column in a table, the lineage is filtered to show only downstream assets that depend on the field (or column) or upstream inputs to the field (or column) as in this 'Superstore' workbook example that shows the lineage filtered for the Commission (Variable) field:



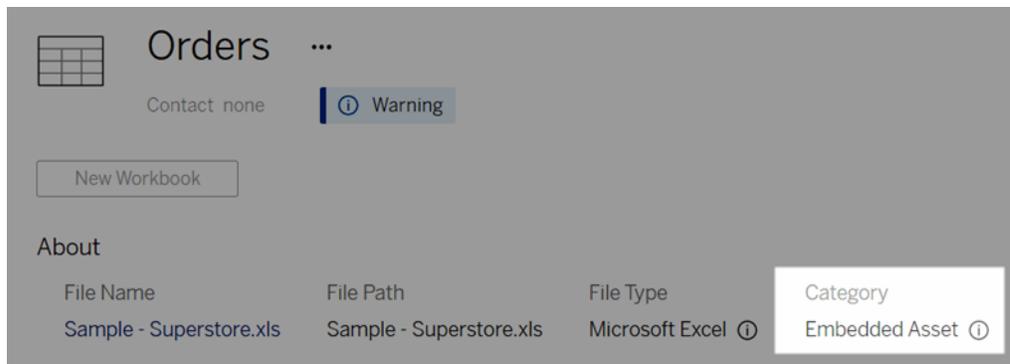
You can select an upstream or downstream asset in the Lineage pane to see its details. For example, when you select Metrics, the list of metrics depending on this workbook appears to the left of the Lineage pane.

From the Lineage pane, you can navigate to any asset related to your initial choice, in this case the workbook, by following the links that interest you.

Embedded asset appears in External Assets

Tableau Catalog treats an external asset as 'embedded' if the **Include external files** check box is selected when a data source or workbook is published. When an external asset (database, table, or file) is embedded in published Tableau content (workbooks, data sources, and flows), the external asset is used by the content, but is not shareable with other users. That embedded external asset appears in the lineage upstream from its Tableau content and is listed in External Assets.

To see if an external asset is embedded, go to the external asset's detail page and see if "Embedded Asset" is listed under **Category**.



For information about embedded data, see [Publishing data separately or embedded in workbooks](#) in Tableau Desktop and Web Authoring Help.

Lineage and custom SQL connections

When you view the lineage of a connection that uses custom SQL, Catalog doesn't support showing column information. For more information, see [Tableau Catalog support for custom SQL](#) in the Tableau Desktop and Web Authoring Help.

Catalog doesn't support cubes

Cube data sources (also known as multidimensional or OLAP data sources) are not supported by Tableau Catalog. Tableau content (such as a data source, view, or workbook) that relies on cube data does not display any cube metadata or cube lineage in Catalog.

Mismatch between lineage count and tab count

You might notice a mismatch in the count of assets between the Tableau Catalog Lineage tool and the tabs in Tableau Server or Tableau Online.

The count mismatch is explained by the fact that each—lineage count vs. tab count—counts assets a different way. For example, at any given point in time, Catalog can count only assets that are indexed, whereas Tableau Server or Tableau Online counts any assets that are published. Other reasons for count differences include whether:

- You have "View" permissions for the asset.
- An asset is hidden.

- Any fields are used in a workbook.
- An asset is directly or indirectly connected to.

Workbook count mismatch example

As an example, here's how the tab count vs. the lineage count is determined for workbooks.

The screenshot displays the Tableau Server interface for a workbook named 'Sample - Superstore'. The 'Connected Workbooks' tab is active, showing a count of 69. The 'Lineage' panel on the right shows a count of 56 workbooks. The 'Fields (26)' table lists the following fields:

Type	Name	Sheets	Description
Abc	Category	24	awd
Abc	City	6	No description
Abc	Country/Region	6	No description
Abc	Customer Name	14	No description
#	Discount	5	No description
👤	Location	0	No description
🔗	Manufacturer	0	No description
#	Migrated Data (Count)	0	No description
+ #	Number of Records	4	No description

The 'Lineage' panel shows the following counts:

- Databases: 1
- Tables: 1
- Sample - Superstore Fields: 26
- Workbooks: 56
- Sheets: 402
- Dashboards: 16
- Owners: 14

Connected Workbooks tab counts workbooks that meet both these criteria:

- Connects to the data source (whether or not any fields are actually used in the workbook).
- The user has permissions to view (whether it's a worksheet, dashboard, or story).

Tableau Catalog Lineage counts workbooks that meet all these criteria:

- Has been indexed by Tableau Catalog.
- Connects to the data source and uses at least one field in the data source.

- Contains worksheets, including dashboards or stories that contain a worksheet, that use at least one field in the data source.

When metadata is blocked because of limited permissions, Catalog still counts the workbook. But instead of seeing some of the sensitive metadata, you see **Permissions required**. For more information, see Access lineage information.

Use email to contact owners

At the end of the lineage is Owners. The list of owners includes anyone assigned as the owner or contact for any content downstream from the lineage anchor.

You can email owners to let them know about changes to the data. (To email owners, you must have the 'Overwrite' (Save) capability on the lineage anchor content.)

1. Select **Owners** to see the list of people who are impacted by the data in this lineage.
2. Select the owners you want to send a message to.
3. Click **Send Email** to open the email message box.
4. Enter the Subject and your message in the text box, and click **Send**.

Set a Data Quality Warning

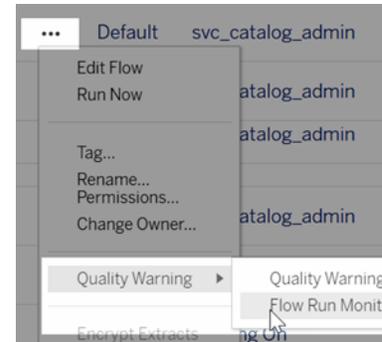
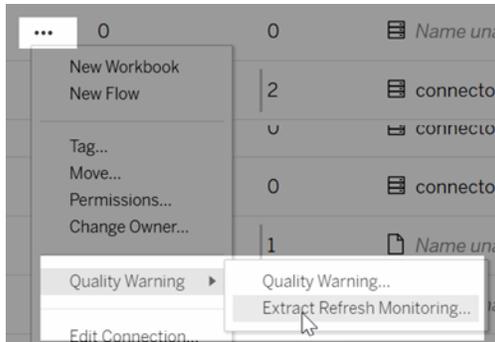
Data quality warnings are a feature of Tableau Catalog. Starting in 2019.3, Tableau Catalog is available in the Data Management Add-on to Tableau Online and Tableau Server. When Tableau Catalog is enabled in your environment, you can set data quality warnings. For more information, see "About Tableau Catalog" in the [Tableau Server](#) or [Tableau Online](#) Help.

You can set a warning message on a data asset so that users of that data are aware of issues. For example, you might want to let users know that a data source has been deprecated, or that a refresh has failed for an extract data source, or that a table contains sensitive data.

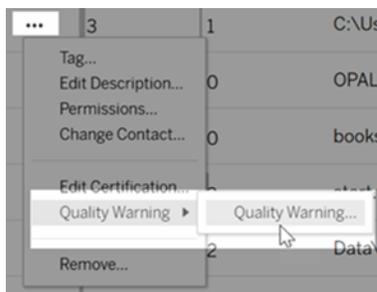
About data quality warnings

There are two kinds of data quality warnings you can set. The first kind is visible to users until you remove it. The second kind is generated by Tableau when you set Tableau to monitor for refresh failures of extract data sources or flow run failures.

For extracts and flows, you can set one data quality warning of each kind per asset.



For live data sources and for databases and tables, you can set only one quality warning.



When you set the first kind, it's visible to users until you remove it using the data quality dialog box. You can also use the dialog box to make any updates to the quality warning you set.

When you set Tableau to monitor for extract data source refresh failures or flow run failures, if a warning is generated, it's visible to users until the extract refresh or flow run is successful. You can update or remove a warning by opening and editing the contents in the dialog box.

Note: You can set both kinds of data quality warnings and enable high visibility using REST API. For more information, see [Add Data Quality Warning](#) in the Tableau REST API Help.

Who can see the data quality warning

When you set a warning, the warning is visible to users of the asset and any assets downstream from it. For example, a warning set on a table is visible to users looking at a dashboard with an upstream dependency on that table - users see a warning icon on the dashboard's Data Details tab and can open the pane to see the details. Warnings are visible elsewhere in Tableau as described in [Set high visibility for a data quality warning](#). This applies to Tableau Server and Tableau Online users.

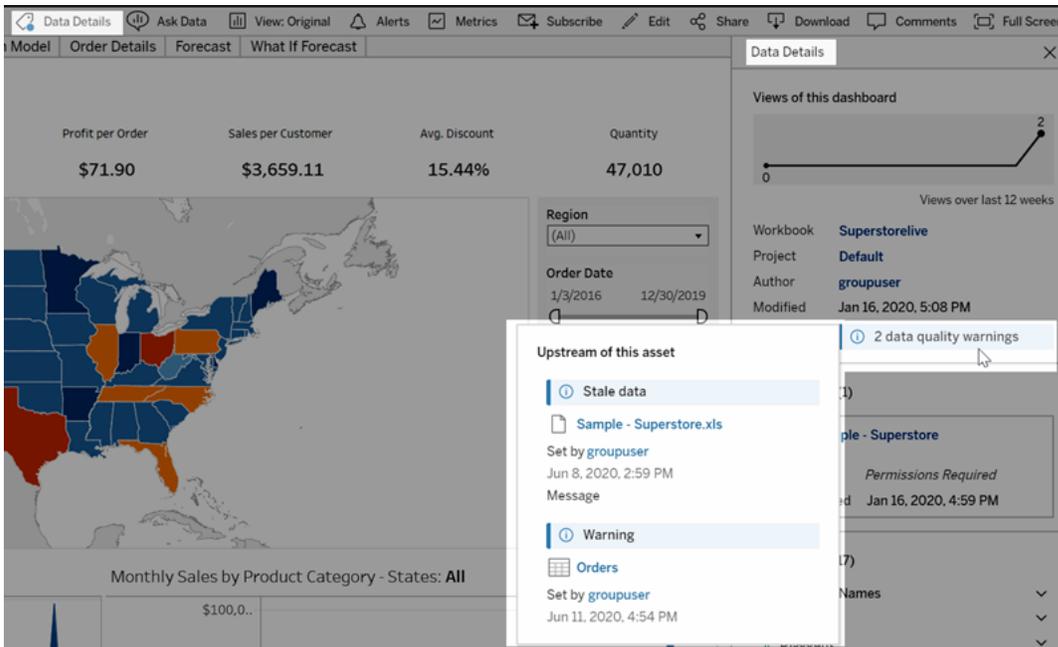
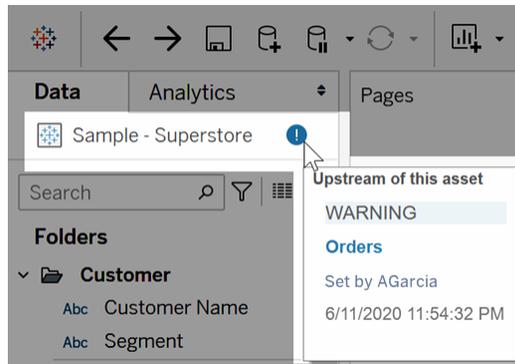


Tableau Desktop users see a warning icon (either a blue circle or a yellow triangle with an exclamation point) on the Data tab in a workbook sheet when a warning is set on a data source, table, or database used in the workbook. To see the details of the warning, hover over the warning icon with a mouse.



How to set a quality warning

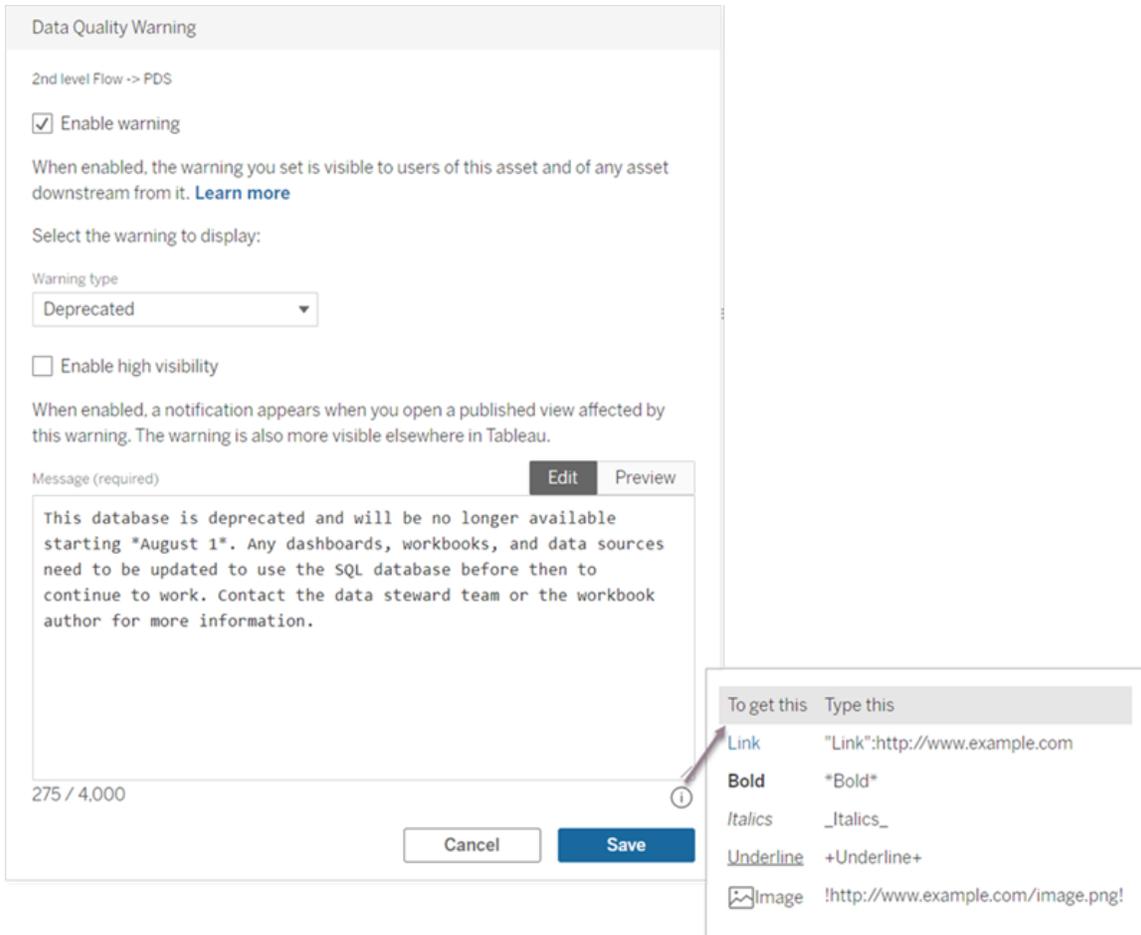
There are several types of data quality warnings you can set on an asset:

- Warning
- Deprecated
- Stale data
- Under maintenance
- Sensitive data

In addition to showing the type of data quality warning, you can include an optional message with more details about the warning.

To set a data quality warning:

1. Select the More actions menu (. . .) next to the asset you want to create a warning for, and then select **Quality Warning**.
2. Select the **Enable warning** check box.
3. Select the **Warning type** from the drop-down list.
4. (Optional) Select the **Enable high visibility** check box.
5. Enter a message to display to users.
6. Click **Save**.



You can format the text in a message with bold, underline, and italics, and include a link or an image. To see text formatting tips, click the information (i) icon above the Save button.

How to set a monitoring quality warning

You can set Tableau to monitor for two events: extract data source refresh failure and flow run failure. When the event occurs, Tableau generates a quality warning that appears in the same places that a manual quality warning appears.

To set Tableau to monitor for either event:

1. Select the More actions menu (. . .) next to the extract data source or flow you want to create a warning for, and then select the appropriate option:

- Quality Warning > Extract Refresh Monitoring
 - Quality Warning > Flow Run Monitoring
2. In the dialog box, select the monitoring check box.
 3. (Optional) Select the Enable high visibility check box.
 4. (Optional) Enter a message to display to users. To see text formatting tips, click the information (i) icon above the Save button.
 5. Click Save.

Flow Run Monitoring

CityFlow

Generate a quality warning when one or more flow outputs fail. Select 'High visibility' to show a notification when users open a published view affected by this warning and to raise visibility elsewhere in Tableau. [Learn more](#)

Flow run monitoring

Enable high visibility

Message (optional) Edit Preview

The flow run failed. If you can't resolve the issue, contact Ashley Garcia.

75 / 4,000 i

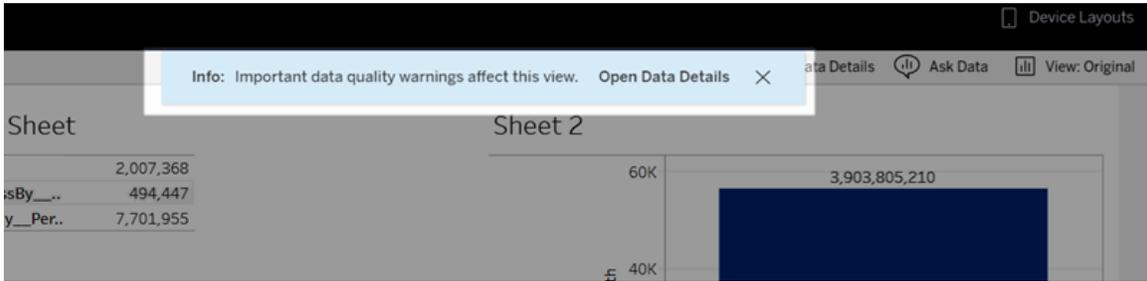
Cancel Save

Set high visibility for a data quality warning

There are times when you want to make sure that users of a visualization know important information about the data they're viewing, whether it be that the data is stale or that the source has been deprecated or that the flow run has failed. You can do this by selecting the **Enable high visibility** check box in the Data Quality Warning dialog box, as described in step 4 of How to set a data quality warning.

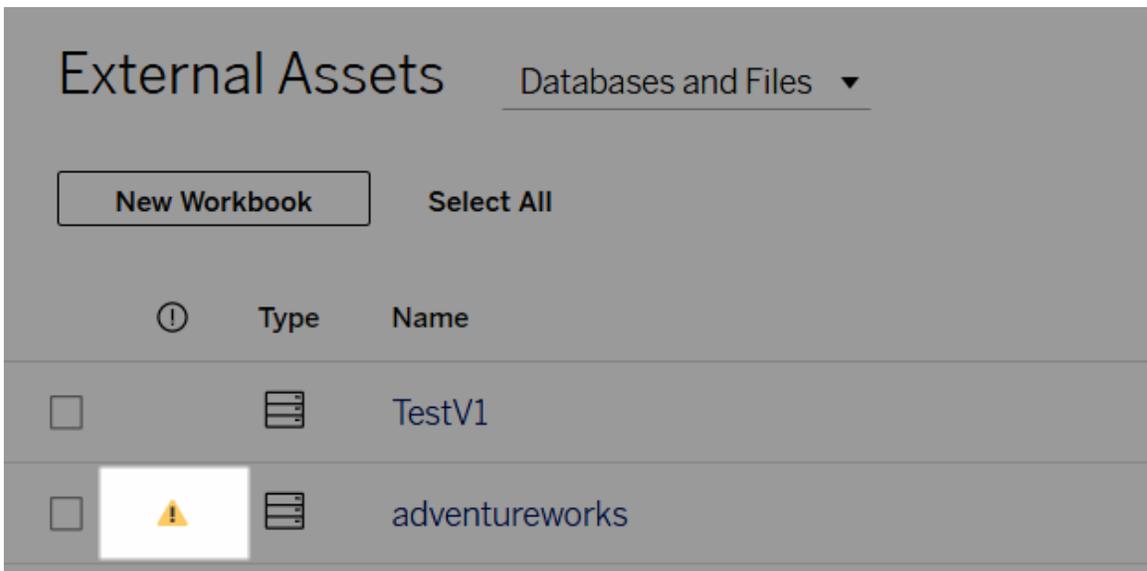
Tableau Server on Linux Administrator Guide

When enabled, a notification appears when anyone opens a published view affected by this warning.

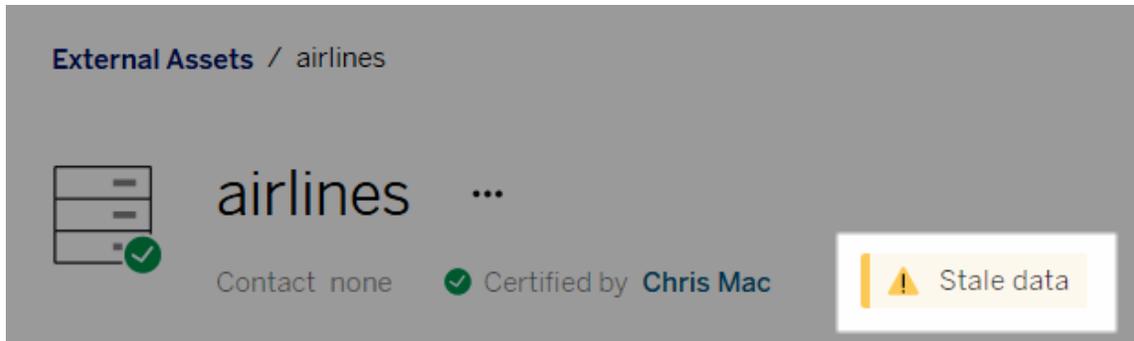


High visibility warnings are identified with a yellow warning symbol, which makes them more visible elsewhere in Tableau, as in these examples:

On the External Assets page:



On a database page:



On the Data Details pane:

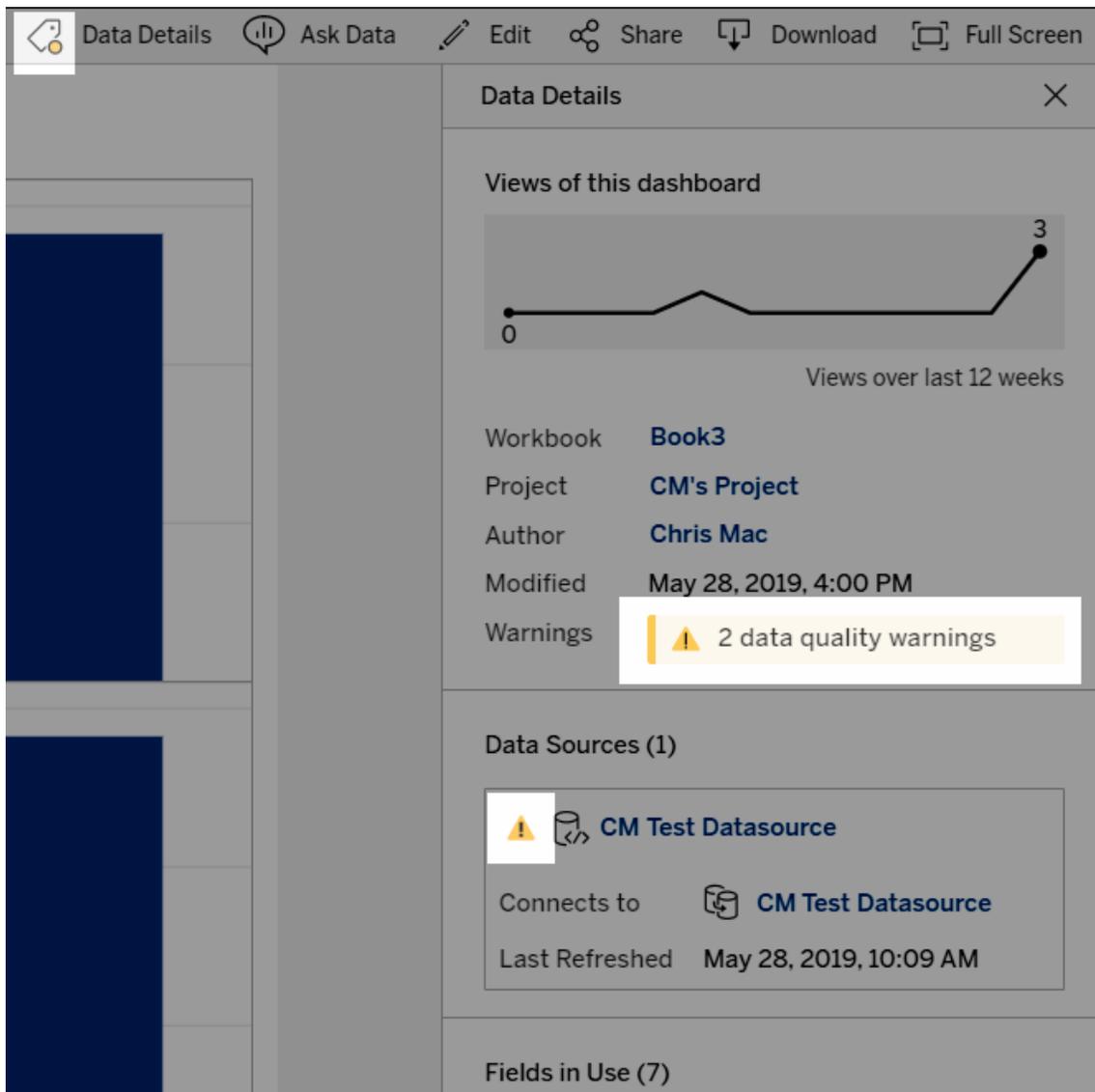
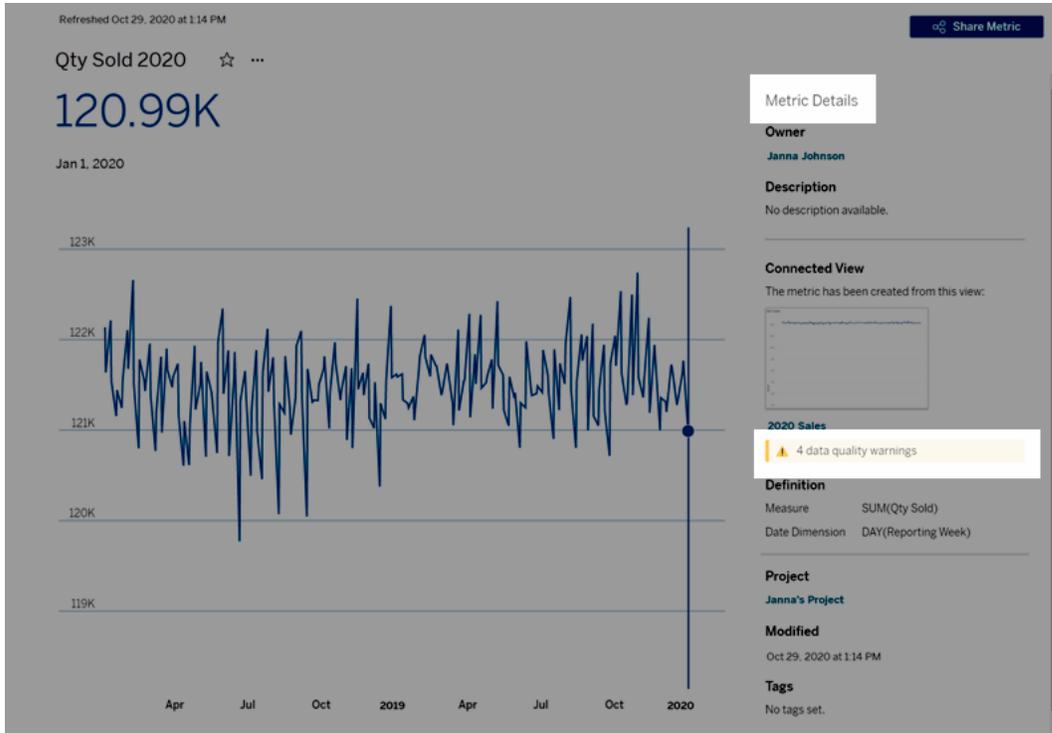


Tableau Server on Linux Administrator Guide

On metrics. Warnings appear when you open a metric in Tableau Mobile, and in Tableau Server and Tableau Online when you hover over a metric in grid view and on the metric details page, as shown below:



You can change a high visibility warning to normal visibility by clearing the **Enable high visibility** check box in the Data Quality Warning dialog box.

Remove a data quality warning

When a warning no longer applies, you can remove it by navigating to the data asset with the warning.

1. Select the More actions menu (. . .) next to the data asset and select **Quality Warning**.
2. Clear the **Enable warning** check box to remove the warning.
3. Click **Save**.

Who can do this

To set a data quality warning, you must either

- be a server or site administrator, or
- have the Save capability for the asset.

Manage Dashboard Extensions in Tableau Server

Dashboard extensions are web applications that run in custom dashboard zones and can interact with the rest of the dashboard using the [Tableau Extensions API](#). Dashboard extensions give users the ability to interact with data from other applications directly in Tableau.

Note: You must be a server administrator to enable dashboard extensions on the server, or to block specific extensions from running. You must be a server administrator to add extensions to the safe list and to control the type of data the extensions can access. The server administrator can also configure whether users on the site will see prompts when they add or view extensions in a dashboard. For information about extension security and recommended deployment options, see [Extension Security - Best Practices for Deployment](#)

For information about using dashboard extensions in Tableau, see [Use Dashboard Extensions](#).

Looking for Tableau Online? See [Manage Dashboard Extensions in Tableau Online](#).

Before you run extensions on Tableau Server

Tableau supports two types of dashboard extensions: Network-enabled extensions, which can be hosted on web servers located inside or outside of your local network and have full access to the web, and Sandboxed extensions, which run in a protected environment without access to any other resource or service on the web.

Note: Beginning with version 2021.1.0 Tableau supports integration with Einstein Discovery through the Einstein Discovery Dashboard extension. This is a special extension that has access to data in Salesforce.com and is allowed by default. It is not considered a Network-enabled extension or a Sandboxed extension. For more information on Einstein Discovery integration, see [What's New and Changed in Tableau Server](#).

Sandboxed extensions are hosted by Tableau and employ W3C standards, such as Content Security Policy (CSP), to ensure the extension can't make network calls outside of the hosting Tableau Server. A Sandboxed extension can query data in the dashboard, but it can't send that data anywhere outside of the sandbox. Sandboxed Extensions are supported in Tableau 2019.4 and later. By default, Sandboxed extensions are allowed to run if extensions are enabled for the site.

Network-enabled dashboard extensions are web applications and could be running on any computer set up as a web server. This includes local computers, computers in your domain, and third-party web sites. Because Network-enabled extensions could be hosted on third-party sites and could have access to the data in the dashboard, you want to only allow the extensions you trust. See [Test Network-enabled extensions for security](#).

For security, you can use the settings for dashboard extensions on Tableau Server to control and limit the dashboard extensions that are allowed to run.

- By default, Sandboxed extensions are allowed to run if extensions are enabled for the site.
- By default, no Network-enabled extensions are allowed unless they have been explicitly added to the safe list.
- By default, only extensions that use the HTTPS protocol are allowed, which guarantees an encrypted channel for sending and receiving data (the only exception is for `http://localhost`).
- If the Network-enabled extension requires full data (access to the underlying data) the

extension will not be able to run on Tableau Server unless you explicitly add the extension to the safe list and grant the extension access to full data.

Control dashboard extensions and access to data

Server administrators can control a global setting to allow extensions for all sites on the server. Server administrators can also put extensions, including Sandboxed extensions, on a global block list to prevent them from running (see Block specific extensions). By default, all Sandboxed extensions are enabled on the server, but site administrators can choose to override the default and prohibit Sandboxed extensions for the site.

Change the global setting enabling extensions on the server

1. To change this setting for the server, go to **Manage All Sites > Settings > Extensions**. If the server just has a single site, the global controls appear on the settings page for the site.
2. Under Dashboard Extensions, select or clear the **Let users run extensions on this server** checkbox. If this option is not selected, extensions are not allowed to run. This global setting overrides the **Let users run extensions on this site** settings for each site.

Change the default settings for a site

Server administrators can control whether to enable extensions for the site and whether to allow Sandboxed extensions on the site. That is, if extensions are enabled on the server, the default site settings allow Sandboxed extensions to run on the site, provided the extension is not specifically blocked on the server. The default site settings allow Network-enabled extensions to run that appear on the safe list for the site. Individual Sandboxed extensions can also be added to the safe list, if Sandboxed extensions are not allowed by default.

1. To change these settings for the site, go to **Settings > Extensions**.
2. Under Dashboard Extensions, configure these options:

- **Let users run extensions on this site**
- **Let Sandboxed extensions run unless they are specifically blocked by a server administrator**

Server administrators can add or remove Network-enabled and Sandboxed extensions from the safe list for a site. When you add an extension to the safe list, you can control whether to allow the extension to have access to full data. See [Add extensions to the safe list and configure user prompts](#).

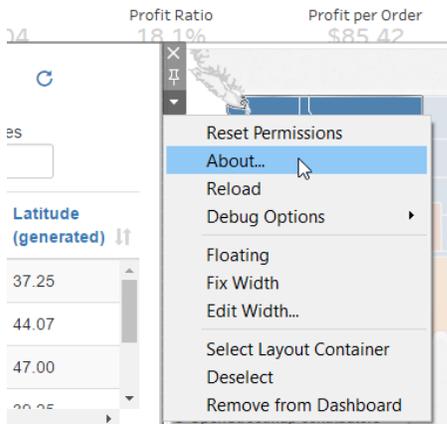
Identifying an extension

As a web application, an extension is associated with a URL. You use this URL to test and verify the extension. You also use the URL to add the extension to the safe list to allow full data access, or to the block list to prohibit any access.

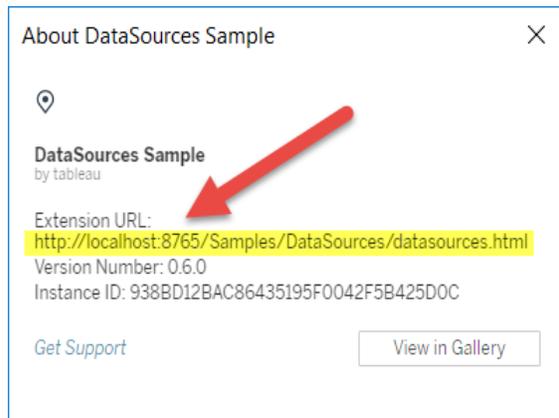
If you have the extension manifest file (.trex), an XML file that defines properties for the extension, you can find the URL from the `<source-location>` element.

```
<source-location>  
  <url>https://www.example.com/myExtension.html</url>  
</source-location>
```

If you have added the extension to the dashboard, you can find the URL from the extension properties. From the **More Options** menu, click **About**.



The About dialog box lists the name of the extension, the author of the extension, the web site of the author, along with the URL of the extension.



Add extensions to the safe list and configure user prompts

To ensure that users can use Network-enabled extensions that are trusted, you can add them to the safe list for the site. You can also add Sandboxed extensions to the safe list, if Sandboxed extensions aren't enabled by default on the site.

On the safe list, you can control whether to grant the extension full data access. By default, when you add an extension to the safe list, the extension only has access to the summary (or aggregated) data. You can also control whether users will see a prompt asking them to allow the extension access to data. You might want to add an extension to the safe list (for example, a Sandboxed extension) so that you can configure whether or not users see the prompts. When you hide the prompt from users, the extension can run immediately.

1. Go to **Settings > Extensions**.
2. Under **Enable Specific Extensions**, add the URL of the extension. See Identifying an extension.
3. Choose to **Allow** or **Deny** the extension **Full Data Access**.

Full data access is access to the underlying data in the view, not just the summary or aggregated data. Full data access also includes information about the data sources,

such as the names of the connection, fields, and tables. In most cases, if you are adding an extension to the safe list so that it can run, you will also want to allow the extension to have access to full data, if the extension requires it. Before adding extensions to the safe list, be sure to Test Network-enabled extensions for security.

4. Choose to **Show** or **Hide** the **User Prompts**.

Users see the prompts by default when they are adding an extension to a dashboard, or when they are interacting with a view that has an extension. The prompt tells users details about the extension and whether the extension has access to full data. The prompt gives users the ability to allow or deny the extension from running. You can hide this prompt from users, allowing the extension to run immediately.

Block specific extensions

The default global policy allows all Sandboxed extensions and those Network-enabled extensions that appear on the safe list for a site. Server administrators can keep specific extensions from running by adding them to the block list for the server. If an extension is on the global block list it overrides any settings for the extension on the safe list for a site.

1. To add an extension to the blocked list for the server, go to **Manage All Sites > Settings > Extensions**. On single-site installations, the block list is on the site **Extensions** settings page.
2. Under **Block Specific Extensions**, add the URL of the extension. See Identifying an extension.

Test Network-enabled extensions for security

Dashboard extensions are web applications that interact with data in Tableau using the Extensions API. Network-enabled dashboard extensions could be hosted on web servers inside or outside of your domain, and can make network calls and have access to resources on the Internet. Because of this and the potential vulnerabilities, such as cross-site scripting, you should test and vet Network-enabled dashboard extensions before users use them in dashboards on Tableau Desktop, and before you allow the extensions on Tableau Server.

Examine the source files

Dashboard extensions are web applications and include various HTML, CSS, and JavaScript files, and an XML manifest file (*.trex) that defines the properties in the extension. In many cases, the code for a dashboard extension is publicly available on GitHub and can be examined there or downloaded. In the manifest file (*.trex), you can find the source location, or URL indicated where the extension is hosted, the name of the author, and the web site of the author or company to contact for support. The `<source-location>` element specifies in the URL, the `<author>` element, specifies the name of the organization and the web site to contact for support (`website="SUPPORT_URL"`). The web site is the **Get Support** link user see in the **About** dialog box for the extension.

Many dashboard extensions reference external JavaScript libraries, such as the jQuery library or API libraries for third parties. Validate that the URL for external libraries points to a trusted location for the library. For example, if the connector references the jQuery library, make sure that the library is on a site that is considered standard and safe.

All extensions are required to use HTTPS protocol (`https://`) for hosting their extensions. You should examine the source files for the extension to ensure that any reference to external libraries is also using HTTPS or is hosted on the same web site as the extension. The one exception to the requirement of HTTPS is if the extension is hosted on the same computer as Tableau (`http://localhost`).

To the extent possible, make sure you understand what the code is doing. In particular, try to understand how the code is constructing requests to external sites, and what information is being sent in the request. In particular, check if any user-supplied data is validated to prevent cross-site scripting.

Understand data access

The Tableau Extensions API provides methods that can access the names of the active tables and fields in the data source, the summary descriptions of the data source connections, and the underlying data in a dashboard. If an extension uses any of these methods in a view, the extension developer must declare that the extension requires full data permission in the manifest file (*.trex). The declaration looks like the following.

```
<permissions>
  <permission>full data</permission>
</permissions>
```

Tableau uses this declaration to provide a prompt to users at run time that gives them the option of allowing this access or not. If the extension uses any one of these four methods, without declaring full-data permission in the manifest file, the extension will load but the method calls will fail.

For information about how an extension accesses data from the dashboard, and the JavaScript methods used, see [Accessing Underlying Data](#) in the Tableau Extensions API. To get a better understanding about what the extension can find out about the data, you can use the [DataSources](#) sample dashboard extension (available from the [Tableau Extensions API GitHub repository](#)) to see what data is exposed when the `getDataSourcesAsync()` method is called.

Test the extension in an isolated environment

If possible, test the dashboard extension in an environment that is isolated from your production environment and from user computers. For example, add a dashboard extensions to a safe list on a test computer or virtual machine that's running a version of Tableau Server that is not used for production.

Monitor traffic created by the dashboard extension

When you test a Network-enabled dashboard extension, use a tool like [Fiddler](#), [Charles HTTP proxy](#), or [Wireshark](#) to examine the requests and responses that the extension makes. Make sure that you understand what content the extension is requesting. Examine the traffic to be sure that the extension is not reading data or code that is not directly related to the purpose of the extension.

Configure Connections with Analytics Extensions

Tableau supports a set of functions that your users can use to pass expressions to analytics extensions for integration with R, Python, and Einstein Discovery.

This topic describes how to configure sites on Tableau Server with analytics extensions.

Because Tableau Server provides an authentication mechanism, it can be more secure to expose analytics extensions functionality to users through Tableau Server than in Tableau Desktop.

For more information about user scenarios and configuring Tableau Desktop, see [Pass Expressions Analytics Extensions](#), in the *Tableau Desktop and Web Authoring Help*.

Feature change history:

- 2021.2 — You can configure multiple analytics extension connections for each site. (You are limited to a single Einstein Discovery connection per site.)

For information about how to determine analytics extension usage in workbooks, see [Determining analytics extensions usage](#).

- 2021.1 — Einstein Discover is included as an analytics extension option. Einstein Discovery in Tableau is powered by [salesforce.com](#). Consult your agreement with [salesforce.com](#) for applicable terms.
- 2020.2 — You can configure a different analytics extension connection for each site on your server. Prior to this change a single analytics extension configuration applied globally to all sites on the server.
- 2020.1 — This functionality is now called *analytics extensions*. Previously the feature was called "external services."

Server SSL

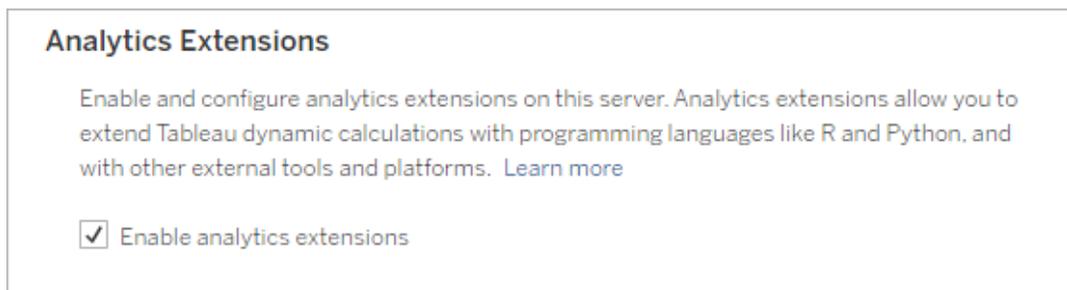
To configure SSL for analytics extensions, you must install a valid certificate on the computer running Tableau Server. The certificate must be trusted by the computer running Tableau Server. The certificate Subject field or one of the SAN entries on must exactly match the URI of the analytics extensions service configuration.

Enable analytics extensions

Before you configure extensions, you must enable analytics extensions server-wide.

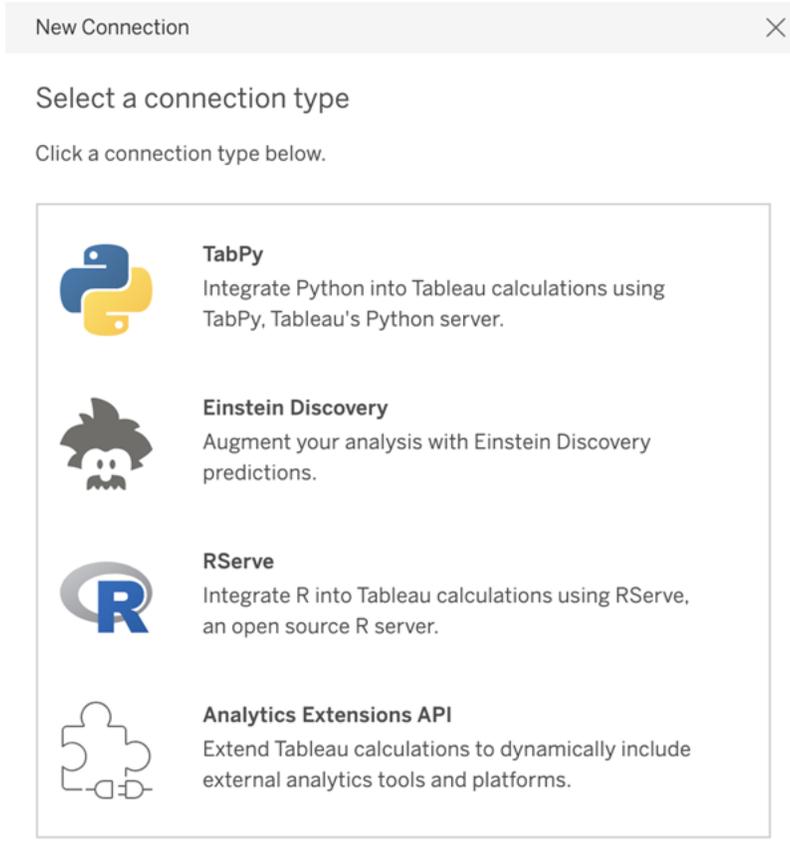
Tableau Server on Linux Administrator Guide

1. Sign in to the Tableau Server Admin Area.
 - If you only have a single site (default) on your server, click **Settings**, and then go to Step 2.
 - If you have multiple sites on your server:
 - a. Under **All Sites**, click **Manage all sites**.
 - b. Click the **Extensions** tab.
2. Scroll to **Analytics Extensions**, select **Enable analytics extensions**, and then click **Save**.



Configure analytics extensions settings

1. Sign in to the Tableau Server Admin Area.
2. On the Settings page, click the **Extensions** tab and then scroll to **Analytics Extensions**. (On multi-site deployments of Tableau Server, navigate to the site where you want to configure analytics extensions, and then click **Settings>Extensions**.)
3. **Multi-site deployments only**: you must enable Analytics Extensions on each site. Under Analytics Extensions, select **Enable analytics extensions for site**.
4. Under Analytics Extensions, click **Create new connection**.
5. In the **New Connection** dialog, click the connection type you want to add, then enter the configuration settings for your analytics service:



6. The options you need to configure depend on the connection type you choose:
- For an Einstein Discovery connection, click **Enable**.
 - For TabPy, RServer and Analytics Extensions API connections, enter the following information:
 - **Connection Name:** (Required) Specify the server type you are connecting to. RSERVE supports connections to R using the RServe package. TABPY supports connections to Python using TabPy, or to other analytics extensions.
 - **Require SSL** (Recommended): Select this option to encrypt the connection to the analytics service. If you specify a `HTTPS` URL in the **Hostname** field, then you must select this option.
 - **Hostname:** (Required) Specify the computer name or URL where the analytics service is running. This field is case sensitive.

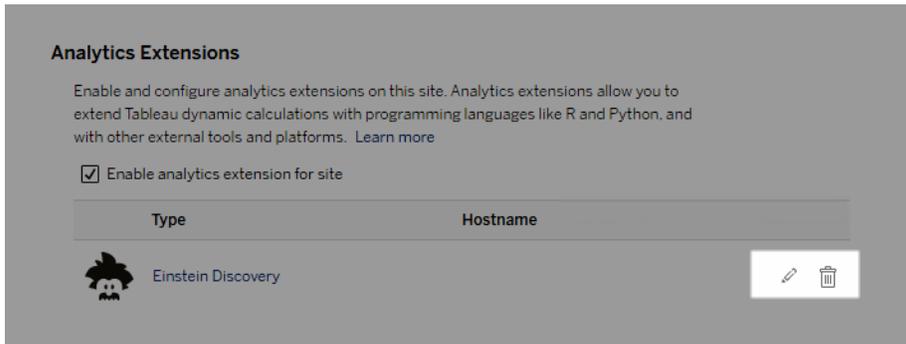
Tableau Server on Linux Administrator Guide

- **Port:** (Required) Specify the port for the service.
- **Sign in with a username and password**(Recommended): Select this option to specify user name and password that is used to authenticate to the analytics service.

7. Click **Create**.

Edit or delete an analytics extension connection

To edit or delete a configuration, navigate to **Analytics Extensions** on the **Extensions** tab of your site.



Click the **Edit** or **Delete** icon and follow the prompts to change the configuration.

Client requirement: Intermediate certificate chain for Rserve external service

As of Tableau Server version 2020.1, you must install a full certificate chain on Tableau Desktop computers (Windows and Mac) that are connecting to a Rserve external connection through Tableau Server. This requirement is due to how Rserve manages the handshake on secure connections.

Importing a root certificate on the Tableau Desktop is not sufficient, the entire certificate chain must be imported onto the client computer.

Script errors

Tableau cannot verify that workbooks that use an analytics extension will render properly on Tableau Server. There might be scenarios where a required statistical library is available on a user's computer but not on the analytics extension instance that Tableau Server is using.

A warning will be displayed when you publish a workbook if it contains views that use an analytics extension.

This worksheet contains external service scripts, which cannot be viewed on the target platform until the administrator configures an external service connection.

Determining analytics extensions usage

Beginning with version 2021.2, analytics extensions configurations are mapped at the workbook level. This allows administrators to use custom views to query the Tableau Repository and determine which workbooks are using which extensions, and how often they are used.

To do this you need to join the workbook connections table to tables showing workbook usage. For details about creating and using custom administrative views, see [Collect Data with the Tableau Server Repository and Create Custom Administrative Views](#).

Configure Einstein Discovery Integration

Beginning with version 2021.1.0, Tableau Server supports integration with Einstein Discovery, making Einstein Discovery predictions available to authors and viewers of workbooks and dashboards. Starting in version 2021.2.0, Einstein Discovery predictions is also now available when authoring flows on the web.

Einstein Discovery in Tableau is powered by [salesforce.com](https://www.salesforce.com). Consult your agreement with [salesforce.com](https://www.salesforce.com) for applicable terms.

To integrate Einstein Discovery with Tableau Server, there are several necessary configuration steps, including some in Tableau Server, and some in the Salesforce org running

Einstein Discovery. This overview outlines these steps for Dashboard extensions, Analytics extensions, and Tableau Prep extensions, and provides links to specific topics with steps for completing the server configuration.

For details on how to use Einstein Discovery predictions in Tableau, including licensing and permission requirements, see [Integrate Einstein Discovery Predictions in Tableau](#) in the Tableau Desktop and Web Authoring Help. For information about adding predictions in flows, see [Add Einstein Discovery Predictions to your flow](#).

Einstein Discovery dashboard extensions

The Einstein Discovery dashboard extension allow workbook authors to surface real-time predictions in Tableau. The dashboard extension delivers predictions interactively, on-demand, using source data in a Tableau workbook and an Einstein Discovery-powered model deployed in Salesforce.

To configure Tableau Server for the Einstein Discovery dashboard extension you need to do the following:

1. In Tableau Server:
 - a. Enable saved OAuth tokens for data connections and extensions in Tableau Server. Allow Saved Access Tokens
 - b. Enable Dashboard extensions for the server. See: Manage Dashboard Extensions in Tableau Server
2. In Salesforce, in the organization running Einstein Discovery:
 - a. Configure CORS in Salesforce.com for Einstein Discover Integration in Tableau Server.
 - b. In Salesforce, in the organization running Tableau CRM, create a connected app. See Create a Connected Salesforce App.
3. In Tableau Server, configure server for saved SF OAuth credentials using information from the connected app. Configure Tableau Server for Salesforce.com OAuth

Einstein Discovery analytics extensions

The Einstein Discovery analytics extension gives your users the ability to embed predictions directly in Tableau calculated fields. A table calc script requests predictions from a model deployed in Salesforce by passing its associated prediction ID and input data that the model requires. Use Model Manager in Salesforce to auto-generate a Tableau table calculation script, and then paste that script into a calculated field for use in a Tableau workbook.

To configure Tableau Server for either the Einstein Discovery analytics extension you need to do the following:

1. In Tableau Server:
 - a. Enable saved OAuth tokens for data connections and extensions in Tableau Server. Allow Saved Access Tokens
 - b. Enable analytics extensions for the server and configure a connection type.
See: [Configure Connections with Analytics Extensions](#)
2. In Salesforce, in the organization running Einstein Discovery, create a connected app. See [Create a Connected Salesforce App](#).
3. In Tableau Server, configure server for saved SF OAuth credentials using information from the connected app. [Configure Tableau Server for Salesforce.com OAuth](#)

Einstein Discovery Tableau Prep extensions

Supported in Tableau Server and Tableau Online starting in version 2021.2.0

The Einstein Discovery Tableau Prep extension enables users to embed Einstein predictions directly in their flows when authoring flows on the web.

To configure Tableau Server or Tableau Online for the Einstein Discovery Tableau Prep extension you need to do the following:

1. In Tableau Server:
 - a. Enable saved OAuth tokens for data connections and extensions in Tableau Server. See [Allow Saved Access Tokens](#)
 - b. Enable Tableau Prep Extensions for the server. See [Enable Tableau Prep Extensions](#).

2. In Salesforce, in the organization running Einstein Discovery, create a connected app. See [Create a Connected Salesforce App](#).
3. In Tableau Server, configure server for saved SF OAuth credentials using information from the connected app. Configure Tableau Server for [Salesforce.com OAuth](#)

Creators: Connect to data on the web

Before you can create a new workbook and build a view on the web to analyze your data on the web, you need to connect to your data. Tableau supports connecting to data sources on the web published through Tableau Desktop, or, connecting to data directly through Tableau Online, Tableau Server, or Tableau Public.

Starting in 2019.3, Tableau Catalog is available in the Data Management Add-on to Tableau Server and Tableau Online. When Tableau Catalog is enabled in your environment, in addition to navigating and connecting to data from Explore, you can navigate and connect to more kinds of data, like databases and tables, from Tableau Catalog. For more information about Tableau Catalog, see "About Tableau Catalog" in the [Tableau Server](#) or [Tableau Online](#) Help.

Note: Data connections created in Tableau Server or Tableau Online are **live connections** only. If you need to use an extract for web authoring, you can publish your data source through Tableau Desktop. To publish through Tableau Desktop, see [Publish Data Sources and Workbooks](#). However, data connections in Tableau Public are extracts only, and there is no way to publish a data source.

Open the Connect to Data page

On the web, you use the Connect to Data page to access data to connect to. After you sign in to Tableau Server or Tableau Online, you can open this page two ways:

- **Home > New > Workbook**
- **Explore > New > Workbook**

If you're on Tableau Public, you can open this page from your author profile:

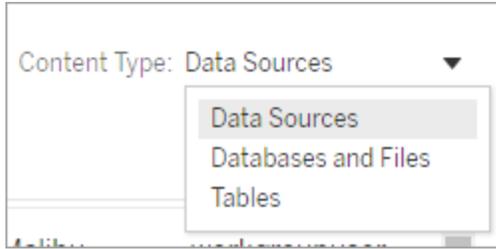
- **My Profile > Create a Viz**

Type	Name	Workbooks	Live/Last Extract	Connects To	Project	Owner
📁	2016 Fitness Challenge	12	Apr 10, 2017, 3:17 AM	fitness-challenge	Fitness Challenge	Jan
📁	TS Events	12	Aug 28, 2019, 12:04 ...	googleapis.com/bigquery	Admin Insights	Ed
📁	TS Users	12	Jan 15, 2020, 11:01 P...	googleapis.com/bigquery	Admin Insights	Ed
📁	Tableau_Builds	12	Jan 16, 2020, 2:06 AM	database	Data Platform	Eug
📁	Orders+ (Sample - Supe...	12	Mar 28, 2019, 3:29 P...	Sample - Superstore.xls	Building the Product	Din
📁	Pipeline	12	Live	branch-incident-manage...	Health & KPIs	Gar
📁	Tableau Public Lifetime I...	12	Live	tableau	Tableau Public	Jan
📁	GA - Help views - v2018	12	Jan 16, 2020, 12:12 ...		Content metrics	Mai

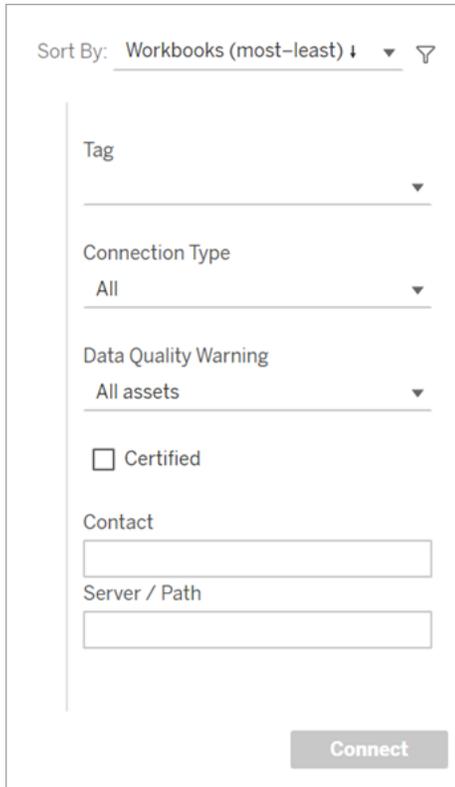
Select the filter icon to filter by Connection Type and by Certified.

When you have Catalog enabled in your environment:

- You can select from **Data Sources**, **Databases and Files**, or **Tables** to search for data.



- When you select **Databases and Files** or **Tables**, you can filter assets by Data Quality Warning: All assets, Assets without warnings, and Assets with warnings.



On the Connect to Data page, the tabs you see depend on the product you have.

Tableau Server

On Tableau Server, select from the following tabs to connect to data: On this site, Files, and Connectors.

Connect to data On this site

1. Select **On this site** to browse to or search for published data sources.
2. Select the data source under **Name** and click the **Connect** button.

Note: When you have the Data Management Add-on with Tableau Catalog enabled, you can use **On this site** to connect to databases and tables, as well as data sources.

Connect to files

Tableau supports uploading Excel, text-based data sources (.xlsx, .csv, .tsv), and spatial file formats that only require one file (.kml, .geojson, .topojson, .json, and Esri shapefiles and Esri File Geodatabases packaged in a .zip) directly in your browser. In the **Files** tab of the **Connect to Data** window, connect to a file by dragging and dropping it into the field or clicking **Upload from Computer**.

Use connectors

From the **Connectors** tab, you can connect to data housed in a cloud database or on a server in your enterprise. You need to supply connection information for each data connection that you make. For example, for most data connections, you need to supply a server name and your sign-in information.

[Supported Connectors](#) has information on how to connect Tableau to each of these connector types to set up your data source. If the connector you need doesn't appear in the Connectors tab, you can connect to data through Tableau Desktop and publish your data source to Tableau Online or Tableau Server for web authoring. Learn more about how to [Publish a Data Source](#) in Tableau Desktop.

When Tableau successfully connects to your data, the Data Source page opens so that you can prepare the data for analysis and begin building your view. To learn more, see [Creators: Prepare Data on the Web](#).

Tableau Server on Linux Administrator Guide

Tableau Server connectors

Action Matrix*	Google Cloud SQL	OData
Alibaba AnalyticDB for MySQL	Google Drive	OneDrive
Alibaba Data Lake Analytics	Kognito*	Oracle
Alibaba MaxCompute	Kyvos	Pivotal Greenplum Database
Amazon Athena	Hortonworks Hadoop Hive	PostgreSQL
Amazon Aurora for MySQL	IBM BigInsights	Progress OpenEdge*
Amazon Redshift	IBM DB2	Qubole Presto
Apache Drill	IBM PDA (Netezza)*	SAP Sybase ASE*
Aster Database*	MapR Hadoop Hive	SAP Sybase IQ*
Box	MariaDB	Salesforce
Cloudera Hadoop	MarkLogic*	SharePoint Lists
Databricks	Microsoft Azure Synapse Analytics	SingleStore (formerly MemSQL)
Denodo	Microsoft SQL Server	Snowflake
Dropbox	MonetDB*	Spark SQL
Exasol	MongoDB BI Connector	Teradata***
Google BigQuery**	MySQL	Vertica

*Not available on Linux servers.

**Google BigQuery requires OAuth when creating data sources from the web. Learn more about how server administrators can [Set up OAuth for Google](#).

***Teradata web authoring currently does not support query banding functionality. See [Teradata](#) for more details.

Tableau Catalog Supported Connectors

Tableau Catalog supports making a connection with a subset of the data connectors that Tableau Server supports. If a data source, database, file, or table is grayed out, you can't connect from Tableau Server. You can, however, connect from the Tableau Desktop **Connect** pane, if you have the correct permissions.

Tableau Online

On Tableau Online, select from the following tabs to connect to data: On this site, Files, Connectors, and Dashboard Starters.

Connect to data On this site

1. Select **On this site** to browse to or search for published data sources.
2. Select the data source under **Name** and click the **Connect** button

Note: When you have the Data Management Add-on with Tableau Catalog enabled, you can use **On this site** to connect to databases and tables, as well as data sources.

Connect to files

Tableau supports uploading Excel or text-based data sources (.xlsx, .csv, .tsv) directly in your browser. In the **Files** tab of the Connect to Data window, connect to an Excel or text file by dragging and dropping it into the field or clicking **Upload from Computer**.

Tableau Server on Linux Administrator Guide

Use connectors

From the **Connectors** tab, you can connect to data housed in a cloud database or on a server in your enterprise. You need to supply connection information for each data connection that you make. For example, for most data connections, you need to supply a server name and your sign-in information.

[Supported Connectors](#) has information on how to connect Tableau to your data using connectors. If the connector you need doesn't appear in the Connectors tab, you can connect to data through Tableau Desktop and publish your data source to Tableau Online or Tableau Server for web authoring. Learn more about how to [Publish a Data Source](#) in Tableau Desktop.

Note: If you're unable to connect to your data from Tableau Online, check to see if the database is publicly accessible. Tableau Online can only connect to data that's accessible from the public internet. If your data is behind a private network, you can connect using Tableau Bridge. To learn more, see [Publishers: Use Tableau Bridge to Keep Tableau Online Data Fresh](#).

Tableau Online Connectors

Amazon Athena	Google BigQuery*	OneDrive*
Amazon Aurora for MySQL	Google Cloud SQL	Oracle
Amazon EMR Hadoop Hive	Google Drive	PostgreSQL
Amazon Redshift	Hortonworks Hadoop Hive	SharePoint Lists
Box	MapR Hadoop Hive	SingleStore (formerly MemSQL)
Cloudera Hadoop	Microsoft Azure Synapse Analytics	Snowflake
	Microsoft SQL Server	Spark SQL

Databricks	MongoDB BI Connector	Teradata**
Denodo	MySQL	Vertica
Dropbox*	OData	
Exasol		

*For more information about using OAuth 2.0 standard for Google BigQuery, OneDrive, and Dropbox connections in Tableau Online, see [OAuth Connections](#).

**Teradata web authoring currently does not support query banding functionality. See [Teradata](#) for more details.

Tableau Catalog Supported Connectors

Tableau Catalog supports making a connection with a subset of data connectors that Tableau Online supports. If a data source, database, file, or table is grayed out, you can't connect from Tableau Online. You can, however, connect from the Tableau Desktop **Connect** pane, if you have the correct permissions.

Use Dashboard Starters

On Tableau Online, you can author and analyze data from LinkedIn Sales Navigator, Oracle Eloqua, Salesforce, ServiceNow ITSM, and QuickBooks Online using Dashboard Starters. On the **Dashboard Starter** tab, from the list of pre-built designs, select an option and click **Use Dashboard**. For more information, see [Dashboard Starters for Cloud-based Data Sources](#).

Tableau Public

On Tableau Public, you can connect to data by uploading a supported file.

Tableau Server on Linux Administrator Guide

Connect to files

Tableau supports uploading Excel or text-based data sources (.xlsx, .csv, .tsv) directly in your browser. In the **Files** tab of the Connect to Data window, connect to an Excel or text file by dragging and dropping it into the field or clicking **Upload from Computer**. If you don't have a data set, check out the free [sample data sets](#) on the Tableau Public website.

Use connectors

From the **Connectors** tab, you can connect to data housed in a cloud database. You need to supply connection information for each data connection that you make. For example, for most data connections, you need to supply your sign-in information.

[Supported Connectors](#) has information on how to connect Tableau to your data using connectors. If the connector you need doesn't appear in the Connectors tab, you can connect to data through Tableau Desktop and create an extract.

Note: If you're unable to connect to your data from Tableau Public, check to see if the database is publicly accessible. Tableau Public can only connect to data that's accessible from the public internet.

Tableau Public Connectors

Google Drive

OData

After you connect

When Tableau connects to your data, the Data Source page opens so that you can prepare the data for analysis and begin building your view. To learn more, see [Creators: Prepare Data on the Web](#).

Keep data fresh in web authoring

Update uploaded files in Tableau Online or Tableau Server: If you manually upload a file (Excel or text) for web authoring, Tableau can't refresh the file automatically. To update your data, select "Edit Connection" to upload a new version of the file.

In Tableau Public, go to your viz and click **Request Update**. You can also keep your data fresh automatically by selecting "Keep this data in sync" in Tableau Desktop Public Edition.

Update file-based published data sources in Tableau Online: If you have a published data source in Tableau Online (published through Tableau Desktop) that uses file-based data, you can keep it fresh using Tableau Bridge. For more information, see [Expand Data Freshness Options by Using Tableau Bridge](#).

Run Initial SQL

Note: Tableau Prep Builder version 2019.2.2 and later supports using Initial SQL, but doesn't yet support all of the same options supported by Tableau Desktop . For information about using Initial SQL with Tableau Prep Builder, see [Use Initial SQL to query your connections](#) in the Tableau Prep Builder online help.

When connecting to some databases, you can specify an initial SQL command that will run when a connection is made to the database, for example, when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. Initial SQL is not run when you refresh your view. Note that this initial SQL is different than a custom SQL connection. A custom SQL connection defines a relation (or table) to issue queries against. For more information, see [Connect to a Custom SQL Query](#).

You can use this command to:

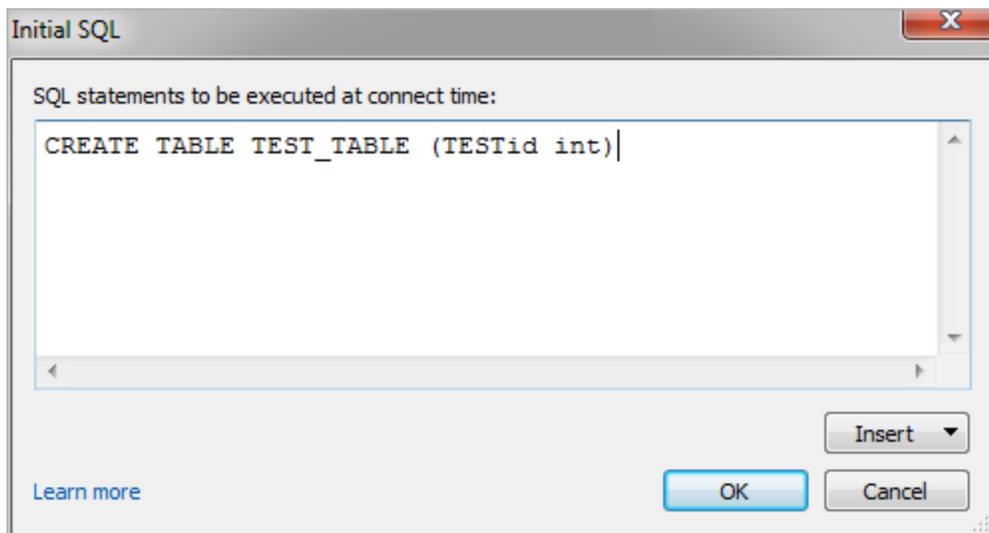
- Set up temporary tables to use during the session.
- Set up a custom data environment.

You have the option to add an initial SQL command in the Server Connection dialog box or on the Data Source page.

Note: If your data source supports running an initial SQL statement, an **Initial SQL** link appears in the lower-left corner of the Server Connection dialog box. For information about your data source, see [Supported Connectors](#).

To use initial SQL

1. In the Server Connection dialog box, click **Initial SQL**. Or, on the Data Source page, select **Data > Initial SQL** or **Data > Query Banding and Initial SQL** depending on the database you connect to.
2. Enter the SQL command into the Initial SQL dialog box. You can use the **Insert** drop-down menu to pass parameters to your data source.



Note: Tableau does not examine the statement for errors. This SQL statement is simply sent to the database when you connect.

Your software license may restrict you from using initial SQL with your connection. If you publish to Tableau Server, the server must be configured to allow Initial SQL statements. By

default, the server software is configured to allow these statements to run when the workbook is loaded in a web browser.

Administrators can configure server to ignore initial SQL statements by using the `tsm configuration set` command:

```
tsm configuration set -k vizqlserver.initialsql.disabled -v true
```

If the server doesn't allow initial SQL statements, the workbook opens, but the initial SQL commands are not sent.

For more information about the `tsm configuration set` command, see the [Tableau Server Help](#).

Parameters in an initial SQL statement

You can pass parameters to your data source in an initial SQL statement. There are several reasons why this is useful:

- You can configure impersonation using the **TableauServerUser** or **TableauServerUserFull** parameters.
- If your data source supports it, you can set up row-level security (for example, for Oracle VPD or SAP Sybase ASE) to make sure that users see only the data that they are authorized to see.
- You can provide more details in logging, for example, the Tableau version or the workbook name.

The following parameters are supported in an initial SQL statement:

Parameter	Description	Example of returned value
TableauServerUser	The user name of the current server user. Use when setting up imper-	jsmith

	sonation on the server. Returns an empty string if the user is not signed in to Tableau Server.	
TableauServerUserFull	The user name and domain of the current server user. Use when setting up impersonation on the server. Returns an empty string if the user is not signed in to Tableau Server.	domain.lan\jsmith
TableauApp	The name of the Tableau application.	Tableau Desktop Professional Tableau Server
TableauVersion	The version of the Tableau application.	9.3
WorkbookName	The name of the Tableau workbook. Use only in workbooks with an embedded data source.	Financial-Analysis

Examples

The following examples show different ways you can use parameters in an initial SQL statement.

- This example sets the security context on Microsoft SQL Server:

```
EXECUTE AS USER = [TableauServerUser] WITH NO REVERT;
```

- This example shows how, on a DataStax data source, you can use parameters to add detail to logging or to set up a session variable to keep track of the data:

```
SET TABLEAUVERSION [TableauVersion];
```

- This example can be used to help set up row-level security for Oracle VPD:

```
begin
```

```
DBMS_SESSION.SET_IDENTIFIER([TableauServerUser]);

end;
```

Note: Oracle PL/SQL blocks require a trailing semicolon to terminate the block. Consult Oracle documentation for the proper syntax.

Defer execution to the server

You can defer an initial SQL statement so that it is executed only on the server. One reason to defer execution to the server is if you don't have permission to execute the commands that set up impersonation. Use `<ServerOnly></ServerOnly>` tags to enclose the commands to be executed only on the server.

Example:

```
CREATE TEMP TABLE TempTable(x varchar(25));
INSERT INTO TempTable VALUES (1);
<ServerOnly>INSERT INTO TempTable Values(2);</ServerOnly>
```

Security and impersonation

If you use the **TableauServerUser** or **TableauServerUserFull** parameter in an initial SQL statement, you will create a dedicated connection that can't be shared with other users. This will also restrict cache sharing, which can enhance security, but may also slow performance.

Troubleshoot 'create table' for MySQL and Oracle connections

For MySQL connections, tables are not listed after using initial SQL to create table

When you connect to MySQL, if you run an initial SQL statement like the following, tables might not show because of the way Tableau constructs the query:

```
CREATE TABLE TestV1.testtable77(testID int);
```

To resolve this issue, add `IF NOT EXISTS` to the SQL statement:

```
CREATE TABLE IF NOT EXISTS TestV1.TestTable(testID int);
```

Tableau Server on Linux Administrator Guide

For Oracle connections, using initial SQL to create table causes Tableau to stall

When you connect to Oracle and run an initial SQL statement like the following, Tableau is stalled with a spinning wheel because of the way Tableau constructs the query:

```
CREATE TABLE TEST_TABLE (TESTid int)
```

To resolve this issue, use the following SQL statement:

```
BEGIN
EXECUTE IMMEDIATE 'create table test_table(testID int)';
EXCEPTION
WHEN OTHERS THEN NULL;
END;
```

Create and Interact with Flows on the Web

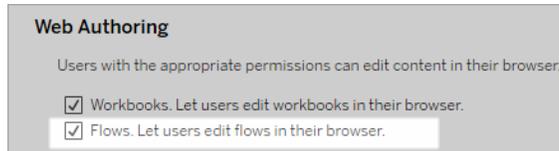
Starting in version 2020.4.0, you can create and interact with flows on Tableau Server to clean and prepare your data. Connect to your data, build a new flow, or edit an existing flow and your work is automatically saved every few seconds as you go. Create draft flows that are only available to you or publish your flow to make it available for others. Run your individual flows right from the web or run your flows automatically on a schedule using Tableau Prep Conductor if you have the Data Management Add-on. For more information, see [Tableau Prep on the Web](#).

For more information about configuring the Tableau Prep Flow Authoring process on Tableau Server, see [Tableau Prep Flow Authoring](#).

Complete the tasks described in this article to enable web authoring for flows and other flow features on the web.

Turn flow web authoring on or off for a site

1. In a web browser, sign in to the server as an administrator and go to the site in which you want web authoring to be enabled. In that site, click **Settings**.
2. In the **Web Authoring** section, select **Flows. Let users edit flows in their browser.** to enable the functionality.



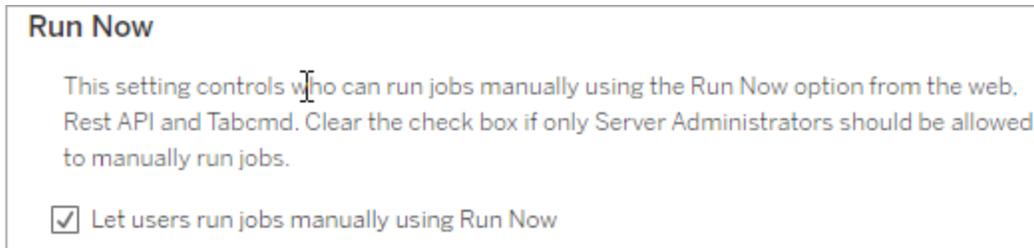
Clear the check box to turn off web authoring for that site.

3. If you want the change to take effect immediately, restart the server. Otherwise, the change takes effect after server session caching expires or the next time users sign in after signing out.

Enable Run Now

Control whether users or only administrators can run flows manually using the **Run Now** option. The Data Management Add-on is not required to run flows manually.

1. In a web browser, sign in to the server as an administrator and go to the site in which you want to enable Run Now for flows. In that site, click **Settings**.
2. In the **Run Now** section, select **Let users run jobs manually using Run Now** to enable the functionality.

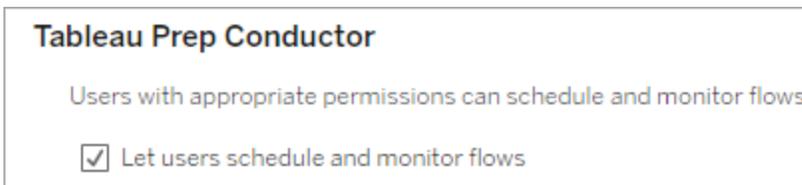


Clear the check box if only Server Administrators can run flows manually.

Enable Tableau Prep Conductor

If the Data Management Add-on is enabled, enable this option to let users schedule and track flows in Tableau Server and Tableau Online. For information about the additional configuration requirements for Tableau Prep Conductor, see [Tableau Prep Conductor](#).

1. In a web browser, sign in to the server as an administrator and go to the site in which you want to enable Tableau Prep Conductor. In that site, click **Settings**.
2. In the **Tableau Prep Conductor** section, select **Let users schedule and monitor flows** to enable the functionality.



Enable Tableau Prep Extensions

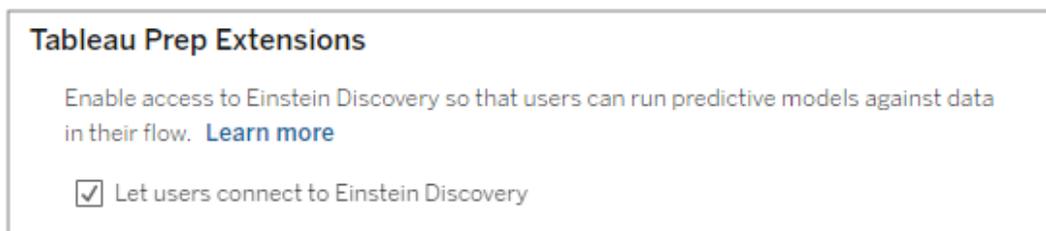
Supported in Tableau Server and Tableau Online starting in version 2021.2.0

Set this option to control whether users can connect to Einstein Discovery to run predictive models against data in their flow.

You can use Einstein Discovery-powered models to bulk score predictions for the data in your flow when authoring flows on the web. Predictions can help you make better informed decisions and take actions to improve your business outcomes.

You'll need to configure additional settings to include predictions in our flow. For more information, see [Add Einstein Discovery Predictions to your flow](#) and [Configure Einstein Discovery Integration](#).

1. In a web browser, sign in to the server as an administrator and go to the site in which you want to enable Tableau Prep Extensions. In that site, click **Settings > Extensions**.
2. In the **Tableau Prep Extensions** section, select **Let users connect to Einstein Discovery** to enable the functionality.



Turn autosave off or on

Enabled by default, this feature automatically saves a user's flow work every few seconds.

While not recommended, administrators can disable autosave on a site using the Tableau Server REST API method "Update Site" and `flowAutoSaveEnabled` setting. For more information, see [Tableau Server REST API Site Methods: Update Site](#). For more information about autosave on the web, see [Turn autosave off or on](#).

For more information about configuring site settings, see [Site Settings Reference](#) in the Tableau Server help.

Use Flows

You can publish flows from Tableau Prep Builder to Tableau Server or Tableau Online, run flows and manage flows.

With the Data Management Add-on you can also schedule and monitor flows. For more information, see [License the Data Management Add-on](#).

Feature	Description
<p>Publish a flow to Tableau Server or Tableau Online</p>	<p>Publish flows to Tableau Server or Tableau Online.</p>
<p>Schedule flow runs</p>	<p>You can create scheduled flow tasks to run a flow at a specific time or on a recurring basis. This feature requires the Data Management Add-on. For more information, see License the Data Management Add-on.</p>
<p>Manage flows</p>	<p>Manage a variety of actions, depending on your permissions, including run flows, tag flows, change owners, set permissions, download flows, move flows between projects, rename or delete flows.</p>
<p>Monitor flow health and performance</p>	<p>Set up email notifications at the site or server level when flows fail, view and resume suspended flow tasks, and view errors and alerts. You can also use Administrative Views to monitor the activities related to flows, performance history, and the disk space used at the server or site level. This feature requires the Data Management Add-on. For more information, see License the Data Management Add-on.</p>
<p>Tableau REST API - flow methods</p>	<p>Programmatically use the flow methods of the Tableau Server REST API.</p>

Create, Interact with, and Embed Views on the Web

You can create and interact with views on Tableau Server. For more information, see the following topics in the Tableau Help for users.

[Using Tableau on the Web](#)

[Tour your Tableau Site](#)

[Edit Views on the Web](#)

[Join your Data](#)

[Create a Dashboard](#)

[Create a Story](#)

[Embed Views and Dashboards in Web Pages](#)

[Make Workbooks Compatible Between Versions](#)

Alerts and subscriptions

[Troubleshoot Subscriptions](#)

[Send Data-Driven Alerts from Tableau Online or Tableau Server](#)

Manage Saved Credentials for Data Connections

Saved credentials enable you to connect to a data source without being prompted for your credentials. The credentials saved for your connection can be OAuth access tokens, or other credentials, such as user name and password.

On Tableau Server, if your server administrator has allowed you to save credentials, you can find and manage them in the **Saved Credentials** section on your Account Settings page.

Note: If you do not see the Saved Credentials section, consult with your server administrator about allowing saving credentials.

Remove saved credentials

To remove Tableau access to data, delete the associated saved credentials for that data from your account.

After you delete the credentials, you will need to sign in to the data the next time you access it. This will create new saved credentials.

Your administrator might choose for all users to use the same shared credentials for connecting to a data source. If this is the case, the saved credential is associated with the data connection for all users, and it does not appear under Saved Credentials on your Account Settings page.

Note: If you're a Tableau Server user and can't delete saved credentials, ask your administrator if they've cleared the **Allow users to save data source access tokens** option in the server settings.

Test connections using saved credentials

If the connector supports test functionality, you can test the connection using saved credentials.

1. While you're signed in to Tableau Server or Tableau Online, display your Account Settings page.

2. In the Saved Credentials section, click the **Test** link next to the stored connection that you want to test.

This test confirms that Tableau Online or Tableau Server can access your account using this corresponding saved credential. If the test succeeds, but you cannot access your data through this managed connection, confirm that the credentials you provided for this connection can access your data.

For example, if you accidentally created the connection using your personal Gmail account, but you use a different account to access a Google Analytics database, you will need to delete the saved credentials and sign in to the data using the appropriate Gmail account.

Update saved credentials

To help ensure uninterrupted data access from existing Tableau content after a custom OAuth client has been configured for your site, we encourage you to update your saved credentials. To update saved credentials, you can delete the previous saved credentials for a particular connector and then add it again.

When you add saved credentials again, both new and existing Tableau content will access the data using the custom OAuth client configured by your server administrator. For more information about custom OAuth clients, see FAYE.

1. Sign in to Tableau Server and navigate to your **My Account Settings** page.
2. Under **Saved Credentials for Data Sources**, do the following:
 1. Click **Delete** next to the saved credentials for a connector.
 2. Next to the same connector, click **Add** and follow the prompts to 1) connect to the custom OAuth client that your site admin notified you about and 2) save the latest credentials.

Clear all saved credentials

When you select **Clear All Saved Credentials**, the following items are removed from your user account:

- All saved credentials for connections that are stored in your account.

Caution: If any of these saved credentials are stored with published workbooks or data sources, deleting them also removes access to the data source from those locations. Effectively, this is like "changing the locks" anywhere the affected saved credentials are used.

- Passwords you have used to access published data extracts or workbooks that connect to them.

Use Relationships for Multi-table Data Analysis

Tables that you drag into this canvas use relationships. Relationships are a flexible way to combine data for multi-table analysis in Tableau.

Think of a relationship as a contract between two tables. When you are building a viz with fields from these tables, Tableau brings in data from these tables using that contract to build a query with the appropriate joins.

We recommend using relationships as your first approach to combining your data because it makes data preparation and analysis easier and more intuitive. **Use joins only when you absolutely need to.** Learn more about the basics of creating relationships in this 5-minute video.

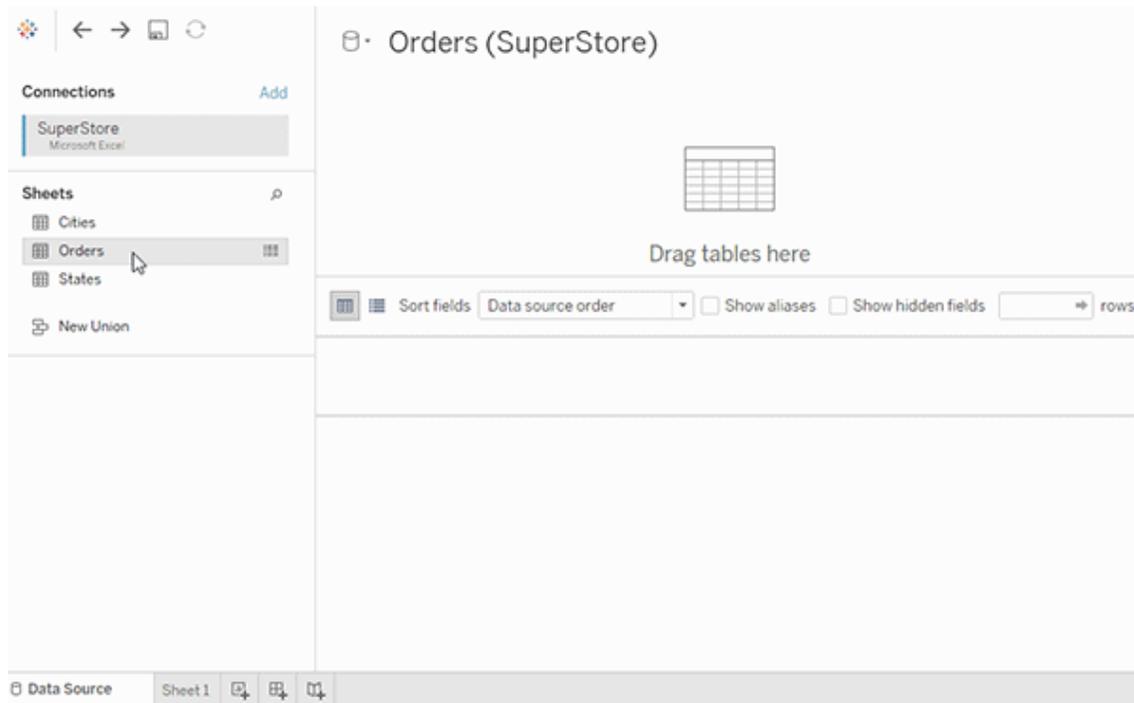
Learn more about how relationships work in these Tableau blog posts:

- [Relationships, part 1: Introducing new data modeling in Tableau](#)
- [Relationships, part 2: Tips and tricks](#)
- [Relationships, part 3: Asking questions across multiple related tables](#)

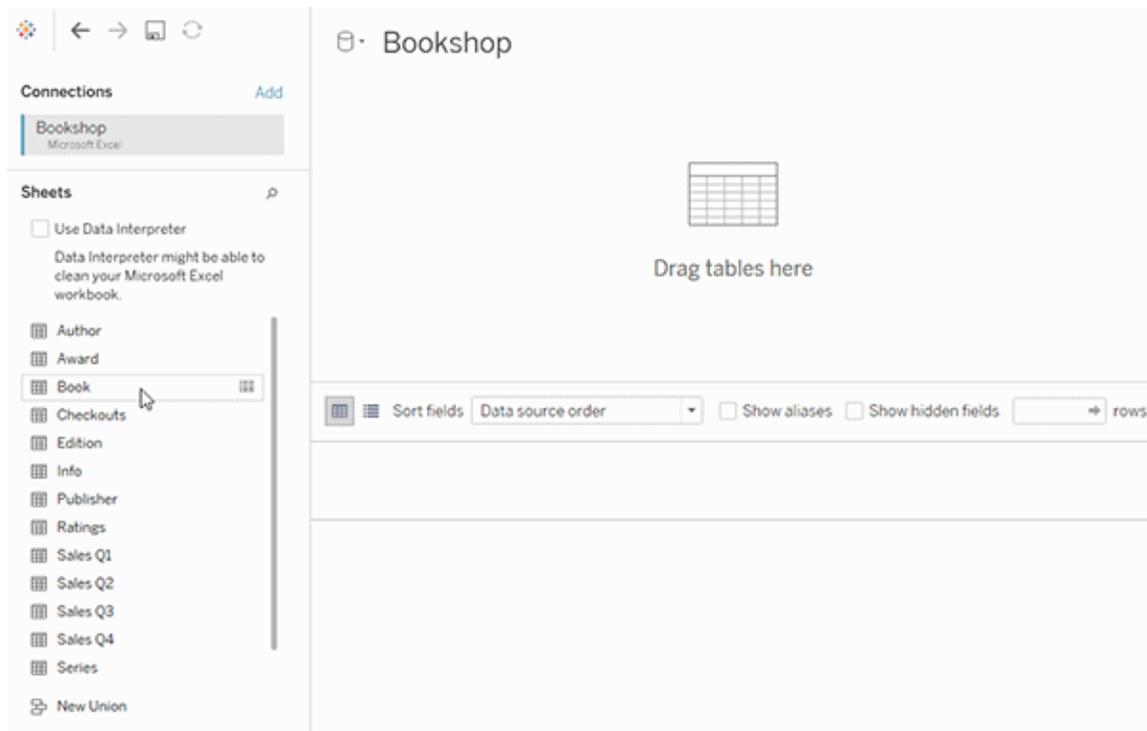
Are you building a new data source and workbook?

Drag a table to the Data Source page canvas to start building your data source.

A data source can be made of a single table that contains all of the dimension and measure fields you need for analysis...



Or, you can create a multi-table data source by dragging out more tables and defining their relationships...



Watch this 1-minute video about getting started with using relationships.

For more information about using relationships, see [Relate Your Data](#), [How Relationships Differ from Joins](#), [The Tableau Data Model](#), and [Create and define relationships](#).

For more information on changes to data sources and analysis in Tableau 2020.2, see [What's Changed with Data Sources and Analysis in 2020.2](#) and [Questions about Relationships, the Data Model, and Data Sources in 2020.2](#).

Are you opening an older workbook or data source?

When you open a pre-2020.2 workbook or data source in 2020.2, your data source will appear as a single logical table in the canvas, with the name "Migrated Data" or the original table name. Your data is preserved and you can continue to use the workbook as you did before.

The screenshot shows the Tableau interface with a logical table named "Customer Orders" displayed in the physical layer. The table is composed of data from multiple CSV files. The interface includes a sidebar with file connections, a top navigation bar, and a data table with columns for Address ID, Country, City, Postal Code, Region, State, Customer ID, Customer Name, and Segment.

Address ID (Address...)	Country	City	Postal Code	Region	State	Customer ID (Cust...)	Customer Name	Segment
232	United States	Henderson	42420	South	Kentucky	CG-12520	Claire Gute	Consumer
232	United States	Henderson	42420	South	Kentucky	CG-12520	Claire Gute	Consumer
232	United States	Henderson	42420	South	Kentucky	CG-12520	Claire Gute	Consumer
232	United States	Henderson	42420	South	Kentucky	CG-12520	Claire Gute	Consumer
232	United States	Henderson	42420	South	Kentucky	CG-12520	Claire Gute	Consumer
232	United States	Henderson	42420	South	Kentucky	CG-12520	Claire Gute	Consumer
232	United States	Henderson	42420	South	Kentucky	CG-12520	Claire Gute	Consumer

To see the physical tables that make up the single logical table, double-click that logical table to open it in the physical layer. You will see its underlying physical tables, including joins and unions.

For more information on changes to data sources and analysis in Tableau 2020.2, see [What's Changed with Data Sources and Analysis in 2020.2](#) and [Questions about Relationships, the Data Model, and Data Sources in 2020.2](#).

The Tableau Data Model

Every data source that you create in Tableau has a data model. You can think of a data model as a diagram that tells Tableau how it should query data in the connected database tables.

The tables that you add to the canvas in the **Data Source** page create the structure of the data model. A data model can be simple, such as a single table. Or it can be more complex, with multiple tables that use different combinations of relationships, joins, and unions.

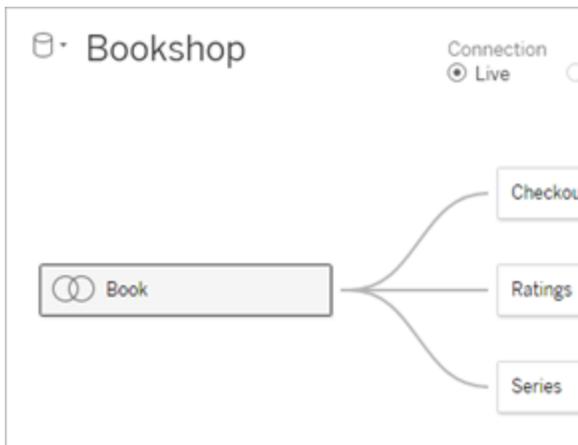
The data model has two layers:

Tableau Server on Linux Administrator Guide

- The default view that you first see in the Data Source page canvas is the *logical layer* of the data source. You combine data in the logical layer using relationships (or noodles). Think of this layer as the Relationships canvas in the Data Source page. For more information, see [Use Relationships for Multi-table Data Analysis](#).
- The next layer is the *physical layer*. You combine data between tables at the physical layer using joins and unions. Each logical table contains at least one physical table in this layer. Think of the physical layer as the Join/Union canvas in the Data Source page. Double-click a logical table to view or add joins and unions.

Logical Layer

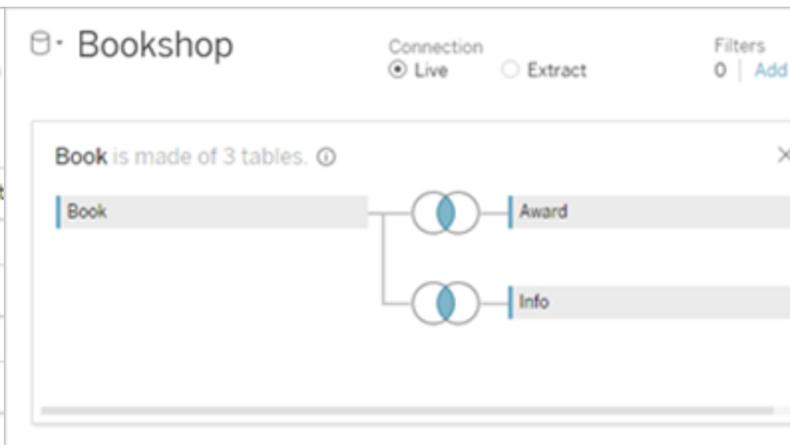
Noodles = Relationships



The top-level view of a data source with multiple, related tables. This is the logical layer. Logical tables can be combined using relationships (noodles). They don't use join types. They act like containers for physical tables.

Physical Layer

Venn diagram = Joins



Double-click a logical table to open it and see its physical tables. Physical tables can be combined using joins or unions. In this example, the Book logical table is made of three, joined physical tables (Book, Award, Info).

Logical Layer

Relationships canvas in the Data Source page

Physical Layer

Join/Union canvas in the Data Source page

Logical Layer

Tables that you drag here are called logical tables

Logical tables can be related to other logical tables

Logical tables are like containers for physical tables

Level of detail is at the row level of the logical table

Logical tables remain distinct (normalized), not merged in the data source

Physical Layer

Tables that you drag here are called physical tables

Physical tables can be joined or unioned to other physical tables

Double-click a logical table to see its physical tables

Level of detail is at the row level of merged physical tables

Physical tables are merged into a single, flat table that defines the logical table

Layers of the data model

The top-level view that you see of a data source is the **logical layer** of the data model. You can also think of it as the Relationships canvas, because you combine tables here using relationships instead of joins.

When you combine data from multiple tables, each table that you drag to the canvas in the logical layer must have a relationship to another table. You do not need to specify join types for relationships; during analysis Tableau automatically selects the appropriate join types based on the fields and context of analysis in the worksheet.

The **physical layer** of the data model is where you can combine data using joins and unions. You can only use pivots in this canvas. You can think of it as the Join/Union canvas. In previous versions of Tableau, the physical layer was the only layer in the data model. Each logical table can contain one or more physical tables.

Important: You can still create single-table data sources in Tableau that use joins and unions. The behavior of single-table analysis in Tableau has not changed. Your upgraded workbooks will work the same as they did before 2020.2.

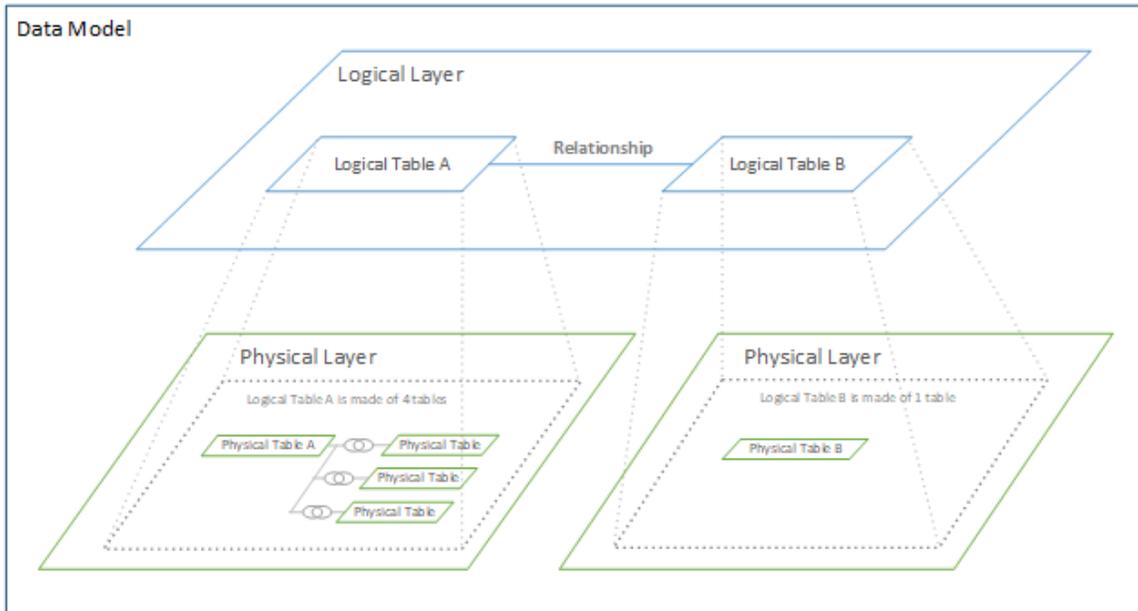
Learn more: For related information on combining data using relationships, also see these

topics and blog posts:

- How Relationships Differ from Joins
- Use Relationships for Multi-table Data Analysis
- [Relate Your Data](#)
- [Relationships, part 1: Introducing new data modeling in Tableau](#)
- [Relationships, part 2: Tips and tricks](#)
- [Relationships, part 3: Asking questions across multiple related tables](#)

Understanding the data model

In previous versions of Tableau, the data model had only the physical layer. In Tableau 2020.2 and later, the data model has the logical (semantic) layer and a physical layer. This gives you more options for combining data using schemas to fit your analysis.

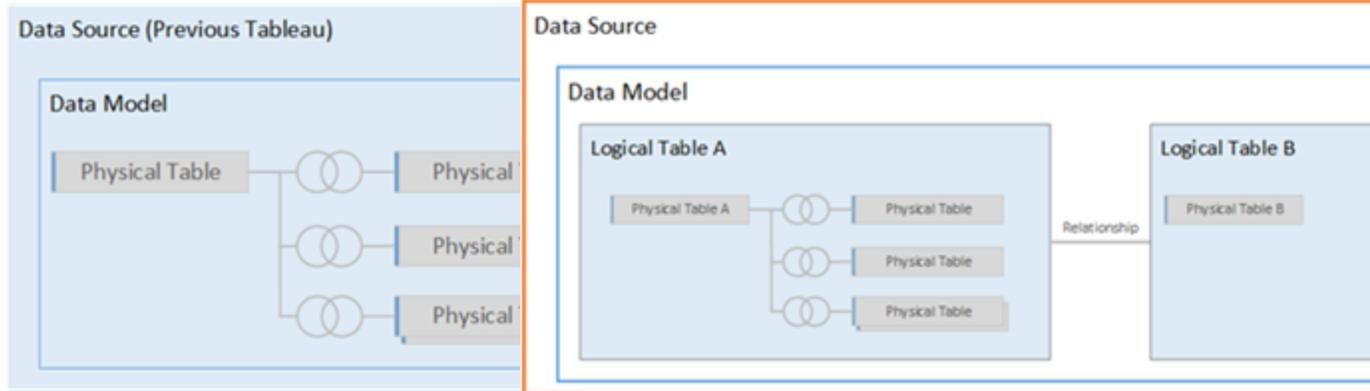


In Tableau 2020.2 and later, a logical layer has been added in the data source. Each logical table contains physical tables in a physical layer.

In previous versions of Tableau, the data model in your data source consisted of a single, physical layer where you could specify joins and unions. Tables added to the physical layer (joined or unioned) create a single, flattened table (denormalized) for analysis.

Previous versions

2020.2 and later



In versions of Tableau before 2020.2, the data model has only the physical layer

In 2020.2 and later, the data model has two layers: the logical layer and the physical layer

In Tableau 2020.2, the data model in your data source includes a new semantic layer above the physical layer—called the logical layer—where you can add multiple tables and relate them to each other. Tables at the logical layer are not merged in the data source, they remain distinct (normalized), and maintain their native level of detail.

Logical tables act like containers for merged physical tables. A logical table can contain a single, physical table. Or it can contain multiple physical tables merged together through joins or unions.

Build a new model

When you add one or more tables to the logical layer, you are essentially building the data model for your data source. A data source can be made of a single, logical table, or you can drag multiple tables to the canvas to create a more complex model.

- The first table that you drag to the canvas becomes the root table for the data model in your data source.
- After you drag out the root table, you can drag out additional tables in any order. You will need to consider which tables should be related to each other, and the matching field pairs that you define for each relationship.

Tableau Server on Linux Administrator Guide

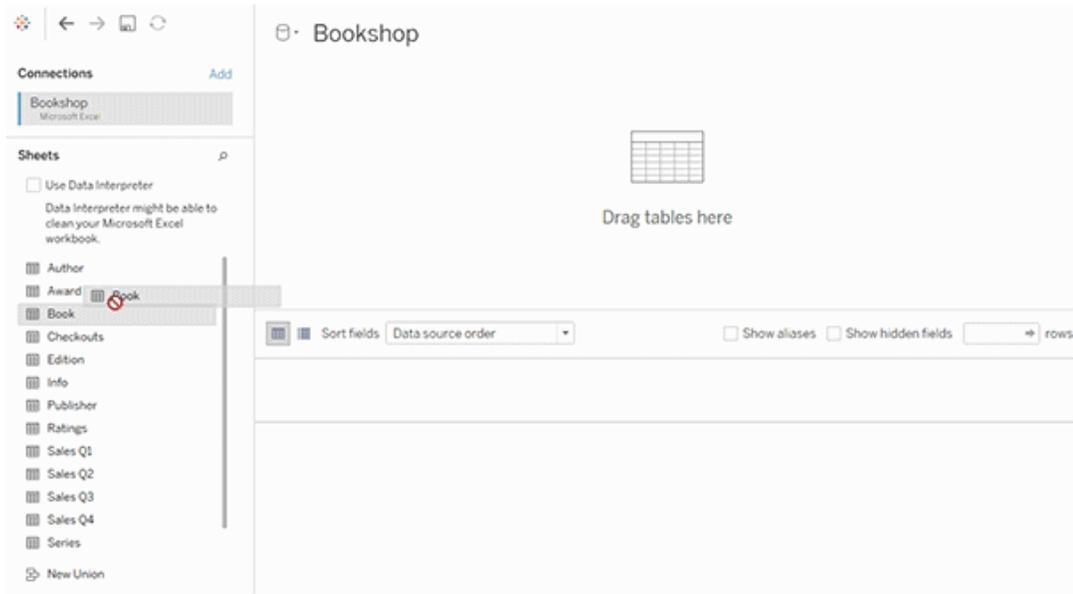
- If you are creating a star schema, it can be helpful to drag the fact table out first, and then relate dimension tables to that table.
- Deleting a table in the canvas automatically deletes its related descendants as well. If you delete the root table, all other tables in the model are also removed.
- Each relationship must be made of at least one matched pair of fields. Add multiple field pairs to create a compound relationship. Matched pairs must have the same data type. Changing the data type in the Data Source page does not change this requirement. Tableau will still use the data type in the underlying database for queries.
- Relationships can be based on calculated fields.
- You can specify how fields used in the relationships should be compared by using operators when you define the relationship.

Note: You can't edit the data model of a published data source.

For more information about relationships, see [Create and define relationships in Relate Your Data](#).

Multi-table model

- To create a multi-table model, drag tables to the logical layer of the Data Source page canvas.



Tables that you drag to the logical layer of the Data Source page canvas must be related to each other. When you drag additional tables to the logical layer canvas, Tableau automatically attempts to create the relationship based on existing key constraints and matching fields to define the relationship. If it can't determine the matching fields, you will need to select them.

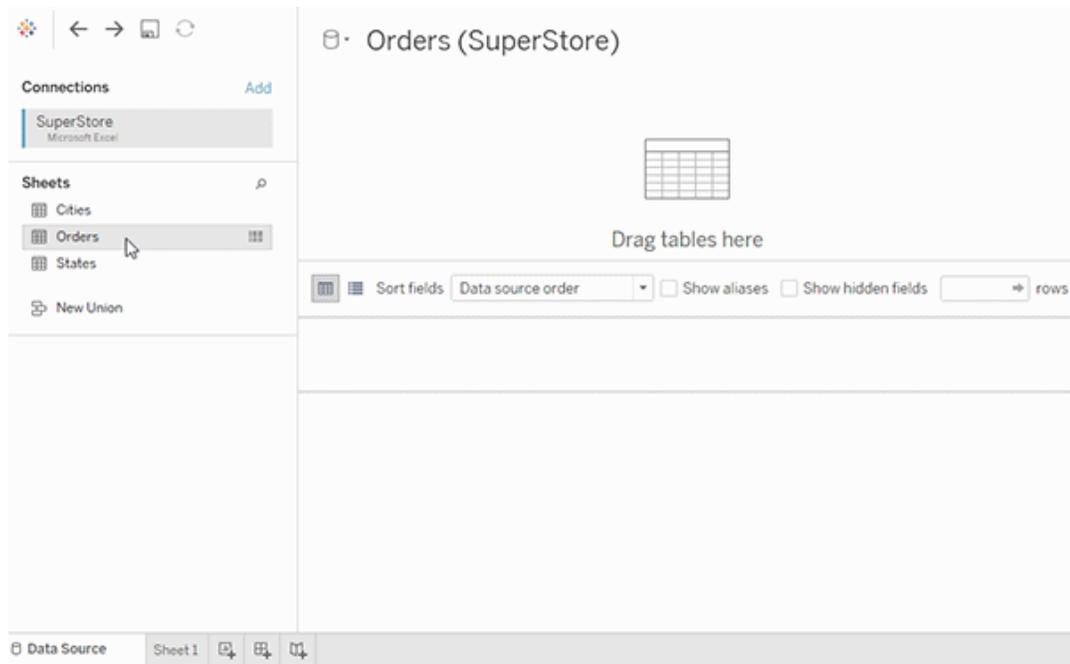
If no constraints are detected, a **Many-to-many** relationship is created and referential integrity is set to **Some records match**. These default settings are a safe choice and provide the most a lot of flexibility for your data source. The default settings support full outer joins and optimize queries by aggregating table data before forming joins during analysis. All column and row data from each table becomes available for analysis.

You can add more data inside any logical table by double-clicking the table. This opens the physical layer of the Data Source page canvas. If you need to use joins or unions, you can drag the tables you want to join or union into the physical layer canvas. The physical tables are merged in their logical table.

Follow the steps in [Create and define relationships](#) to combine multiple tables.

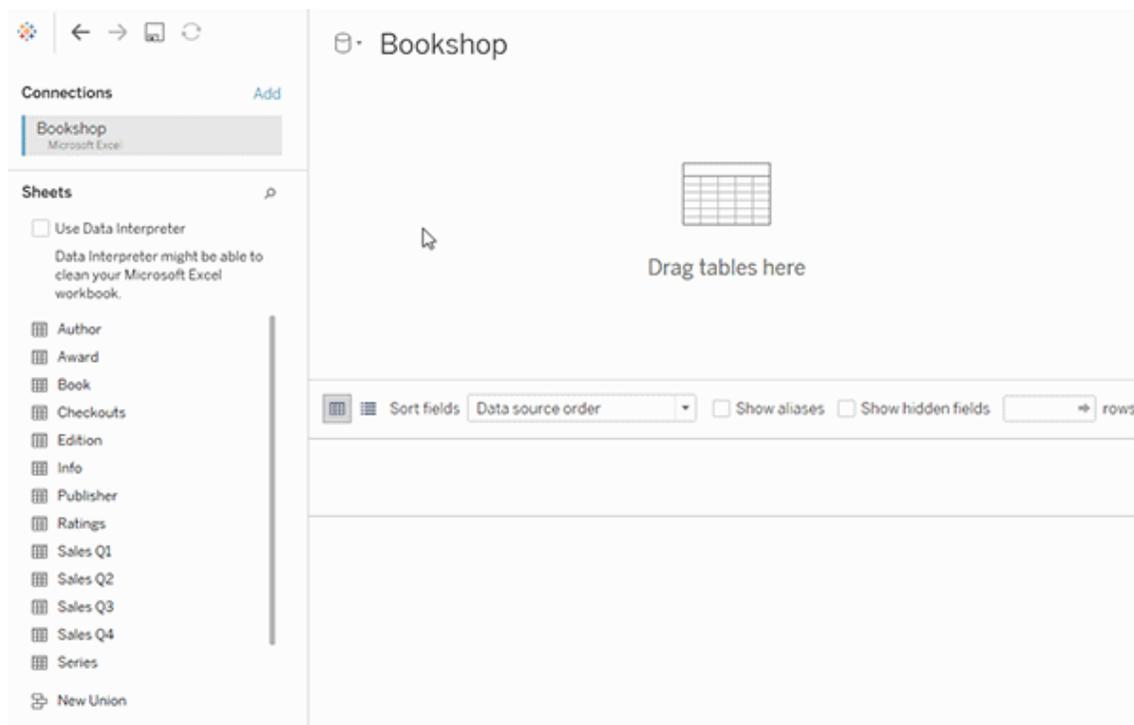
Single-table model

- To create a single-table model, drag a table into the logical layer canvas of the Data Source page. You can then use the fields from that table in the Data pane for analysis.



Single-table model that contains other tables

You can add more data inside the single, logical table by double-clicking the table. This opens the physical layer of the Data Source page canvas. If you need to use joins or unions, you can drag the tables you want to join or union into the physical layer canvas. The physical tables are merged in their logical table.



This example shows the Book table in the Relationships canvas (logical layer) of the data source. Double-clicking the Book logical table opens the Join/Union canvas (physical layer).

In this example, the joins merge the Award and Info tables with the Book table. In this case, the join between Book and Award will be one-to-many, at the level of detail of awards. This would duplicate measure values for Book and Info. To avoid duplication, you could relate Award and Info to Book instead of joining them inside of the Book logical table.

Supported data model schemas

The data modeling capabilities introduced to Tableau in 2020.2 are designed to make analysis over common multi-table data scenarios—including star and snowflake data models—easy. The following types of models are supported in Tableau data sources.

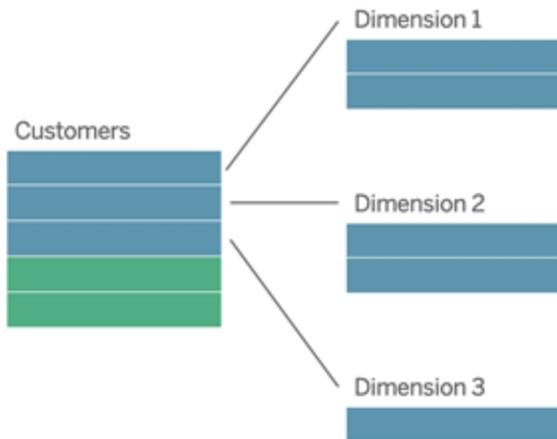
Single-table

Analysis over a single logical table that contains a mixture of dimensions and measures works just as in Tableau pre-2020.2. You can build a logical table using a combination of joins, unions, custom SQL, and so on.



Star and snowflake

In enterprise data warehouses, it is common to have data structured in star or snowflake schemas where measures are contained in a central fact table and dimensions are stored separately in independent dimension tables. This organization of data supports many common analysis flows including rollup and drill down.



These models can be directly represented with relationships in the data modeling capabilities available starting with Tableau 2020.2.

Drag the fact table into the model first and then relate the dimension tables to the fact table (in a star schema) or to other dimension tables (in a snowflake).

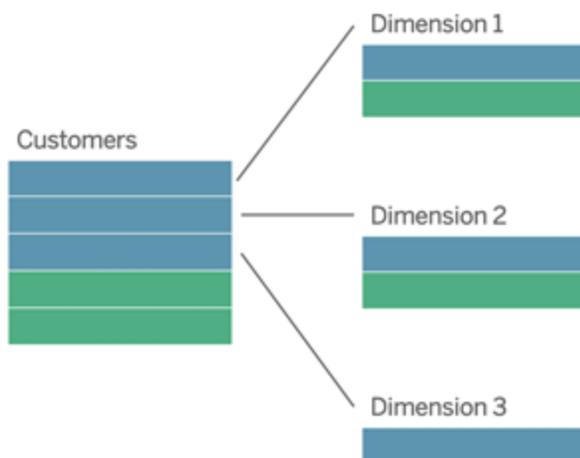
Typically, in a well-modeled star or snowflake schema, the relationships between the fact table and the dimension tables will be many-to-one. If this information is encoded in your data warehouse, Tableau will automatically use this to set the relationship's Performance

Options. If not, you can set this information yourself. For more information, see [Optimize Relationship Queries Using Performance Options](#).

In a well-modeled star or snowflake schema, every row in the fact table will have a matching entry in each of the dimension tables. If this is true and captured in your data warehouse integrity constraints, Tableau will automatically use this information to set the referential integrity setting in Performance Options. If some fact table rows do not have a matching row in a dimension table (sometimes called “late-arriving dimensions” or “early-arriving facts”), Tableau will default to retaining all rows when computing measures, but may drop values when showing dimension headers. For more information, see [Optimize Relationship Queries Using Performance Options](#).

Star and snowflake with measures in more than one table

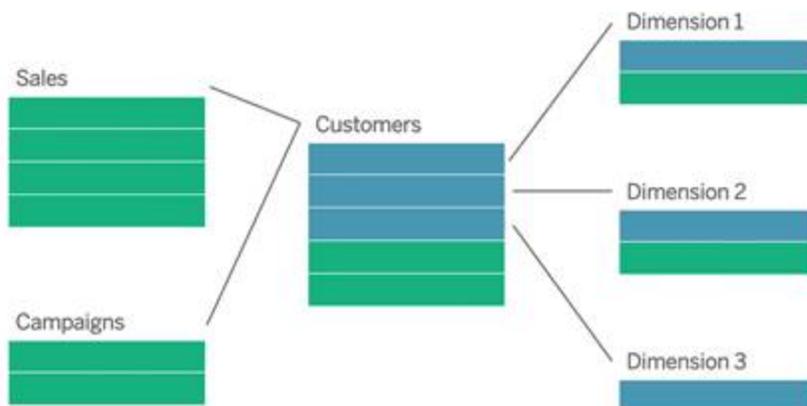
In some star or snowflake schemas, all the measures for your analysis are contained in the fact table. However, it is often true that additional measures of interest may be related to the dimension tables in your analysis. Even if the dimension tables do not contain measures, it is common in analysis to want to count or otherwise aggregate dimension values. In these cases, the distinction between fact tables and dimension tables is less clear. To create clarity when viewing your data model, we recommended adding the finest grain table to the data source canvas first, and then relating all other tables to that first table.



If you were to join these tables together into a single logical table, the measures in the dimension tables would be replicated, resulting in distorted aggregates unless you took precautions to deduplicate the values using LOD calculations or COUNT DISTINCT. However, if you instead create relationships between these tables, Tableau will aggregate measures before performing joins, avoiding the problem of unnecessary duplication. This relieves you of the need to carefully track the level of detail of your measures.

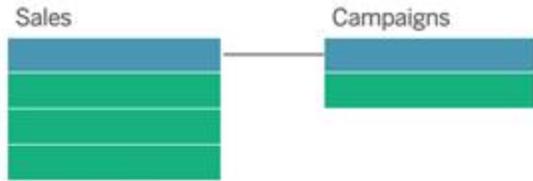
Multi-fact analysis

Tableau’s data modeling capabilities support some forms of multi-fact analysis. Additional fact tables (containing measures) can be added to any of the previously mentioned models as long as they only relate to a single dimension table. For example, you can bring two or more fact tables together to analyze a shared dimension, such as in Customer 360-like analyses. These fact tables can be at a different level of detail than the dimension table, or from each other. They can also have a many-to-many relationship with the dimension table. In these scenarios, Tableau will ensure that values are not replicated before aggregation.



If you don’t have a shared dimension table that relates your fact tables, you can sometimes dynamically build one using custom SQL or by using joins or unions of other dimension tables.

Two fact tables can be related directly to each other on a common dimension. This type of analysis works best when one of the fact tables contains a superset of the common dimension.



Unsupported models

- **Multiple fact tables related to multiple shared dimension tables.** In some use cases it is common to have multiple fact tables related to multiple shared dimension tables. For example, you might have two fact tables, Store Sales and Internet Sales, related to two common dimension tables, Date and Customer. Typically, such scenarios would require creating a circular relationship in your data model. Circular relationships are not supported in 2020.2.

You can approximate this type of model by merging some of the tables in the physical layer. For example, you might be able to union Store Sales and Internet Sales into a single table, which can then be related to Date and Customer. Alternatively, you might be able to cross-join Date and Customer to create a single dimension table which can then be related to Store Sales and Internet Sales.

- **Directly relating 3 or more fact tables on shared dimensions.** While it is possible to build this model in the logical layer, you might see unwanted results, unless you only use dimensions from a single table.

Requirements for relationships in a data model

- When relating tables, the fields that define the relationships must have the same data type. Changing the data type in the Data Source page does not change this requirement. Tableau will still use the data type in the underlying database for queries.
- You can't define relationships based on geographic fields.
- Circular relationships aren't supported in the data model.
- You can't edit relationships in a published data source.
- You can't define relationships between published data sources.

Tableau Server on Linux Administrator Guide

- Your workbook must use an embedded data source for you to be able to edit relationships and performance options in the Data Source page in Tableau Online or Tableau Server.

Factors that limit the benefits of using related tables

- Dirty data in tables (i.e. tables that weren't created with a well-structured model in mind and contain a mix of measures and dimensions in multiple tables) can make multi-table analysis more complex.
- Using data source filters will limit Tableau's ability to do join culling in the data. Join culling is a term for how Tableau simplifies queries by removing unnecessary joins.
- Tables with a lot of unmatched values across relationships.
- Interrelating multiple fact tables with multiple dimension tables (attempting to model shared or conformed dimensions).

How Relationships Differ from Joins

Relationships are a dynamic, flexible way to combine data from multiple tables for analysis. You don't define join types for relationships, so you won't see a Venn diagram when you create them.

Think of a relationship as a contract between two tables. When you are building a viz with fields from these tables, Tableau brings in data from these tables using that contract to build a query with the appropriate joins.

- **No up-front join type.** You only need to select matching fields to define a relationship (no join types). Tableau first attempts to create the relationship based on existing key constraints and matching field names. You can then check to ensure they are the fields you want to use, or add more field pairs to better define how the tables should be related.
- **Automatic and context-aware.** Relationships defer joins to the time and context of analysis. Tableau automatically selects join types based on the fields being used in the visualization. During analysis, Tableau adjusts join types intelligently and preserves the native level of detail in your data. You can see aggregations at the level of detail of the fields in your viz rather than having to think about the underlying joins. You don't need to use LOD expressions such as FIXED to deduplicate data in related tables.
- **Flexible.** Relationships can be many-to-many and support full outer joins. When you combine tables using relationships, it's like creating a custom, flexible data source for

every viz, all in a single data source for the workbook. Because Tableau queries only tables that are needed based on fields and filters in a viz, you can build a data source that can be used for a variety of analytic flows.

For more information, see [Relate Your Data](#) and [Don't Be Scared of Relationships](#).

Joins are still available as an option for combining your data. Double-click a logical table to go to the join canvas. For more information, see [Where did joins go?](#)

Watch a video: For an overview of data source enhancements and an introduction to using relationships in Tableau, see this 5-minute video.

For related information about how relationship queries work, see these Tableau blog posts:

- [Relationships, part 1: Introducing new data modeling in Tableau](#)
- [Relationships, part 2: Tips and tricks](#)
- [Relationships, part 3: Asking questions across multiple related tables](#)

Characteristics of relationships and joins

Relationships are a dynamic, flexible way to combine data from multiple tables for analysis. We recommend using relationships as your first approach to combining your data because it makes data preparation and analysis easier and more intuitive. [Use joins only when you absolutely need to.](#)

Here are some advantages to using relationships to combine tables:

- Make your data source easier to define, change, and reuse.
- Make it easier to analyze data across multiple tables at the correct level of detail (LOD).
- Do not require the use of LOD expressions or LOD calculations for analysis at different levels of detail.
- Only query data from tables with fields used in the current viz.

Relationships

- Are displayed as flexible noodles between logical tables
- Require you to select matching fields between two logical tables

Tableau Server on Linux Administrator Guide

- Do not require you to select join types
- Make all row and column data from related tables potentially available in the data source
- Maintain each table's level of detail in the data source and during analysis
- Create independent domains at multiple levels of detail. Tables aren't merged together in the data source.
- During analysis, create the appropriate joins automatically, based on the fields in use.
- Do not duplicate aggregate values (when Performance Options are set to Many-to-Many)
- Keep unmatched measure values (when Performance Options are set to Some Records Match)

Joins

Joins are a more static way to combine data. Joins must be defined between physical tables up front, before analysis, and can't be changed without impacting all sheets using that data source. Joined tables are always merged into a single table. As a result, sometimes joined data is missing unmatched values, or duplicates aggregated values.

- Are displayed with Venn diagram icons between physical tables
- Require you to select join types and join clauses
- Joined physical tables are merged into a single logical table with a fixed combination of data
- May drop unmatched measure values
- May duplicate aggregate values when fields are at different levels of detail
- Support scenarios that require a single table of data, such as extract filters and aggregation

Requirements for using relationships

- When relating tables, the fields that define the relationships must have the same data type. Changing the data type in the Data Source page does not change this requirement. Tableau will still use the data type in the underlying database for queries.
- You can't define relationships based on geographic fields.
- Circular relationships aren't supported in the data model.
- You can't edit relationships in a published data source.
- You can't define relationships between published data sources.

- Your workbook must use an embedded data source for you to be able to edit relationships and performance options in the Data Source page in Tableau Online or Tableau Server.

Factors that limit the benefits of using related tables

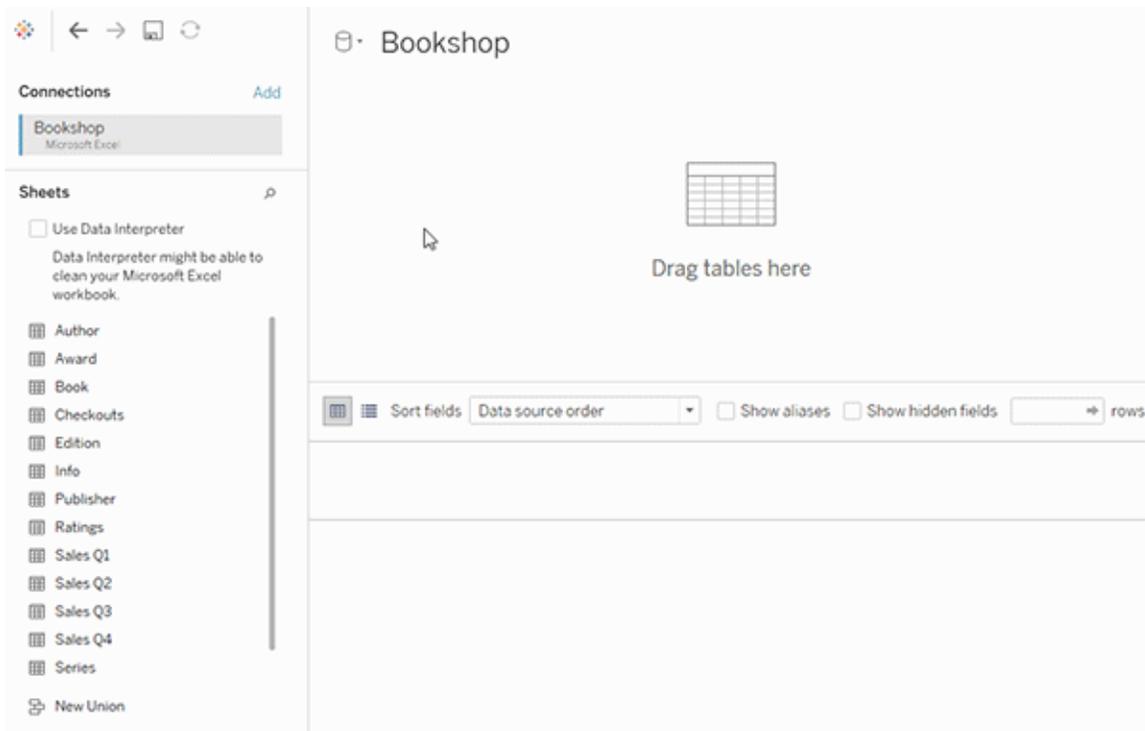
- Dirty data in tables (i.e. tables that weren't created with a well-structured model in mind and contain a mix of measures and dimensions in multiple tables) can make multi-table analysis more complex.
- Using data source filters will limit Tableau's ability to do join culling in the data. Join culling is a term for how Tableau simplifies queries by removing unnecessary joins.
- Tables with a lot of unmatched values across relationships.
- Interrelating multiple fact tables with multiple dimension tables (attempting to model shared or conformed dimensions).

Where did joins go?

You can still specify joins between tables in the physical layer of a data source. Double-click a logical table to go to the Join/Union canvas in the physical layer and add joins or unions.

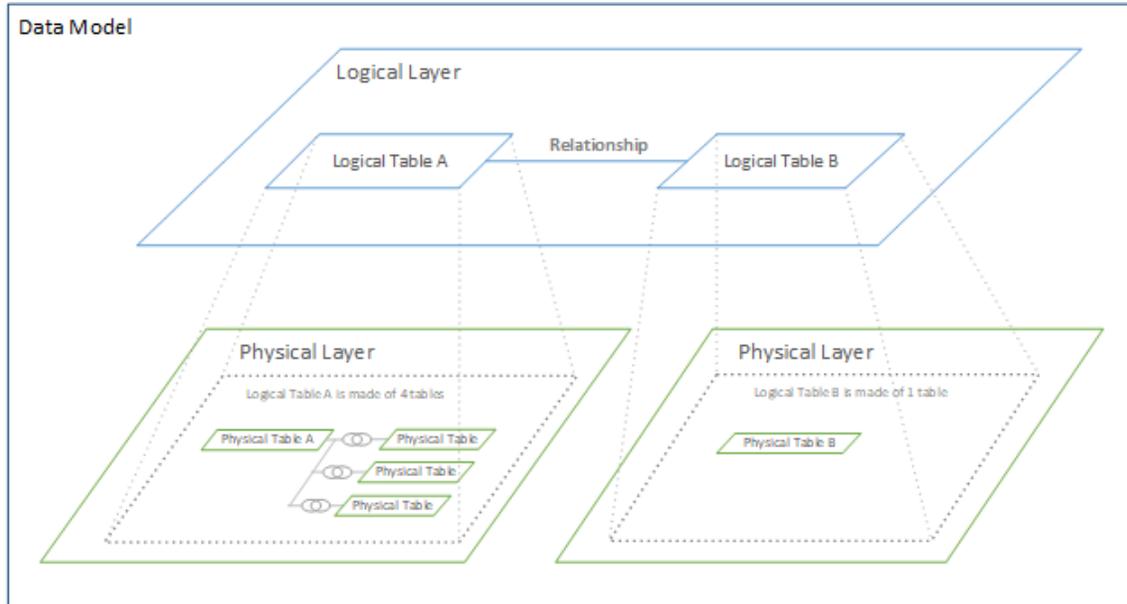
Every top-level, logical table contains at least one physical table. Open a logical table to view, edit, or create joins between its physical tables. Right-click a logical table, and then click **Open**. Or, just double-click the table to open it.

Tableau Server on Linux Administrator Guide



When you create a data source, it has two layers. The top-level layer is the logical layer of the data source. You combine data between tables in the logical layer using relationships.

The next layer is the physical layer of the data source. You combine data between tables at the physical layer using joins. For more information, see [Logical and physical tables in the data model](#).



Optimize Relationship Queries Using Performance Options

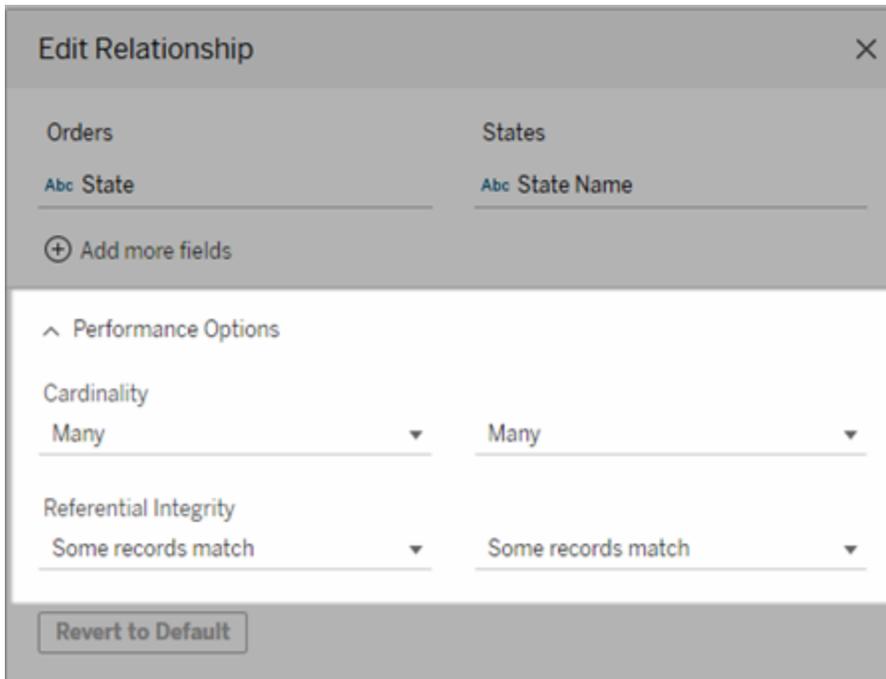
Performance Options are optional settings that define the cardinality (uniqueness) and referential integrity (matching records) between the two tables in a relationship. These settings help Tableau optimize queries during analysis.

- **If you aren't sure what to choose**, use the recommended default settings indicated by Tableau. Using the defaults is safe and will automatically generate correct aggregations and joins during analysis. If you don't know the cardinality or referential integrity, you don't need to change these settings.
- **If you know the shape of your data**, you can optionally change these settings to represent uniqueness and matching between the records in the two tables.

In many analytical scenarios, using the default settings for a relationship will give you all of the data you need for analysis. In some scenarios, you might want to adjust the Performance Options settings to describe your data more accurately. For more details about using relationships to combine and analyze data, see [Relate Your Data](#) and this Tableau blog post: [Relationships, part 1: Introducing new data modeling in Tableau](#).

Note: In web authoring in Tableau Server or Tableau Online: Your workbook must be connected to a live data source for you to be able to edit relationships and performance options in the Data Source page. You cannot edit the data model of a published data source.

What the Cardinality and Referential Integrity settings mean



Cardinality options

Cardinality settings determine if Tableau aggregates table data before or after automatically joining the data during analysis.

- Select **Many** if the field values aren't unique, or you don't know. Tableau will aggregate the relevant data before forming joins during analysis.
- Select **One** if field values are unique. During analysis, the relevant data will be joined before aggregation. Setting this option correctly optimizes queries in the workbook when the field values in the relationship are unique. However, selecting **One** when field values aren't unique can result in duplicate aggregate values being shown in the view.

Note: Selecting **One** treats records as if each key value is unique and there is at most only one row with a null value.

Referential Integrity options

Referential Integrity settings determine the type of join used to get the dimension values for a measure during analysis.

- Select **Some Records Match** if some values in the field don't have a match in the other table, or you don't know. During analysis, Tableau uses outer joins to get dimensions values for a measure. All measure values will be shown in the view, even unmatched measures.
- Select **All Records Match** if values in the field are guaranteed to have a match in the other table. This setting generates fewer and simpler joins during analysis, and optimizes queries. You might see inconsistent results during analysis (unmatched values removed or missing in view) if there are unmatched values in this table.

Notes: Selecting **All Records Match** treats records as if no null values exist in the fields used for the relationship. During analysis, Tableau will use inner joins to get dimension values for a measure. By default, Tableau will never join null keys.

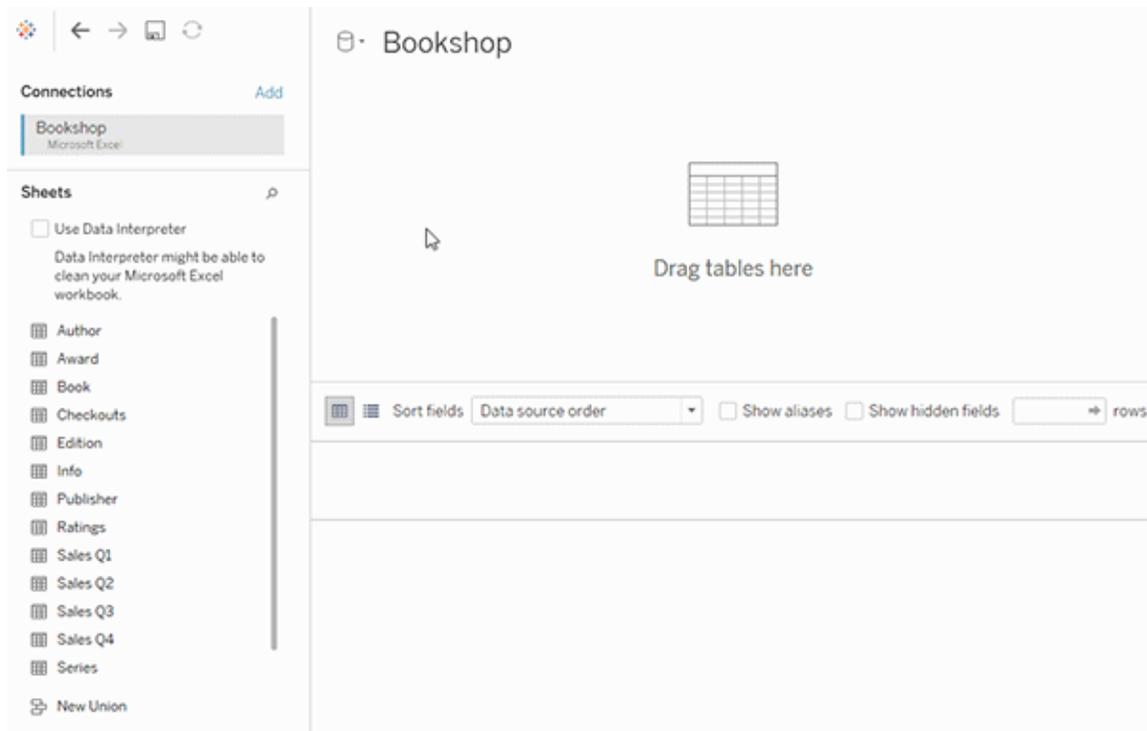
For more information about Cardinality and Referential integrity as concepts, see [Cardinality and Referential Integrity](#).

Where did joins go?

You can still specify joins between tables in the physical layer of a data source. Double-click a logical table to go to the join canvas.

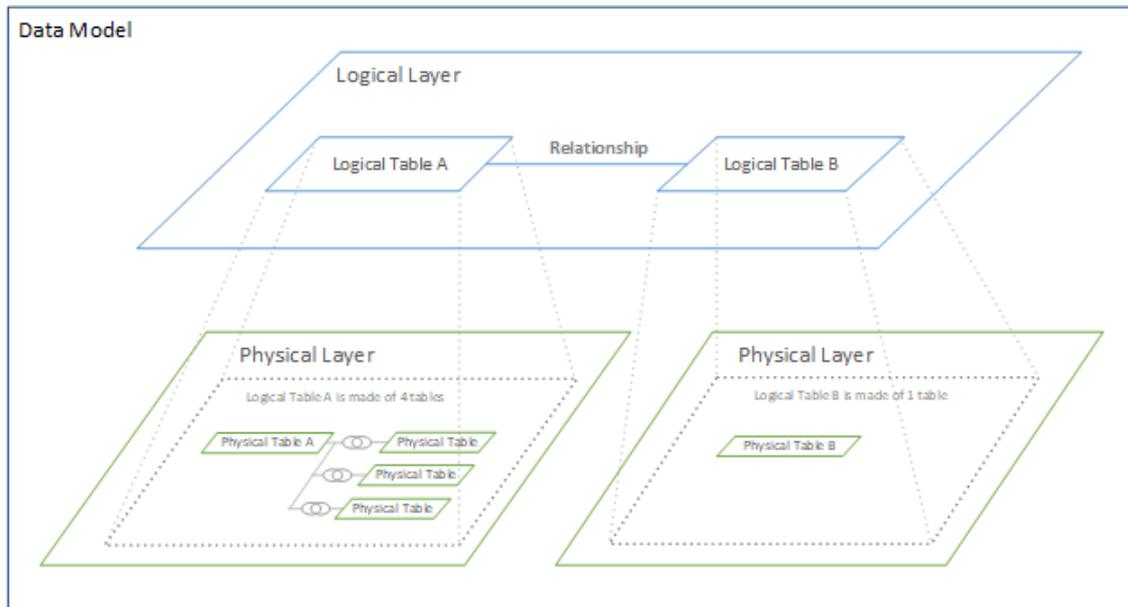
Every top-level, logical table contains at least one physical table. Open a logical table to view, edit, or create joins between its physical tables. Right-click a logical table, and then click **Open**. Or, just double-click the table to open it.

Tableau Server on Linux Administrator Guide



When you create a data source, it has two layers. The top-level layer is the logical layer of the data source. You combine data between tables in the logical layer using relationships.

The next layer is the physical layer of the data source. You combine data between tables at the physical layer using joins. For more information, see [Logical and physical tables in the data model](#).



Tips on using Performance Options

If you know the shape of your data, you can use the optional settings in Performance Options to establish the cardinality of the tables to each other (one-to-one, one-to-many, many-to-many) and indicate referential integrity (values from one table will always have match in the other table).

Instead of thinking of the settings in Performance Options as “yes” and “no”, think of them as “yes” and “I don’t know”. If you are sure that a table’s values are unique, select **One**. If you are sure that each record in one table matches one or more records in the other table, select **All Records Match**. Otherwise, leave the default settings as they are.

If you aren't sure about the shape of your data, use the default settings. When Tableau can't detect these settings in your data, the default settings are:

- Cardinality: Many-to-Many
- Referential integrity: Some Records Match

If Tableau detects key relationships or referential integrity in your data, those settings will be used and indicated as "detected".

To reapply the default settings, click **Revert to Default**.

Terms defined

Cardinality refers to the uniqueness of data contained in a field (column) or combination of fields. When the tables you want to analyze contain many rows of data, queries can be slow (and performance of the overall data source is affected) so we recommend choosing a method for combining data based on the cardinality of the related columns between tables.

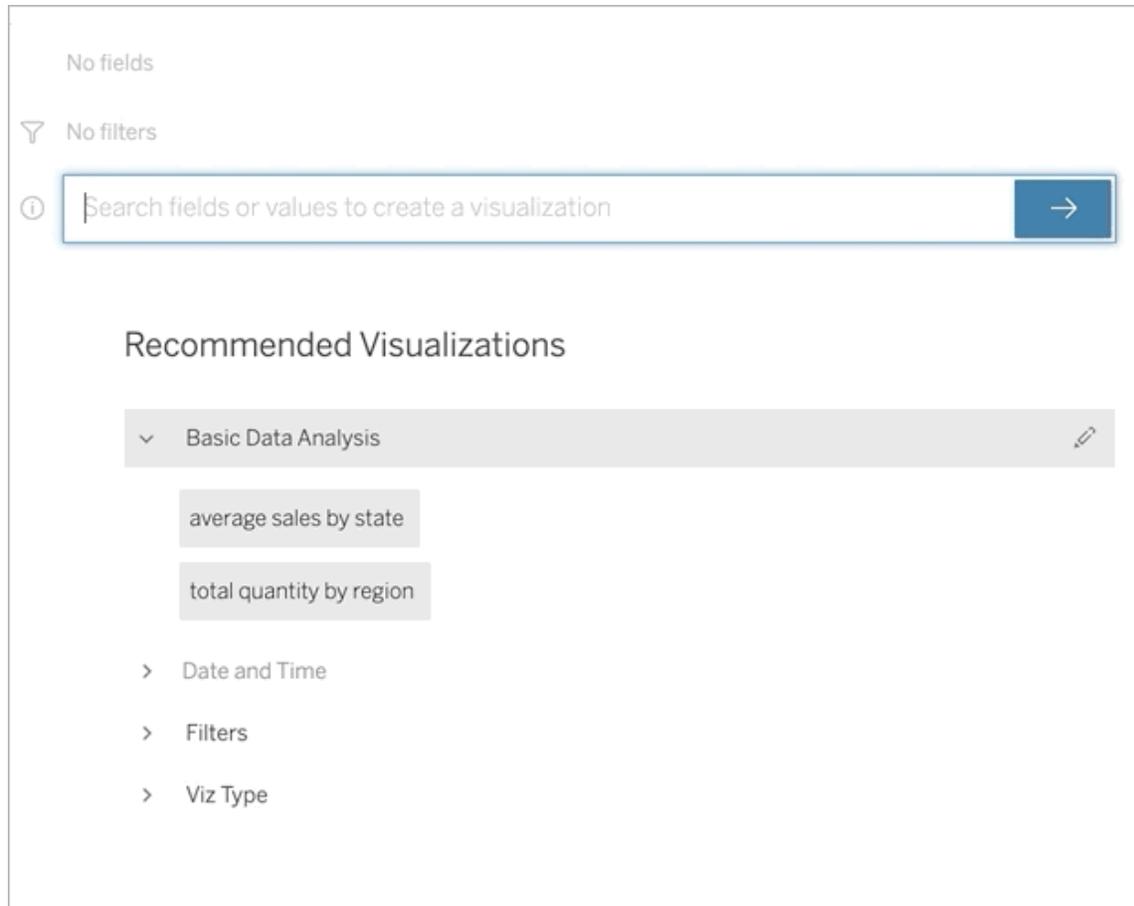
- Low cardinality: When related columns have a lot of repeated data. For example, a table called Products might contain a Category column that contains three values: Furniture, Office Supplies, and Technology.
- High cardinality: When related columns have highly unique data. For example, a table called Orders might contain an Order ID column that contains a unique value for every order of product.

Referential integrity means that one table will always have a matching row in the other table. For example, a Sales table will always have a matching row in the Product Catalog table.

Automatically Build Views with Ask Data

Ask Data lets you type a question in common language and instantly get a response right in Tableau. Answers come in the form of automatic data visualizations, with no need to manually drag-and-drop fields or understand the nuances of your data's structure.

Ask Data lets you ask sophisticated questions naturally, with support for key analytical concepts like time series and spatial analysis, and an understanding of conversational phrases like "last year", "earliest", and "most popular".



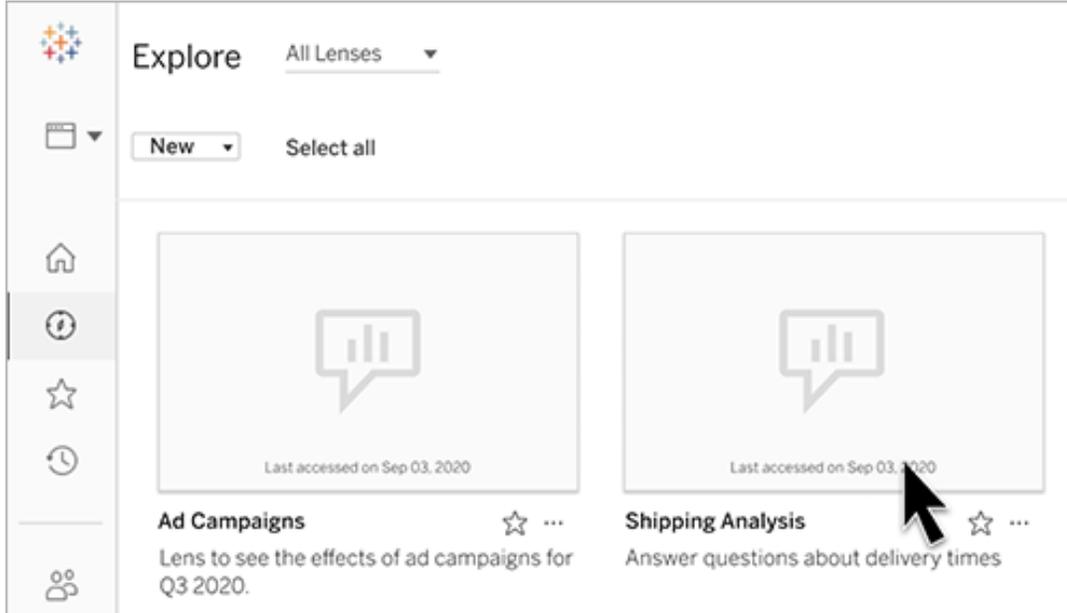
Navigating to Ask Data lenses

Before you can query a data source with Ask Data, **a Tableau author must first create a lens** that specifies the subset of data fields the lens uses.

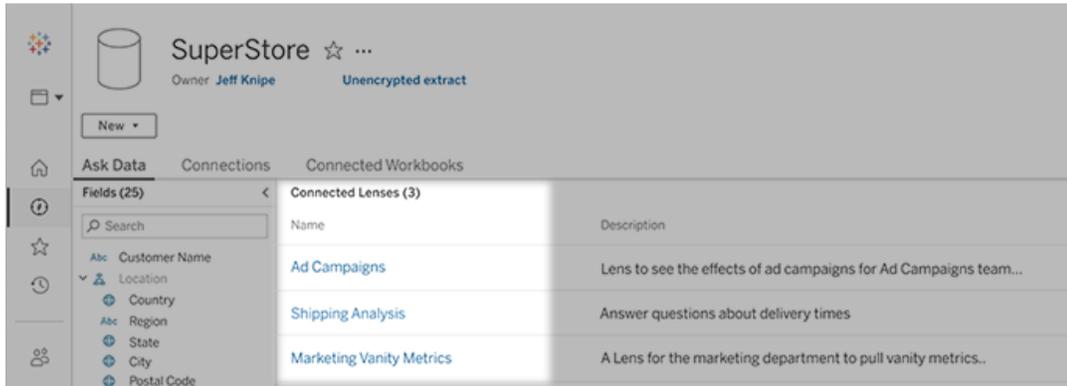
In Tableau, here are all the places where you can access an Ask Data lens:

Tableau Server on Linux Administrator Guide

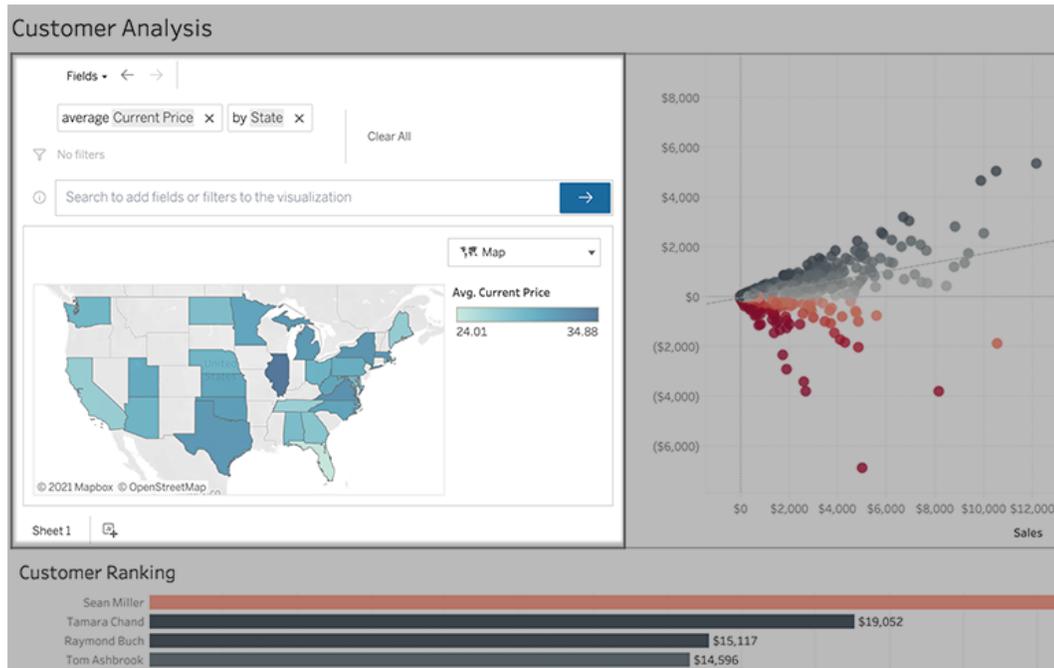
- On the All Lenses page at the top level of your Tableau Online or Tableau Server site.



- On the Ask Data tab for a data source for which lenses have been created.



- In an Ask Data object on a dashboard.

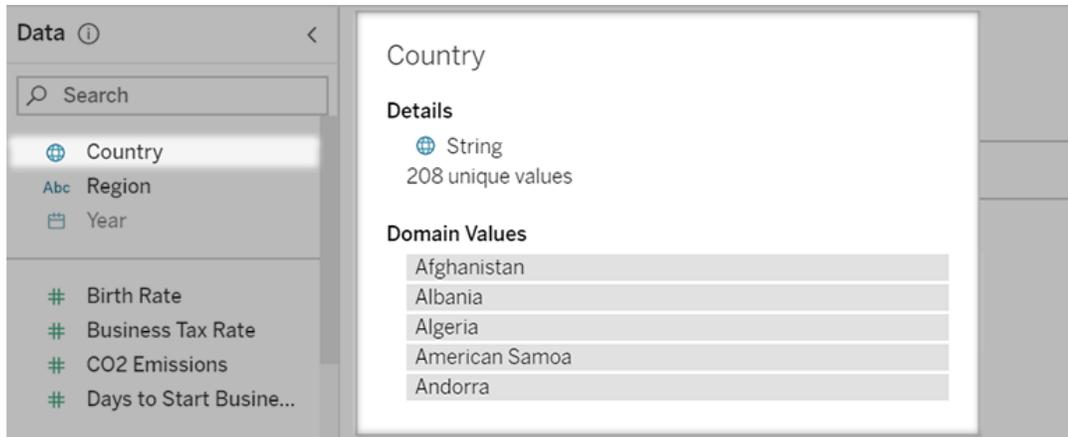


Ask Data from a lens page or dashboard object

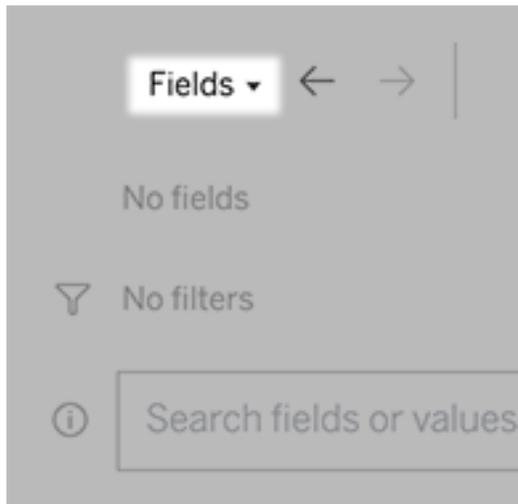
1. Navigate to a lens via the All Lenses page for your Tableau site, the Ask Data tab for a data source, or an Ask Data object on a dashboard.
2. (Optional) Under **Recommended Visualizations**, click an entry to quickly see visualizations the lens author has created for your organization.

If a recommendation addresses your data analysis needs, you can stop here. Otherwise, continue on to quickly create your own question.

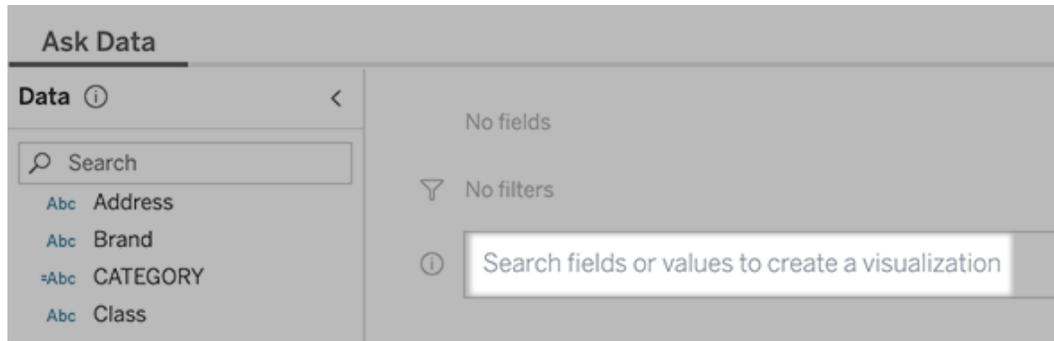
3. (Optional) In the Data pane at left, briefly hover over each field to learn more about the data it contains.



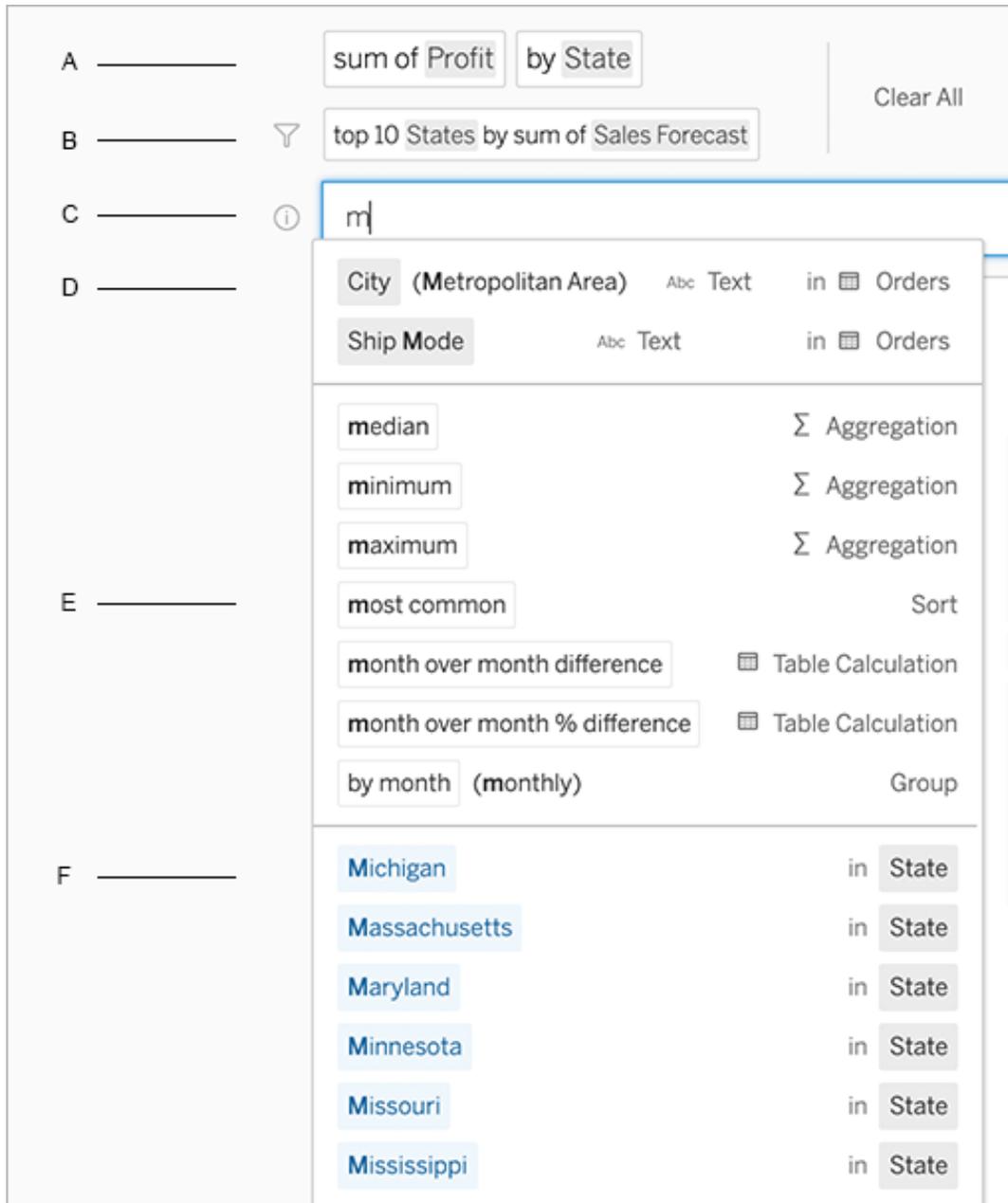
In a narrower dashboard object, the Data pane may be hidden, but you can see the same information by clicking the **Fields** drop-down menu.



4. In the box reading “**Search fields or values to create a visualization**”, enter your query about the data.



5. As you type, Ask Data searches data fields, functions, and string values and displays results in a drop-down list. Click items in the list to add them to your current entry, shown above the search box. To automatically create a viz using the current entry, press **Enter** at any time.



Modifying a query by searching for fields and analytical functions

A. Current entry B. Current filters C. Search box D. Returned fields E. Returned analytical functions F. Returned field values

Tip: (Tableau Online only) Hover over elements in the text box to see how they're incorporated in the interpretation above. Words that aren't used are grayed out, helping you rephrase your query in a way that's clearer to Ask Data.



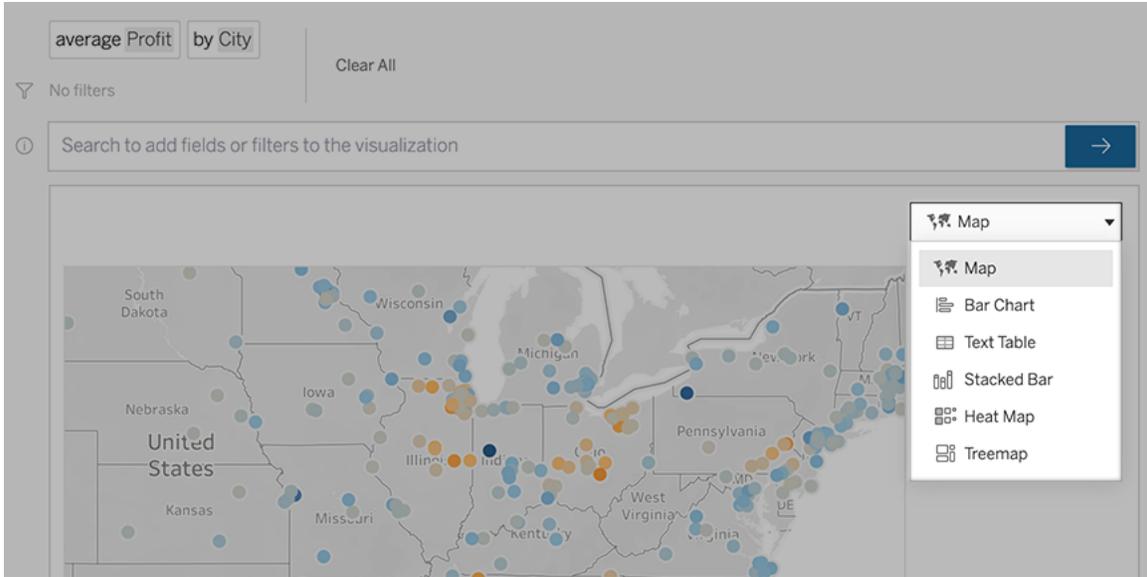
Rephrase your question

You can rephrase questions by clicking options, data fields, and filters in the user interface.

Change the viz type

If the default viz doesn't fully reveal your data, click the menu at upper right, and choose from these supported viz types:

- Bar Chart
- Gantt Bar
- Heat Map
- Histogram
- Line Chart
- Map
- Pie Chart
- Scatter
- Stacked Bar Chart
- Text Table
- Treemap

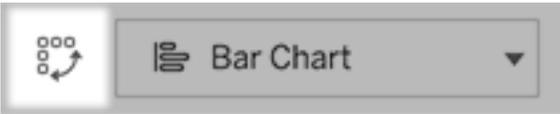


Note: To automatically create certain viz types, Ask Data sometimes adds fields like “Number of Records” to your entries.

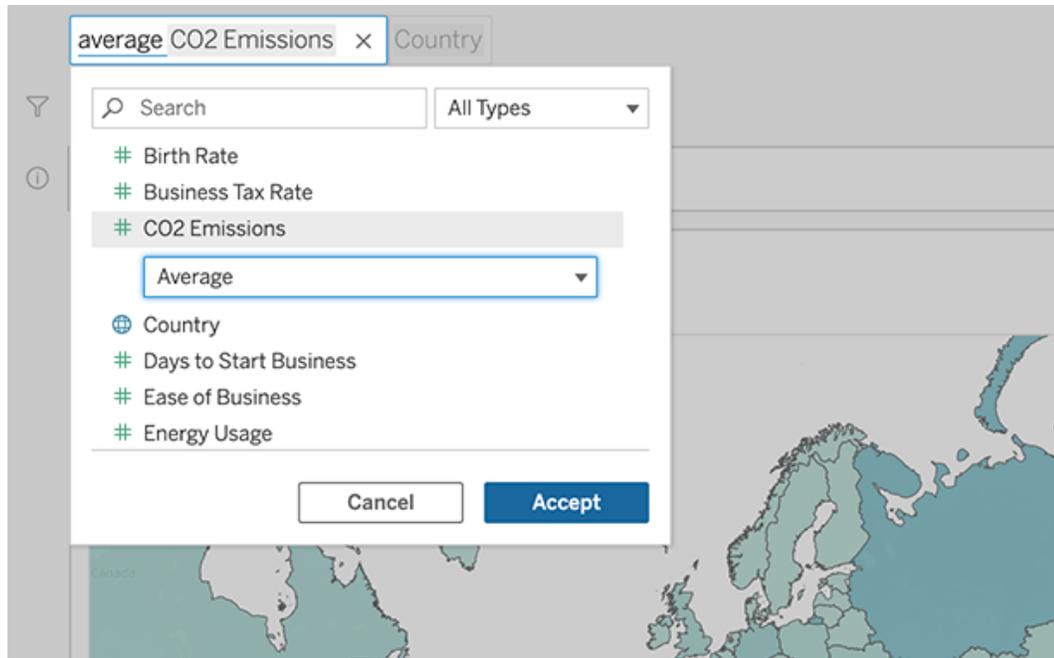
Change fields, filters, and displayed data

Ask Data gives you several ways to fine-tune how field values are displayed.

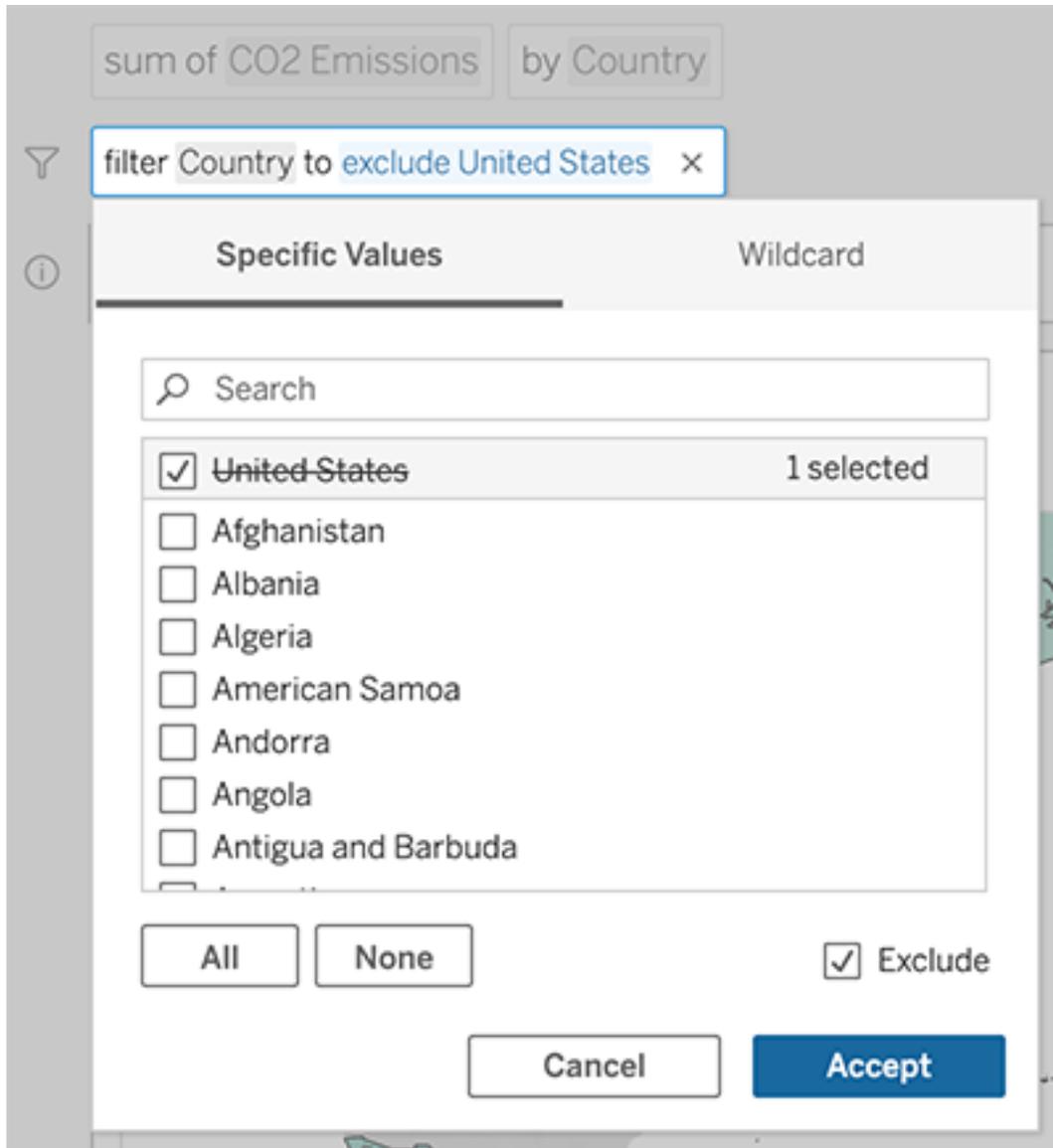
- To switch the fields used for the vertical and horizontal axes, click the Swap Axes button  to the left of the viz selection menu:



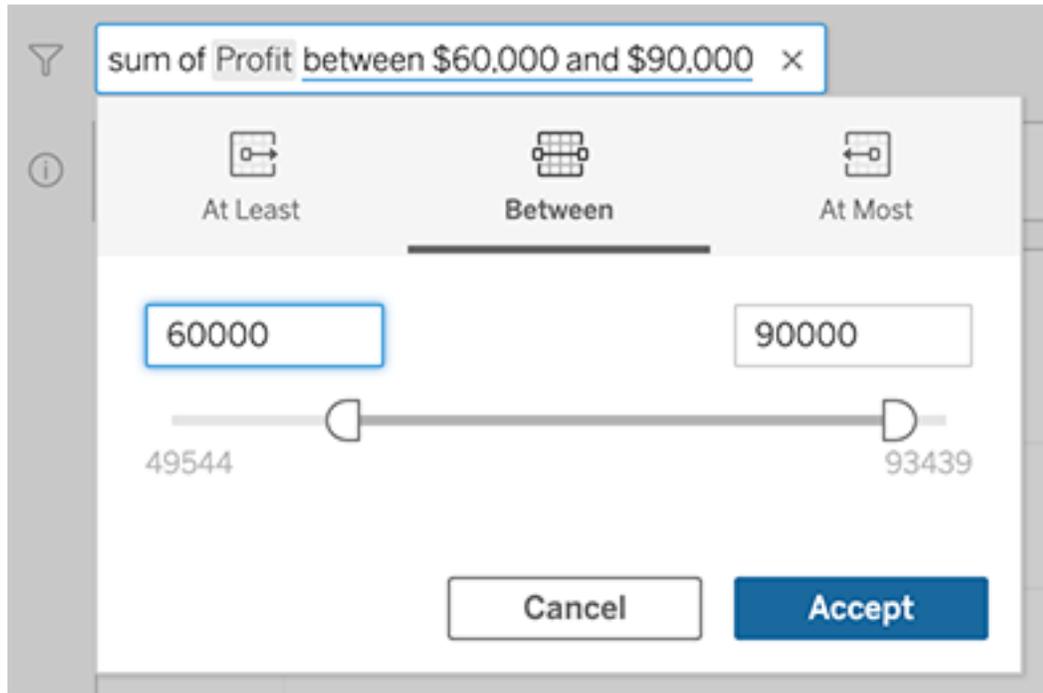
- To change a field or its aggregation or grouping type (for example, from sum to average), click the field name in the text box.



- For categorical filters, click values (like "exclude United States" in the example below) to change specific values or enter wildcard parameters.



- To adjust a numeric range, click words like “high”, “between”, or “cheap”.



- To delete a field or filter, hover over it and click the **X**.

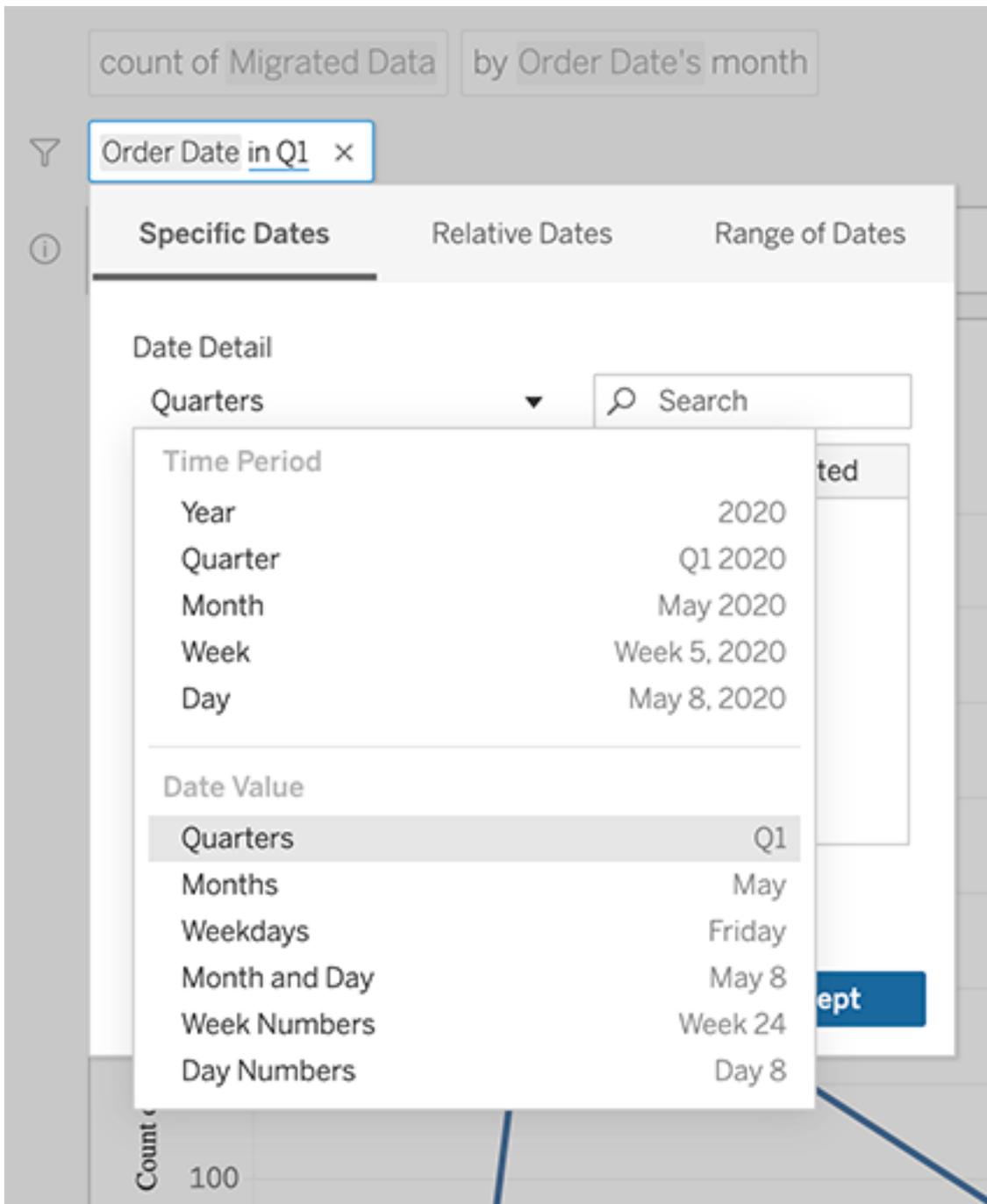
Adjust date filters

To adjust a date filter, click words like “in”, “last”, “previous”, or “between”. Then click one of the following:

- **Specific Dates** to enter a specific time period or date value
- **Relative Dates** to show a date range relative to the present day
- **Range of Dates** to enter specific start and end points

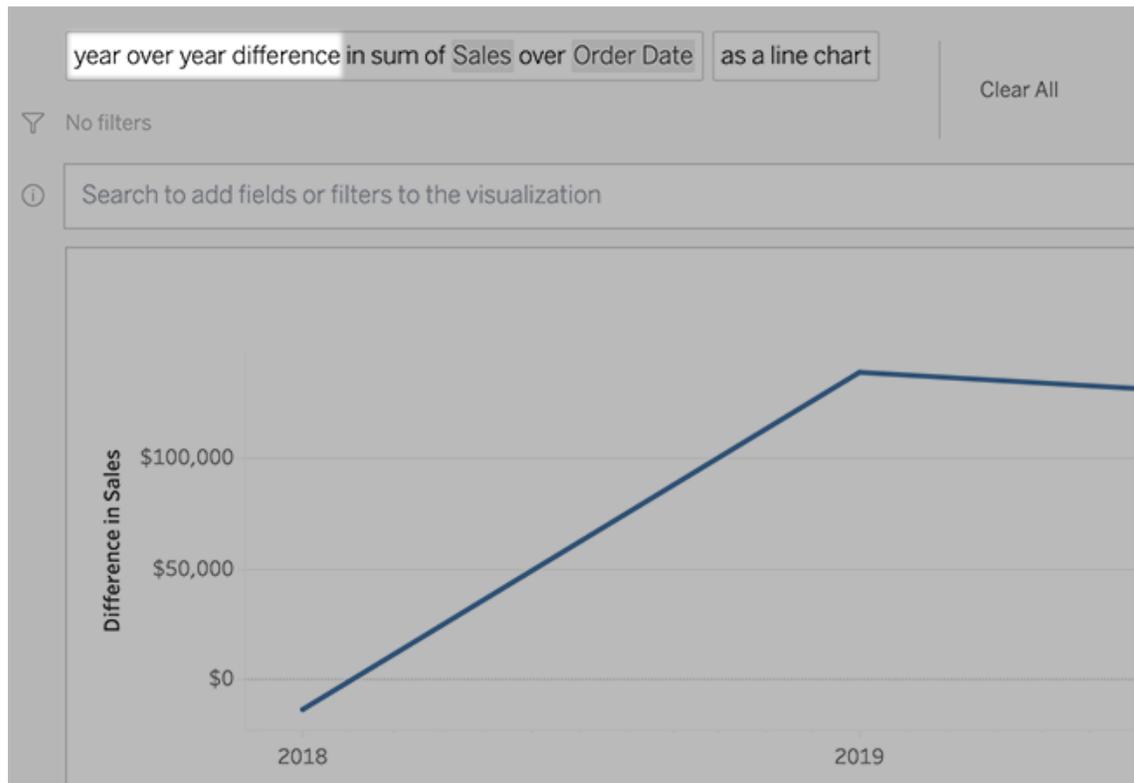
Specific Dates offers some unique options in the **Date Detail** menu:

- **Time Period** options show a single continuous date range
- **Date Value** options show ranges that can repeat in multiple time periods. For example, to see combined sales performance for Q1 across multiple years, under Date Value, you would choose Quarters.

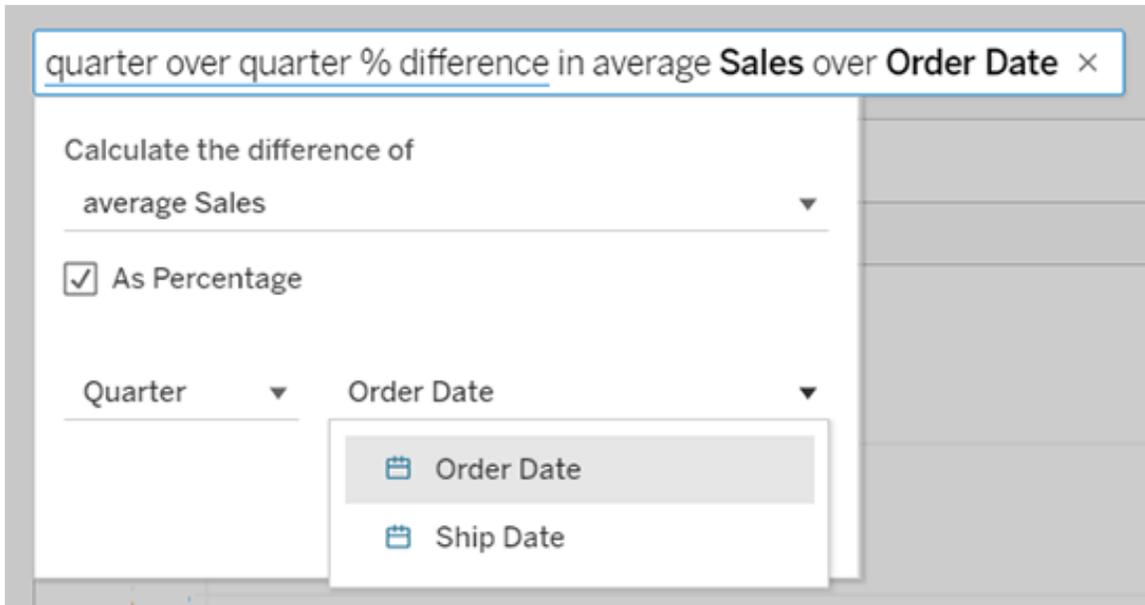


Compare differences over time

Ask Data lets you compare time periods with phrases like “year over year”, “quarter over quarter”, or similar comparisons by month, week, or day. The results appear as difference or percent difference table calculations in workbooks you save from Ask Data.



In the text box, click a difference calculation to choose other fields, aggregation methods, and time periods.

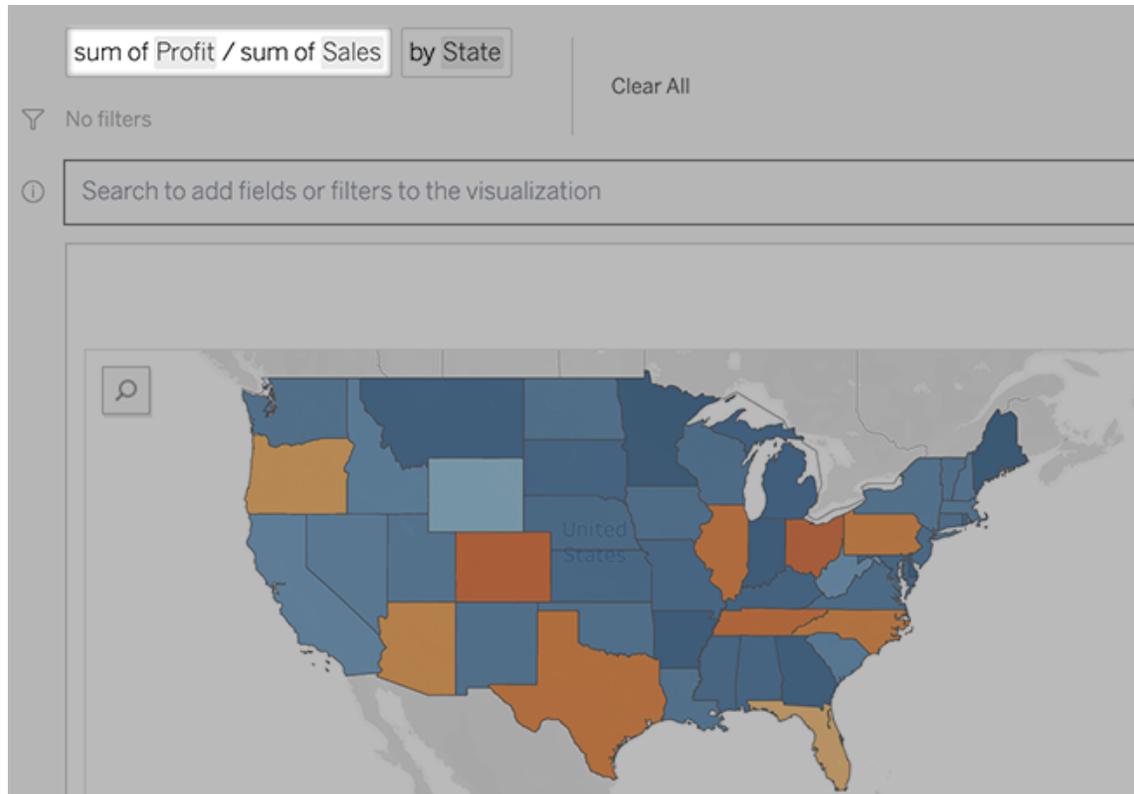


Apply simple calculations

Ask Data supports simple calculations between two measures, which you can apply using these symbols:

- + sums the measures
- produces the difference between them
- * multiplies
- / divides

In workbooks you save from Ask Data, these calculations don't become calculated fields but instead ad hoc calculations on the Columns, Rows, or Marks shelves.



Add sheets with other vizzes

To quickly create multiple different vizzes from a lens, add sheets in Ask Data.

At the bottom of the web page, do any of the following:

- Click the **Add Sheet** icon to the right of named sheets.

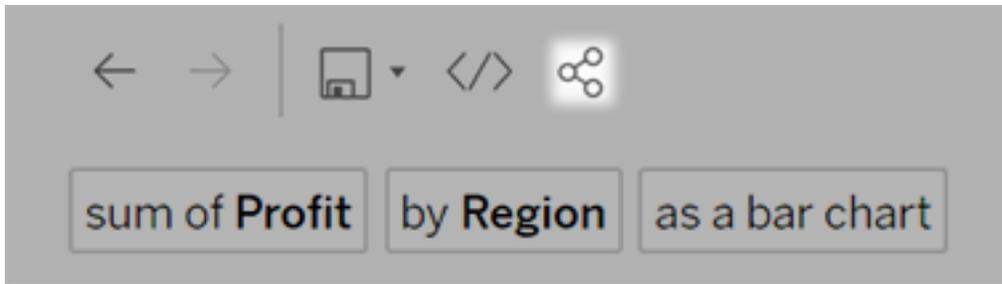


- Right-click a sheet name, and choose either **Duplicate** or **Delete**.

(To rename sheets from Ask Data, you need to save them in a new workbook.)

Share links to Ask Data vizzes

You can quickly share Ask Data vizzes. Simply click the link icon above the query box, and copy a link viewable by anyone with access to the lens.



Tips for success

As you structure questions for Ask Data, apply these tips to get better results.

- **Use keywords** — For example, instead of “I want to see all the countries that these airports are in”, try “by airport and country”.
- **Use exact wording for field names and values** — For example, if your lens includes Airport Code, Airport Name, and Airport Region fields, specify those by name.
- **See a ranked list** — Ask Data maps terms like “best” and “worst” to Top 1 and Bottom 1, respectively. If you want to see broader rankings, use “high” and “low” instead. For example, enter “houses with low sale prices”.
- **Surround unusually long values with quotation marks** — To analyze long field values that contain line returns, tabs, more than five words, or more than 50 characters, surround them with quotation marks. To improve performance, Ask Data doesn't index fields of that length.

Create Lenses that Focus Ask Data for Specific Audiences

Most people don't need information from an entire data source but instead want data visualizations relevant to their job function, like sales, marketing, or support. To optimize Ask Data for different audiences like these, Tableau authors create separate Ask Data *lenses*, which query a selected subset of fields. For the selected fields, authors can specify synonyms for field names and values, reflecting terms the lens audience uses in common language (for example, "SF" for "San Francisco"). Lens authors then customize the

recommended visualizations that appear below the Ask Data query box, which provide answers to users with a single click.

Note: Ask Data lenses can be created only for data sources published separately to a Tableau site, not data sources embedded in workbooks.

Create or configure a lens page on your Tableau site

On your Tableau site, each lens has a separate page where users can query Ask Data and authors can configure lens fields, synonyms, and suggested questions.

The screenshot shows a Tableau lens page titled "Shipping Analytics". At the top, it displays the owner "Jared", the modification date "Apr 26, 2021, 2:09 PM", and the data source "RetailSample Extract". Below this is a description: "A lens for ad-hoc analytics from the logistics team." The main section is titled "Ask Data" and is divided into two panes. The left pane, labeled "Data", contains a search bar and a list of fields: Address, Brand, CATEGORY, Class, Department, Distributioncenter, District, Division, and Geographic Location (which is expanded to show Region, State, City, and Zip). The right pane shows "No fields" and "No filters" selected. Below this is a search box with the placeholder text "Search fields or values to create a visualization". Underneath is a section titled "Recommended Visualizations" with a dropdown menu for "FAQs" and two suggested visualizations: "Prices by category" and "Shipping Costs over time".

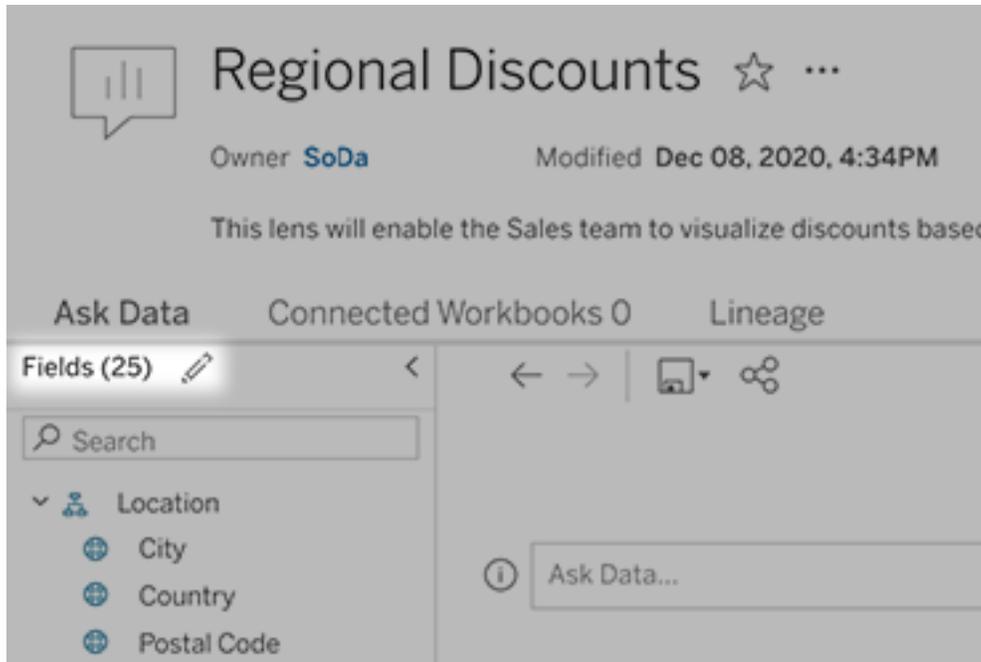
A lens page on a Tableau site

1. To create a lens page on your Tableau site, go to a data source page, and choose **New > Ask Data Lens**.

Tableau Server on Linux Administrator Guide

To configure an existing lens, go to the lens page on your site. (From an Ask Data object in a dashboard, you can click the pop-up menu in the upper corner and choose **Go to Lens Page**.)

2. If you're creating a new lens, enter a name, description, and project location, and then click **Publish Lens**.
3. At the top of the Fields pane at left, click the pencil icon. Then select the relevant fields for lens users, and click **Save**.



4. At left, hover over individual fields, and click the pencil icon to edit either of the following:
 - Common synonyms for field names and values that lens users may enter in their queries.

- (Tableau Online only) Descriptions that appear when users hover over fields



5. To edit recommended visualizations below the query box so they address common queries from lens users, do the following:

- To edit a section title, click the pencil icon  to the right of the title.
- To add a recommendation to a section, click the plus sign (+) to the right of the title. Then enter Text to Display and the related query in the Matching Result field.

Tip: Ask Data creates a viz for the matching result when users either click the displayed text or enter it as a query.

- To change an existing recommendation, hover over it and click the pencil icon . To remove a question, click the X.

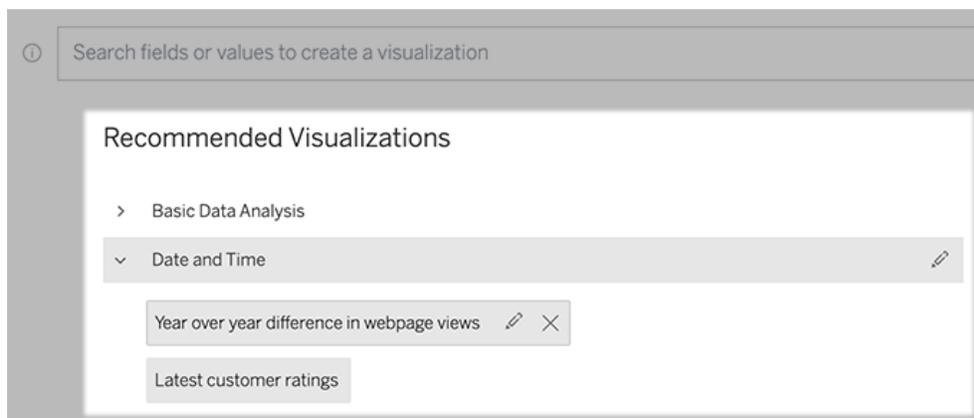


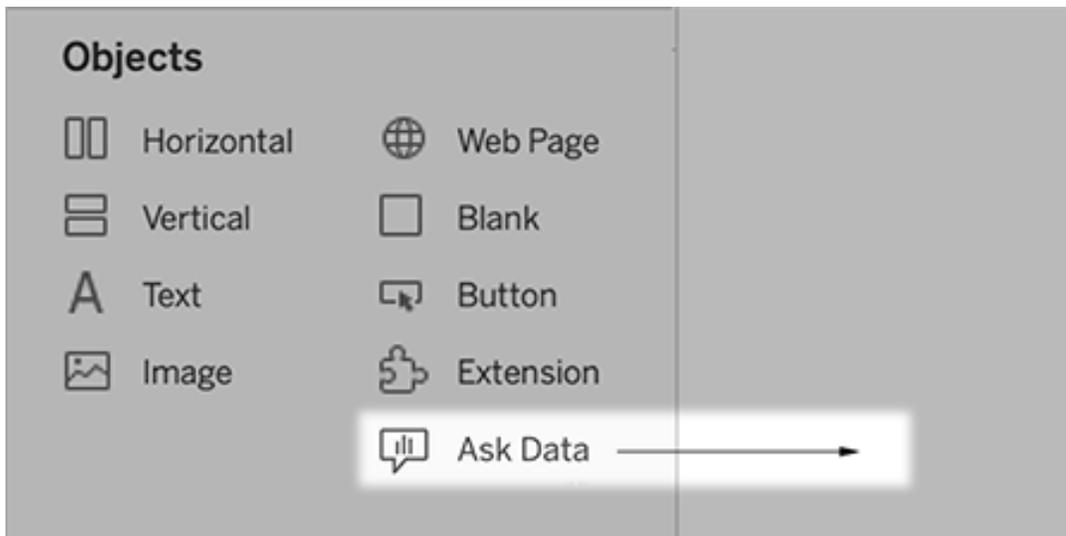
Tableau Server on Linux Administrator Guide

Add an Ask Data lens to a dashboard

On a dashboard, you can add an Ask Data object that lets users query a published data source via a lens on your Tableau site.

1. While editing a dashboard in Tableau Online, or Tableau Server, drag the Ask Data object to the canvas.

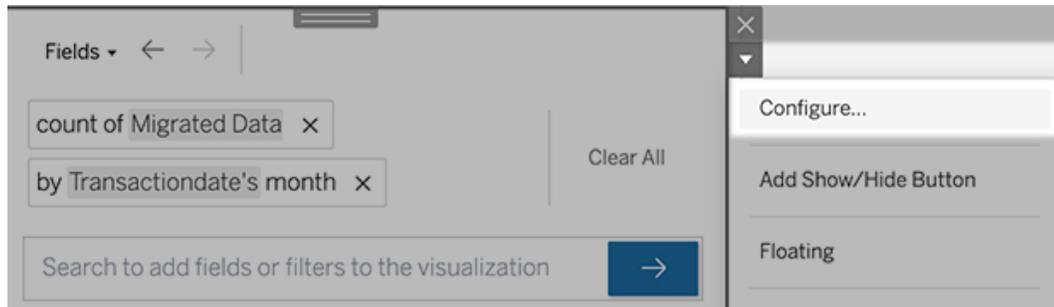
Note: In Tableau Desktop, you can also drag an Ask Data object to the canvas for placement purposes. But to select a lens, you will need publish to Tableau Online or Tableau Server and edit the object there.



2. Select a published data source previously connected to the workbook.
 3. To use an existing lens, select it, and click **Use Lens**. Then skip remaining steps.
- Or, to create a new lens, do one of the following:
- If there are no lenses for the data source, click **Go to Data Source Page**.
 - If lenses already exist, click the data source name at the bottom of the dialog.
4. Complete the steps in Create or configure a lens page on your Tableau site.
 5. After you finish creating the lens, return to the Lens object in your dashboard, and click **Refresh**. Then select the new lens, and click **Use Lens**.

Apply a different lens to an Ask Data dashboard object

1. From the pop-up menu at the top of the object, choose **Configure**.



2. Go to Add an Ask Data lens to a dashboard, and repeat steps 2 onward.

Change a lens name, description, or project location

1. Navigate to the lens page on Tableau Online or Tableau Server.
2. To the right of the lens name at the top of the page, click the three dots (...), and choose **Edit Workbook**.
3. Click **Edit Lens Details**.

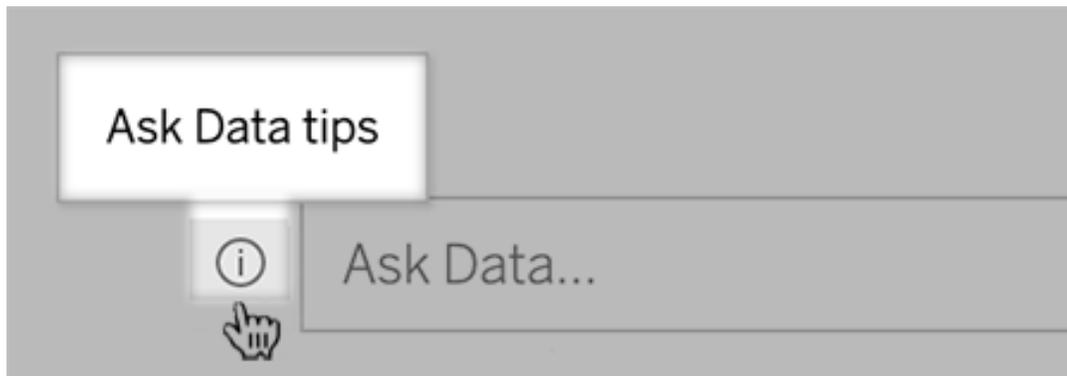
See how people use Ask Data with a lens

For data source owners and lens authors, Ask Data provides a dashboard that reveals the most popular queries and fields, the number of visualization results that users clicked, and other helpful information. Filters let you narrow data down to specific users and time ranges. These stats help you further optimize a lens to increase the success of your users.

Note: If you use Tableau Server, you can access this data in the Tableau Server Repository to create custom dashboards.

Tableau Server on Linux Administrator Guide

1. In Tableau Server or Tableau Online, navigate to a lens page.
2. To the left of the Ask Data text box, click the "i".



3. In the lower-left corner of the tips dialog, click **Usage Analytics**.

Permissions for publishing and viewing lenses

For Ask Data objects in dashboards, no change to permissions should be required: by default, existing workbook authors can create lenses, and existing dashboard audiences can view them. But for reference, here's a detailed outline of required lens permissions for both dashboards and direct access via a data source page.

To create and publish a lens, a user needs:

- The Creator or Explorer user role
- Lens Creation permission for the data source (inherited by default from the Connect permission)
- Write permission for the parent project to which the lens is published

To access and interact with a published lens, a user needs:

- The Viewer role or above
- Connect permission for the data source
- View permission for the lens

Note: By default, lens permissions like View reflect a project's permissions for workbooks. If Tableau administrators want to change default lens permissions, they can do so either individually for each project, or in bulk using the permissions API.

Disable or Enable Ask Data for a Site

Ask Data is enabled for sites by default, but Tableau administrators may disable it.

1. Go to the **General** site settings.
2. (Tableau Server only) In the **Web Authoring** section, select **Let users edit workbooks in their browser**.
3. In the **Availability of Ask Data** section, choose from these options:
 - **Enabled** enables creation of Ask Data lenses for all published data sources.
 - **Disabled** hides Ask Data throughout the site, while preserving information about previously created lenses so they can be restored if Ask Data is re-enabled.

Note: Beginning with version 2019.4.5, Tableau Server administrators can configure whether or not Ask Data is enabled by default.

Optimize Data for Ask Data

If you manage and publish data sources, here are some tips to help make users of Ask Data more successful. By spending a little extra time on this process, you'll open up data analysis to a wider range of people at your organization, helping them independently answer questions and gain deeper insights.

Optimize data in Ask Data

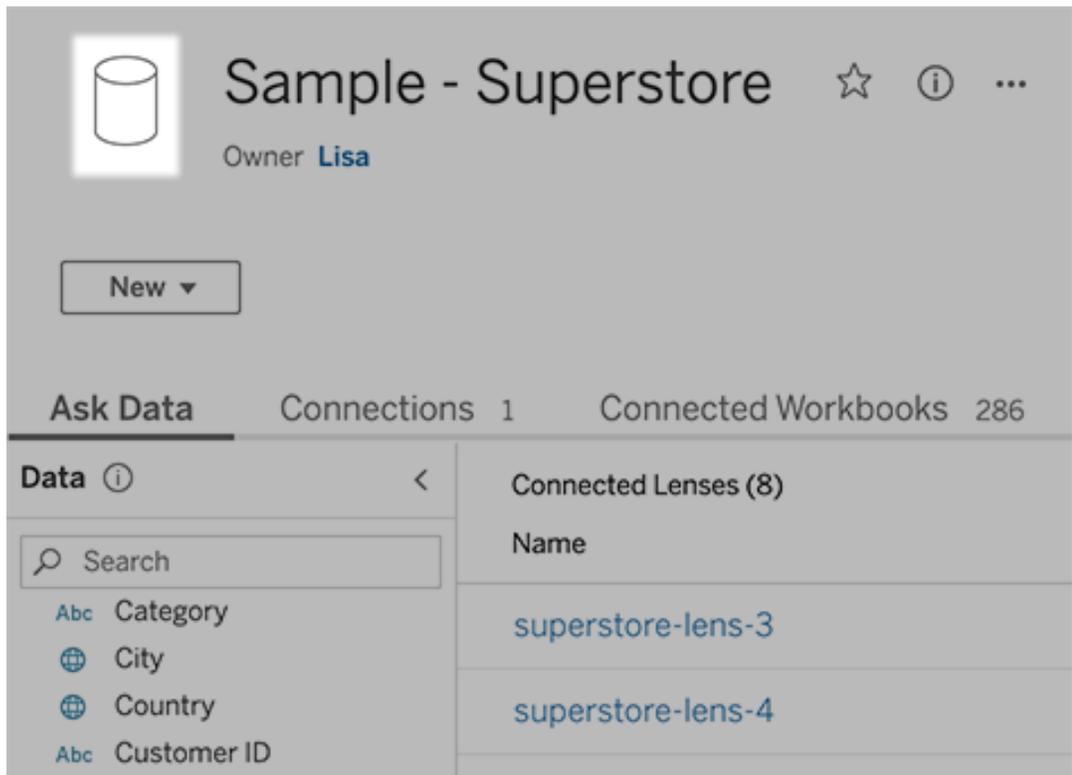
In the Data pane on the left of the Ask Data interface, data source owners can add synonyms for fields and exclude irrelevant values.

Changing settings at the data source or lens level

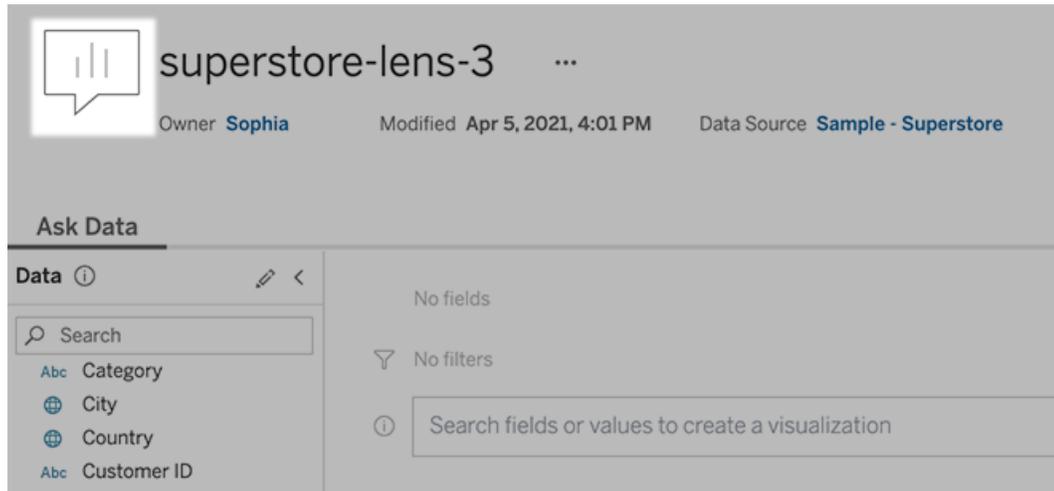
When changing settings in the Data pane for Ask Data, pay close attention to whether you're at the data source or lens level. (For more information, see [Create Lenses that Focus Ask Data for Specific Audiences.](#))

- At the data source level, you'll see the cylindrical data source icon in the upper left corner. Here, changes you make in the Data pane will apply by default to all subsequently created lenses.

Note: For extracts, two cylinders will appear.



- For an individual lens, you'll see the quotation icon in the upper left corner. Here, changes you make in the Data pane will apply to this lens alone.



Add synonyms for field names and values

People may not use the same terminology found in your data source, so data source owners and Tableau administrators can [add synonyms for specific data field names and values](#). Synonyms you enter are available throughout your organization, making data analysis quicker and easier for everyone.

Exclude values of specific fields from search results

To improve the usability of search results in Ask Data, you can exclude the values of specific fields from indexing. Though Ask Data doesn't add non-indexed values to search results, the values still appear in visualization results when relevant. For example, if you don't index values from a "Product" field because they add unnecessary detail to search results, Ask Data can still display values such as "iPhone 12" in resulting data visualizations. And users can manually add non-indexed values to queries by surrounding them with quotation marks (for example, "Sales for Product containing "iPhone 12"").

Note: This field-level setting is ignored if [the value indexing setting for the data source](#) is set to Disabled. Field names and related synonyms are always indexed.

Tableau Server on Linux Administrator Guide

1. Go to the Ask Data tab for a data source or individual lens.
2. Hover over a data source field at left, and click the **Edit Field Details** icon (the pencil).



3. Deselect **Index field values**.

Either click the text box that appears to reindex the data source now, or let it reindex based on its regular indexing schedule.

Optimize data sources

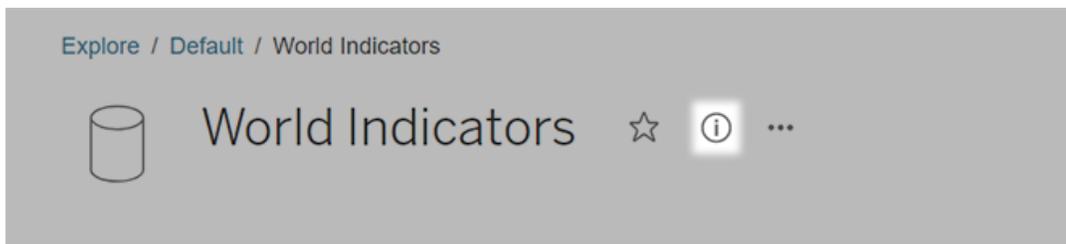
To create the best experience for Ask Data users, optimize the original data source.

Note: Ask Data doesn't support multidimensional cube data sources, or non-relational data sources like Google Analytics.

Optimize indexing for Ask Data

Data source owners can change how often field values are indexed for Ask Data, optimizing system performance.

1. At the top of a data source page, click the Details icon:

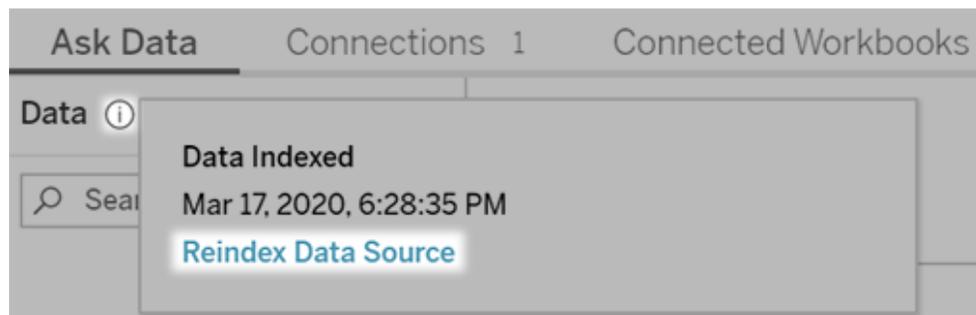


2. In the Ask Data section, click **Edit**.

3. Choose an indexing option for field values:

- **Automatic** checks for changes every 24 hours and analyzes the data source if it is live, has had an extract refreshed, or has been republished. Choose this option for a data source frequently used with Ask Data, so it will be ready before users query it.
- **Manual** analyzes the data source only when Tableau creators manually trigger indexing on the data source page. Choose this option if the data source changes frequently but users query it with Ask Data only occasionally.

To trigger manual indexing, go to the data source page, click the circled “i” in the Data pane at left, and then click **Reindex Data Source**.



- **Disabled** analyzes only field names, not values.

4. Click **Save**

Use data extracts for faster performance

For improved performance and support for large data sets, use Ask Data with published extracts rather than live data sources. For more information, see [Create an extract](#).

Ensure that users can access the data source

To use Ask Data, users must have permission to connect to the individual data source. If a data source has row-level permissions, those permissions also apply to Ask Data, which won't recognize secure values or make related statistical recommendations.

Tableau Server on Linux Administrator Guide

Be aware of unsupported data source features

Ask Data supports all Tableau data source features except the following. If your data source contains these, Ask Data users won't be able to query related fields.

- Sets
- Combined fields
- Parameters
- Table calculations in calculated fields

Anticipate user questions

Anticipate the kinds of questions your users will ask, and then optimize your data source for those questions using these techniques:

- Clean and shape data in [Tableau Prep](#) or a similar tool.
- [Join data](#) to include all fields users may have questions about in one table, improving performance.
- Add [calculated fields](#) that answer common user questions.
- Create [bins with appropriate sizes](#) for quantitative variables that users are likely to want to see as a histogram or another binned form.

Simplify the data

To make data easier to understand by both users and Ask Data, simplify the data source as much as possible during the data prep process.

1. Remove any unnecessary fields to improve performance.
2. Give each field a unique and meaningful name.
 - For example, if there are five field names that start with “Sales ...”, better distinguish them so Ask Data can properly interpret the term “sales”.
 - Rename “Number of records” to something more meaningful. For example, use “Number of earthquakes” in a data source where each record is an earthquake.
 - Avoid field names that are numbers, dates, or boolean (“true” or “false”) values.
 - Avoid names which resemble analytical expressions such as “Sales in 2015” or “Average Products Sold”.
3. Create meaningful [aliases for field values](#), reflecting terms people would use in conversation.

Set appropriate field defaults

To help Ask Data analyze data correctly, ensure that default field settings reflect the content of each field.

- **Set data types** for text, time, date, geographic, and other values.
- **Assign the proper data role:** dimension or measure, continuous or discrete.
- For each measure, **assign appropriate default settings** in Tableau Desktop, such as color, sort order, number format (percentage, currency, etc.), and aggregation function. For example, SUM may be appropriate for “Sales”, but AVERAGE might be a better default for “Test Score”.

Tip: It's particularly helpful to set a default comment for each field, because these comments appear as informative descriptions when users hover over fields in Ask Data.

Create hierarchies for geographic and categorical fields

For time data, Tableau automatically creates hierarchies, which let users quickly drill up and down in vizzes (for example, from day to week to month). For geographic and categorical data, however, we recommend that you **create custom hierarchies** to help Ask Data produce visualizations that reflect the relationships between fields. Be aware that Ask Data won't show the hierarchies in the data pane.

Ask Data doesn't index hierarchy names, only names of fields within hierarchies. For example, if a geographic hierarchy named “Location” contains “Country” and “City” fields, users should enter “Country” and “City” in their questions for Ask Data.

Avoid field names that Ask Data uses for analytical functions

To let users ask questions about data in a conversational style, Ask Data reserves the following words for analytical functions. If any of these words appear as field names in your data, we recommend that you revise them.

Note: In addition to the terms listed below, Ask Data reserves all conjunctions, prepositions, and visualization types such as “Bar chart”.

Aggregation terms

- aggregate
- average
- avg
- maximum
- max
- mean
- median
- minimum
- min
- most
- sum
- count
- distinct count

Grouping terms

- group
- by
- per

Filtering and limiting terms

- filter
- limit
- at least
- at most
- between
- top
- bottom
- contains
- starts with

- ends with
- include

Sorting terms

- sort
- alphabetical
- ascending
- descending

Time terms

- date
- second
- minute
- hour
- day
- week
- month
- quarter
- year
- second over second (sos)
- minute over minute
- hour over hour (hoh)
- day over day (dod)
- week over week (wow)
- month over month (mom)
- quarter over quarter (qoq)
- year over year (yoy)
- today
- tomorrow
- yesterday
- next
- last

Qualifying terms

- best
- expensive

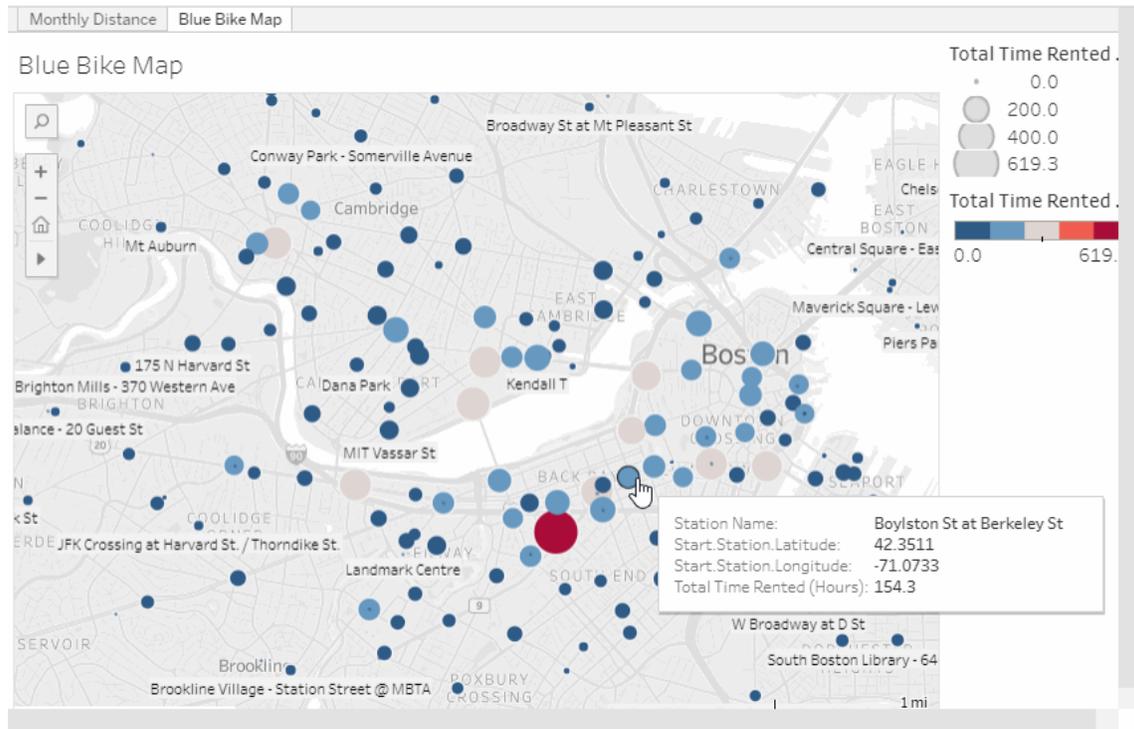
- most expensive
- cheap
- cheapest
- big
- small

Discover Insights Faster with Explain Data

Explain Data gives you a new window into your data. Use it to inspect, uncover, and dig deeper into the marks in a viz as you build, explore, and analyze your data. When you select a mark while editing a view and run Explain Data, Tableau builds statistical models and proposes possible explanations for the selected mark, including potentially related data from the data source that isn't used in the current view.

For information on running Explain Data and exploring explanations, see [Get Started with Explain Data](#).

Note: This topic describes how Explain Data works in Tableau 2021.2 and later versions. If you have a previous version of Tableau, read this topic in [version 2021.1 of Explain Data help](#).



Select a mark and run Explain Data from the tooltip menu. The Explain Data pane opens with possible explanations

As you build different views, use Explain Data as a jumping-off point to help you explore your data more deeply and ask better questions. For more information, see How Explain Data helps to augment your analysis. For information on what characteristics make a data source more interesting for use with Explain Data, see Requirements and Considerations for Using Explain Data.

Access to Explain Data for Authors vs. Viewers

Explain Data is always available to authors who can edit a workbook.

Authors can control whether Explain Data is available in viewing mode in published workbooks and which explanation types are displayed in the Explain Data pane.

When Explain Data is enabled by the author of a published workbook, all users can run Explain Data on a mark in viewing mode.

For information on controlling access to Explain Data, explanation types, and fields, see [Control Access to Explain Data](#).

How Explain Data helps to augment your analysis

Explain Data is a tool that uncovers and describes relationships in your data. It can't tell you what is causing the relationships or how to interpret the data. **You are the expert on your data.** Your domain knowledge and intuition are key in helping you decide what characteristics might be interesting to explore further using different views. For related information, see [How Explain Data Works and Requirements and Considerations for Using Explain Data](#).

For more information on how Explain Data works and how to use Explain Data to augment your analysis, see these Tableau Conference presentations:

- [From Analyst to Statistician: Explain Data in Practice \(1 hour\)](#)
- [Leveraging Explain Data \(45 minutes\)](#)
- [Explain Data Internals: Automated Bayesian Modeling \(35 minutes\)](#)
- [Machine Learning, Explainable AI, and Tableau \(45 minutes\)](#), [Session Materials](#)
- [Making Business More Bayesian \(45 minutes\)](#)

Get Started with Explain Data

Use Explain Data in your flow of analysis as you are exploring the marks in a viz. The best way to get started with Explain Data is to select a mark, run Explain Data, and start exploring explanations.

Use Explain Data

- [Run Explain Data on a mark](#)
- [Drill into explanations](#)
- [View analyzed fields](#)
- [Why am I seeing a Reselect button?](#)
- [Terms and concepts in explanations](#)
- [Explanation Types](#)

Author Workbooks and Control

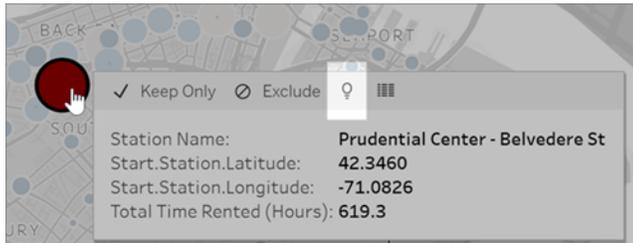
Access

- [Requirements and Considerations for Using Explain Data](#)
- [Change Explain Data Settings \(Authors-only\)](#)
- [Control Access to Explain Data](#)
- [How Explain Data Works](#)

Run Explain Data on a mark

The basic steps to run Explain Data are:

1. Select a mark in a viz.
2. Hover the cursor over the mark, and then click **Explain Data** in the tooltip menu.



The Explain Data pane opens with possible explanations for the value of the explained mark. Click different explanation names to expand the details and start exploring.

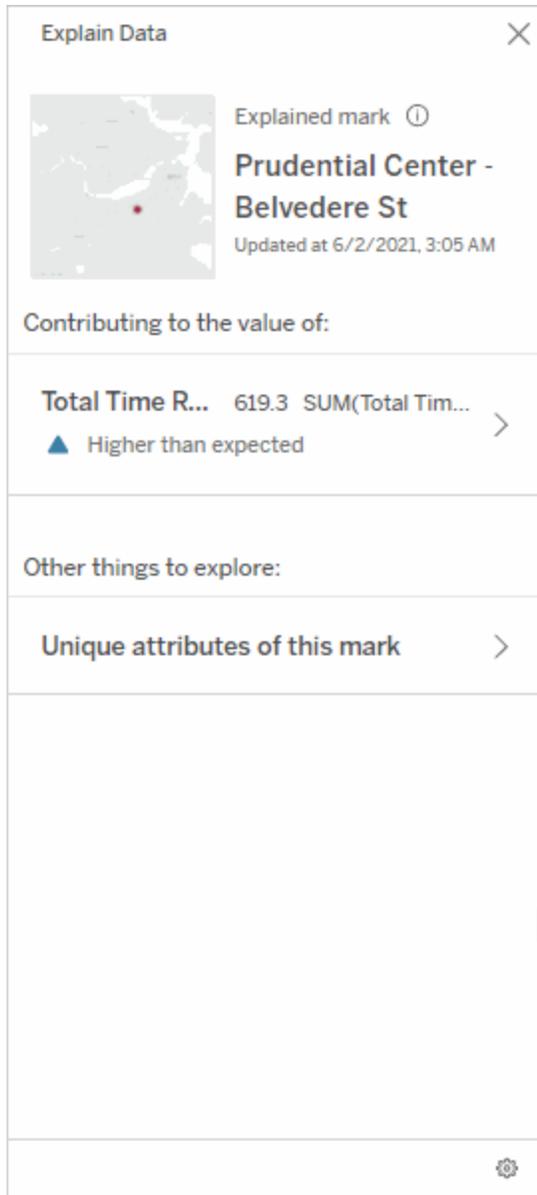
Tips for using Explain Data

- You must select a single mark—only one mark can be explained at a time.
- The view must contain marks that are aggregated using SUM, AVG, COUNT, COUNTD, or AGG (a calculated field).
- If Explain Data cannot analyze the type of mark selected, the Explain Data icon and context menu command will not be available. For more information, see [Situations where Explain Data is not available](#).
- The data you analyze must be drawn from a single, primary data source. Explain Data does not work with blended or cube data sources.
- For information on what characteristics make a data source more interesting for use with Explain Data, see Requirements and Considerations for Using Explain Data.

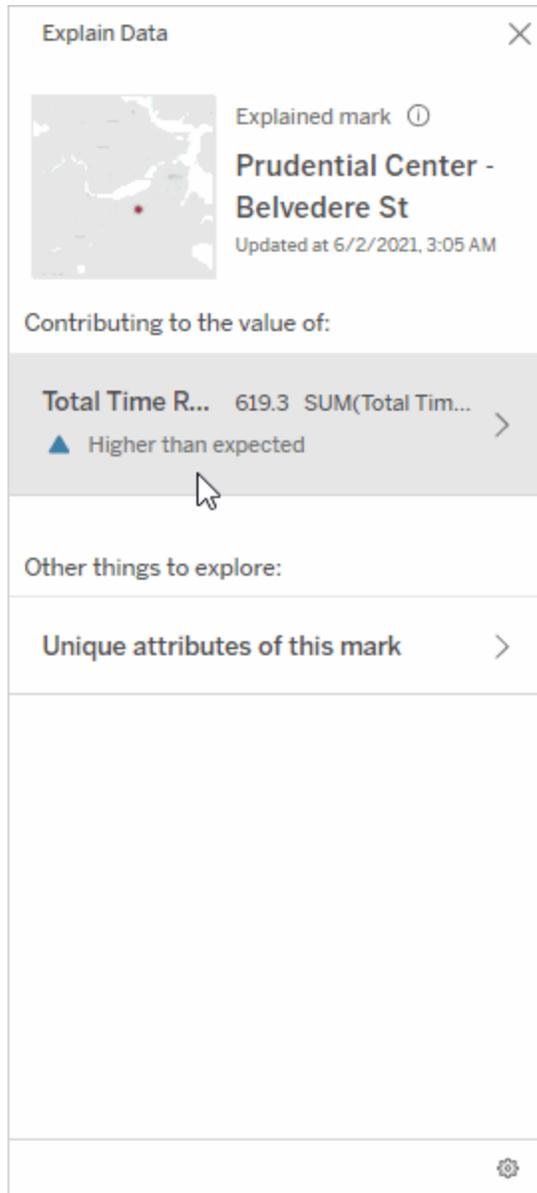
Drill into explanations

1. In the Explain Data pane, click an explanation name to see more details.

Click the arrows to expand or contract explanations.

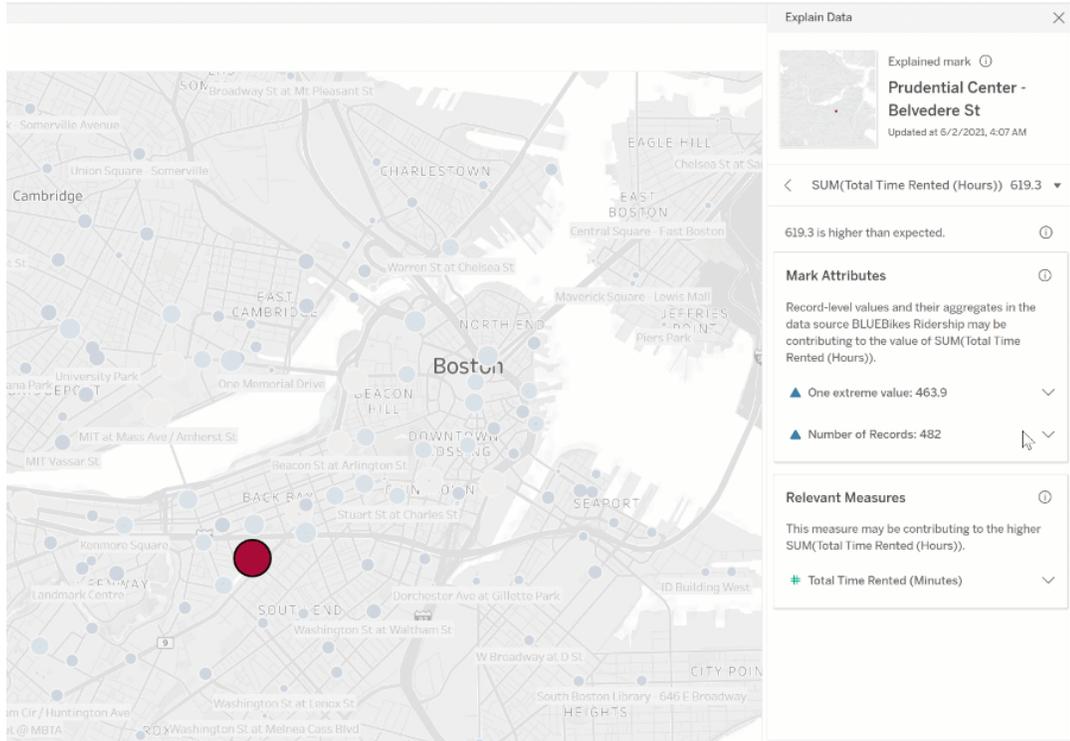


2. Scroll to see explanation details.



Some explanations offer different layers of details for further exploration

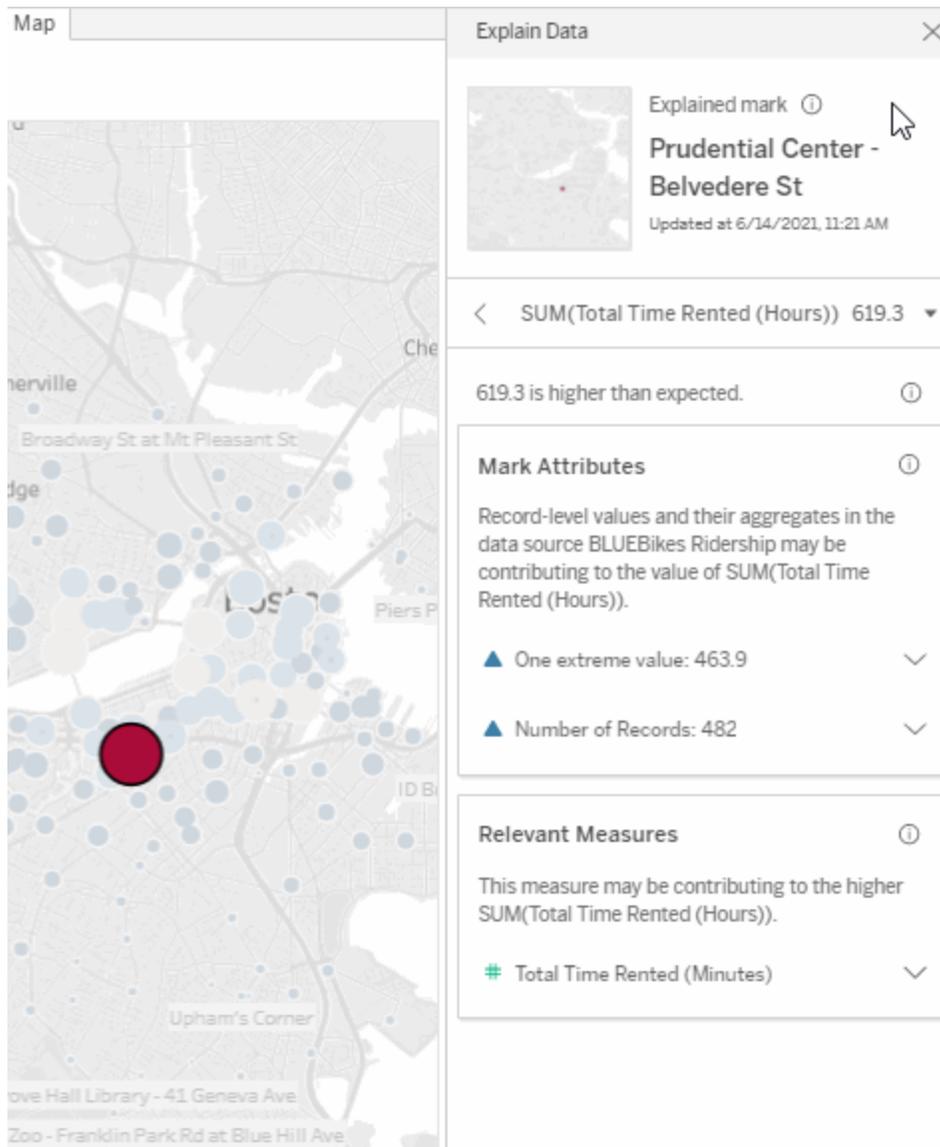
3. Hover over charts in the explanations to see more detail. Click the **Open**  icon to see a larger version of the visualization.



Creators or Explorers who open the view for editing can click the **Open**  icon to open the visualization as a new worksheet and explore the data further.

Note: Creators and Explorers who have editing permissions can also control Explain Data Settings. For more information, see [Control who can use Explain Data and what they can see](#).

4. Hover over a Help icon  to see tooltip help for an explanation. Click the Help icon  to keep the tooltip open. Click a **Learn More** link to open the related help topic.

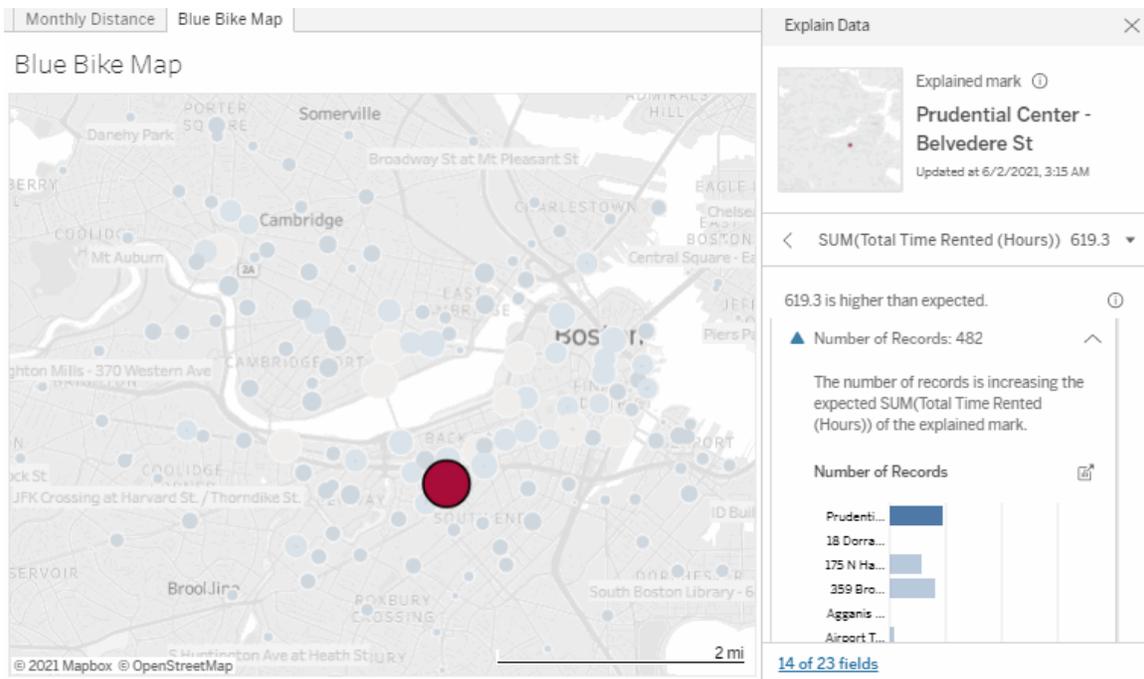


Why am I seeing a Reselect button?

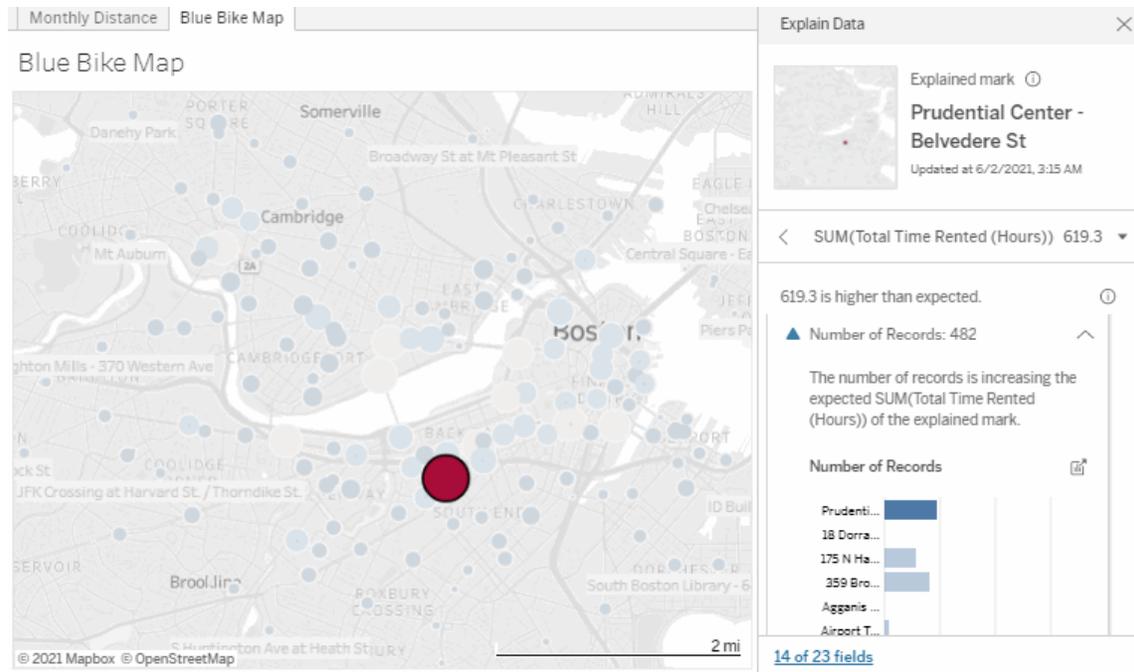
When you run Explain Data, the explanations that are presented in the Explain Data pane specifically apply to the mark you selected. If you click a different mark, deselect the explained mark, or navigate to a different sheet in the workbook, you might see a message asking if you want to reselect the same mark.

Tableau Server on Linux Administrator Guide

If you click **Reselect**, Tableau returns you to the original view and worksheet, and reselects the explained mark.



If you want to explore a new mark, ignore the Reselect message, click another mark, and then run Explain Data again.



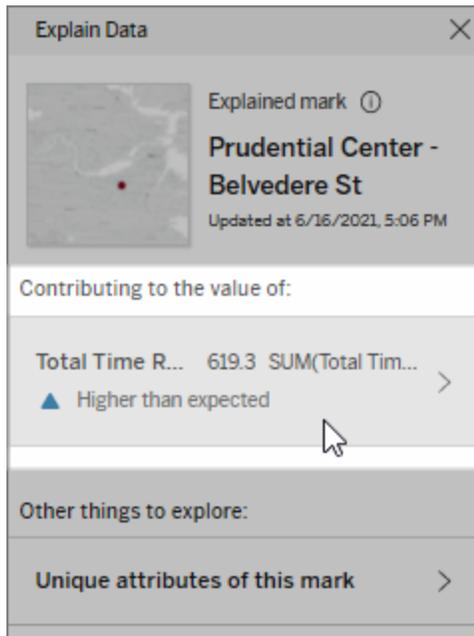
Explain Data will also display messages if the view has changed (such new fields or filters added or removed from the view), if the data source has changed, or if Explain Data settings have changed.

View analyzed fields

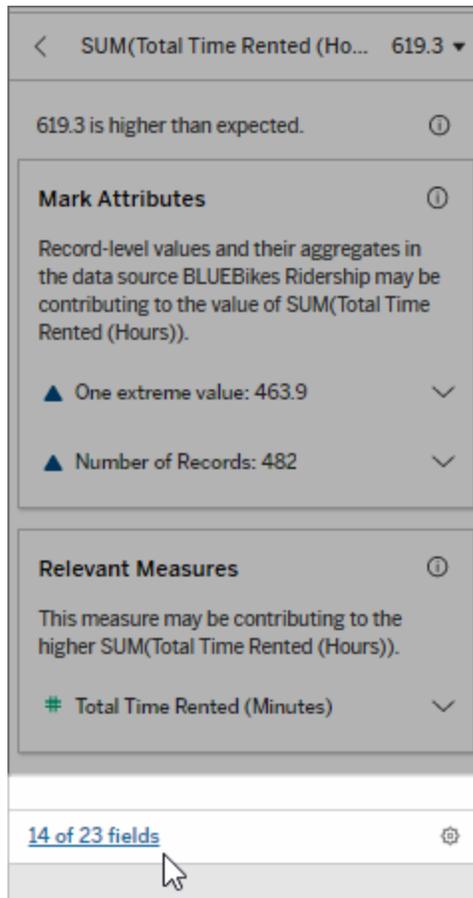
1. Run Explain Data on a mark.

Tableau Server on Linux Administrator Guide

2. In the Explain Data pane, under **Contributing to the value of**, click a measure name.



3. Click the *number-of-fields* link at the bottom of the pane.



Authors have the option to open Explain Data Settings to control which fields are included in the analysis. For more information, see [Change fields used for statistical analysis](#).

Terms and concepts in explanations

The following terms and concepts appear frequently in explanations. You may find it helpful to become acquainted with their meaning in the context of using Explain Data.

What is a mark?

A mark is a selectable data point that summarizes some underlying record values in your

data. A mark can be made of a single record or multiple records aggregated together. Marks in Tableau can be displayed in many different ways such as lines, shapes, bars, and cell text.

Tableau gets the records that make up the mark based on the intersection of the fields in the view.

The **explained mark** refers to a mark that you selected in the view that was analyzed by Explain Data.

For more information on marks, see [Marks](#).

What does expected mean?

The expected value for a mark is the median value in the expected range of values in the underlying data in your viz. The expected range is the range of values between the 15th and 85th percentile that the statistical model predicts for the explained mark. Tableau determines the expected range each time it runs a statistical analysis on a selected mark.

If an expected value summary says the mark is *lower than expected* or *higher than expected*, it means the aggregated mark value is outside the range of values that a statistical model is predicting for the mark. If an expected value summary says the mark is *slightly lower* or *slightly higher* than expected or *within the range of natural variation*, it means the aggregated mark value is within the range of predicted mark values, but is lower or higher than the median.

For more information, see [What is an expected range?](#)

What are dimensions and measures?

Each column name in a database is a field. For example, Product Name and Sales are each fields. In Tableau, fields like Product Name that categorize data are called dimensions; fields with quantifiable data like Sales are called measures. Tableau aggregates measures by

default when you drag them into a view.

Some explanations describe how the underlying record values and the aggregations of those values may be contributing to the value of the explained mark. Other explanations may mention the distribution of values across a dimension for the explained mark.

When you run Explain Data on mark, the analysis considers dimensions and measures in the data source that aren't represented in the view. These fields are referred to as unvisualized dimensions and unvisualized measures.

For more information on dimensions and measures, see [Dimensions and Measures](#).

What is an aggregate or aggregation?

An aggregate is a value that is a summary or total. Tableau automatically applies aggregations such as SUM or AVG whenever you drag a measure onto Rows, Columns, a Marks card option, or the view. For example, measures are displayed as SUM(Sales) or AVG(Sales) to indicate how the measure is being aggregated.

To use Explain Data, your visualization must use a measure that is aggregated with SUM, AVG, COUNT, COUNTD, or AGG.

For more information about aggregation, see [Data Aggregation in Tableau](#).

What is a record value?

A record is a row in a database table. A row contains values that correspond to each field. In this example, Category, Product Name, and Sales are fields (or columns). Furniture, Floor Lamp, and \$96 are the values.

Category	Product Name	Sales
Furniture	Floor Lamp	\$96.00

What is a distribution?

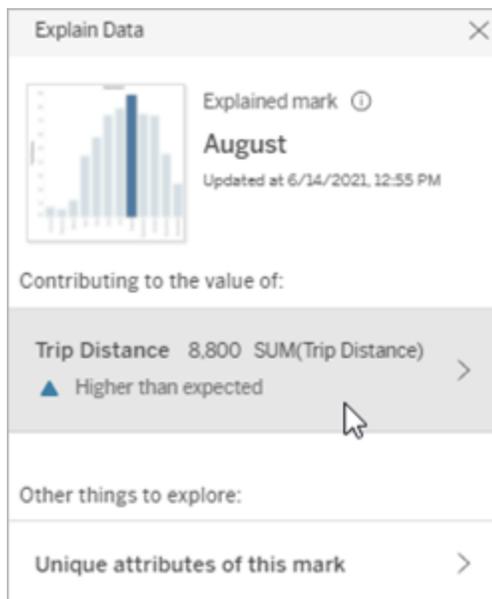
A distribution is a list of all the possible values (or intervals) of the data. It also indicates how often each value occurs (frequency of occurrence).

Explanation Types in Explain Data

Each time you select a new mark in a viz or dashboard and run Explain Data, Tableau runs a new statistical analysis considering that mark and the underlying data in the workbook. Possible explanations are displayed in expandable sections for the Explain Data pane. For information about how Explain Data analyzes and evaluates explanations, see How Explain Data Works.

Contributing to the value of the mark

The **Contributing to the value of** section of the Explain Data pane lists explanations for each measure that can be explained (referred to as *target measures*). Each explanation listed here describes a relationship with the values of the target measure that are tested on the explained mark. Use your real-world, practical understanding of the data to determine if the relationships found by Explain Data are meaningful and worth exploring.



In this example, Trip Distance is the target measure

Mark Attributes

These explanations describe how underlying records of the marks in the view may be contributing to the aggregated value of the measure being explained. Mark attributes can include **Extreme Values**, **Number of Records**, or the **Average Value** of the mark.

Note: For definitions of common terms used in explanations, see Terms and concepts in explanations.

Extreme Values

This explanation type indicates if one or more records have values that are significantly higher or lower than most records. If the explanation is supported by a model, it indicates the extreme value is affecting the target measure of the explained mark.

When a mark has extreme values, it doesn't automatically mean it has outliers or that you should exclude those records from the view. That choice is up to you depending on your analysis. The explanation is simply pointing out an extreme value in the mark. For example, it could reveal a mistyped value in a record where a banana cost 10 dollars instead of 10 cents. Or, it could reveal that a particular sales person had a great quarter.

Note: This explanation must be enabled by the author to be visible in viewing mode for a published workbook. For more information, see Control Access to Explain Data.

This explanation shows:

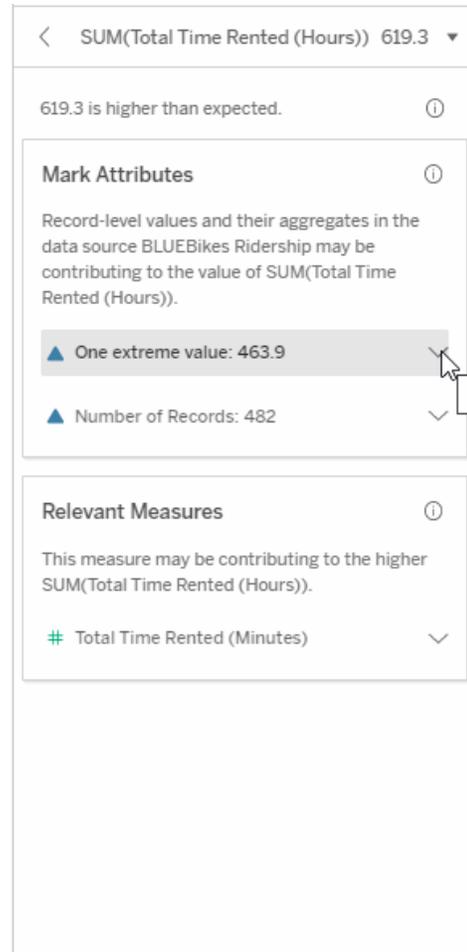
- The number of underlying records in the explained mark.
- The extreme value or values contributing to the value of the target measure.
- The distribution of values in the mark.
- The record details that correspond to each distribution value.

Exploration options:

- Hover over a circle in the chart to see its corresponding value.
- Click the left or right arrow below the details list to scroll through record details.
- If available, click **View Full Data**, and then click the **Full Data** tab to see all records in a table.
- Click the **Open**  icon to see a larger version of the visualization.

Next steps for analysis:

- If the number of records is low, examine these values compared to the extreme value.
- If the extreme value is significantly higher or lower than the other record values, exclude it and consider how it changes the value of the explained mark.
- When considering the data with and without the extreme value, use this as an opportunity to apply your practical



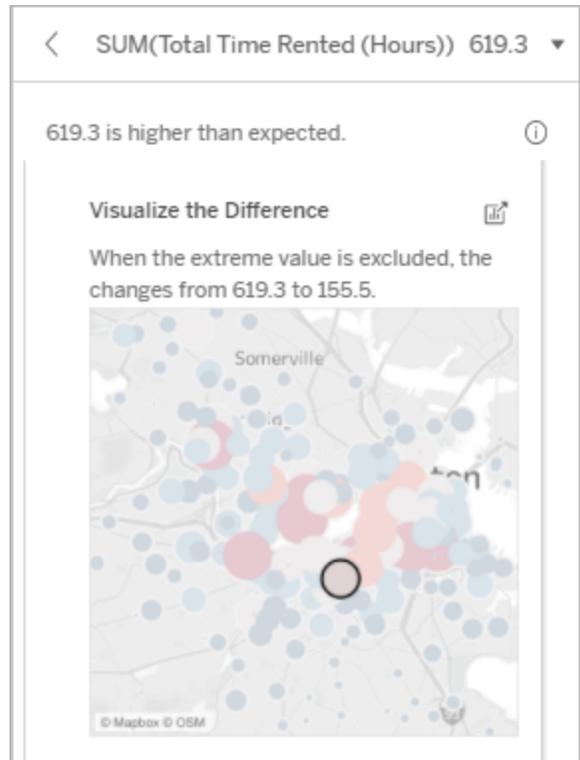
In this example, a single extreme value of 463 hours rented is contributing to the higher than expected sum of Total Time Rented of 613 hours.

A likely reason for this high value could be that someone forgot to dock the bike when they returned it. In this case, the author might want to exclude this value for future analysis.

knowledge about the data.

This section shows:

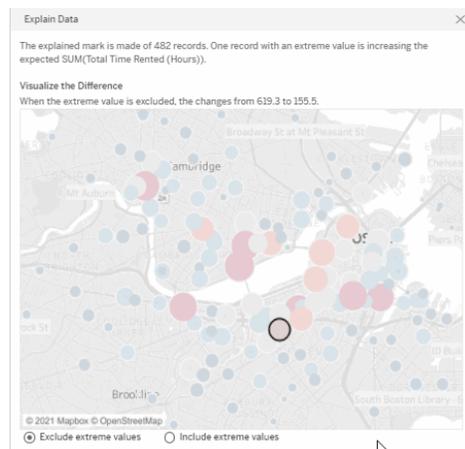
- How the explained mark value changes when the extreme value is excluded.



Exploration options:

- Click the **Open**  icon to see a larger version of the visualization.
- Explore the difference with and without the extreme value (or values).
- Authors can open the view as a new sheet and apply a filter to exclude the extreme value.

Next steps for analysis:



- If the extreme value is significantly higher or lower than the other record values, exclude it and see how it changes the value of the explained mark.
- When considering the data with and without the extreme value, use this as an opportunity to apply your practical knowledge about the data.

In this example, when the extreme value of 483 is excluded, the explained mark is no longer high compared to other marks in the view. Other marks now stand out. The author might want to explore the other marks to consider why these other locations have higher hours for bike rentals.

Number of Records

This explanation type describes when the count of the underlying records is correlated to the sum. The analysis found a relationship between the number of records that are being aggregated in a mark and the mark's actual value.

While this might seem obvious, this explanation type helps you explore whether the mark's value is being affected by the magnitude of the values in its records or simply because of the number of records in the explained mark.

This explanation shows:

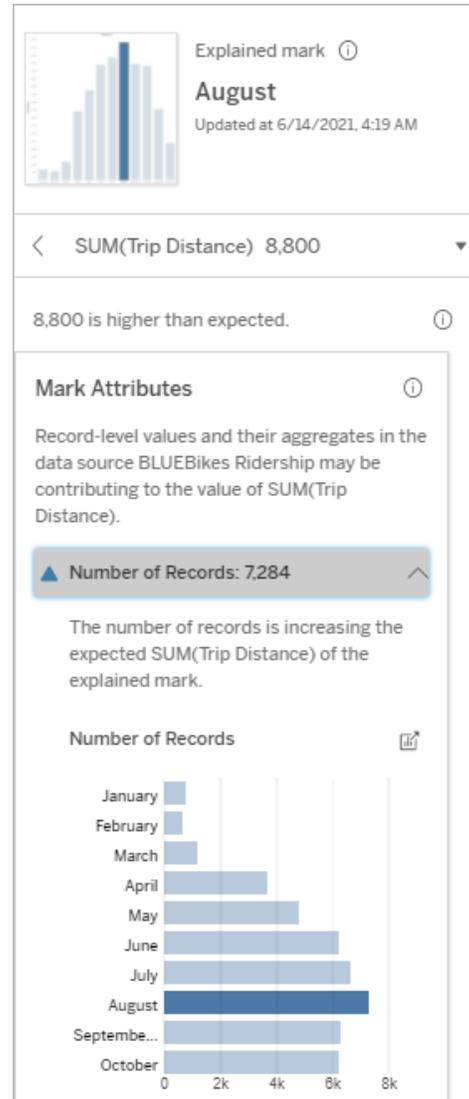
- The number of records in the target measure for the explained mark (dark blue bar).
- The number of records in the target measure for other marks in the source visualization (light blue bar).

Exploration options:

- Hover over each bar to see its details.
- Scroll to see more of the chart.
- Click the **Open**  icon to see a larger version of the visualization.

Next steps for analysis:

- Compare whether the individual values of records are low or high, or the number of records in the explained mark is low or high.
- Authors, if you are surprised by a high number of records, you might need to normalize the data.



In this example, the number of records for Trip Distance is listed for each value of Ride Month, which is a dimension in the original visualization. August has the highest total trip distance value.

You might explore whether August

has the highest value for trip distance because more rides occurred in August, or if it has the highest trip distance because some rides were longer.

Average Value of Mark

This explanation type describes when the average of a measure is correlated to the sum. Compare whether the average value is low or high, or the number of records is low or high.

This explanation shows:

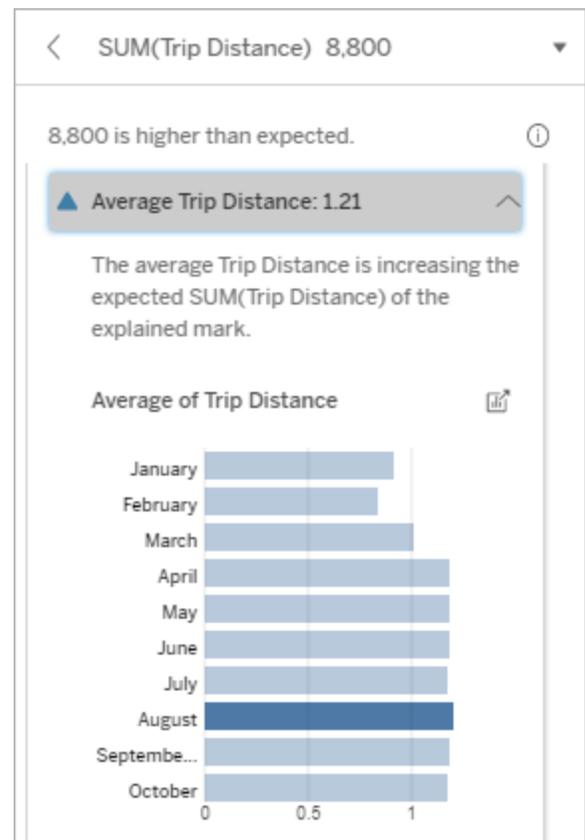
- The average of the target measure for each value of a dimension used in the source visualization.

Exploration options:

- Hover over each bar to see its details.
- Scroll to see more of the chart.
- Click the **Open**  icon to see a larger version of the visualization.

Next steps for analysis:

- Compare whether the average value is low or high, or the number of records is low or high. For example, are profits high because you sold a lot of items or because you sold expensive items?
- Try to figure out why the explained mark has a significantly higher or



In this example, the average trip dis-

lower average value.

tance for August is not significantly higher or lower than most months. This suggests that trip distance is higher for August because there were more rides in August, rather than from people taking longer rides.

Relevant Single Value

Use this explanation to understand the composition of the record values that make up the explained mark.

This explanation type identifies when a single value in an unvisualized dimension may be contributing to the aggregate value of the explained mark. An unvisualized dimension is a dimension that exists in the data source, but isn't currently being used in the view.

This explanation indicates when every underlying record of a dimension has the same value, or when a dimension value stands out because either many or few of the records have the same single value for the explained mark.

Note: For definitions of common terms used in explanations, see [Terms and concepts in explanations](#).

This explanation shows:

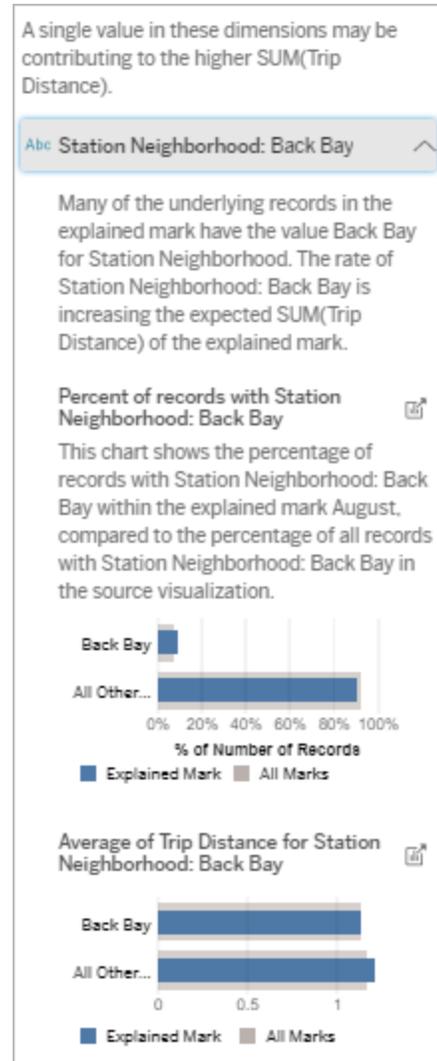
- The percent of the number of records for a single value of a dimension for the explained mark (blue bar) versus all marks (gray bar) in the source visualization.
- The percent of the number of records for all other values of a dimension for the explained mark (blue bar) versus all marks (gray bar) in the source visualization.
- The average of the target measure for the single value of a dimension in the explained mark (blue bar) versus all marks (gray bar).
- The average of the target measure for all other values of a dimension for the explained mark (blue bar) versus all marks (gray bar) in the source visualization.

Exploration options:

- Hover over each bar to see its details.
- Click the **Open**  icon to see a larger version of the visualization.

Next steps for analysis:

- Use this explanation to understand the composition of the record values that make up the explained mark.
- Authors might want to create a new visualization to explore any unvisualized dimension surfaced in this



In this example, the statistical analysis has exposed that many of the rides come from the station neighborhood of Back Bay. Note that Station Neighborhood is an unvisualized dimension that has some relationship to Trip Distance in the underlying data for the source visualization.

explanation.

Relevant Dimensions

Use this explanation to understand the composition of the record values that make up the explained mark.

This explanation type shows that the distribution of an unvisualized dimension may be contributing to the aggregate value of the explained mark. This type of explanation is used for target measure sums, counts, and averages. An unvisualized dimension is a dimension that exists in the data source, but isn't currently being used in the view.

Note: For definitions of common terms used in explanations, see [Terms and concepts in explanations](#).

This explanation shows:

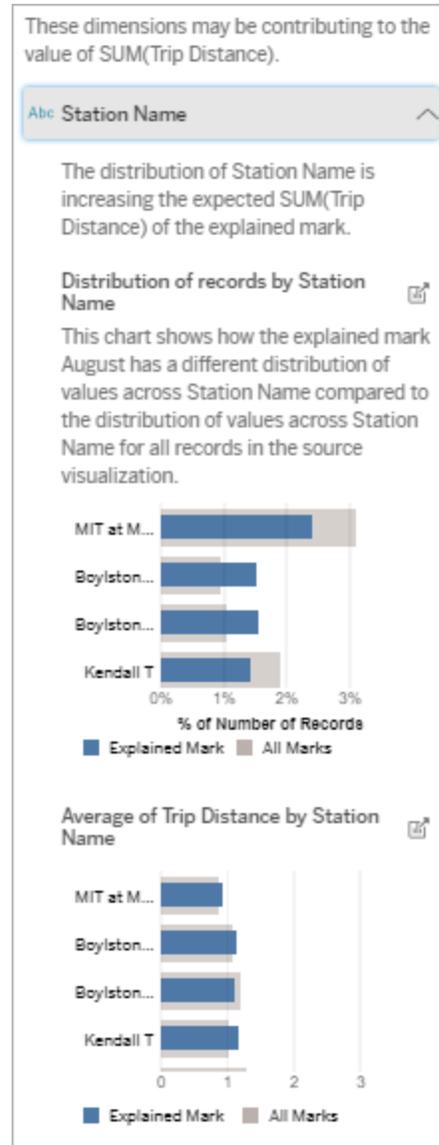
- The percent of the number of records for all values of a dimension for the explained mark (blue bar) versus all values of a dimension for all marks (gray bar) in the source visualization.
- The average of the target measure for all values of a dimension for the explained mark (blue bar) all values of a dimension for all marks (gray bar).

Exploration options:

- Hover over each bar to see its details.
- Scroll to see more of the chart.
- Click the **Open** icon to see a larger version of the visualization.

Next steps for analysis:

- Use this explanation to understand the composition of the record values that make up the explained mark.
- Authors might want to create a new visualization to explore any unvisualized dimensions surfaced in this explanation.



In this example, the statistical analysis has exposed that more rides were taken from Boylston stations and fewer rides were taken from MIT and Kendall, compared to rides taken for marks overall.

Note that Station Name is an unvisualized dimension that has some relationship to Trip Distance in the underlying data for the source visualization.

Relevant Measures

This explanation type shows that the average of an unvisualized measure may be contributing to the aggregate value of the explained mark. An unvisualized measure is a measure that exists in the data source, but isn't currently being used in the view.

This explanation can reveal a linear or quadratic relationship between the unvisualized measure and the target measure.

Note: For definitions of common terms used in explanations, see [Terms and concepts in explanations](#).

This explanation shows:

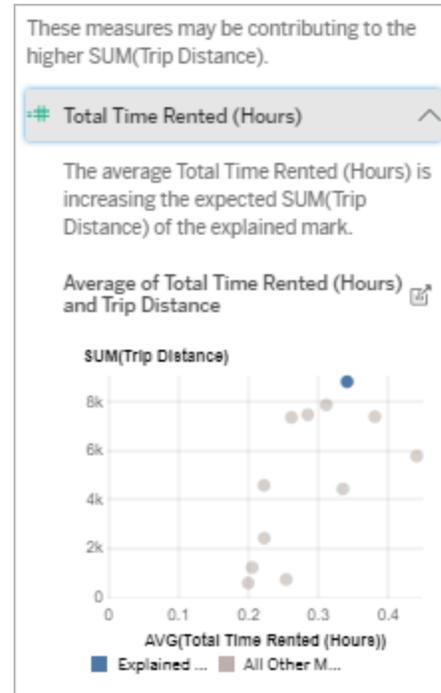
- The relationship between the sum of the target measure and the average of an unvisualized measure for the explained mark (blue circle) and all marks (gray circles) in the view.
- If the sum of the target measure is high or low because the average value of the unvisualized measure is high or low.

Exploration options:

- Hover over each circle to see its details.
- Click the **Open** icon to see a larger version of the visualization.

Next steps for analysis:

- Authors might want to create a new visualization to explore any unvisualized measures surfaced in this explanation.



In this example, one possible reason why trip distance is high is because the average total time rented is also high.

Other things to explore (Unique attributes of a mark)

This section of the Explain Data pane shows possible reasons why the explained mark is unique or unusual. These explanations:

- Do not explain why the value of this mark is what it is.
- Are not related in any way to the value of the measures in the source visualization.
- Do not take any target measures into account.

Relevant Single Value

The explanation type indicates when all records in the explained mark have the same single value in the unvisualized dimension, which is unusual compared to the distribution of values for all other marks in the view.

An unvisualized dimension is a dimension that exists in the data source, but isn't currently being used in the view

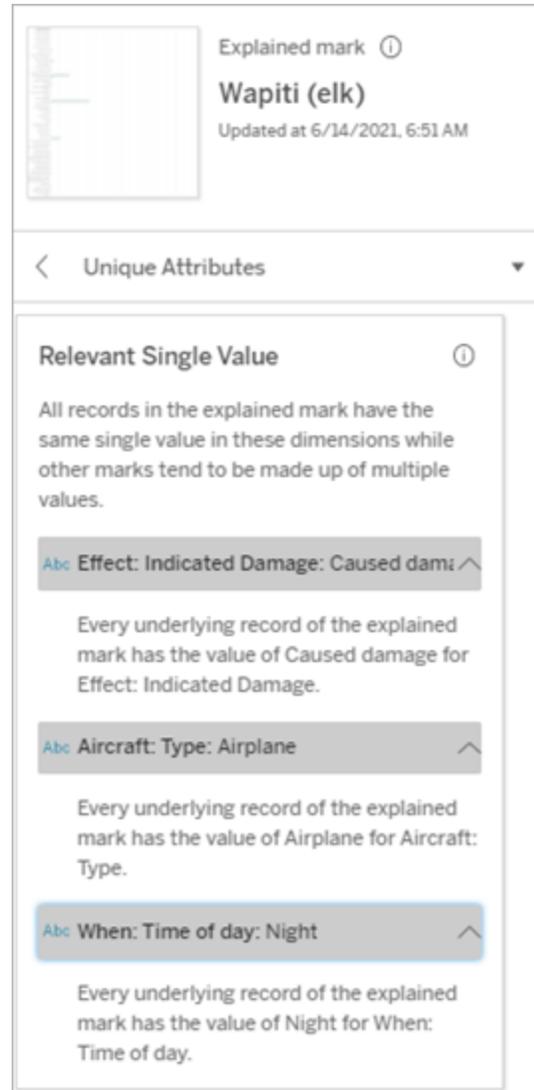
Note: For definitions of common terms used in explanations, see [Terms and concepts in explanations](#).

This explanation shows:

- When every underlying record has the same single value for a dimension.

Next steps for analysis:

- If all records in the explained mark have the same single value, you might want to check the number of records in that mark.
- You might check if the unvisualized dimension is a proxy for a dimension that is being used in the source visualization.



This example shows an explanation for data about incidents related to birds and other wildlife colliding with aircraft. In the unfortunate case of a wapiti (elk), three underlying dimensions each had underlying records with single values: Aircraft, Indicated Damage, and Time of Day.

Relevant Dimensions

Use this explanation to understand the composition of the record values that make up the explained mark.

The distribution of an unvisualized dimension in the explained mark is unusual compared to the distribution of values for all other marks in the view. An unvisualized dimension is a dimension that exists in the data source, but isn't currently being used in the view.

Note: For definitions of common terms used in explanations, see [Terms and concepts in explanations](#).

This explanation shows:

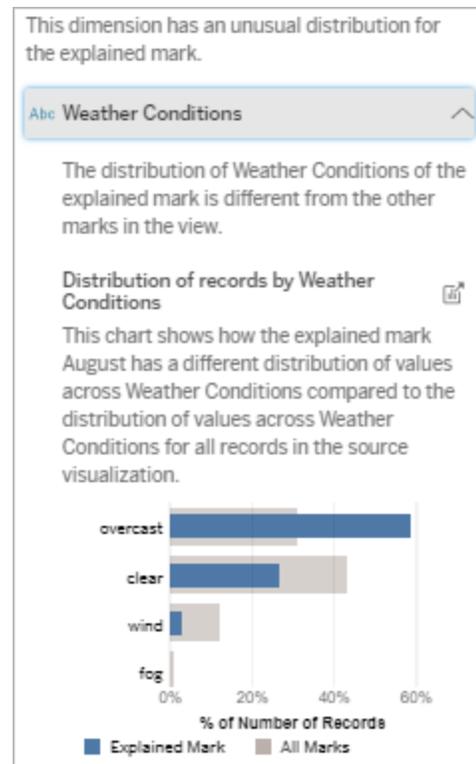
- The percent of the number of records for all values of a dimension for the explained mark (blue bar) versus all values of a dimension for all marks (gray bar) in the source visualization.

Exploration options:

- Hover over each bar to see its details.
- Scroll to see more of the chart.
- Click the **Open**  icon to see a larger version of the visualization.

Next steps for analysis:

- Use this explanation to understand the composition of the record values that make up the explained mark.



In this example, a high percentage of

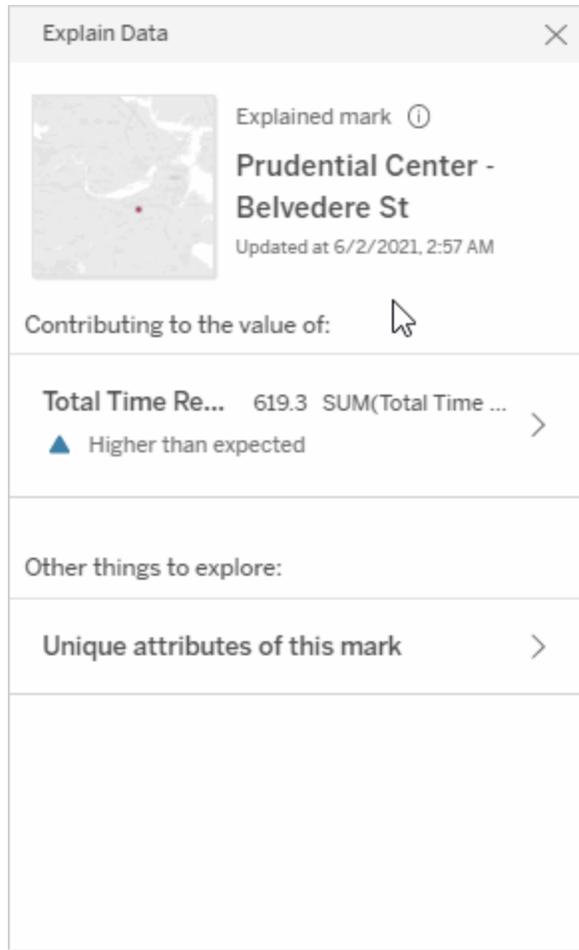
- Authors might want to create a new visualization to explore any unvisualized dimensions surfaced in this explanation.

records are associated with overcast weather. Because the data is about bike rentals in Boston, and the explained mark is Trip Distance for August, we can assume that the weather is typically warm and humid. People might have rented bikes more often on overcast days to avoid the heat. It's also possible there were more overcast days in August.

Analyzed Fields in Explain Data

When you run Explain Data on a mark, a statistical analysis is run on the aggregated mark, and then on potentially related fields from the data source that aren't represented in the current view.

Explain Data might not include every column from the data source in the analysis. In many cases, certain types of fields will be automatically excluded from the analysis. For more information, see [Fields excluded by default](#).



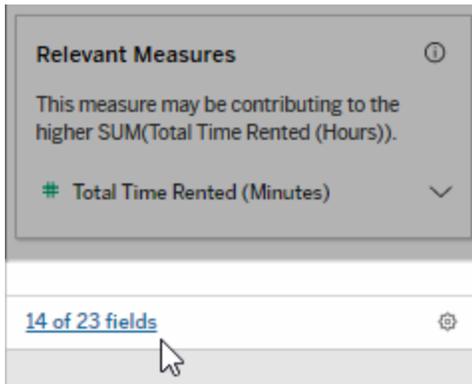
Note: Dimensions with more than 500 unique values won't be considered for analysis (unless allowed by the author in Explain Data Settings).

All users can view information on which fields are included or excluded in the current analysis. Creators and Explorers who have editing permissions can edit the fields used by Explain Data for statistical analysis.

View fields analyzed by Explain Data

When you expand an explanation for a measure that is contributing to the value of the mark, a link that indicates the number of fields considered in the analysis is displayed at the bottom

of the Explain Data pane.

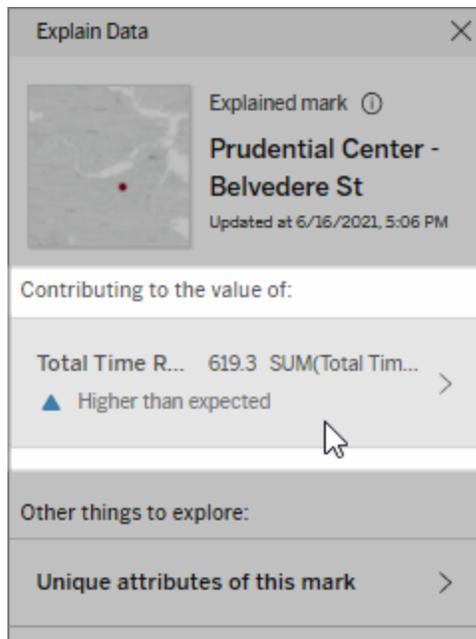


Click the link to see the list of fields included in or excluded from the current statistical analysis.

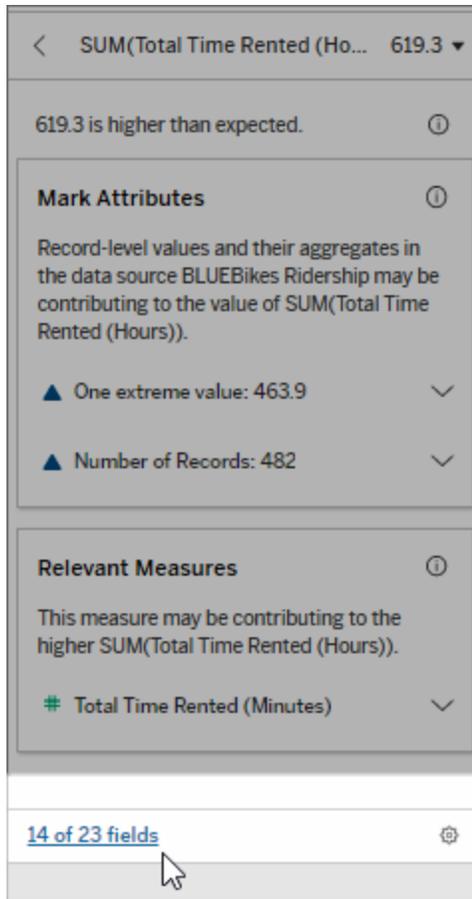
When a data source contains more than 1000 unvisualized dimensions or measures, you might see an alert asking if you want Explain Data to consider more fields. Click **Explain All** to run an analysis that includes more fields. The analysis may take longer to complete.

To view fields used by Explain Data for statistical analysis

1. [Run Explain Data on a mark.](#)
2. In the Explain Data pane, under **Contributing to the value of**, click a measure name.

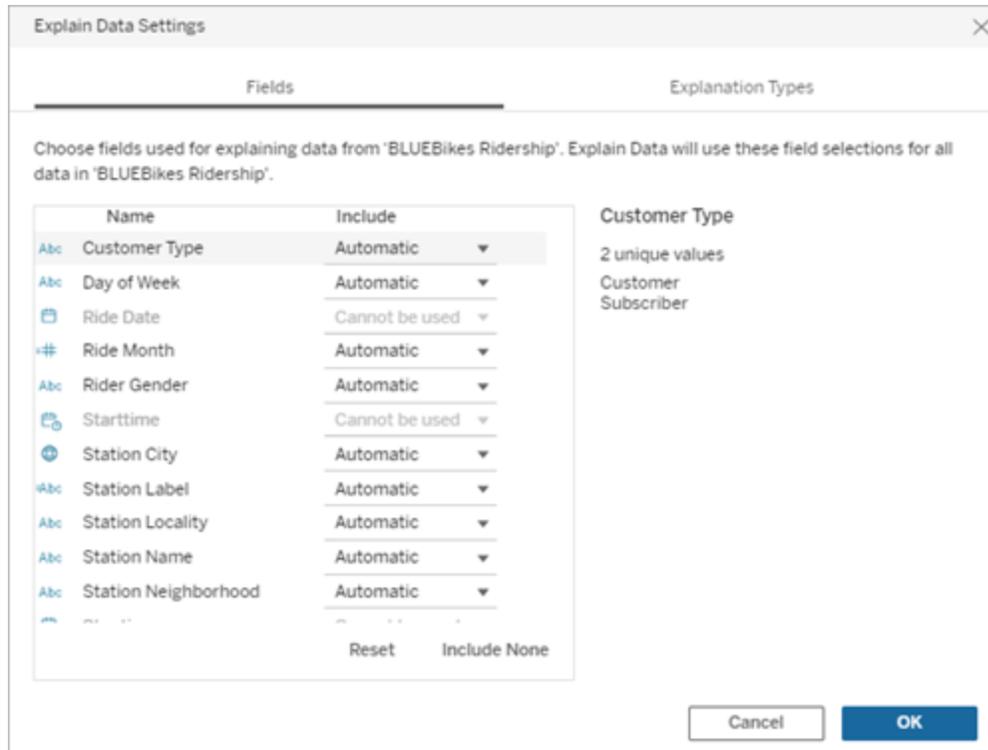


3. Click the *number-of-fields* link at the bottom of the pane.



Change fields used for statistical analysis

Creators and Explorers who have editing permissions can select fields to be included or excluded from the statistical analysis in the Fields tab of the Explain Data Settings dialog box.

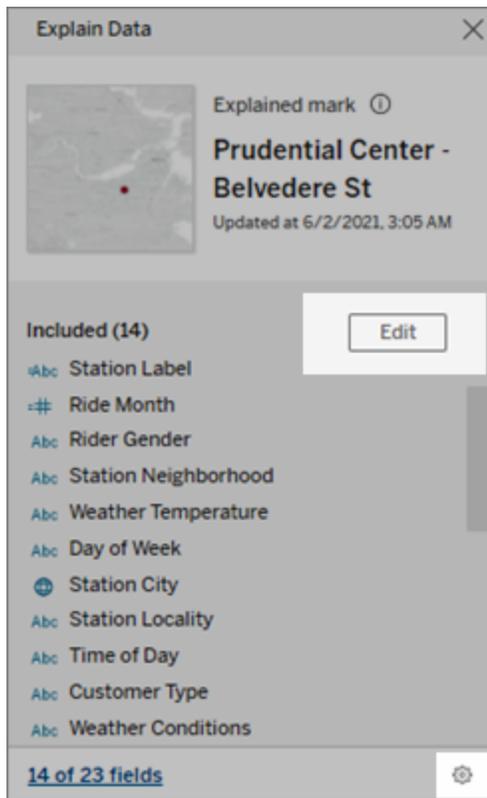


When a data source contains dimensions with a large number of unique values (up to 500), those fields won't be considered for analysis.

To edit the fields used by Explain Data for statistical analysis

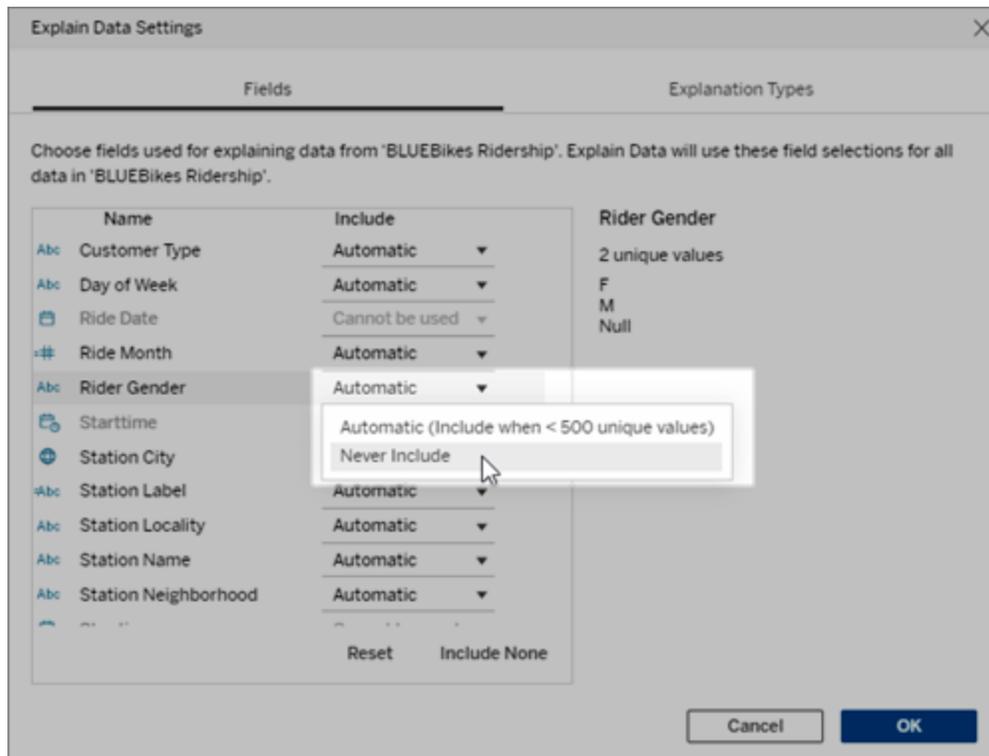
Settings for analyzed fields are applied at the data source level.

1. Run Explain Data on a mark when editing a view.
2. In the Explain Data pane, click the settings icon at the bottom of the Explain Data pane. Or, click the **Edit** button in the Analyzed Fields view ([how to open analyzed fields](#)).



3. In the Explain Data Settings dialog box, click the **Fields** tab.
4. Click a drop-down arrow next to a field name, select **Automatic** or **Never Include**, and then click **OK**.

Note that fields must have less than 500 unique values to be included in the analysis.



Fields excluded by default

Fields excluded by default

All unvisualized measures when there are more than 1,000 measures in the data source.

All unvisualized dimensions when there are more than 1,000 dimensions in the data source.

Reasons for exclusion

Computing explanations for more than 1000 unvisualized measures or dimensions can take longer to compute, sometimes several minutes. These fields are excluded by default for initial analysis, but you can choose to include them for further analysis.

In this situation, you might see an alert asking if you want Explain Data to consider more fields. Click the alert link to get more information. Click **Explain All** to run an analysis that includes more fields.

Fields excluded by default	Reasons for exclusion
Fields that use geometry, latitude, or longitude	Geometry, latitude, or longitude by themselves can never be explanations. It is highly likely that an explanation that calls out the latitude or the longitude as an explanation is due to a spurious correlation and not a probable explanation.
Dimensions with high cardinality (dimensions with > 500 members)	High cardinality dimensions take longer to compute. Dimensions with more than 500 unique values will not be considered for analysis.
Groups, bins, or sets	Not currently supported.
Table calculations	Table calculations cannot be analyzed when table calculations are at a different level of detail than the view.
Unvisualized measures that can't be averaged	Unvisualized measures that can't be averaged include measures that are calculated fields where the calculation expression includes aggregations (display as AGG() fields when added to the sheet).
Discrete measures and continuous dimensions	Not currently supported.
Hidden fields	Not available.
Calculated fields with errors	No values present to analyze.

Requirements and Considerations for Using Explain Data

Explain Data is always available to authors in Tableau Desktop. In Tableau Online and Tableau Server, Creators and Explorers with editing permissions can run Explain Data when editing a workbook. Authors can allow any user to run Explain Data in viewing mode in published workbooks. For more information, see [Control Access to Explain Data](#).

What makes a viz a good candidate for Explain Data

Explain Data works best on visualizations that require deeper exploration and analysis, rather than infographic-style, descriptive vizzes that communicate summarized data.

- Row-level data is necessary for Explain Data to create models of your data and generate explanations. Vizzes with underlying, row-level data, where relationships might exist in unvisualized fields are good candidates for running Explain Data.
- Vizzes based on pre-aggregated data without access to row-level data are not ideal for the statistical analysis performed by Explain Data.

What data works best for Explain Data

When you are using Explain Data in a worksheet, remember that Explain Data works with:

- **Single marks only**—Explain Data must be run on a single mark. Multiple mark analysis is not supported.
- **Aggregated data**—The view must contain one or more measures that are aggregated using SUM, AVG, COUNT, or COUNTD. At least one dimension must also be present in the view.
- **Single data sources only**—The data must be drawn from a single, primary data source. Explain Data does not work with blended or cube data sources.

When preparing a data source for a workbook, keep the following considerations in mind if you plan to use Explain Data during analysis.

- Use a data source with underlying data that is sufficiently wide. An ideal data set has at least 10-20 columns in addition to one (or more) aggregated measures to be explained.
- Give columns (fields) names that are easy to understand.
- Eliminate redundant columns and data prep artifacts. For more information, see [Change fields used for statistical analysis](#).
- Don't discard unvisualized columns in the data source. Explain Data considers fields in the underlying data when it analyzes a mark.

Tableau Server on Linux Administrator Guide

- Low cardinality dimensions work better. The explanation of a categorical dimension is easier to interpret if its cardinality is not too high (< 20 categories). Dimensions with more than 500 unique values will not be considered for analysis.
- Don't pre-aggregate data as a general rule. But if the data source is massive, consider pre-aggregating the data to an appropriate level of detail.
- Use extracts over live data sources. Extracts run faster than live data sources. With live data sources, the process of creating explanations can create many queries (roughly one query per each candidate explanation), which can result in explanations taking longer to be generated.

Situations where Explain Data is not available

Sometimes Explain Data will not be available for a selected mark, depending on the characteristics of the data source or the view. If Explain Data cannot analyze the selected mark, the Explain Data icon and context menu command will not be available.

Explain Data can't be run in views that use:

- Map coordinate filters
- Blended data sources
- Data sources with parameters
- Data sources that don't support COUNTD or COUNT(DISTINCT ...) syntax, such as Access.
- Filters on aggregate measures
- Disaggregated measures

Explain Data can't be run if you select:

- Multiple marks
- Axis
- Legend
- Grand total
- Trend line or reference line
- A mark in a view that contains a very low number of marks

Explain Data can't be run when the measure to be used for an explanation:

- Isn't aggregated using SUM, AVG, COUNT, COUNTD
- Is a table calculation
- Is used in measure values

Explain Data can't offer explanations for a dimension when it is:

- A calculated field
 - A parameter
 - Used in Measure Names and Measure Values
 - A field with more than 500 unique values.
- Dimensions with more than 500 unique values will not be considered for analysis.

Control Access to Explain Data

Explain Data is always available to Creators and Explorers in authoring mode. Authors can control whether Explain Data is available in viewing mode in published workbooks and which explanation types are displayed in the Explain Data pane.

Be aware that Explain Data can surface values from dimensions and measures in the data source that aren't represented in the view. As an author, you should run Explain Data and test the resulting explanations to make sure that sensitive data isn't being exposed in your published workbooks.

Because Extreme Values explanations display record-level details, workbook authors must explicitly select an option to display Extreme Values explanations in published workbooks. For more information, see [Show or hide Extreme Values explanations in viewing mode](#).

Who can access Explain Data

Your access to Explain Data will vary depending on how you interact with Tableau workbooks, and whether you are viewing or editing a published workbook.

Mode	Who Can Access
Viewing mode	When an author makes Explain Data available in viewing mode in a published workbook, Tableau Viewers , Explorers , and Creators can run and explore Explain Data explanations in viewing mode.
Editing mode	Tableau Creators can run Explain Data when editing a view in Tableau Desktop, Tableau Online, or Tableau Server.

Mode

Who Can Access

Explorers who have editing permissions can run Explain Data when editing a workbook in Tableau Online or Tableau Server.

Creators and Explorers who have editing permissions can open new worksheets for further analysis.

They also can use Explain Data Settings to control who can use Explain Data and what they can see.

Control who can use Explain Data and what they can see

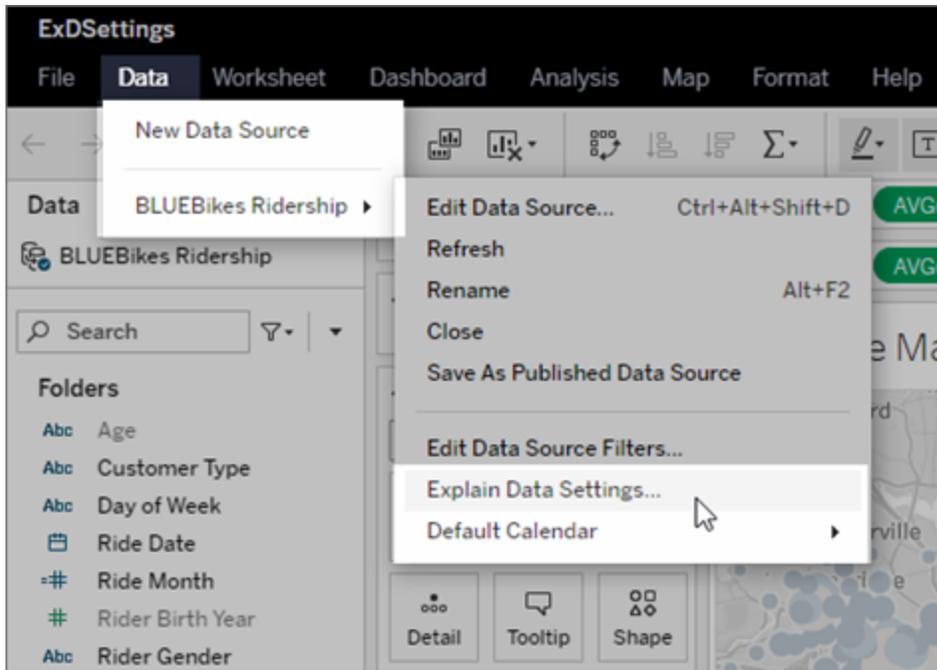
Creators and Explorers who have editing permissions can access **Explain Data Settings**, which provide options for controlling:

- Whether all users, including Viewers, are allowed to run Explain Data and **see explanations while viewing published workbooks** and dashboards. This option is not enabled by default, so authors need to explicitly allow it.
- Whether **Extreme Values explanations and record-level details** are displayed in viewing mode in published workbooks. This option is not enabled by default, so authors need to explicitly allow it.
- The **explanation types that are displayed** in the Explain Data pane.
- The **fields that are included in, or excluded from** statistical analysis.

These options are set for the entire workbook and can only be set in the Explain Data Settings dialog box.

Open the Explain Data Settings dialog box

1. From the **Data** menu, select *data-source-name* > **Explain Data Settings**. For example:



Or, in the Explain Data pane, click the settings icon (bottom right).

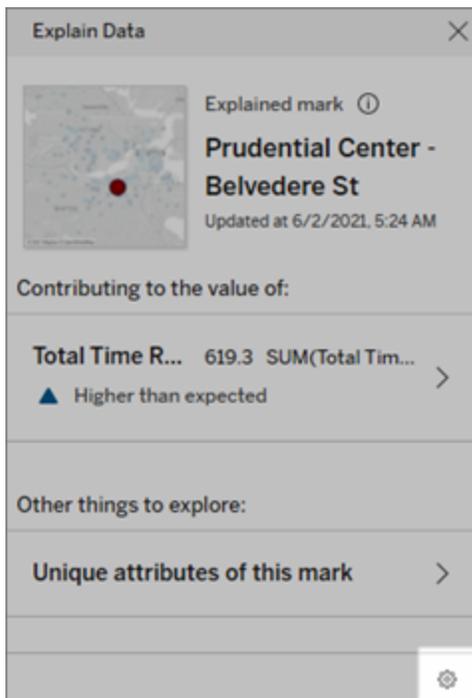
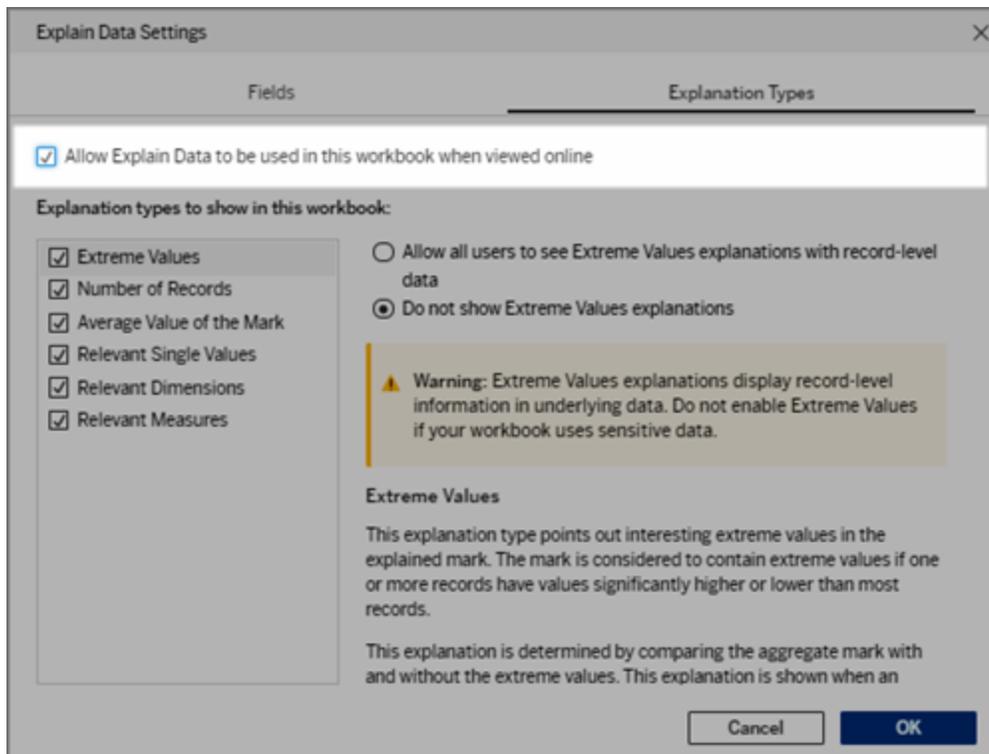


Tableau Server on Linux Administrator Guide

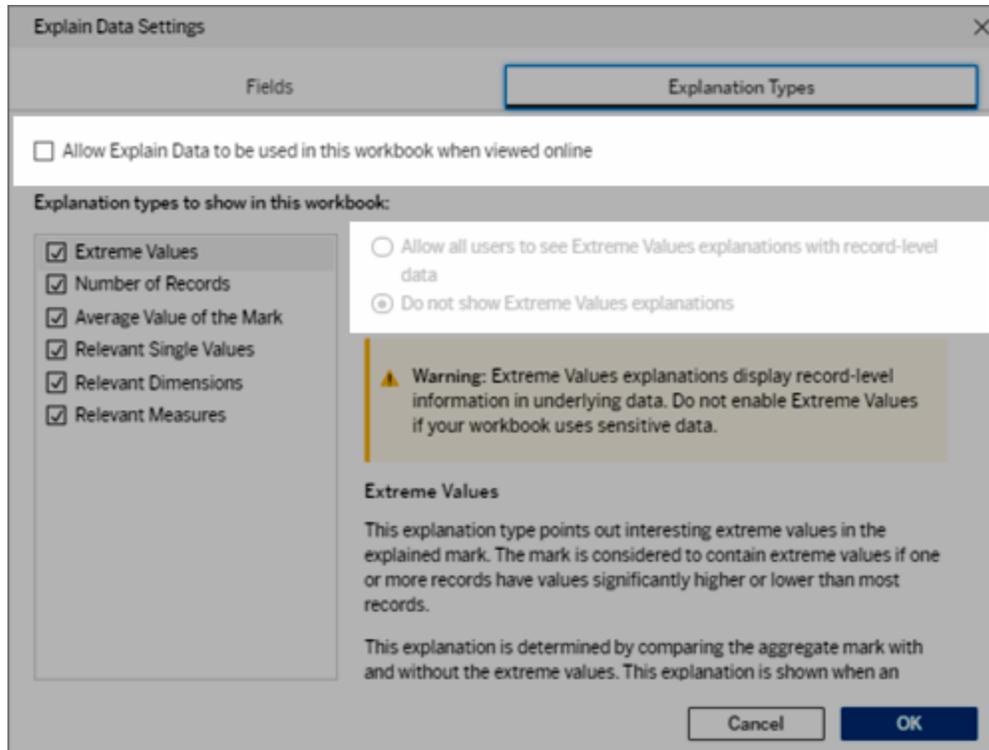
Enable Explain Data in viewing mode in published workbooks and dashboards

By default, Explain Data is not available in viewing mode. To turn Explain Data on for viewing mode, authors must do the following:

1. In the **Explain Data Settings** dialog box, click the **Explanation Types** tab.
2. Select **Allow Explain Data to be used in this workbook when viewed online**.



Alternatively, leave this option unchecked to return to the default setting and make Explain Data unavailable in viewing mode. Creators and Explorers who have editing permissions will still be able to run Explain Data when editing a workbook.



This image shows the default settings for the Explanation Types tab

3. Click **OK**.

Test the setting by saving and closing the published workbook, and then open a view from the workbook in viewing mode. Select a mark, and then run Explain Data to check the explanation results.

Show or hide Extreme Values explanations in viewing mode

The Extreme Values explanation type includes record-level details. When editing a workbook, Creators and Explorers can always see Extreme Values explanations and record-level details. However, in viewing mode for published workbooks, Extreme Values explanations are hidden by default.

Tableau Server on Linux Administrator Guide

619.3 is higher than expected. ⓘ

Mark Attributes ⓘ

Record-level values and their aggregates in the data source BLUEBikes Ridership may be contributing to the value of SUM(Total Time Rented (Hours)).

▲ One extreme value: 463.9 ▼

▲ Number of Records: 482

Open e:

Relevant Measures ⓘ

This measure may be contributing to the higher SUM(Total Time Rented (Hours)).

Total Time Rented (Minutes) ▼

[14 of 23 fields](#)

Note that only Creators and Explorers with Download Full Data permissions can see the View Full Data option for Extreme Values explanations.

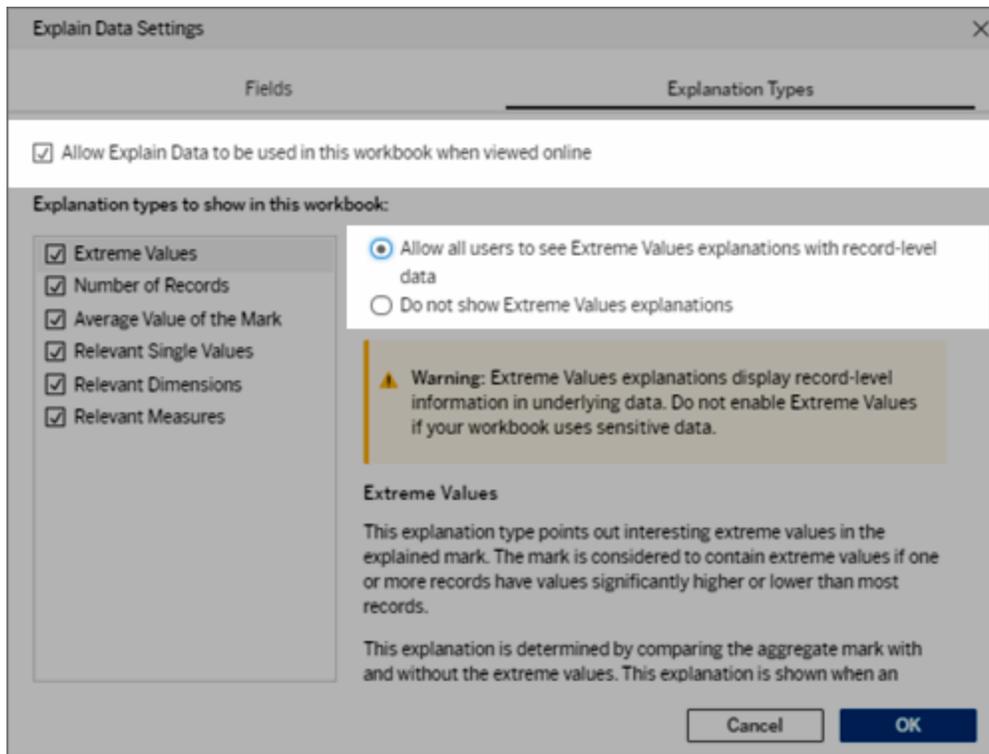


If you have sensitive data in your workbook, you can keep the Extreme Values explanation type hidden for viewing mode.

To show Extreme Values explanations in viewing mode in published workbooks

As an author, if you want Extreme Values explanations to be displayed in viewing mode, do the following:

1. In the **Explain Data Settings** dialog box, click the **Explanation Types** tab.
2. Select **Allow Explain Data to be used in this workbook when viewed online**.
3. Make sure that the Extreme Values explanation type is selected.
4. Select **Allow all users to see Extreme Values explanations with record-level data**.



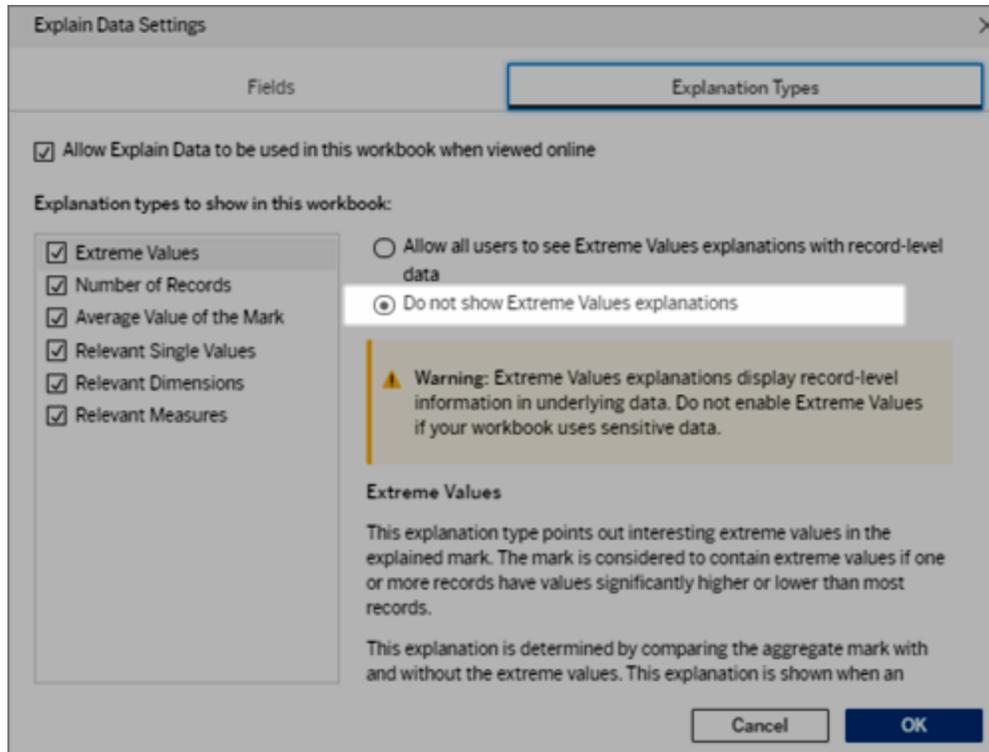
5. Click **OK**.

Test the setting by saving and closing the published workbook, and then opening a view from the workbook in viewing mode. Select a mark that typically has Extreme Value explanations, and then run Explain Data to check the explanation results.

To hide Extreme Values explanations and record-level details in viewing mode (published workbooks)

By default, Extreme Values explanations and record-level details are not available in viewing mode. Use these steps to return to the default state.

1. In the **Explain Data Settings** dialog box, click the **Explanation Types** tab.
2. Make sure that the Extreme Values explanation type is selected (allows Creators and Explorers to see these explanations in editing mode).
3. Select the option **Do not show Extreme Values explanations** to hide Extreme Values explanations and record-level details from all users in viewing mode.



4. Click **OK**.

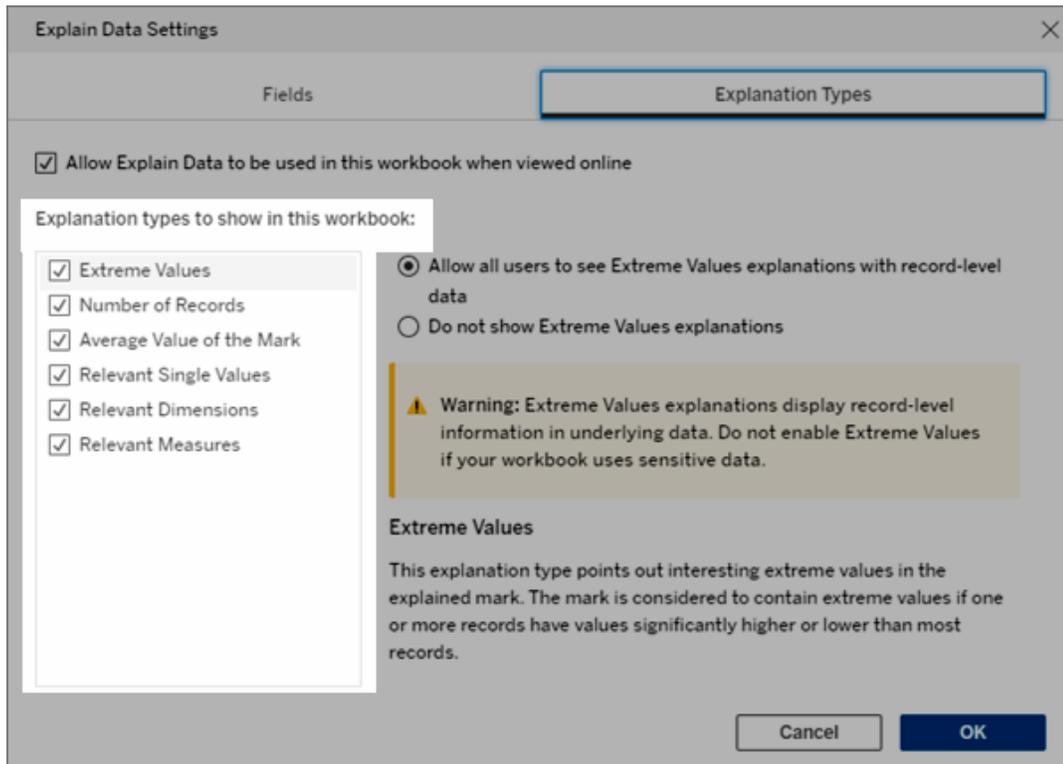
Test the setting by saving and closing the workbook, and then opening a view from the workbook in viewing mode. Select a mark that typically has Extreme Value explanations, and then run Explain Data to check the explanation results.

Include or exclude explanation types displayed by Explain Data

Creators and Explorers who have editing permissions can choose to exclude (or include) explanation types displayed for all workbook users.

1. In the **Explain Data Settings** dialog box, click the **Explanation Types** tab.

2. In the list of explanation types, select or clear an explanation type.



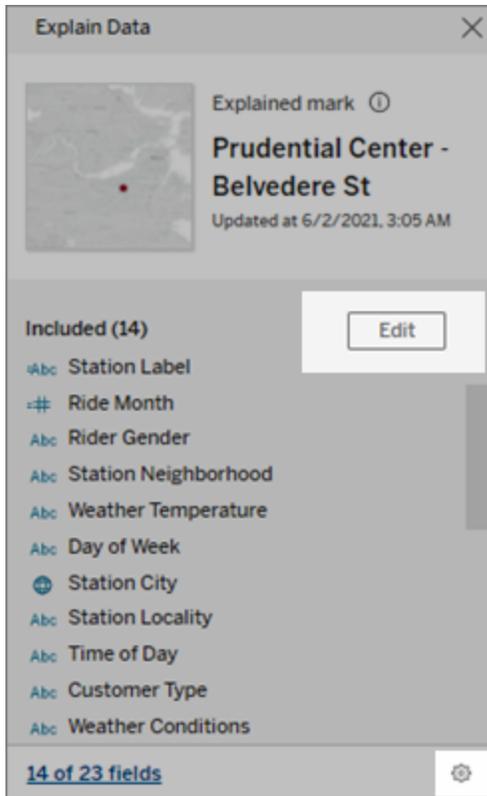
3. Click **OK**.

Test the setting by saving and closing the published workbook, and then opening a view from the workbook in viewing mode. Select a mark that typically has Extreme Value explanations, and then run Explain Data to check the explanation results.

Include or exclude fields used for statistical analysis

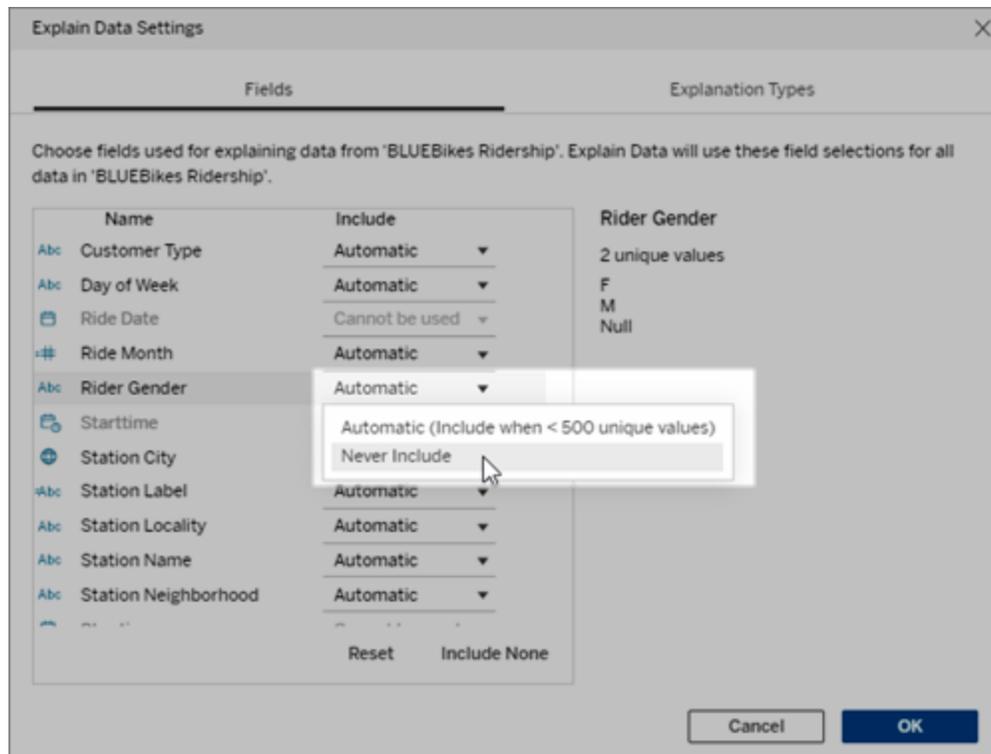
Creators or Explorers who have editing permissions can choose to exclude (or include) fields that are eligible for analysis.

1. In the Explain Data pane (bottom right), click the settings icon. Or, click the **Edit** button in the [Analyzed Fields view](#).



2. In the **Explain Data Settings** dialog box, click the **Fields** tab.
3. In the list of fields under **Include**, click the drop-down arrow and select **Automatic** to include an eligible field every time Explain Data runs for that workbook.

Note that fields must have less than 500 unique values to be included in the analysis.



Select **Never Include** to explicitly exclude the field.

Select **Include None** to run a statistical analysis on the data without considering fields.

Select **Reset** to return to the default settings.

4. Click **OK**.

Test the setting by saving the published workbook. Select a mark, and then run Explain Data to check the explanation results.

How Explain Data Works

Use Explain Data as an incremental, jumping-off point for further exploration of your data.

The possible explanations that it generates help you to see the different values that make up or relate to a selected mark in a view. It can tell you about the characteristics of the data points in the data source, and how the data might be related (correlations) using statistical modeling. These explanations give you another tool for inspecting your data and finding interesting clues about what to explore next.

Note: Explain Data is a tool that uncovers and describes relationships in your data. It can't tell you what is causing the relationships or how to interpret the data. **You are the expert on your data.** Your domain knowledge and intuition is key in helping you decide what characteristics might be interesting to explore further using different views.

For related information on how Explain Data works, and how to use Explain Data to augment your analysis, see these Tableau Conference presentations:

- [From Analyst to Statistician: Explain Data in Practice \(1 hour\)](#)
- [Leveraging Explain Data \(45 minutes\)](#)
- [Explain Data Internals: Automated Bayesian Modeling \(35 minutes\)](#)
- [Machine Learning, Explainable AI, and Tableau \(45 minutes\)](#), [Session Materials](#)
- [Making Business More Bayesian \(45 minutes\)](#)

What Explain Data is (and isn't)

Explain Data is:

- A tool and a workflow that leverages your domain expertise.
- A tool that surfaces relationships in your data and recommends where to look next.
- A tool and a workflow that helps expedite data analysis and make data analysis more accessible to a broader range of users.

Explain Data is not:

- A statistical testing tool.
- A tool to prove or disprove hypotheses.
- A tool that is giving you an answer or telling you anything about causality in your data.

When running Explain Data on marks, keep the following points in mind:

- **Consider the shape, size, and cardinality of your data.** While Explain Data can be used with smaller data sets, it requires data that is sufficiently wide and contains enough marks (granularity) to be able to create a model.
- **Don't assume causality.** Correlation is not causation. Explanations are based on models of the data, but are not causal explanations.

A correlation means that a relationship exists between some data variables, say A and B. You can't tell just from seeing that relationship in the data that A is causing B, or B is causing A, or if something more complicated is actually going on. The data patterns are exactly the same in each of those cases and an algorithm can't tell the difference between each case. Just because two variables seem to change together doesn't necessarily mean that one causes the other to change. A third factor could be causing them both to change, or it may be a coincidence and there might not be any causal relationship at all.

However, you might have outside knowledge that is not in the data that helps you to identify what's going on. A common type of outside knowledge would be a situation where the data was gathered in an experiment. If you know that B was chosen by flipping a coin, any consistent pattern of difference in A (that isn't just random noise) must be caused by B. For a longer, more in-depth description of these concepts, see the article [Causal inference in economics and marketing](#) by Hal Varian.

How explanations are analyzed and evaluated

When you run Explain Data on a mark, a statistical analysis is run on the aggregated mark and then on possibly related data points from the data source that aren't represented in the current view.

Explain Data first predicts the value of a mark using only the data that is present in the visualization. Next, data that is in the data source (but not in the current view) is considered and added to the model. The model determines the range of predicted mark values, which is within one standard deviation of the predicted value.

What is an expected range?

The expected value for a mark is the median value in the expected range of values in the underlying data in your viz. The expected range is the range of values between the 15th and 85th percentile that the statistical model predicts for the explained mark. Tableau determines the expected range each time it runs a statistical analysis on a selected mark.

Possible explanations are evaluated on their explanatory power using statistical modeling. For each explanation, Tableau compares the expected value with the actual value.

value	Description
Higher than expected / Lower than expected	If an expected value summary says the mark is <i>lower than expected</i> or <i>higher than expected</i> , it means the aggregated mark value is outside the range of values that a statistical model is predicting for the mark. If an expected value summary says the mark is <i>slightly lower</i> or <i>slightly higher</i> than expected, or <i>within the range of natural variation</i> , it means the aggregated mark value is within the range of predicted mark values, but is lower or higher than the median.
Expected Value	If a mark has an expected value, it means its value falls within the expected range determined by Tableau when it runs a statistical analysis on the mark you selected.
Random Variation	When the explained mark has a low number of records, there may not be enough data available for Explain Data to form a statistically significant explanation. If the mark's value is outside the expected range, Explain Data can't determine whether this unexpected value is being caused by random variation or by a meaningful difference in the underlying records.
No Explanation	When the explained mark value is outside of the expected range and it does not fit a statistical model used for Explain Data, no explanations are generated.

Models used for analysis

Explain Data builds models of the data in a view to predict the value of a mark and then determines whether a mark is higher or lower than expected given the model. Next, it considers additional information, like adding additional columns from the data source to the view, or flagging record-level outliers, as potential explanations. For each potential

explanation, Explain Data fits a new model, and evaluates how unexpected the mark is given the new information. Explanations are scored by trading off complexity (how much information is added from the data source) against the amount of variability that needs to be explained. Better explanations are simpler than the variation they explain.

Explanation type	Evaluation
Extreme values	<p>Extreme values are aggregated marks that are outliers, based on a model of the visualized marks. The selected mark is considered to contain an extreme value if a record value is in the tails of the distribution of the expected values for the data.</p> <p>An extreme value is determined by comparing the aggregate mark with and without the extreme value. If the mark becomes less surprising by removing a value, then it receives a higher score.</p> <p>When a mark has extreme values, it doesn't automatically mean it has outliers, or that you should exclude those records from the view. That choice is up to you depending on your analysis. The explanation is simply pointing out an interesting extreme value in the mark. For example, it could reveal a mistyped value in a record where a banana cost 10 dollars instead of 10 cents. Or, it could reveal that a particular sales person had a great quarter.</p>
Number of records	<p>The number of records explanation models the aggregate sum in terms of the aggregate count; average value of records models it in terms of the aggregate average. The better the model explains the sum, the higher the score.</p> <p>This explanation describes whether the sum is interesting because the count is high or low, or because the average is high or low.</p>
Average value of	<p>This type of explanation is used for aggregate marks that are</p>

Explanation type	Evaluation
the mark	<p>sums. It explains whether the mark is consistent with the other marks because in terms of its aggregate count or average, noting the relation $SUM(X) = COUNT(X) * AVG(X)$.</p> <p>This explanation describes whether the sum is interesting because the count is high or low, or because the average is high or low.</p>
Relevant Dimensions	<p>This explanation models the target measure of the explained mark in terms of the breakdown among categories of the unvisualized dimension. The analysis balances the complexity of the model with how well the mark is explained.</p> <p><i>An unvisualized dimension</i> is a dimension that exists in the data source, but isn't currently being used in the view. This type of explanation is used for sums, counts and averages.</p> <p>The model for unvisualized dimensions is created by splitting out marks according to the categorical values of the explaining column, and then building a model with the value that includes all of the data points in the source visualization. For each row, the model attempts to recover each of the individual components that made each mark. The analysis indicates whether the model predicts the mark better when components corresponding to the unvisualized dimension are modeled and then added up, versus using a model where the values of the unvisualized dimension are not known.</p> <p>Aggregate dimension explanations explore how well mark values can be explained without any conditioning. Then, the model conditions on values for each column that is a potential explanation. Conditioning on the distribution of an explanatory column should result in a better prediction.</p>

Explanation type	Evaluation
Relevant Measures	<p>This explanation models the mark in terms of this unvisualized measure, aggregated to its mean across the visualized dimensions. An <i>unvisualized measure</i> is a measure that exists in the data source, but isn't currently being used in the view.</p> <p>The Relevant Measures explanation can reveal a linear or quadratic relationship between the unvisualized measure and the target measure.</p>

Use Dashboard Extensions

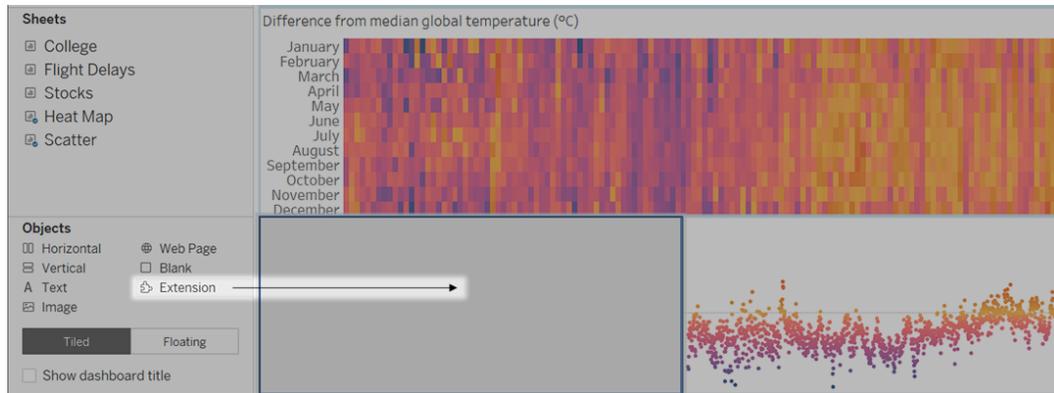
Extensions let you add unique features to dashboards or directly integrate them with applications outside Tableau. Adding extensions is easy; you incorporate them into dashboard layouts just like other dashboard objects.

Extensions expand dashboard functionality with the help of web applications created by third-party developers. If you're a developer and want to create your own extensions, see the [Tableau Extensions API documentation](#) on GitHub.

Note: Tableau administrators can turn off dashboard extensions for [Tableau Desktop](#), [Tableau Server](#), and [Tableau Online](#).

Add an extension to a dashboard

1. In a Tableau workbook, open a dashboard sheet.
2. From the **Objects** section, drag **Extension** to the dashboard.



3. In the “Add an Extension” dialog box, do either of the following:
 - Search for and select an extension.
 - Click **Access Local Extensions**, and navigate to a .trex file you previously downloaded.
4. If prompted, allow or deny the dashboard extension access to data in the workbook. For more information, see [Data security](#), [Network-enabled](#), and [Sandboxed extensions](#).

If you allow access, follow any on screen instructions for configuring the extension.

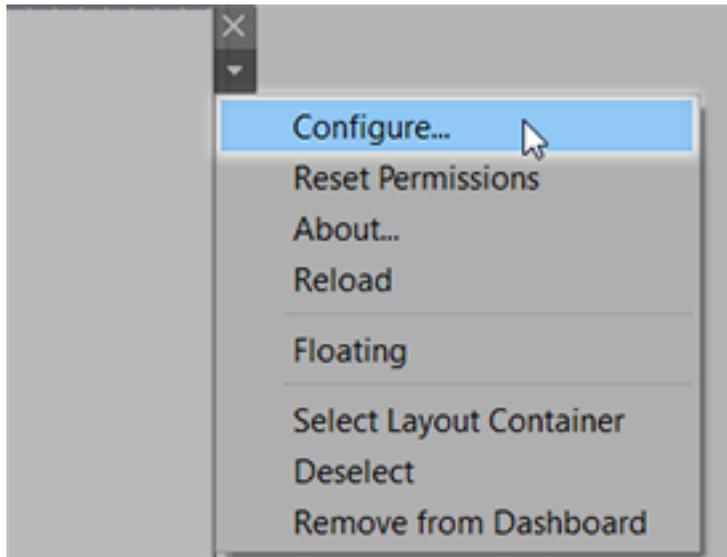
Note: If you're using Tableau Server or Tableau Online, Extension objects will appear blank in prints, PDFs, and images of dashboards (including images in subscription emails).

Configure a dashboard extension

Some dashboard extensions provide configuration options that let you customize features.

1. Select the extension in the dashboard, and from the drop-down menu in the upper-right corner, choose **Configure**.

2. Follow the on-screen instructions to configure the extension.



Reload a dashboard extension

If a dashboard extension becomes unresponsive, you might need to reload it, which is similar to refreshing a web page in a browser.

1. Select the extension in the dashboard, and from the drop-down menu in the upper-right corner, choose **Reload**.

The dashboard extension is refreshed and set to its original state.

2. If reloading the extension fails to return it to a useable state, try removing it from the dashboard and adding it again.

Data security, Network-enabled, and Sandboxed extensions

Dashboard extensions are web applications that come in two forms:

- *Network-enabled extensions* run on web servers located outside of your local network.
- *Sandboxed extensions* run in a protected environment without access to any other resource or service on the web.

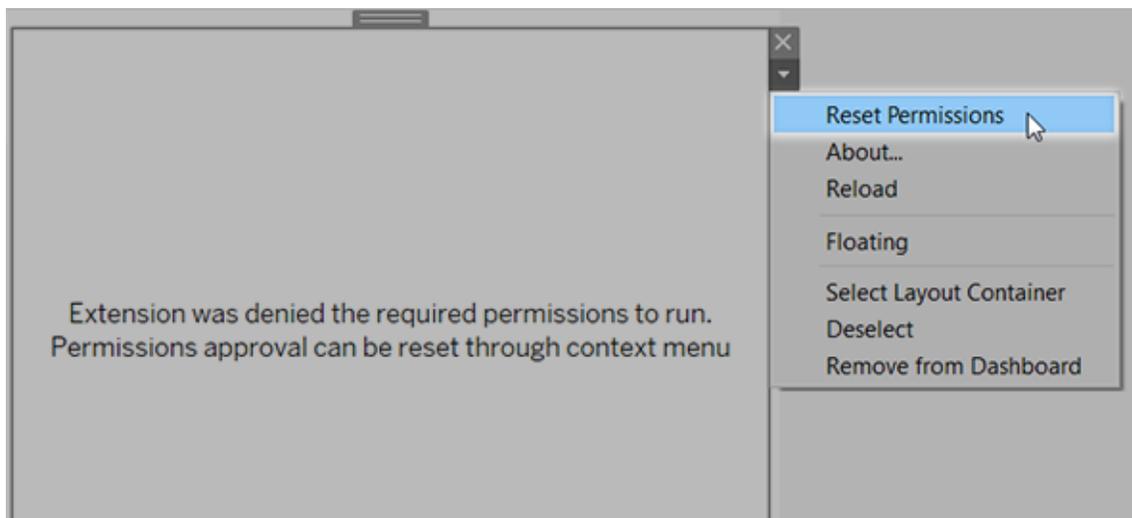
Before adding a Network-enabled extension or viewing a dashboard with one, be certain that you trust the website that hosts it. By default, dashboard extensions use the HTTPS protocol, which guarantees an encrypted channel for sending and receiving data, and ensures some privacy and security.

For more information about data security when using dashboard extensions, see [Extension Security - Best Practices for Deployment](#).

Allow or deny data access to a Network-enabled extension

Depending on how an extension is designed, it can access either visible data in a view, or full underlying data, table and field names from data sources, and information about data source connections. When you add an extension, or view a dashboard with one, you're given an opportunity to allow or deny the extension to run and access this data.

If you're viewing a dashboard with an extension that requires full data access, and that access has been denied, a message appears in place of the extension. If you trust the extension and want to use it, you can reset permissions and allow the extension to run.



1. Select the extension in the dashboard, and from the drop-down menu in the upper-right corner, choose **Reset Permissions**.

2. Click either **Allow** to let the extension run and access data, or **Deny** to prevent the extension from running.

Ensure that JavaScript is enabled in Tableau Desktop

Dashboard extensions interact with data using the Tableau Extensions API library, a JavaScript library. If you want to use extensions, be sure that JavaScript is enabled in the dashboard security settings:

Choose **Help > Settings and Performance > Set Dashboard Web View Security > Enable JavaScript**.

Ensure that extensions run on Tableau Online or Tableau Server

You can add extensions to workbooks you publish from Tableau Desktop or directly in the web-authoring mode of Tableau Online and Tableau Server. A Tableau administrator must allow extensions to run on a site and add Network-enabled extensions to a safe list. Administrators should only allow extensions that you have tested and trust.

If you want to use a dashboard extension on Tableau Online or Tableau Server, direct your administrator to [Manage Dashboard Extensions in Tableau Online](#) or [Manage Dashboard Extensions in Tableau Server](#).

Supported web browsers for Sandboxed extensions

Sandboxed extensions run in all browsers supported [Tableau Server](#) and [Tableau Online](#) except Internet Explorer 11.

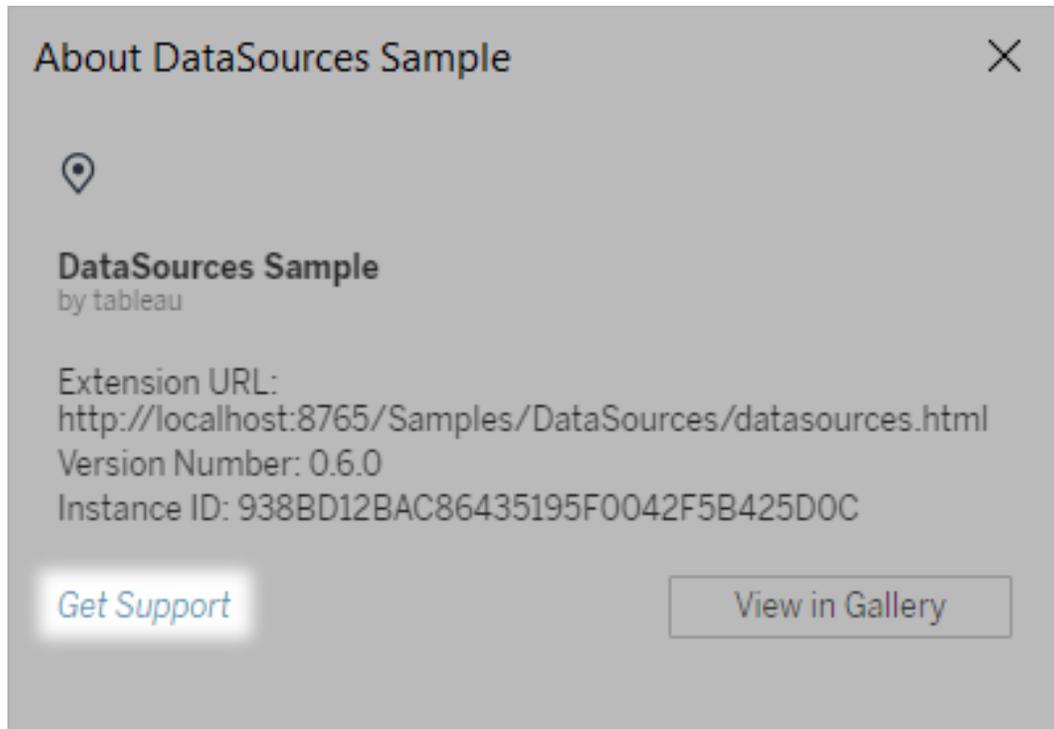
Supported versions of Tableau Server for Sandboxed extensions

You can use Sandboxed extensions in Tableau Server 2019.4 and later.

Get support for dashboard extensions

To get help for an extension, you'll need to contact the developer or company who created it.

1. Select the extension in the dashboard, and from the drop-down menu in the upper-right corner, choose **About**.
2. Click **Get Support** to go to the support page of the extension developer.



Note: Tableau doesn't provide support for extensions or for other programs that interface with the Extensions API. However, you can submit questions and ask for help in the [Tableau developer community](#).

Format Animations

Animate visualizations to better highlight changing patterns in your data, reveal spikes and outliers, and see how data points cluster and separate.

Animations visually transition between filter, sort, and zoom settings, different pages, and changes to filter, parameter, and set actions. As visualizations animate in response to these

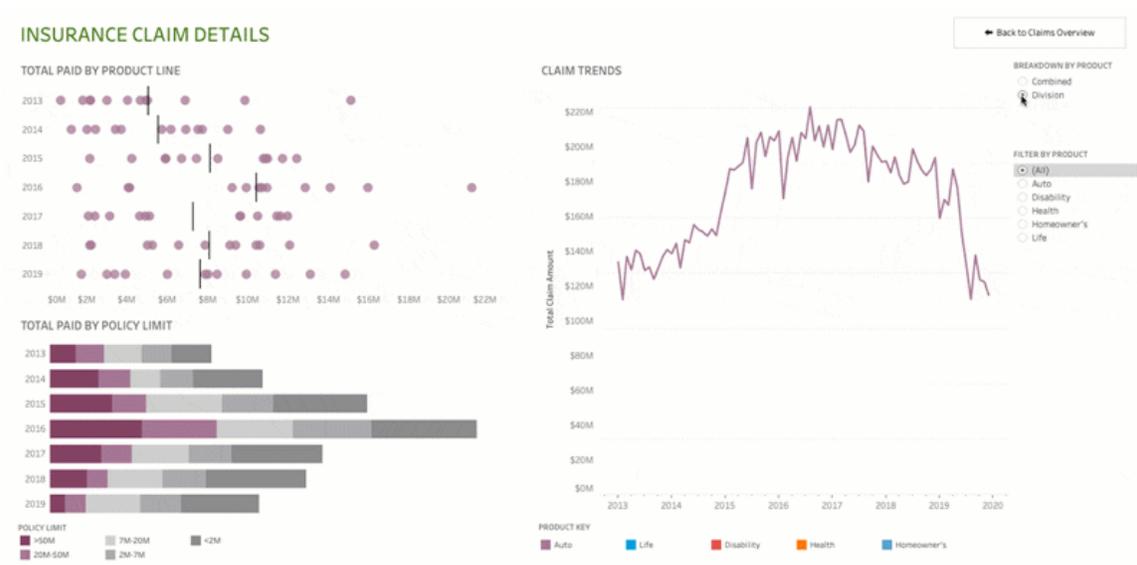
changes, viewers can more clearly see how data differs, helping them make better informed decisions.

Understanding simultaneous and sequential animations

When you author animations, you can choose between two different styles: simultaneous or sequential. Here are examples of each type.

Simultaneous animations

The default simultaneous animations are faster and work well when showing value changes in simpler charts and dashboards.



Click the image above to replay the animation.

Sequential animations

Sequential animations take more time but make complex changes clearer by presenting them step-by-step.



Click the image above to replay the animation.

Animate visualizations in a workbook

When you create a new workbook, Tableau enables animations for your viz by default. You can turn animations on or off at the user and workbook level.

1. Choose **Format > Animations**.
2. If you want to animate every sheet, under **Workbook Default**, click **On**. Then do the following:
 - For **Duration**, choose a preset, or specify a custom duration of up to 10 seconds.
 - For **Style**, choose **Simultaneous** to play all animations at once or **Sequential** to fade out marks, move and sort them, and then fade them in.
3. To override workbook defaults for a particular sheet, change the settings under **Selected Sheet**.

Note: In the Selected Sheet section, “(Default)” indicates a setting that auto-

atically reflects the related Workbook Default setting.

Animations ×

Workbook Default

On Off

Duration
1.00 seconds (Slow) ▼

Style
Simultaneous ▼

Reset All Sheets

Selected Sheet

Heat Map

Animation
On (Default) ▼

Duration
0.30 seconds (Fast) ▼

Style
Sequential ▼

Reset animation settings for a workbook

You can reset animations to return an entire workbook to the default animation settings. Be aware that this turns animations off by default.

1. Choose **Format > Animations**.
2. In the middle of the **Animations** pane, click **Reset All Sheets**.

Completely disable all animations

When you create a new workbook, animations are enabled by default. If you find animations distracting while viewing vizzes, you can completely disable them so they never play. (This isn't a system-wide setting; each user needs to apply it separately.)

- In Tableau Desktop, choose **Help > Settings and Performance**, and deselect **Enable Animations**.
- In Tableau Online or Tableau Server, click your profile image or initials in the top right corner of the browser, and choose **My Account Settings**. Then scroll down to the bottom of the page, deselect **Enable animations**, and click **Save Changes**.

Note: When animations are disabled, you can still choose **Format > Animations** in authoring mode and adjust settings—but they will have no effect.

Why animations won't play

Server rendering

Animations won't play if a viz is server-rendered. To ensure that vizzes render on a client computer or mobile device, use these techniques:

- If you're a viz author, **reduce viz complexity**.
- If you're a Tableau Server administrator, **increase the complexity threshold for client-**

side rendering.

Note: On computers with lower processing power, animations may appear choppy, but users can continue to interact with vizs without any delays in responsiveness.

Unsupported browsers and features

Animations are supported by all web browsers except Internet Explorer.

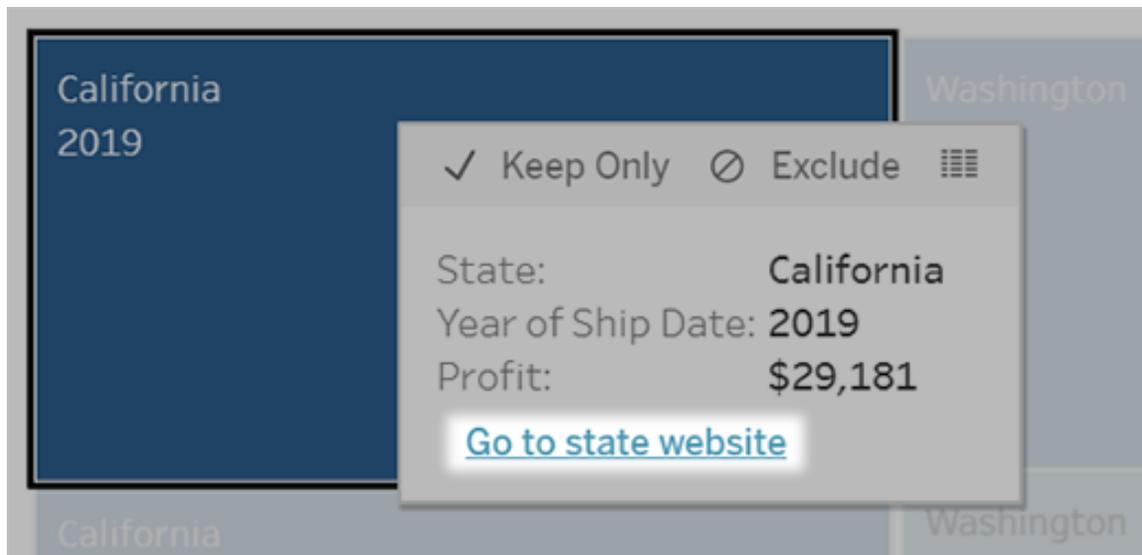
The following Tableau features don't animate:

- Maps, polygons, and density marks in web browsers
- Pie and text marks
- Axes and headers
- Forecasts, trends, and reference lines
- Page history trails (If a viz includes these, turn off animations to avoid unexpected behavior.)

URL Actions

A URL action is a hyperlink that points to a web page, file, or other web-based resource outside of Tableau. You can use URL actions to create an email or link to additional information about your data. To customize links based on your data, you can automatically enter field values as parameters in URLs.

Tip: URL actions can also point to a web page object in a dashboard. See [Actions and Dashboards](#) to learn more.



A URL action run from a tooltip menu. The link reflects the action name, not the target URL.

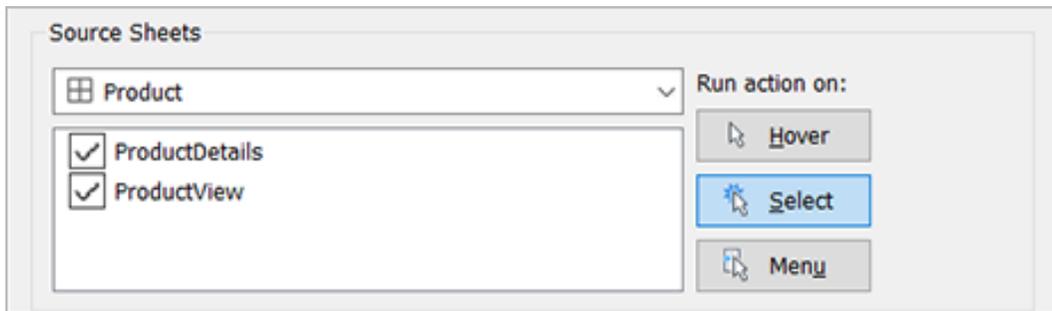
Open a web page with a URL action

Note: Steps on this page refer to the interface of Tableau Online and Tableau Server. The sequence and naming of options differs slightly in Tableau Desktop.

1. On a worksheet, select **Worksheet > Actions**. From a dashboard, select **Dashboard > Actions**.
2. In the Actions dialog box, click **Add Action** and then select **Go to URL**.
3. In the next dialog box, enter a name for the action. To enter field variables in the name, click the **Insert** menu to the right of the **Name** box.

Note: Give the action a descriptive name, because in tooltip menus the link reflects that name, not the URL. For example, when linking to more product details, a good name could be “Show More Details”.

- Use the drop-down list to select a source sheet or data source. If you select a data source or dashboard you can select individual sheets within it.



- Select how users will run the action.

If you choose this option...

Hover	The action is run when the user...
Hover	Mouses over a mark in the view. This option works best for highlight and filter actions within a dashboard.
Select	Clicks a mark in the view. This option works well for all types of actions.
Menu	Right-clicks (control-clicks on Mac) a selected mark in the view, then clicks an option in a tooltip menu. This option works particularly well for URL actions.

- For URL Target, specify where the link will open:

- **New Tab if No Web Page Object Exists** — Ensures that the URL opens in a browser on sheets that lack web page objects. This is a good choice when Source Sheets is set to All or a data source.
- **New Browser Tab** — Opens in the default browser.

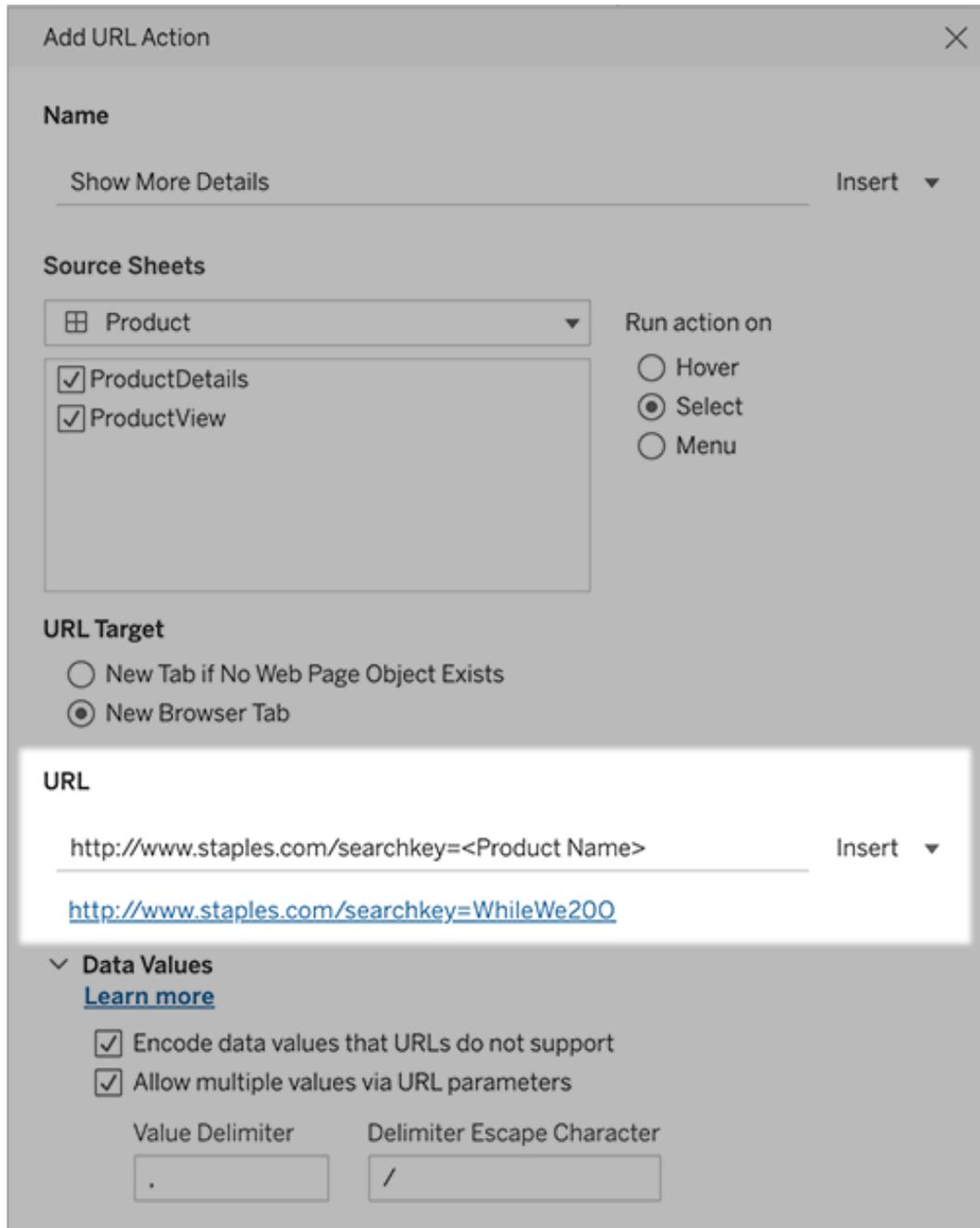
- **Web Page Object** — (Available only for dashboards with Web Page objects)
Opens in the web page object you select.

7. Enter a URL with one of the following prefixes:

- `http, https, ftp, mailto, news, gopher, tsc, tsl, sms, tel, file`
- Tableau Desktop also supports local paths like `C:\Example folder-\example.txt`

To enter field and filter values as parameters in the URL, click the **Insert** menu to the right of the URL. Be aware that any referenced fields must be present in the view, not just a related data source. For details, see [Using field and filter values in URLs](#).

Note: On a dashboard, you can specify an ftp address only if the dashboard doesn't contain a web object. If a web object exists, the ftp address won't load.



Add URL Action

Name

Show More Details Insert ▼

Source Sheets

Product ▼

- ProductDetails
- ProductView

Run action on

Hover

Select

Menu

URL Target

New Tab if No Web Page Object Exists

New Browser Tab

URL

http://www.staples.com/searchkey=<Product Name> Insert ▼

<http://www.staples.com/searchkey=WhileWe200>

▼ **Data Values**

[Learn more](#)

- Encode data values that URLs do not support
- Allow multiple values via URL parameters

Value Delimiter:

Delimiter Escape Character:

Below the URL you enter is a hyperlinked example you can click for testing.

8. (Optional) In the Data Values section, select any of the following options:

- **Encode Data Values that URLs Do Not Support** — Select this option if your data contains values with characters that browsers don't allow in URLs. For example, if one of your data values contains an ampersand, such as “Sales & Finance,” the ampersand must be translated into characters that your browser understands.
- **Allow Multiple Values via URL Parameters** — Select this option if you are linking to a web page that can receive lists of values via parameters in the URL. For example, say you select several products in a view and you want to see each product's details hosted on a webpage. If the server can load multiple product details based on a list of identifiers (product ID or product name), you could use multi-select to send the list of identifiers as parameters.

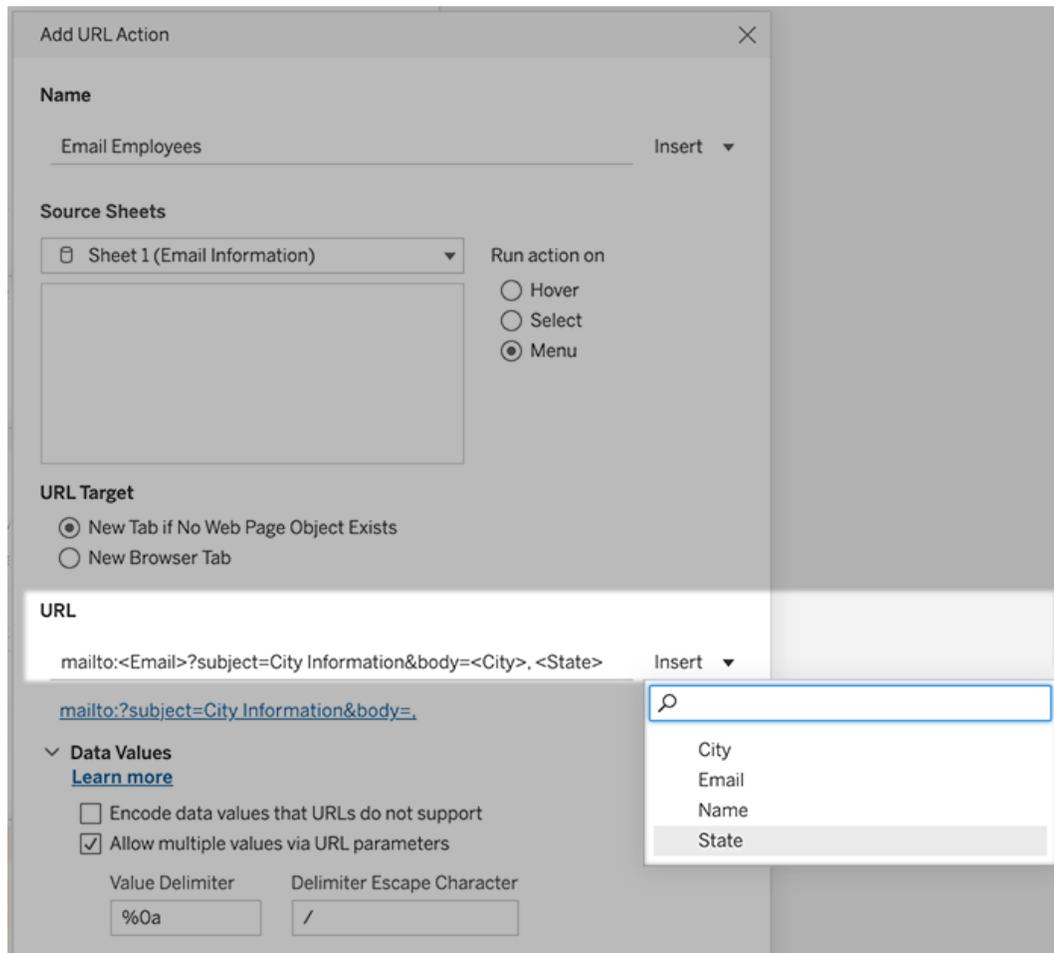
When you allow multiple values, you must also define the delimiter escape character, which is the character that separates each item in the list (for example, a comma). You must also define the Delimiter Escape, which is used if the delimiter character is used in a data value.

Create an email with a URL action

1. On a worksheet, select **Worksheet > Actions**. From a dashboard, select **Dashboard > Actions**.
2. In the Actions dialog box, click **Add Action**, and select **Go to URL**.
3. In the Source Sheets drop-down list, select the sheet that contains the field with the email addresses you want to send to.
4. In the URL box, do the following:
 - Type **mailto:**, and click the **Insert** menu at right to select the data field that contains email addresses.
 - Type **?subject=**, and enter text for the Subject line.

- Type **&body=**, and click the **Insert** menu at right to select the fields of information that you want to include in the body of the email.

In the example below, the “Email” field contains the email addresses, the subject is “City Information”, and the body text of the email consists of the city and state data that is associated with the email address.



:

5. (Optional) Display data from your workbook in the body of your email as a vertical list instead of the default horizontal list. For example, suppose you have a horizontal list of cities, such as Chicago, Paris, Barcelona, which you would rather display

vertically, like this:

Chicago
Paris
Barcelona

To make the list vertical, in the Data Values section, do the following:

- Deselect **Encode Data Values that URLs Do Not Support**
- Select **Allow Multiple Values via URL Parameters**.
- Type **%0a** in the **Value Delimiter** text box to add line breaks between each item in the list. (These are the URL-encoded characters for a line break.)

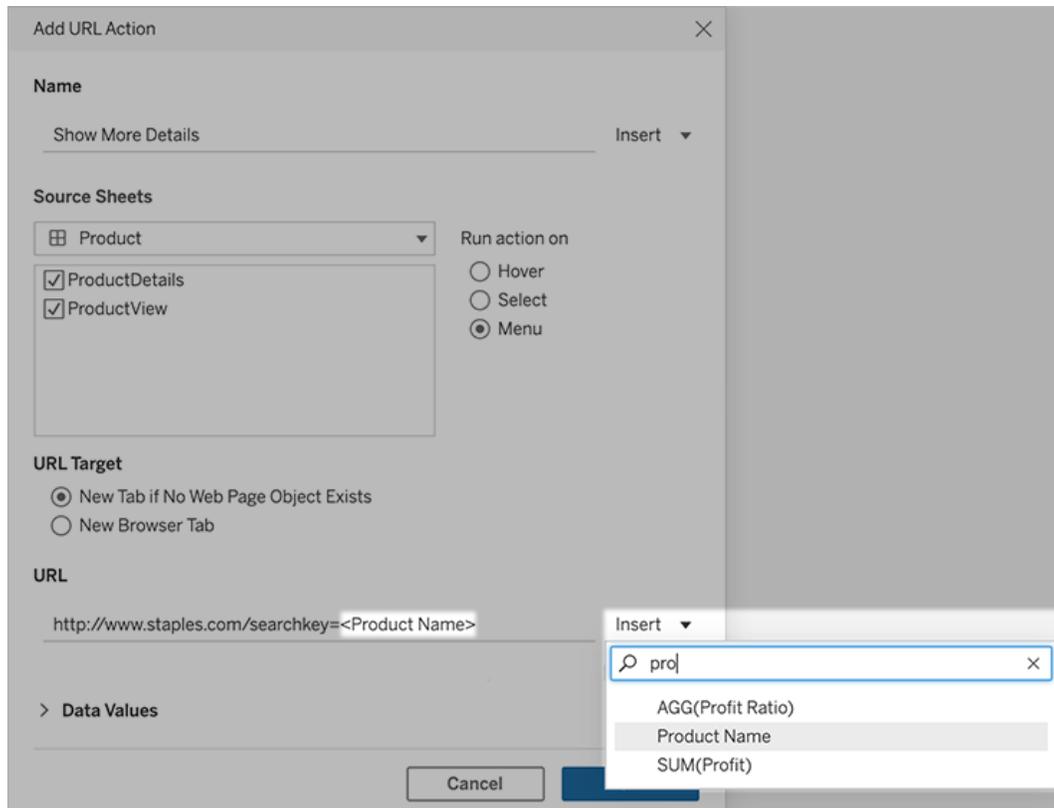
Using field and filter values in URLs

When users trigger URL actions from selected marks, Tableau can send field, filter, and parameter values as variables in the URL. For example, if a URL action links to a mapping website, you could insert the address field to automatically open the currently selected address on the website.

1. In the Edit URL Action dialog box, begin typing the URL for the link.
2. Place the cursor where you want to insert a field, parameter, or filter value.
3. Click the **Insert** menu to the right of the text box and select the field, parameter, or filter you want to insert. The variable appears within angle brackets. You can continue adding as many variables as you need.

Note: Any referenced fields must be present in the view, not just a related data source. Otherwise, the link won't display in the viz, even if it functions when you

click Test Link.



Including aggregated fields

The list of available fields includes only non-aggregated fields. To use aggregated field values as link parameters, first create a related calculated field, and add that field to the view. (If you don't need the calculated field in the visualization, drag it to Detail on the Marks card.)

Inserting parameter values

When inserting parameter values, URL actions send the Display As value by default. To instead send the actual value, add the characters `~na` after the parameter name.

For example, say you have a parameter that includes IP addresses, with Actual Value strings such as 10.1.1.195 and Display As strings with more friendly values such as

Computer A (10.1.1.195). To send the actual value, you'd revise the parameter in the URL to look like this: `http://<IPAddress~na>/page.htm`.

Create a Subscription to a View or Workbook

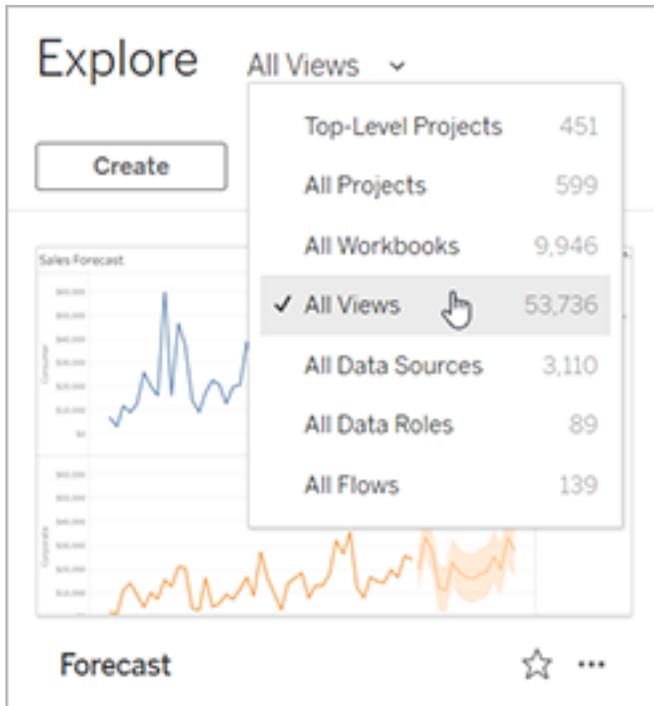
Subscriptions email you an image or PDF snapshot of a view or workbook at regular intervals—without requiring you to sign in to Tableau Server or Tableau Online.

Note: In Tableau Server, administrators determine whether subscriptions are enabled for a site.

Set up a subscription for yourself or others

When you open a view in Tableau Server or Tableau Online, if you see a subscription icon (✉+) in the toolbar, you can subscribe to that view or to the entire workbook. You can subscribe other users who have permission to view the content if you own a workbook, if you are a project leader with an appropriate site role, or if you are an administrator.

1. From the Explore section of your site, select **All Workbooks** or **All Views**, or open the project that contains the view you want to subscribe to.



2. Open a view either directly, or after opening the containing workbook.
3. In the toolbar above the view, click **Subscribe**.

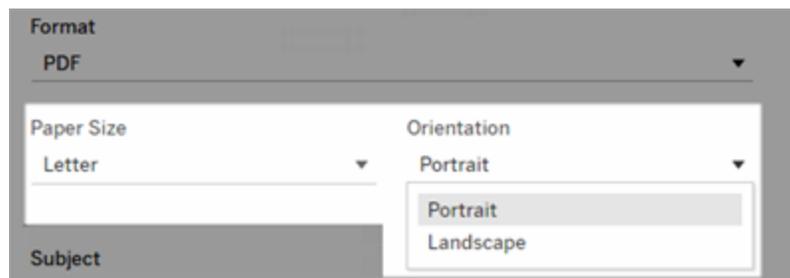


4. Add the Tableau users or groups you want to receive the subscription. To receive a subscription, users must have the **View** and **Download Image/PDF** permissions. If they use Tableau Server, their accounts must also have email addresses.

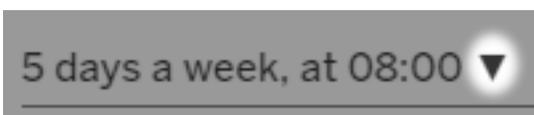
Note: When you subscribe a group, each user is added individually at the time the subscription is created. If more users are added to the group later, you must re-subscribe the group for those new users to receive the subscription. Likewise,

users later removed from the group will not have their subscriptions removed automatically unless their permissions to the subscribed view are removed.

5. Choose whether subscription emails include the current view or the entire workbook.
6. Choose the format for your snapshot: as a PNG image, a PDF attachment, or both.
 - If PDFs, choose the paper size and orientation you'd like to receive.



7. When the workbook uses one data extract on a published connection, you can pick a frequency:
 - **When Data Refreshes:** sends only when data in the view or workbook is refreshed by running refresh schedules.
 - **On Selected Schedule:** Pick a schedule for the subscription.
8. If frequency is not set to When Data Refreshes, pick a schedule:
 - For Tableau Server, choose from subscription schedules established by your administrator.
 - For Tableau Online and Tableau Server with **custom schedules enabled**, click the drop-down arrow to the right of the current settings.



Then specify a custom schedule that sends subscription emails whenever you wish. (The precise delivery time may vary if server load is high.)

Repeats
Hourly ▼

Every:
Hour ▼

From: 09:00 ▼ To: 17:00 ▼

[\(UTC-08:00\) America/Los Angeles](#)

On

S M T W Th F S

To change the time zone, click the Time Zone link it to go to your account settings page.

9. To clarify subscription emails, customize the subject line, and add a message.
10. If the view contains data only when high-priority information exists, select **Don't send if view is empty**.
11. If you own the workbook, select **Subscribe me**.
12. Click **Subscribe**.

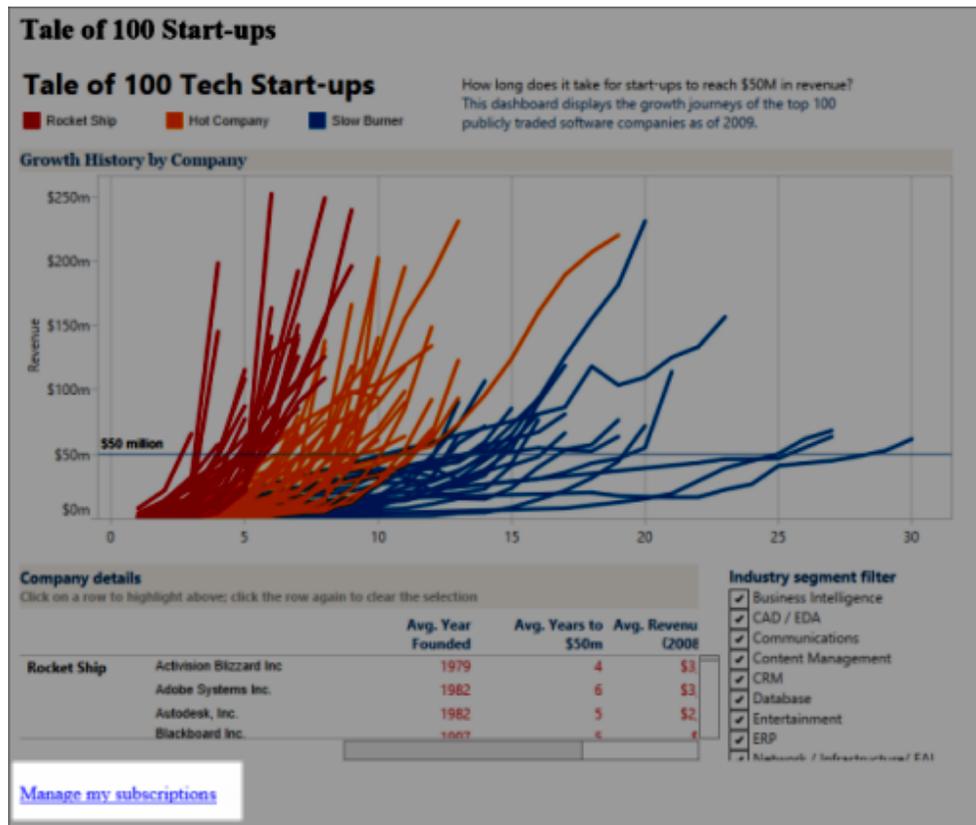
When you receive a subscription email, you can select the image (or the link in the message body for PDF subscriptions) to be taken to the view or workbook in Tableau Online or Tableau Server.

Update subscription settings

Update or unsubscribe from a subscription

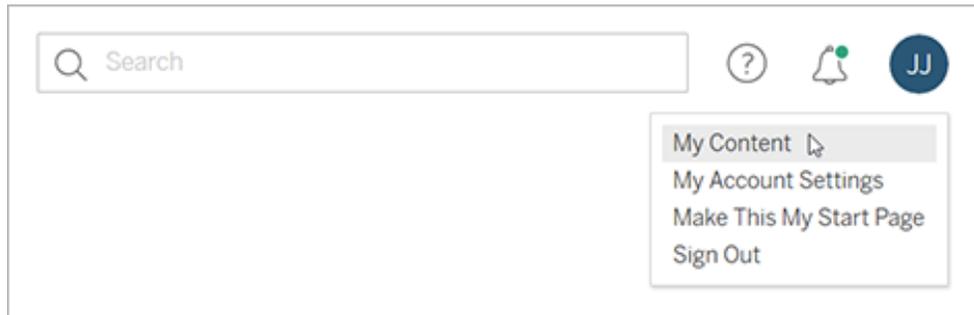
You can unsubscribe from an existing subscription, or make changes to a subscription's format, schedule, subject, or empty view mode.

1. Access your Tableau Server or Tableau Online account settings by doing one of the following:
 - Click **Manage my subscriptions** at the bottom of a subscription email.



- Sign in to Tableau Server or Tableau Online. At the top of the page, select your

user icon, and then select **My Content**.



2. Click **Subscriptions**.
3. Select the check box next to the view you want to unsubscribe from, click **Actions**, and then click **Unsubscribe**, or select the subscription option you'd like to change.

Resume or delete suspended subscriptions

Sometimes, subscriptions fail because of an issue with the workbook or a problem loading the view. If a subscription fails more than five times, you'll receive a notification email that your subscription has been suspended. There are a few ways to resume a suspended subscription if you're a subscription owner or administrator:

- From the My Content area of Tableau web pages, an icon appears in the Last update column to indicate that the subscription is suspended. Select ... > **Resume Subscription** to resume.
- From the Subscriptions tab of the affected workbook, an icon appears in the last update column to indicate that the subscription is suspended. Select ... > **Resume Subscription** to resume.

You'll receive an email notification when the subscription is working again.

See also

[Change Subscription Settings](#) in the Tableau Desktop and Web Authoring Help.

[Project-level administration](#) in the Tableau Online Help, to learn which site roles allow full Project Leader capabilities.

Interact with Data in Tableau

This tutorial walks you through some of the basics of viewing and interacting with data visualizations, or views, in Tableau Server.

Tableau is a tool that lets you interact with published visualizations to explore insights, ask questions, and stay on top of your data. Here's how to get started.

Go ahead. It's safe to click around

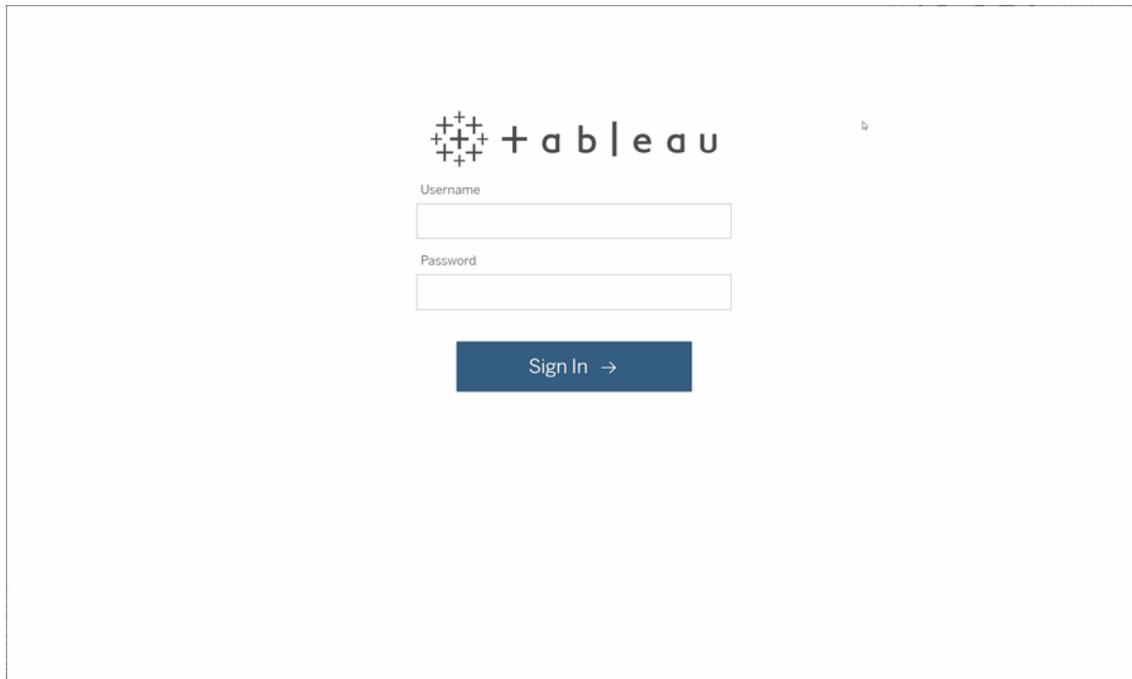
Tableau is built for interaction. What you do to a visualization changes how it looks for you, just for now.

Others will still see the visualization as it originally appeared. And the data used to build it stays the same, too.

1: What is a Tableau Site?

A Tableau Site is a place for your team to share data and data visualizations with each other. You can explore what they've published and made available to you.

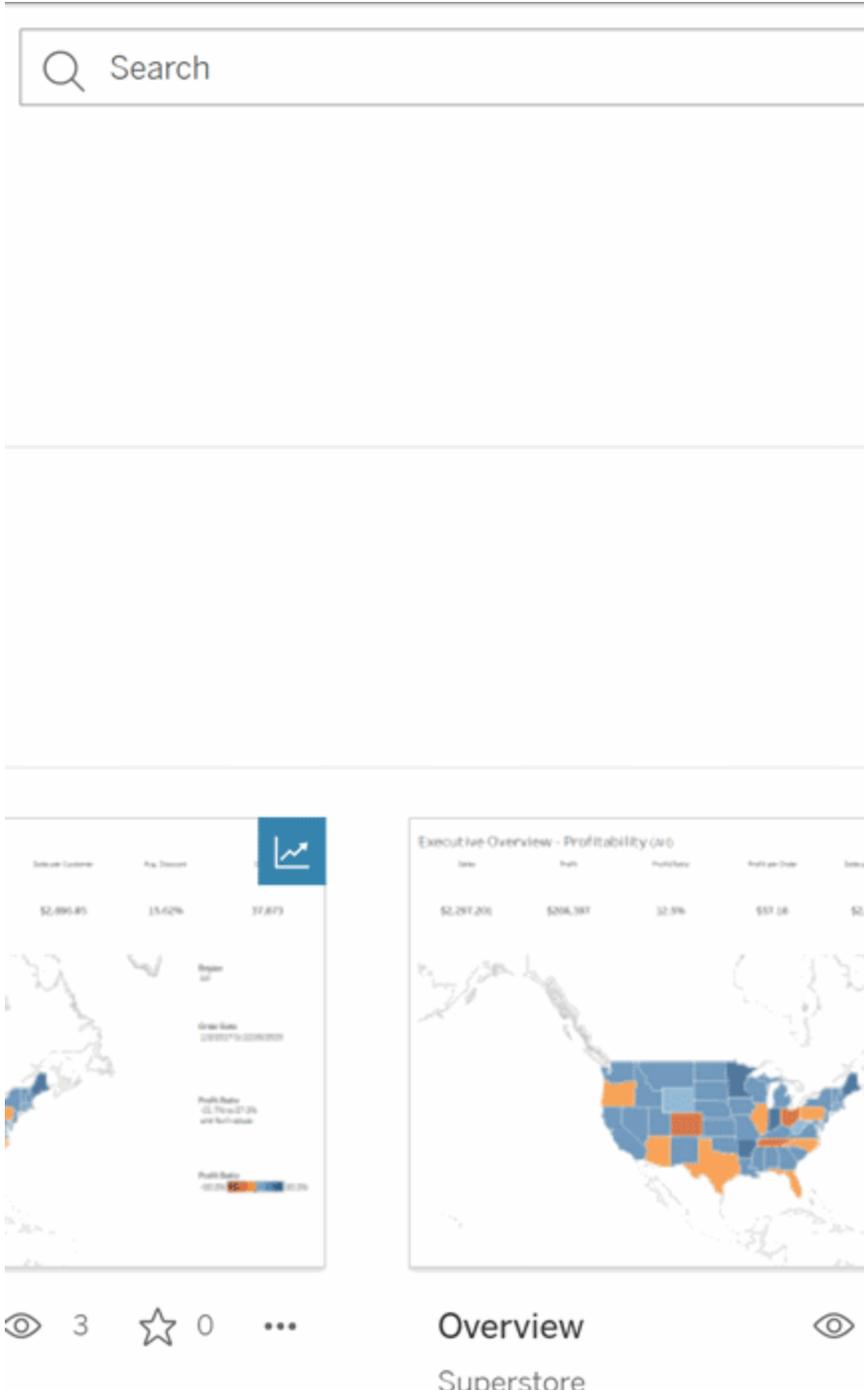
When you sign in to your Tableau site, you'll land on the home page.



2: Search for a viz

Tableau calls visualizations on a site Views. Use search to find views or workbooks (a package of views in a single file).

Search results will show all the different content types relevant to your query.



You can select See All for all search results if the views in the quick search aren't what you were looking for, or use the Explore page to browse. There you'll see all the different types of content a Tableau site can host.

3: Interact with Content

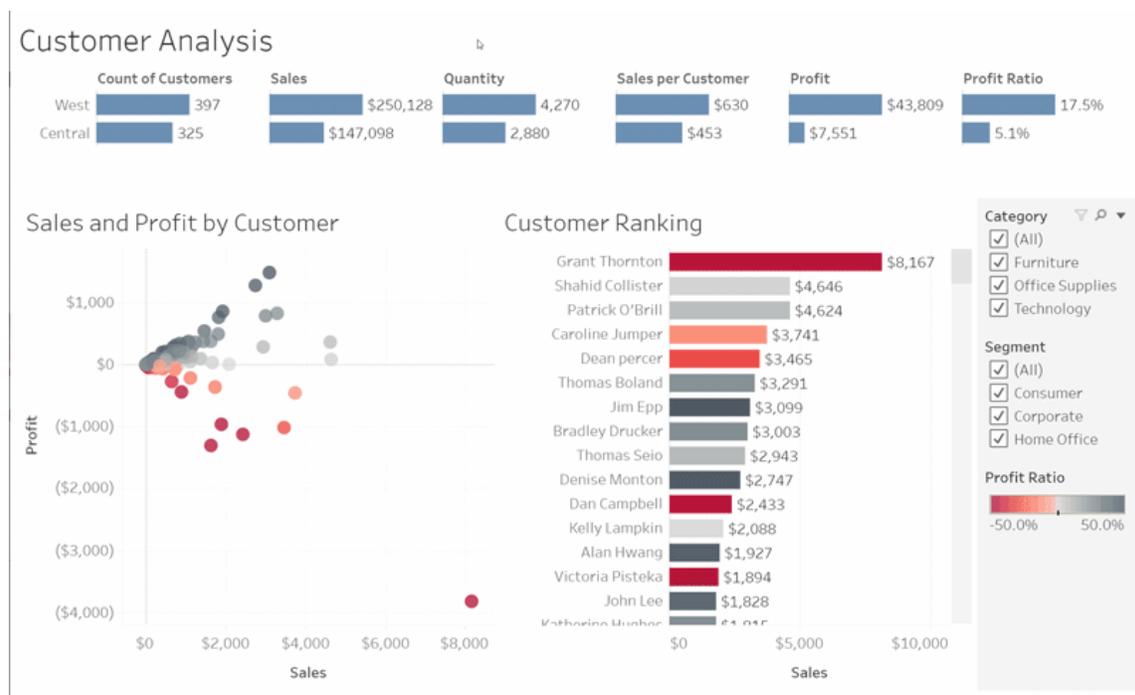
A published view is a canvas for you to interact and understand your data. Remember, you won't hurt or change the underlying data, or change what others see.

Here are some of the tools in your toolbox to find data insights.

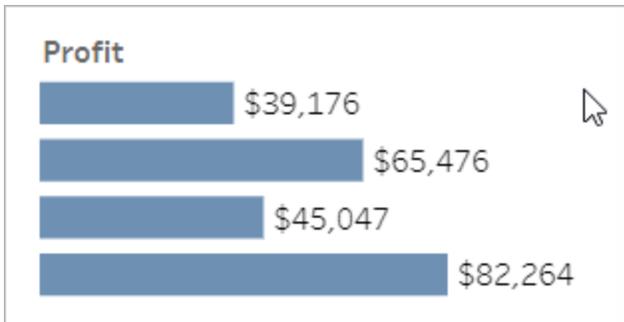
See Details and Sort Data

Now that you know you can click on the data, let's check it out.

As you move the mouse across a view, you might see tooltips that reveal details about each data point, or mark. You can also select multiple marks.

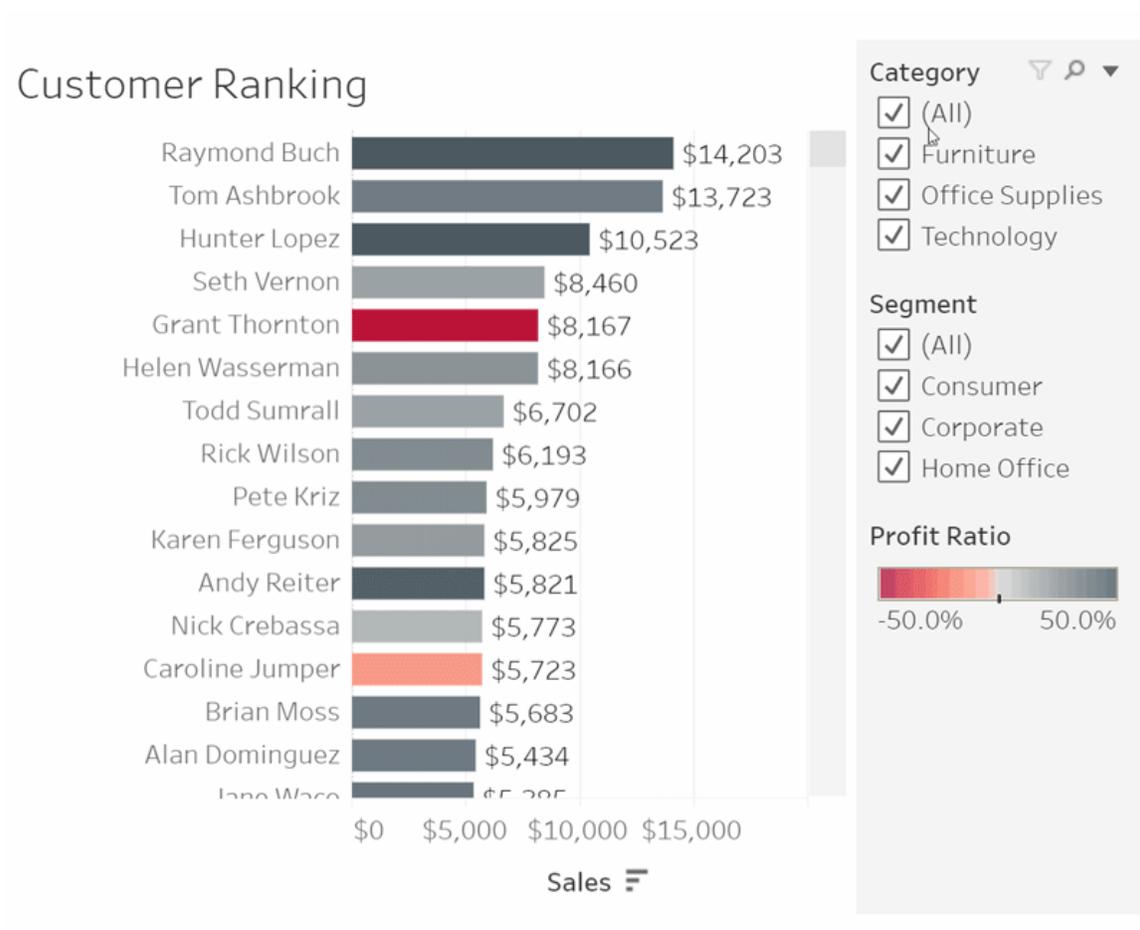


Sort tables alphabetically or numerically by hovering over a column header and clicking the sort icon.



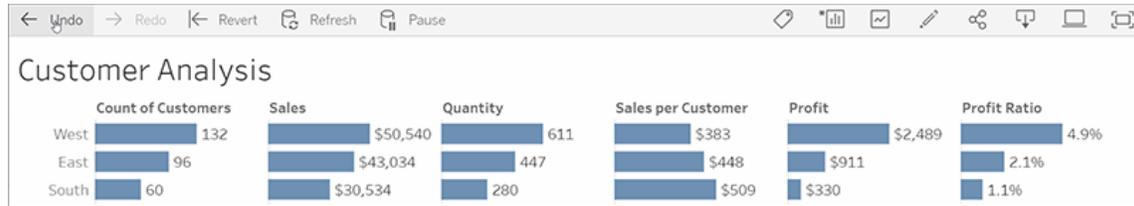
Filter Data

Trim or limit the visible data to a specific area, date, or category.



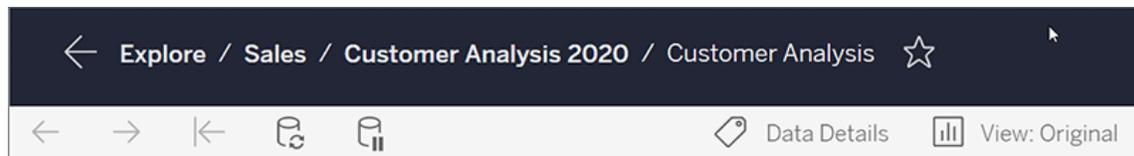
Undo/Revert

Maybe you didn't mean to exclude everything but one area. Click Undo to remove the last change, or use Revert to undo all your selections.

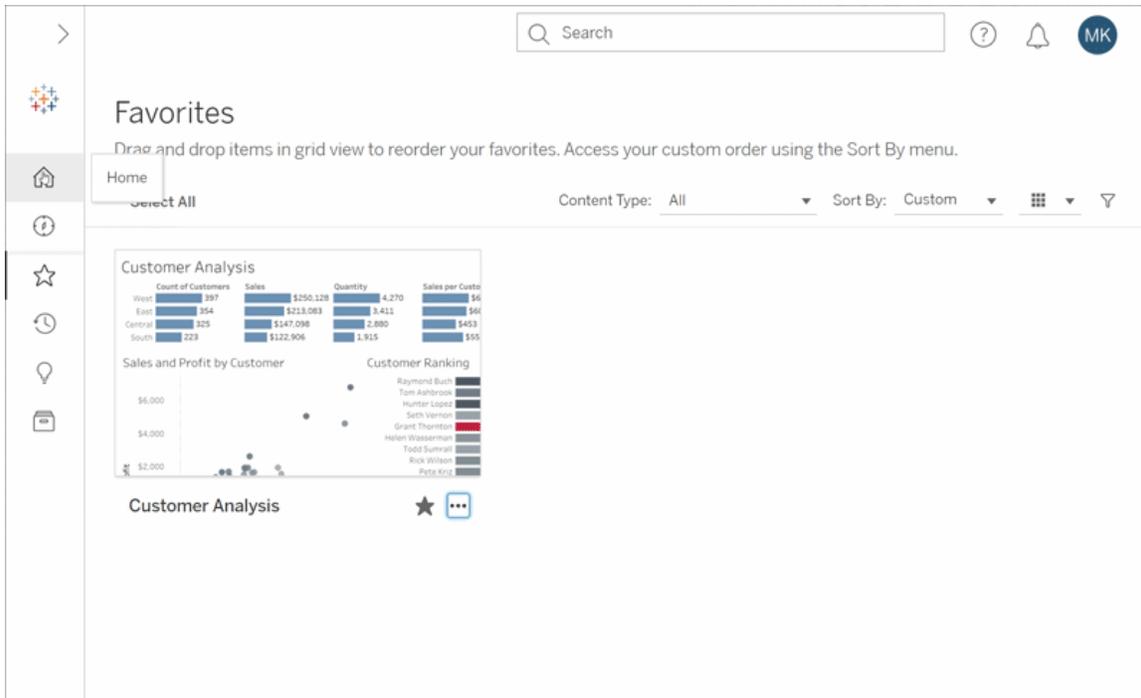


4: Keep up

This view can automatically update with new data, so you don't need to go searching for new charts to get the latest information. Keep it handy by clicking the star icon to add it to your favorites.



All favorites are added to the Favorites page in the navigation panel. And any recent dashboard or view you explored also appears in the Home page, waiting for you next time.



There's a lot you can do in Tableau Server, and this just explored some of the basics. For more information, see [What Can I Do with a Tableau Web View?](#)

Happy exploring!

Select Background Maps

Tableau comes with a set of online and offline background maps that you can access to create map views.

By default, Tableau connects to an online map provider.

The three background maps that come with Tableau are described below. They can be found in the **Maps > Background Maps** menu. If the workbook author has added a WMS server or Mapbox map to the workbook, they will be listed using the name the author gave them in the **Maps > Background Maps** menu as well.

None Displays data between latitude and longitude axes.

- Offline** This background map stores the images that make up the map in a cache on your machine for improved performance and offline access. For more information, see the offline maps section [below](#).
- Tableau** Connects to the Tableau background map. By default, all map views connect to this background map, unless you specify otherwise.

To select a new background map:

- In Tableau Desktop, Tableau Server, or Tableau Online, select **Map > Background Maps** and then select the background map you want to use.

Note: In Tableau Server, your administrator can restrict you to using only Offline maps. The specific server commands to configure this behavior can be found [here](#).

Set a default background map in Tableau Desktop

By default, all map views connect to the Tableau background map. In Tableau Desktop, if you do not want maps views to automatically be created with the Tableau background map, you can specify a different background map as the default.

To specify a default background map:

1. In Tableau Desktop, select **Map > Background Maps**, and then select the background map that you want to make the default.
2. Select **Map > Background Maps > Set as Default** to set the selected background map to the default.

The background map is automatically saved as a Tableau Map Source (.tms) and placed in the Mapsources folder of your My Tableau Repository. It is now the default background map for all new worksheets.

Use the Offline background map in Tableau Desktop, Online, or Server

You can create and inspect data in a map view offline using the offline background map that comes with Tableau Desktop.

To use the offline background map:

- In Tableau, select **Map > Background Maps > Offline**

Note: The offline background map uses map images stored on your machine. You can find these images in the following locations:

- **On Windows:** `C:\Program Files\Tableau\<Tableau Version>\Local\Maps`
- **On Mac:** `//Applications/<Tableau Version>.app/Contents/install/local/maps`

There are several actions, however, that require Tableau to retrieve a map image that may not be stored. If the new map image is not stored on your machine, you won't be able to load the map until you reconnect to the online map that comes with Tableau.

You may need to reconnect to the online map if you would like to do one or more of the following:

- **Turn layers on or off** - if you decide to turn on a layer that isn't stored in the cache, Tableau will need to connect to retrieve the necessary information.
- **Zoom in or out** - zooming in or out on a map requires different map images. If the images at the specified zoom level don't exist in the cache, Tableau will need to retrieve the updated maps.
- **Pan** - panning sometimes requires new map images. If you are working offline and don't have the necessary map images and legends stored in the cache, the new images and legends will not load.

To reconnect to the Tableau online map:

- On Tableau Desktop, select **Map > Background Maps > Tableau**

Note: If you set a map to Offline and then publish the workbook, the published workbook will still use the Offline, stored maps, with all of the functionality and limitations of Offline maps noted above.

About the Tableau background map

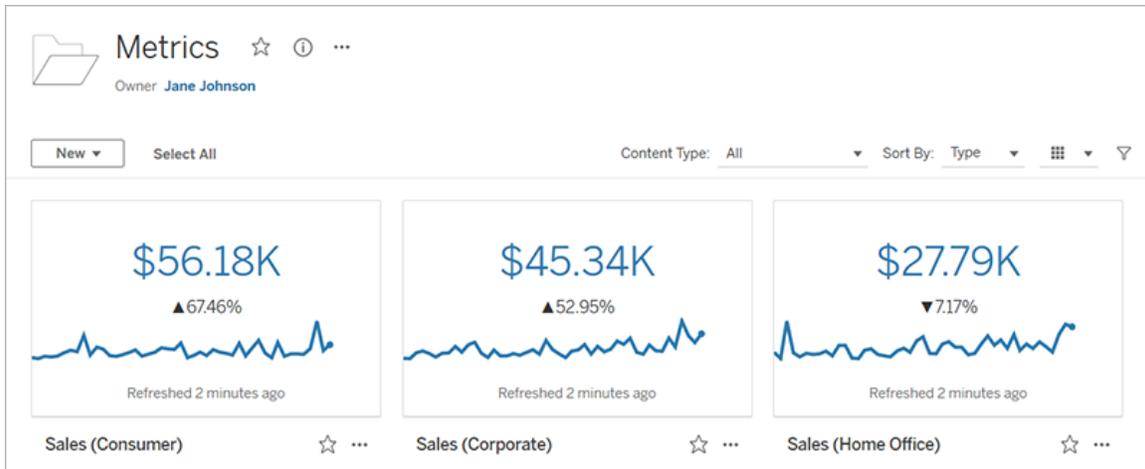
The Tableau background map includes updated map and demographic data, as well as stylistic improvements. It has taken the place of the Tableau classic background map. Any map view created with an earlier version of Tableau Desktop, and with a connection to the Tableau classic background map, will automatically connect to the Tableau background map when opened in Tableau Desktop version 9.0 or later.

Note: If you create a map view with data layers in Tableau Desktop version 8.3 or earlier using the Tableau classic background map, and then open that view in Tableau Desktop 9.0 or later, the color of the data layers may appear differently due to data layering and color ramp (color palette) improvements made to the Tableau background map.

Create and Troubleshoot Metrics

Metrics help you track changes to your data. You can quickly monitor key numbers, because metrics update automatically and display their current value on the grid and list view for your content. This means that you don't need to open a metric to see your data.

If you have multiple dashboards you frequently check, create metrics for the numbers you want to monitor, then track them in one place by adding them to your favorites or creating them in the same project.



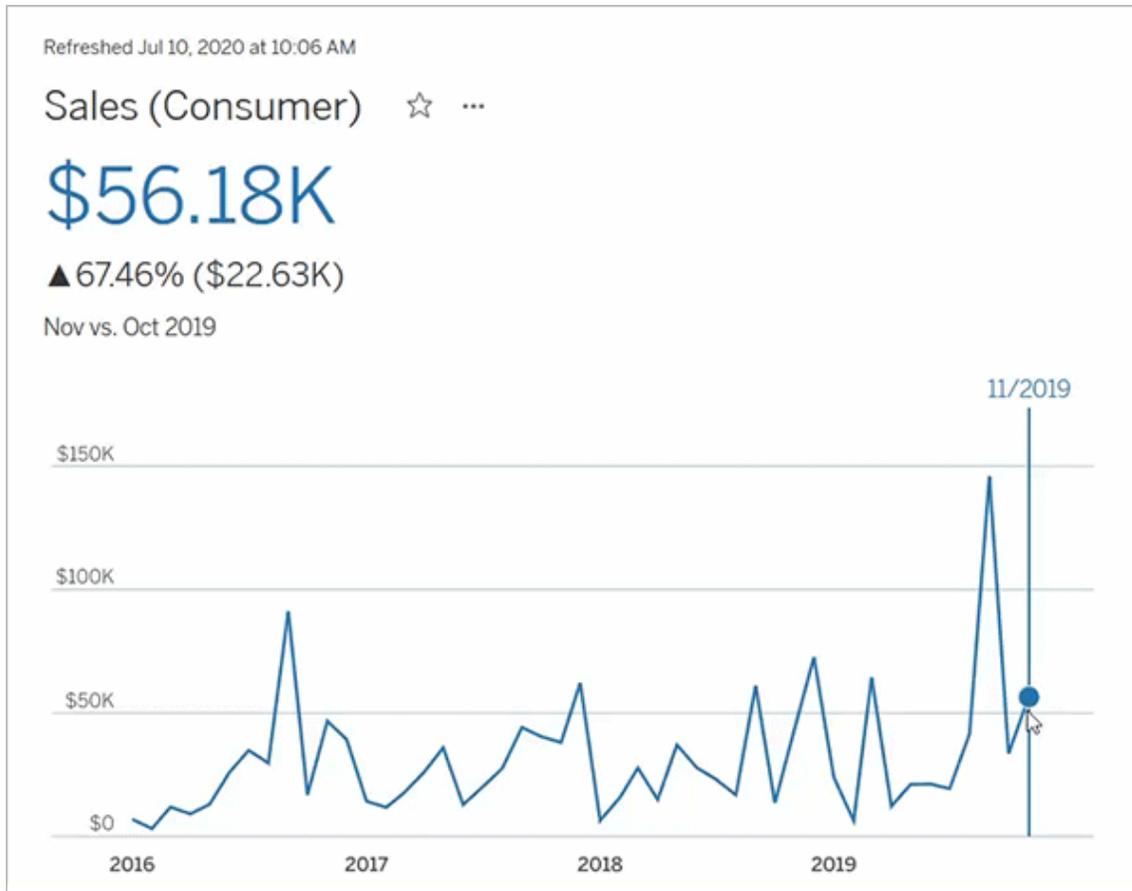
How metrics are defined

A metric's value is defined by an aggregate measure, such as sum of sales, and an optional date dimension, such as order date. For information about dimensions and measures, see [Dimensions and Measures, Blue and Green](#). Any filters applied to the measure before the metric was created will apply to the metric value, for example, filters for segment or region.

Timeline metrics and single-number metrics

A metric that has a date dimension as part of its definition will have a timeline. Because metrics are defined based on the mark you select when you create a metric, you include a timeline by selecting a mark on a chart with a date axis or header.

When you open a metric, the timeline shows the value of the measure based on the granularity of the date dimension, for example, daily sales or monthly users. Move your mouse along the timeline to see historical values.



If the mark you select doesn't have a date dimension associated with it, your metric will be a single-number metric. Single-number metrics still refresh, and their values update when the data changes, but they don't show previous values on a timeline.



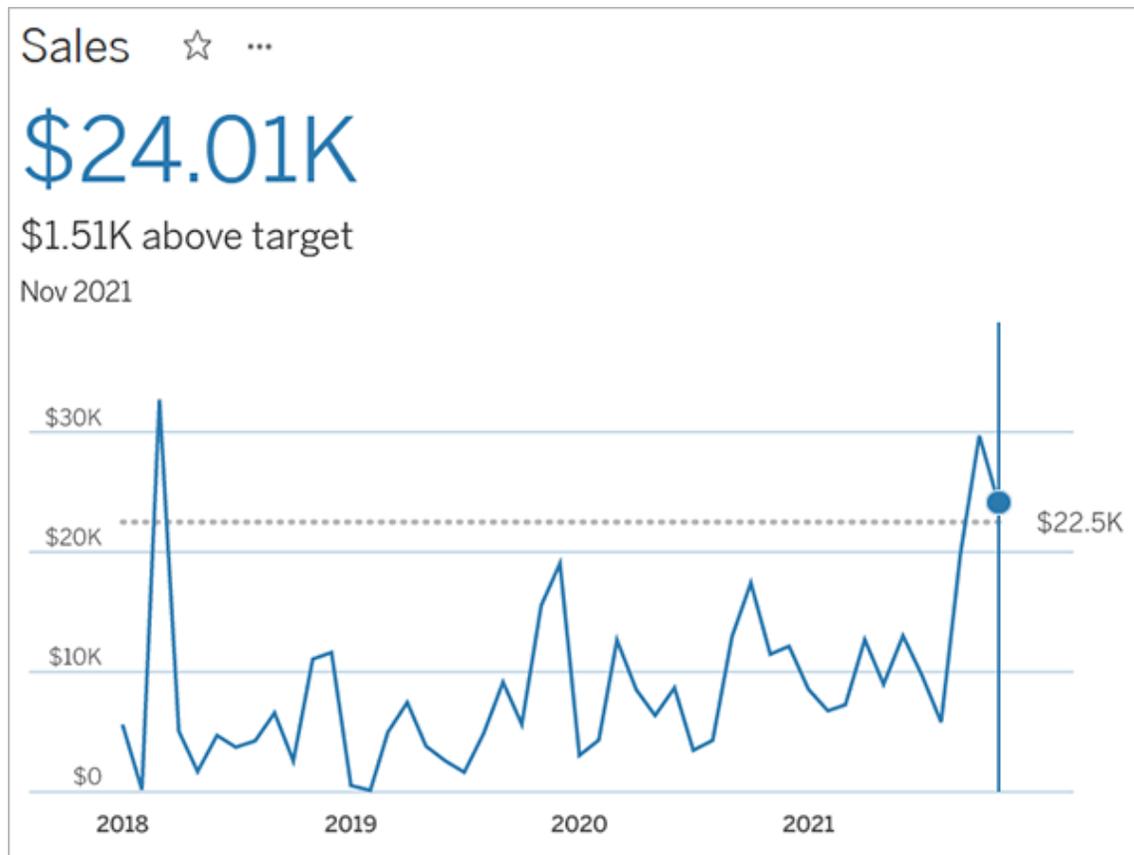
Tableau Server on Linux Administrator Guide

Metrics with target values

If a metric has a target value assigned to it, you will see a comparison to the target value below the current metric value, both on the metric card and on the metric details. The creator of a metric might include a target value to represent a threshold to stay above, for example, if you need to maintain a 90% on-time delivery rate. Or a target value might represent a cumulative goal you are working toward, for example, a monthly sales target.

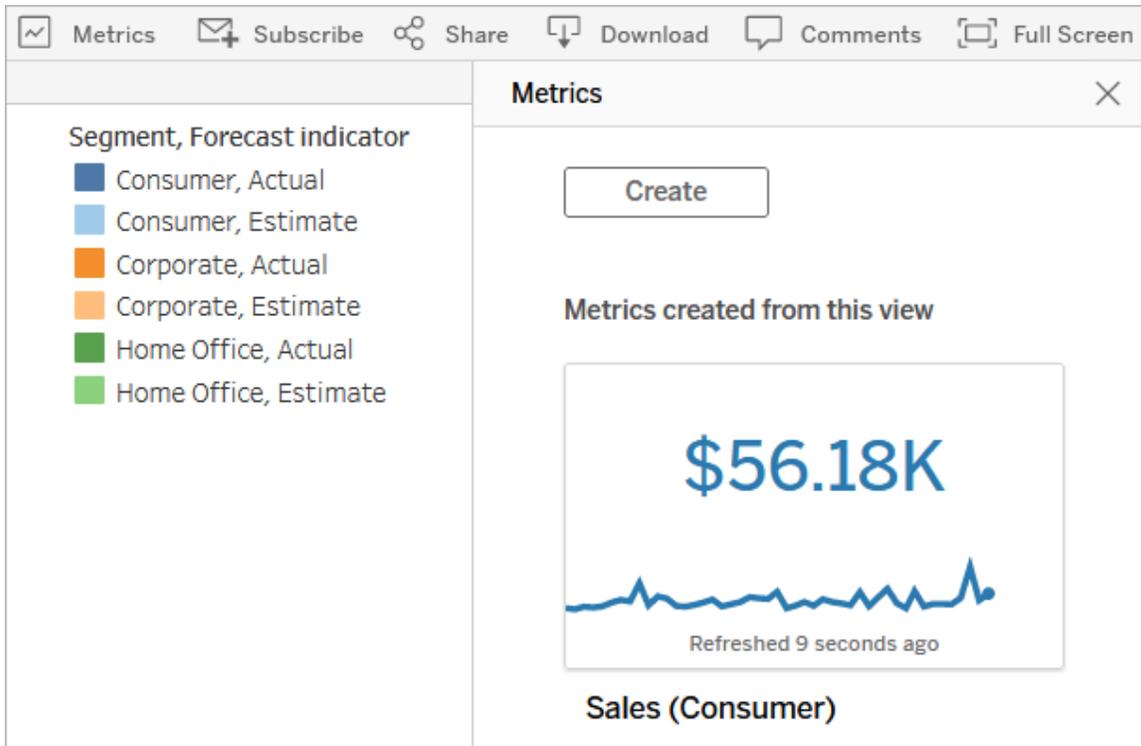


Timeline metrics display the target value as a reference line on the timeline. When a timeline metric includes a target value, the comparisons shown for each point on the timeline are to the target value, rather than to the previous point in time.

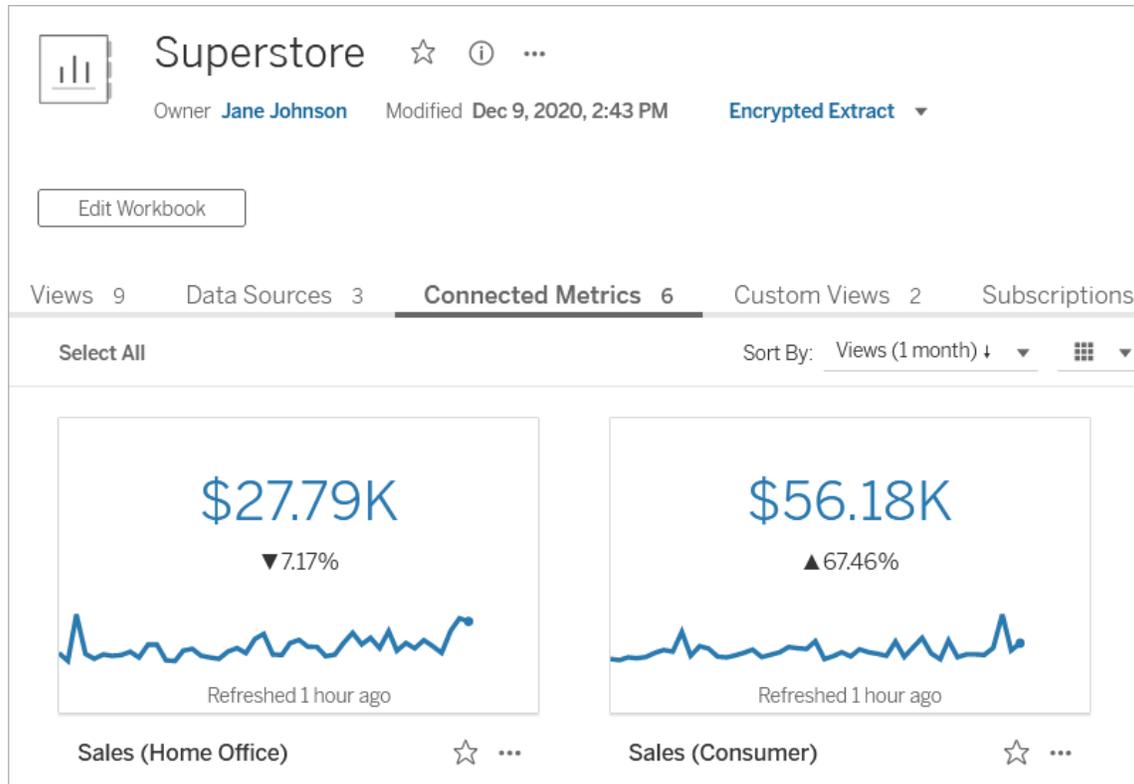


See connected metrics

Connected metrics are the metrics that were created from a particular view or workbook. To see connected metrics for a view, click the **Metrics** button in the view toolbar. The metrics displayed are ordered from the newest creation date to the oldest. Users with all site roles can see connected metrics for a view. However, the Create button will appear only if you have a site role of Creator or Explorer (can publish). For more information, see [Create a metric](#).



To see connected metrics for a workbook, click the **Connected Metrics** tab. You can sort these metrics using the Sort By menu.



Before creating your own metric from a view, it's a good idea to check the connected metrics. Another user might have already created the metric you want. Instead of creating a duplicate metric, open the existing metric and add it as a favorite.

Create a metric

If you have a site role of Creator or Explorer (can publish), you can create metrics on Tableau Online or Tableau Server. Metrics are formatted automatically and optimized for viewing on both mobile and desktop devices.

1. Navigate to the view that you want to create a metric from.
2. On the view toolbar, click the **Metrics** button.

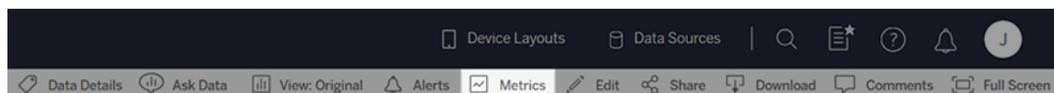
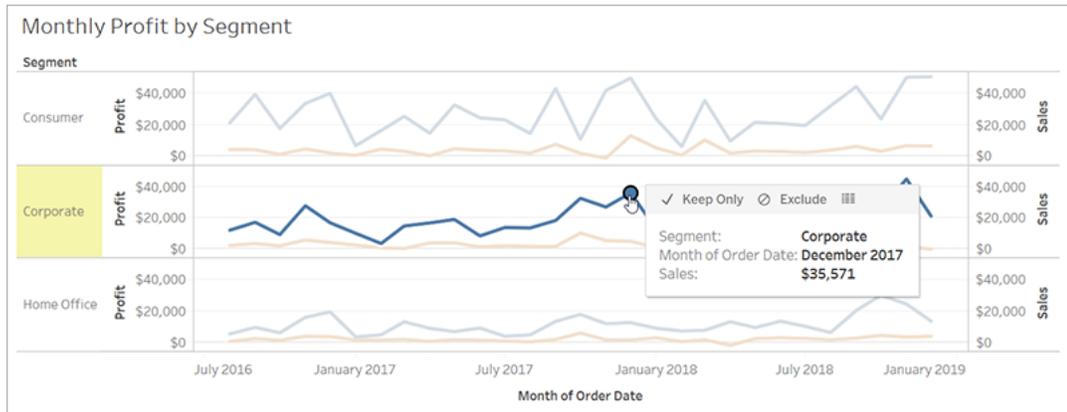


Tableau Server on Linux Administrator Guide

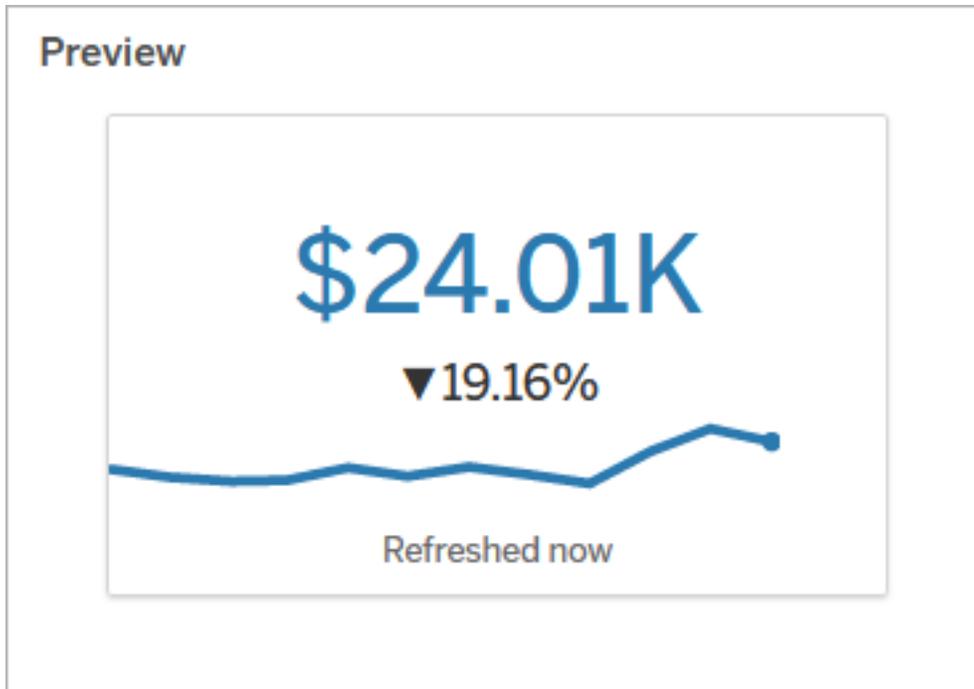
The metrics pane opens.

3. If the pane shows connected metrics, select the **Create** button to enter authoring mode.
4. Select a mark. If you encounter an error, see *When you can't create a metric*.

The measure, and optional date dimension, associated with this mark will define your metric. Any filters you apply to this mark will be applied to your metric.



The metrics pane shows a preview of your metric. The value in the preview is the most recent value for the metric, which may differ from the value of the mark you selected in the case of metrics with a date dimension.

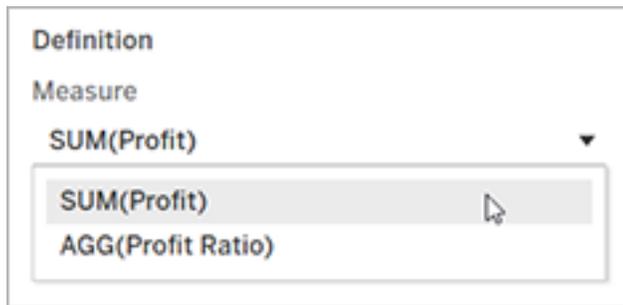


5. The **Name** field is pre-populated based on your selection. You can give the metric a different name. A metric must have a unique name within the project it belongs to.
6. Under **Description**, enter an optional message to help others understand your metric. For example, describe filters applied to the metric or indicate the data source used by the metric.
7. Under **Target value**, enter an optional number to compare the metric against. Don't include commas or symbols in this field. If your target is a percentage, simply type the number without the percent sign, for example, enter 25 instead of 0.25 for a target of 25%.

When you enter a valid target value, the preview updates to show how far above or below the target the current value is. Make sure to note what your target value represents in the metric description.

8. Under **Definition** > **Measure**, select the measure to use for your definition from the

drop-down. This option appears only if the mark you selected has more than one measure associated with it.



9. Under **Project**, click **Change Location** to choose a different project for the metric. By default, your metric will be added to the same project that the view belongs to.
10. Click the **Create** button.

A message appears with a link to the metric in the project you added it to.



Though metrics are created from a view, they exist independent of that view. This means you manage metrics similar to how you manage workbooks. For example, you can set permissions for a metric or move it to another project. For information about managing content on your Tableau site, see [Manage Web Items](#).

Overwrite a metric

1. Create a metric with the same name in the same project as the metric you want to overwrite.

The Overwrite Metric dialog appears.

2. Click the **Overwrite** button.

When you overwrite a metric, the metric continues to appear for those who have added it to their favorites, and any changes made to permissions for the previous metric will apply to the new metric.

When you can't create a metric

If you select a mark on a chart that doesn't support metrics, you will get an error message explaining why you can't create a metric. The table below summarizes these scenarios.

Reason	Scenarios
You don't have access to the full data for the chart.	<ul style="list-style-type: none"> • Row level security or user filters limit the data you can see. For more information, see Restrict Access at the Data Row Level. • You don't have permission to see the full data. For more information, see View Underlying Data.
The password for the workbook's data source is not embedded or is no longer valid. For more information, see Set Credentials for Accessing Your Published Data .	<ul style="list-style-type: none"> • The workbook prompts for a password. • Credentials are passed using single sign-on (SSO).
The data isn't at the correct level of granularity.	<ul style="list-style-type: none"> • The data in the chart isn't aggregated. Metrics use aggregations, such as sum or average. For more information, see Data Aggregation in Tableau. • There are multiple values per cell of data—a result of data blending. For more information, see Troubleshoot Data Blending.
The date dimension is not supported.	<ul style="list-style-type: none"> • The chart includes both date parts and date values. For more information, see Change Date Levels.

Reason

Scenarios

- The date dimension uses the ISO 8601 calendar rather than the standard Gregorian calendar. For more information, see [ISO-8601 Week-Based Calendar](#).
- The date dimension is aggregated at the custom level of Month / Year or Month / Day / Year. For more information, see [Custom Dates](#).

How metrics refresh

When a metric refreshes, it checks the connected view (the view the metric was created from) for new data. A refresh doesn't necessarily update the value of a metric, because there may be no changes to the data.

Metrics refresh at a frequency either based on an extract's refresh schedule or, for live data, every 60 minutes. The time of the last refresh is displayed on the metric.

Fix failing refreshes

If a metric isn't able to access the connected view or its underlying data, the refresh will fail. If the refresh for your metric fails, you'll receive an email notification, which notes the time of the failure and the affected metric.

Metric refreshes may fail for one of the following reasons.

- The connected view was deleted or modified.
- Permissions changed for the connected view.
- The password for the data source is no longer embedded or is no longer valid.
- The metric owner doesn't have the required site role to refresh the metric. A site role of Creator or Explorer (can publish) is required.
- There was a temporary connectivity issue, which will resolve itself.

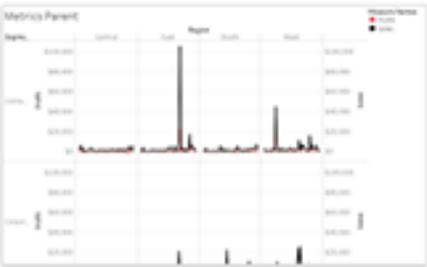
To identify the cause of the failure, look at the metric details. Make sure that the metric owner has the required site role to refresh the metric. Then inspect the **Connected View**.

Metric Details

Owner
Jane Johnson

Connected View

The metric has been created from this view:



Weekly Sales and Profit

Definition

Measure	SUM(Profit)
Date Dimension	WEEK(Order Date)

If the connected view is still listed

Open the view to investigate the cause of the failure.

Evidence	Cause	Solution
The workbook prompts for a password or displays an error when connecting to the data source.	The password for the data source is not embedded or is no longer valid.	The content owner or a Tableau administrator can edit the data source connection to embed the password. For more information, see Edit Connections .

Evidence	Cause	Solution
You can't view the full data, following the instructions on View Underlying Data .	You no longer have permission to access to the full data.	The content owner or a Tableau administrator can change the permissions for the view. For more information, see Permissions .
The measure or date dimension that defines the metric is no longer present.	The view has been modified so the metric can't connect to the data needed to refresh.	The content owner or a Tableau administrator can check the revision history and restore previous versions. For more information, see View Revision History

If there is no connected view listed

The view was deleted or you no longer have permission to access the view. Contact your Tableau administrator for assistance.

Resume suspended refreshes

If a refresh fails enough times, the refresh is suspended. You'll receive an email notification if the refresh for your metric is suspended.

When a metric refresh is suspended, Tableau no longer attempts to get new data for the metric. Metrics with suspended refreshes continue to present historical data.

If the cause of the failure is fixed, you can resume the refresh.

1. Open the affected metric.
2. On the warning message, click **Resume refresh**.

Tableau attempts to perform the refresh. If this attempt succeeds, you'll receive a notification, and the refresh will resume on schedule. If the attempt doesn't succeed, your refresh remains suspended.

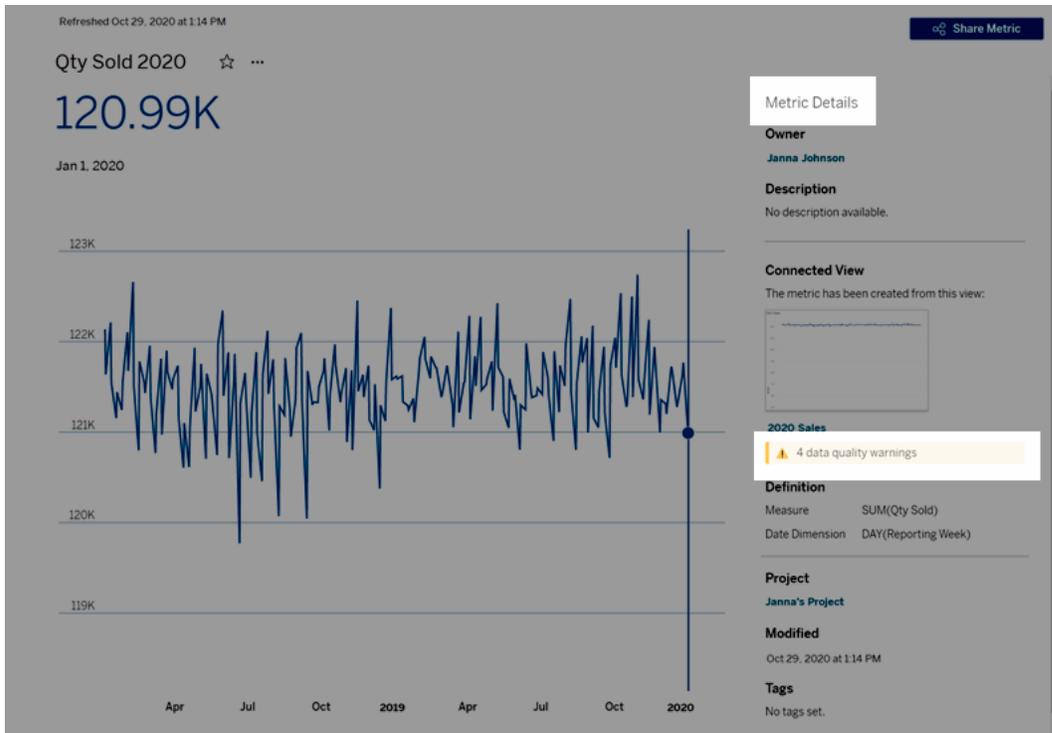
Try overwriting the metric if the connected view is still available. For more information, see [Overwrite a metric](#). Otherwise, you can keep the metric to reference past data or delete the metric.

Note: If the metric refresh is suspended because you don't have the required site role for it to refresh, you won't be able to resume the refresh or delete the metric.

Metrics Appear in Tableau Catalog

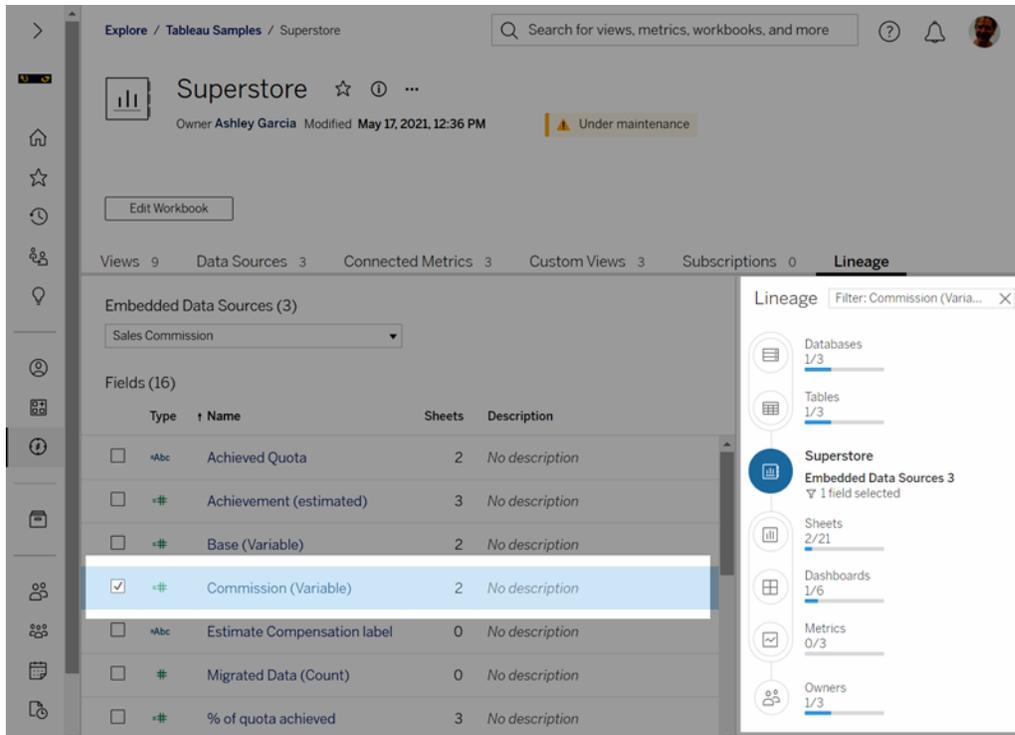
Starting in 2019.3, Tableau Catalog is available in the Data Management Add-on to Tableau Online and Tableau Server. When the Tableau Catalog is enabled in your environment, you can see metrics in the Catalog lineage tool, and metrics affected by data quality warnings display those warnings. For more information about Tableau Catalog, see "About Tableau Catalog" in the [Tableau Server](#) or [Tableau Online](#) Help.

When you have metrics defined for the numbers you want to monitor, it's important to know if the data that the metrics rely on are impacted in any way. You can use Catalog in a couple of ways to know this. First, data quality warnings set on the data your metric is based on show on the metric. These warnings appear when you open a metric in Tableau Mobile, and in Tableau Server and Tableau Online when you hover over a metric in grid view and on the metric details page, as shown below:



For more information, see "Set a Data Quality Warning" in the [Tableau Server](#) or [Tableau Online](#) Help.

You can also use the lineage tool in Tableau Catalog to see the upstream sources that your metric relies on. When doing an impact analysis, you can see which metrics are affected if a certain column or a table is changed or deprecated, or if certain workbooks are removed. Including metrics in the lineage means that Catalog gives a full picture of the impact changes can have on the assets in your environment.



For more information, see "Use Lineage for Impact Analysis" in the [Tableau Server](#) or [Tableau Online](#) Help.

Manage Server

After installing Tableau Server, you can customize and manage your server. For example, you can manage security, licenses, sites, subscriptions and data-driven alerts, and more.

Security	1072
Supported data sources	1223
Create a Connected Salesforce App	1232
Configure Tableau Server for Salesforce.com OAuth	1234
Configure custom OAuth for a site	1235
Manage access tokens	1238
Obtain a Client ID and Enable Google APIs	1239
Configure Tableau Server for Google OAuth	1241
Configure custom OAuth for a site	1241
Create and edit Google data source	1244
Managing access tokens	1245
Step 1. Register OAuth application for Azure	1245
Step 2: Configure Tableau Server for Azure	1245
Configure custom OAuth for a site	1248
Step 1: Obtain a client ID with Snowflake	1251
Step 2: Configure Tableau Server for Snowflake OAuth	1252
Configure custom OAuth for a site	1253
Create a Connected Intuit App	1256

Configure Tableau Server for Intuit QuickBooks Online	1257
Managing access tokens	1257
Managing credentials centrally	1259
See also	1260
Conflict error	1260
Supported data sources	1267
Manage Licenses	1382
Add Users to Tableau Server	1438
Sign in to the Tableau Server Admin Area	1442
Sign in to Tableau Services Manager Web UI	1449
Customize Your Server	1451
Manage Sites Across a Server	1455
Extract Refresh Schedules	1477
Managing Background Jobs in Tableau Server	1506
Tableau Service Manager Jobs	1513
Administrative Views	1516
Performance	1566
Monitoring Tableau Server	1615
Maintenance	1631
tsm Command Line Reference	1692
Entity Definitions and Templates	1920
tabcmd	1967

Troubleshooting	2108
Server Administrator Reference	2171
About Tableau Advanced Management	2313
Who can do this	2402
Agents Unlicensed	2403
Incompatible Agent Version	2404
Agent Down	2407
Who can do this	2408
Who can do this	2409
Use the Master Server web interface	2409
Use the configuration file (config.json)	2412
Who can do this	2414
Use the Master Server web interface	2415
Use the configuration file (config.json)	2415
Who can do this	2417
Use the Master Server web interface	2417
Use the configuration file (config.json)	2418
Who can do this	2420
About Data Management Add-on	2688

Security

As a part of managing Tableau Server, you can configure authentication, data security, and network security.

Authentication

Authentication verifies a user's identity. Everyone who needs to access Tableau Server—whether to manage the server, or to publish, browse, or administer content—must be represented as a user in the Tableau Server repository. The method of authentication may be performed by Tableau Server (“local authentication”), or authentication may be performed by an external process. In the latter case, you must configure Tableau Server for external authentication technologies such as Kerberos, SSPI, SAML, or OpenID. In all cases, whether authentication takes place locally or is external, each user identity must be represented in the Tableau Server repository. The repository manages authorization meta data for user identities.

Looking for Tableau Server on Windows? See [Authentication](#).

Although all user identities are ultimately represented and stored in the Tableau Server repository, you must manage user accounts for Tableau Server in an identity store. There are two, mutually exclusive, identity store options: LDAP and local. Tableau Server supports arbitrary LDAP directories, but it's been optimized for Active Directory LDAP implementation. Alternatively, if you are not running an LDAP directory, you can use the Tableau Server local identity store. For more information see [Identity Store](#).

As shown in the following table, the type of identity store you implement, in part, will determine your authentication options.

Identity Store	Authentication Mechanism							
	Basic	SAML	Site SAML	Kerberos	Automatic Login Windows (SSPI)	OpenID Connect	Trusted Auth	Mutual SSL
Local	X	X	X			X	X	X
Active Directory	X	X		X	X		X	X
LDAP	X	X					X	X

Access and management permissions are implemented through site roles. Site roles define which users are administrators, and which users are content consumers and publishers on the server. For more information about administrators, site roles, groups, Guest User, and user-related administrative tasks, see [Users](#) and [Site Roles for Users](#).

Note: In the context of authentication, it's important to understand that users are not authorized to access external data sources through Tableau Server by virtue of having an account on the server. In other words, in the default configuration, Tableau Server does not act as a proxy to external data sources. Such access requires additional configuration of the data source on Tableau Server or authentication at the data source when the user connects from Tableau Desktop.

Add-on authentication compatibility

Some authentication methods can be used together. The following table shows authentication methods that can be combined. Cells marked with an "X" indicate a compatible authentication set. Blank cells indicate incompatible authentication sets.

	Trusted Authentication	Server-wide SAML	Site SAML	Kerberos	Automatic Login Windows (SSPI)	Mutual SSL	OpenID Connect
Trusted Authentication	N/A	X	X	X		X	X
Server-wide SAML	X	N/A	X				
Site SAML	X	X	N/A				
Kerberos	X			N/A			
Automatic Login Windows (SSPI)					N/A		
Mutual SSL	X					N/A	
OpenID Connect	X						N/A

Client authentication compatibility

Clients	Authentication Mechanism							
	Basic	SAML	Site SAML	Kerberos	Automatic Login Windows (SSPI)	OpenID Connect	Trusted Auth	Mutual SSL

Tableau Server on Linux Administrator Guide

Tableau Desktop	X	X	X	X	X	X		X
Tableau Prep	X	X	X	X	X	X		X
Tableau Mobile	X	X	X	X (iOS only)	X *	X		X
Tabcmd	X							
Web Browser-s	X	X	X	X	X	X	X	X

* SSPI is not compatible with the Workspace ONE version of the Tableau Mobile app.

Local authentication

If the server is configured to use local authentication, then Tableau Server authenticates users. When users sign-in and enter their credentials, either through Tableau Desktop, tabcmd, API, or web client, Tableau Server verifies the credentials.

To enable this scenario, you must first create an identity for each user. To create an identity, you specify a username and a password. To access or interact with content on the server, users must also be assigned a site role. User identities can be added to Tableau Server in the server UI, using [tabcmd Commands](#), or using the [REST API](#).

You can also create groups in Tableau Server to help manage and assign roles to large sets of related user groups (e.g., “Marketing”).

When you configure Tableau Server for local authentication, you can set password policies and account lockout on failed password attempts. See [Local Authentication](#).

Note: Tableau with multi-factor (MFA) authentication is available for Tableau Online only.

External authentication solutions

Tableau Server can be configured to work with a number of external authentication solutions.

NTLM and SSPI

If you configure Tableau Server to use Active Directory during installation, then NTLM will be the default user authentication method.

When a user logs onto Tableau Server from Tableau Desktop or a web client, the credentials are passed through to Active Directory, which then verifies them and sends an access token to Tableau Server. Tableau Server will then manage user access to Tableau resources based on the site roles stored in the repository.

If Tableau Server is installed on a Windows computer in Active Directory, then you may optionally enable automatic logon. In this scenario, Tableau Server will use Microsoft SSPI to automatically sign in your users based on their Windows username and password. This creates an experience similar to single sign-on (SSO).

Do not enable SSPI if you plan to configure Tableau Server for SAML, trusted authentication, a load balancer, or for a proxy server. SSPI is not supported in these scenarios. See `tsm authentication sspi <commands>`.

Kerberos

You can configure Tableau Server to use Kerberos for Active Directory. See Kerberos.

SAML

You can configure Tableau Server to use SAML (security assertion markup language) authentication. With SAML, an external identity provider (IdP) authenticates the user's credentials, and then sends a security assertion to Tableau Server that provides information about the user's identity.

For more information, see SAML.

OpenID Connect

OpenID Connect is a standard authentication protocol that lets users sign in to an identity provider (IdP) such as Google. After they've successfully signed in to their IdP, they are automatically signed in to Tableau Server. To use OpenID Connect (OIDC) on Tableau Server, the server must be configured to use the local identity store. Active Directory or LDAP identity stores are not supported with OIDC. For more information, see [OpenID Connect](#).

Mutual SSL

Using mutual SSL, you can provide users of Tableau Desktop, Tableau Mobile, and other approved Tableau clients a secure, direct-access experience to Tableau Server. With mutual SSL, when a client with a valid SSL certificate connects to Tableau Server, Tableau Server confirms the existence of the client certificate and authenticates the user, based on the user name in the client certificate. If the client does not have a valid SSL certificate, Tableau Server can refuse the connection. For more information, see [Configure Mutual SSL Authentication](#).

Trusted authentication

Trusted authentication (also referred to as "Trusted tickets") lets you set up a trusted relationship between Tableau Server and one or more web servers. When Tableau Server receives requests from a trusted web server, it assumes that the web server has already handled whatever authentication is necessary. Tableau Server receives the request with a redeemable token or ticket and presents the user with a personalized view which takes into consideration the user's role and permissions. For more information, see [Trusted Authentication](#).

LDAP

You can also configure Tableau Server to use LDAP for user authentication. Users are authenticated by submitting their credentials to Tableau Server, which will then attempt to bind to the LDAP instance using the user credentials. If the bind works then the credentials are valid and Tableau Server grants the user a session.

“Binding” is the handshake/authentication step that happens when a client tries to access an LDAP server. Tableau Server does this for itself when it makes various non-authentication related queries (such as importing users and groups).

You can configure the type of bind you want Tableau Server to use when verifying user credentials. Tableau Server supports GSSAPI and simple bind. Simple bind passes credentials directly to the LDAP instance. We recommend that you configure SSL to encrypt the bind communication. Authentication in this scenario may be provided by the native LDAP solution, or with an external process, like SAML.

For more information about planning for and configuring LDAP, see [Identity Store and External Identity Store Configuration Reference](#).

Other authentication scenarios

- REST API: [Signing In and Out \(Authentication\)](#)
- Mobile device authentication: [Single sign-on for Tableau Mobile](#)
- Certificate trust for TSM clients: [Connecting TSM clients](#)
- PAM integration for TSM administration: [TSM Authentication](#)

Data access and source authentication

You can configure Tableau Server to support a number of different authentication protocols to various different data sources. Data connection authentication may be independent of Tableau Server authentication.

For example, you may configure user authentication to Tableau Server with local authentication, while configuring Kerberos delegation, OAuth, or SAML authentication to specific data sources. See [Data Connection Authentication](#).

Local Authentication

If the server is configured to use local identity store, then Tableau Server authenticates users. When users sign-in and enter their credentials, either through Tableau Desktop,

tabcmd, API, or web client, Tableau Server verifies the credentials. Tableau user names stored in the identity store are associated with rights and permissions for Tableau Server. After authentication is verified, Tableau Server manages user access (authorization) for Tableau resources.

To use local authentication, you must configure Tableau Server with a local identity store during Setup. You cannot use local authentication if your Tableau Server has been configured with an external identity store (LDAP, Active Directory, etc).

Password storage

When local authentication is used, the user's salted and hashed password is stored in the repository. Passwords are never stored directly, rather, the result of salting and hashing the password is stored. Server uses the PBKDF2 derivation function with the HMAC SHA512 hashing function.

Configure password settings

After you install Tableau Server with local authentication, you can use Tableau Server Manager (TSM) to configure a number of password-related settings:

- Password policies: these policies define the requirement for password structure, such as length, character types, and other requirements.
- Password expiration: enable and specify password expiry.
- Login rate limit: Tableau Server throttles the time between sign-in attempts after users enter 5 incorrect passwords. Users will need to wait a few seconds before attempting another sign-in. If users continue to enter incorrect passwords, then they must wait for exponentially longer periods of time in between sign-in attempts. By default, the maximum time between sign-in attempts is 60 minutes.

Lock out account access after too many failed attempts. You can specify how many failed attempts users are allowed to enter before they are locked out. For information on how to unlock access to a locked account, see [View and manage users on a site](#) .

- User password reset: Enable users to reset passwords. Enabling password reset will

configure Tableau Server to display a link on the sign-in page. Users who forget passwords or want to reset a password can click the link to initiate a reset-password workflow. Password reset must be configured using TSM CLI, as described below.

Use the TSM web interface

1. Open TSM in a browser:

`https://<tsm-computer-name>:8850`. For more information, see [Sign in to Tableau Services Manager Web UI](#).

2. Click on **User Identity & Access** on the **Configuration** tab and then click **Authentication Method**.
3. Select Local authentication from the drop-down menu to display the password settings.
4. Configure the password settings and then click **Save Pending Changes**.
5. Click **Pending Changes** at the top of the page:



6. Click **Apply Changes and Restart**.

Use the TSM CLI

For the initial configuration of password policies, we recommend that you use the configuration file template below to create a json file. You can also set any single configuration key listed below with the syntax described in `tsm configuration set`.

1. Copy the following json template to a file. Fill in the key values with your password policy configuration. See the reference section that follows for more information about key options.

```
{
```

Tableau Server on Linux Administrator Guide

```
"configKeys": {  
  "wgserver.localauth.policies.mustcontainletters.enabled":  
  false,  
  "wgserv-  
er.localauth.policies.mustcontainuppercase.enabled": false,  
  "wgserver.localauth.policies.mustcontainnumbers.enabled":  
  false,  
  "wgserver.localauth.policies.mustcontainsymbols.enabled":  
  false,  
  "wgserv-  
er.localauth.policies.minimumpasswordlength.enabled":  
  false,  
  "wgserv-  
er.localauth.policies.minimumpasswordlength.value": 8,  
  "wgserv-  
er.localauth.policies.maximumpasswordlength.enabled":  
  false,  
  "wgserv-  
er.localauth.policies.maximumpasswordlength.value": 255,  
  "wgserver.localauth.passwordexpiration.enabled": false,  
  "wgserver.localauth.passwordexpiration.days": 90,  
  "wgserver.localauth.ratelimiting.maxbackoff.minutes": 60,  
  "wgserver.localauth.ratelimiting.maxattempts.enabled":  
  false,  
  "wgserver.localauth.ratelimiting.maxattempts.value": 5,  
  "features.PasswordReset": false  
}
```

```
}
```

2. Run the `tsm settings import -f file.json` to pass the json file with the appropriate values to Tableau Services Manager to configure Tableau Server.
3. Run the `tsm pending-changes apply` command to apply the changes. See `tsm pending-changes apply`.

Configuration file reference

This section lists all of the options that can be used to configure password policies.

```
wgserver.localauth.policies.mustcontainletters.enabled
```

Default value: `false`

Require at least one letter in passwords.

```
wgserver.localauth.policies.mustcontainuppercase.enabled
```

Default value: `false`

Require at least one upper-case letter in passwords.

```
wgserver.localauth.policies.mustcontainnumbers.enabled
```

Default value: `false`

Require at least one number letter in passwords.

```
wgserver.localauth.policies.mustcontainsymbols.enabled
```

Default value: `false`

Require at least one special character in passwords.

```
wgserver.localauth.policies.minimumpasswordlength.enabled
```

Default value: `false`

Tableau Server on Linux Administrator Guide

Enforce minimum-length passwords.

`wgserver.localauth.policies.minimumpasswordlength.value`

Default value: 8

The minimum number of characters passwords must have. Enter a value between 4 and 255, inclusive. You must set `wgserver.localauth.policies.minimumpasswordlength.enabled` to `true` to enforce this value.

`wgserver.localauth.policies.maximumpasswordlength.enabled`

Default value: `false`

Enforce maximum-length passwords.

`wgserver.localauth.policies.maximumpasswordlength.value`

Default value: 255

The maximum number of characters passwords may have. Enter a value between 8 and 225, inclusive. You must set `wgserver.localauth.policies.maximumpasswordlength.enabled` to `true` to enforce this value.

`wgserver.localauth.passwordexpiration.enabled`

Default value: `false`

Enforce password expiry.

`wgserver.localauth.passwordexpiration.days`

Default value: 90

The number of days before a password expires. Enter a value between 1 and 365, inclusive. You must set `wgserver.localauth.passwordexpiration.enabled` to `true` to enforce this value.

`wgserver.localauth.ratelimiting.maxbackoff.minutes`

Default value: 60

Maximum time between sign-in attempts after a user enters multiple incorrect passwords. Enter a value between 5 and 1440, inclusive.

`wgserver.localauth.ratelimiting.maxattempts.enabled`

Default value: `false`

Enforce account lock out after 5 incorrect passwords are entered. To change the number of incorrect passwords that will trigger account lock out, you set `wgserver.localauth.ratelimiting.maxattempts.value`.

`wgserver.localauth.ratelimiting.maxattempts.value`

Default value: 5

The number of incorrect passwords that a user may enter to trigger account lock out. Enter a value between 5 and 100, inclusive. You must set `wgserver.localauth.ratelimiting.maxattempts.enabled` to `true` to enforce this value.

`features.PasswordReset`

Default value: `false`

Enable users to reset passwords. Tableau Server must be configured to send email for this feature to operate. See [Configure SMTP Setup](#).

SAML

SAML (Security Assertion Markup Language) is an XML standard that allows secure web domains to exchange user authentication and authorization data. You can configure Tableau Server to use an external identity provider (IdP) to authenticate users over SAML 2.0. No user credentials are stored with Tableau Server, and using SAML enables you to add Tableau to your organization's single sign-on environment.

You can use SAML server wide, or you can configure sites individually. Here's an overview of those options:

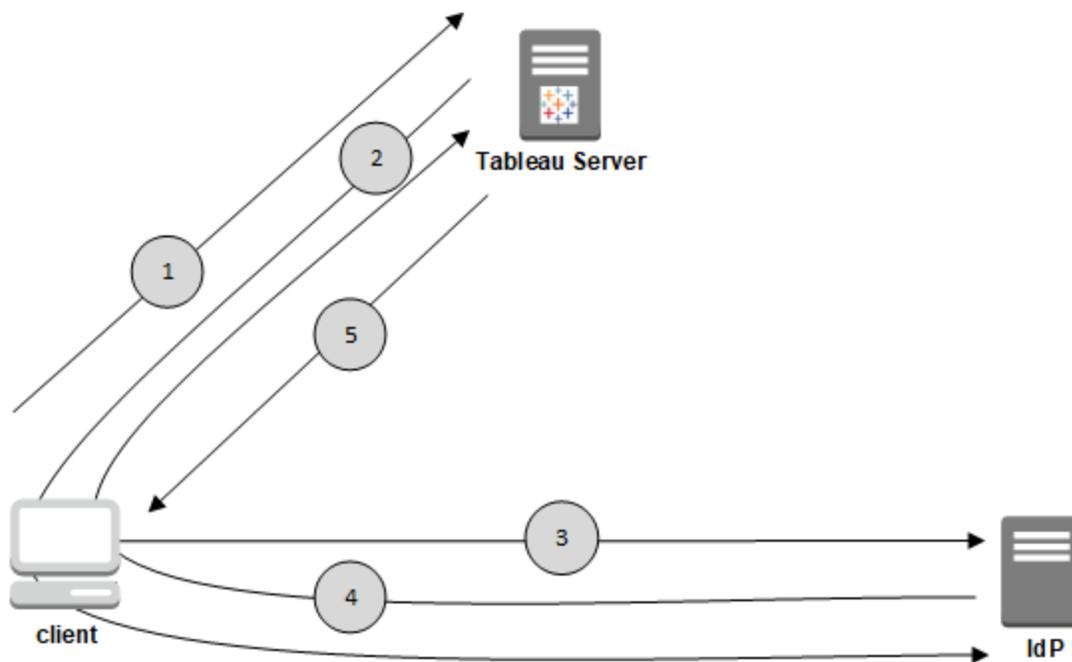
- **Server-wide SAML authentication.** A single SAML IdP application handles authentication for all Tableau Server users. Use this option if your server has only the Default site, as it is unnecessary to configure site specific SAML in this case. You may also use Server-wide SAML in multisite environments, but users are limited to a single IdP to across all sites.
- **Server-wide local authentication and site-specific SAML authentication.** In a multi-site environment, users who are not enabled for SAML authentication at the site level can sign in using local authentication.
- **Server-wide SAML authentication and site-specific SAML authentication.** In a multi-site environment, all users authenticate through a SAML IdP configured at the site level, and you specify a server-wide default SAML IdP for users that belong to multiple sites.

If you want to use site-specific SAML, you must configure server-wide SAML before you configure individual sites. Server-side SAML does not need to be enabled for site-specific SAML to function, but it must be configured.

User authentication through SAML does not apply to permissions and authorization for Tableau Server content, such as data sources and workbooks. It also does not control access to underlying data that workbooks and data sources connect to.

Note: Tableau Server supports both service provider initiated and IdP initiated SAML in browsers and in the Tableau Mobile app. SAML connections from Tableau Desktop must be service provider initiated.

The following image shows the steps to authenticate a user with single sign-on in a typical service provider initiated flow:



1. User navigates to the Tableau Server sign-in page or clicks a published workbook URL.
2. Tableau Server starts the authentication process by redirecting the client to the configured IdP.
3. The IdP requests the user's username and password from the user. After the user submits valid credentials, the IdP authenticates the user.
4. The IdP returns the successful authentication in the form of a SAML Response to the

client. The client passes the SAML Response to Tableau Server.

5. Tableau Server verifies that the username in the SAML Response matches a licensed user stored in the Tableau Server Repository. If a match is verified, then Tableau Server responds to the client with the requested content.

SAML Requirements

Before you configure SAML on Tableau Server, make sure your environment meets the requirements.

Important: SAML configurations, both with the IdP and on Tableau Server, are case sensitive. For example, URLs configured with the IdP and on Tableau Server must match exactly.

Certificate and identity provider (IdP) requirements

To configure Tableau Server for SAML, you need the following:

- **Certificate file.** A PEM-encoded x509 certificate file with a **.crt** extension. This file is used by Tableau Server, not the IdP. If you have an SSL certificate, it is possible in some circumstances to use the same certificate with SAML. For more information, see [Using SSL certificate and key files for SAML](#) later in this article.

Tableau Server requires a certificate-key pair to sign the request that is sent to the IdP. This reduces the threat of a man-in-the-middle attack given the difficulty of spoofing a signed request. Additionally, Tableau Server verifies that the AuthNResponse it receives from the trusted IdP. Tableau Server verifies the AuthNResponse by using the signature produced by the IdP. The IdP certificate metadata is provided to Tableau Server as part of the initial SAML configuration process.

Signed requests are not always necessary for all IdPs. By default, Tableau Server requires signed requests. We recommend this configuration to ensure a more secure communication transmission with the IdP. Work with your IdP team to understand if disabling signed requests is necessary. To disable signed requests see `samlSettings` Entity.

- **Signature algorithm.** The certificate must use a secure signature algorithm, for

example SHA-256. If you attempt to configure Tableau Server for SAML with a certificate that uses SHA-1 signature hash, Tableau Server will reject the certificate. You can configure Tableau Server to accept the less-secure SHA-1 hash by setting the `tsm wgserver.saml.blocklisted_digest_algorithms` configuration key.

- **RSA key and ECDSA curve sizes.** The certificate must either have an RSA key strength of 2048, or ECDSA curve size of 256.

You can configure Tableau Server to accept the less-secure sizes by setting the respective configuration keys, `wgserver.saml.min.allowed.rsa_key_size` and `wgserver.saml.min.allowed.elliptic_curve_size`.

- **Certificate key file.** An RSA or DSA private key file that has the `.key` extension. RSA keys must be in in PKCS#1 or PKCS#8 format.

Password protection requirements are as follows:

- The PKCS#1 RSA key file cannot be password protected.
 - To use a password-protected key file, you must configure SAML with a RSA PKCS#8 file. Note that a PKCS#8 file with a null password is not supported.
 - Password-protected key files are not supported in site-specific SAML deployments.
- **IdP must sign SAML assertions with a secure signature algorithm.** By default, Tableau Server will reject SAML assertions signed with the SHA-1 algorithm. You can configure Tableau Server to accept assertions signed with the less-secure SHA-1 hash by setting the `tsm wgserver.saml.blocklisted_digest_algorithms` configuration key.
 - **IdP account that supports SAML 2.0 or later.** You need an account with an external identity provider. Some examples are PingFederate, SiteMinder, and Open AM.
 - **IdP provider that supports import and export of XML metadata.** Although a

manually created metadata file might work, Tableau Technical Support cannot assist with generating the file or troubleshooting it.

SSL Off-loading

If your organization terminates SSL connections from the IdP at a proxy server before sending the authentication request to Tableau Server, then you may need to make a proxy configuration. In this scenario, SSL is "off-loaded" at the proxy server, which means the https request is terminated at the proxy server and then forwarded to Tableau Server over http.

Tableau Server validates the SAML response message returned from the IdP. Since SSL is off-loaded at the proxy, Tableau Server will validate with the protocol that it receives (http), but the IdP response is formatted with https, so validation will fail unless your proxy server includes the X-Forwarded-Proto header set to `https`. See [Configuring Proxies for Tableau Server](#).

Using SSL certificate and key files for SAML

If you are using a PEM-encoded x509 certificate file for SSL, you can use the same file for SAML. For SSL, the certificate file is used to encrypt traffic. For SAML, the certificate is used for authentication.

In addition to the requirements listed in [Certificate and identity provider \(IdP\) requirements](#) above, to use the same certificate for both SSL and SAML, the certificate must also meet the following condition to work for SAML:

- Confirm that the certificate includes only the certificate that applies to Tableau Server and not any other certificates or keys.

To do this, you can create a backup copy of the certificate file, and then open the copy in a text editor to review its contents.

User management requirements

When you enable SAML, user authentication is performed outside of Tableau, by the IdP. However, user management is performed by an identity store: either an external identity store (Active Directory or LDAP) or by Tableau Server in a local identity store. For more information about planning for user management with Tableau Server, see Identity Store.

When you configure the identity store during Setup, you must select the option that reflects how you want to use SAML. If you want to use site-specific SAML, you must configure server-wide SAML before you configure individual sites.

- **For site-specific SAML:** If you have multiple sites on Tableau Server and want to set up each site for a particular IdP or IdP application (or configure some sites not to use SAML), configure Tableau Server to manage user with a local identity store. For site-specific SAML, Tableau Server relies on the IdP for authentication and does not use passwords.
- **For server-wide SAML:** If you configure server-wide SAML with a single IdP, you can configure Tableau Server to use the local identity store or an external identity store.
- **Server-wide SAML authentication and site-specific SAML authentication.** In a multi-site environment, all users authenticate through a SAML IdP configured at the site level. In this scenario, you specify a server-wide default SAML IdP for users who belong to multiple sites. To configure this scenario, Tableau Server must be configured with a local identity store.

Note: The [REST API](#) and `tabcmd` do not support SAML single-sign (SSO). To sign in, you must specify the name and password of a user who has been created on the server. The user may be managed by the local identity store or an external identity store, depending on how you have configured Tableau Server. For Tableau Online, you can specify the TableauID credentials of the user. REST API or `tabcmd` calls will have the permissions of the user you sign in as.

SAML compatibility notes and requirements

- **Matching usernames:** The user name stored in Tableau Server must match the configured user name attribute sent by the IdP in the SAML assertion. By default, Tableau Server expects the incoming assertion to contain an attribute called "username" with that user's information. For example, if the user name for Jane Smith is stored in PingFederate as **jsmith**, it must also be stored in Tableau Server as **jsmith**.

When configuring SAML during authentication

If you are configuring SAML as part of the initial Tableau Server setup, make sure the account you plan to use exists in your IdP before you run setup. During Tableau Server setup you create the server administrator account.

When running multiple domains

If you use an Active Directory or LDAP external identity store and you are running in multiple domains (that is, users belong to multiple domains, or your Tableau Server installation includes multiple domains), the IdP must send both the user name *and* domain attributes for a user in the SAML assertion. Both these user name and domain attributes must match exactly the user name and domain stored in Tableau Server. Use the `domain\username` format for the user attribute and the fully qualified domain name (FQDN) for the domain attribute. If the domain attribute isn't included, the domain of the Tableau Server computer, also referred to as the default domain, will be used for all users. If users are not in the same domain as the Tableau Server computer, the exclusion of the domain attribute will cause sign in errors.

For more information, see Support for multiple domains and the "Match Assertions" section in the **Use TSM CLI** tab of Configure Server-Wide SAML.

- **Signature algorithm:** Many IdPs default to SHA256. A mismatch between the IdP and Tableau Server signature algorithms will cause SAML authentication failures. For example, your idP may have required SHA-256 signed assertions, but your incoming assertions or uploaded certificates are signed with SHA-1, you can force outgoing

SAML assertions to be signed with SHA-256.

To change to SHA256, set the `sHA256Enabled` configuration entity to `true` during your initial configuration. See `samlSettings` Entity. You can also change to SHA256 by running the following TSM command:

```
tsm configuration set -k wgserver.saml.sha256 -v true
```

- **Single Log Out (SLO):** Tableau Server only supports SP-initiated SLO for server SAML. IdP-initiated SLO is not supported. Site specific SAML does not support SP or IdP-initiated SLO.
- **External authentication types:** Tableau Server supports using one external authentication type at a time.
- **Mutual SSL:** Tableau Server does not support mutual SSL (two-way SSL) and SAML together. If you want to use mutual SSL, you can configure it on the IdP.
- **Assertions encoding:** Assertions must be UTF-8 encoded.
- **Encryption and SAML assertions:** When configured for server-wide SAML, Tableau Server supports encrypted assertions from the IdP. Encryption assertions are enabled by the certificate that you upload as part of the initial configuration for server-wide SAML.

When configured for site-specific SAML, Tableau Server does not support encrypted assertions from the IdP. However, all SAML requests and responses are sent over HTTPS.

- **User identity in Tableau Server for `tabcmd` users:** As described in User management requirements section above, to use `tabcmd`, you must sign in as a user defined on the server. You cannot use SAML accounts with `tabcmd`.
- **Using SAML SSO with Tableau Desktop:** By default, Tableau Desktop allows SP-initiated SAML authentication.

If your IdP does not support this functionality, you can disable SAML sign-in for Tableau Desktop using the following command:

```
tsm authentication saml configure --desktop-access disable
```

For more information, see `tsm authentication saml <commands>`.

- **Distributed installations:** TSM versions of Tableau Server (2018.2 and newer) use the Client File Service to share files in a multi node cluster. After you have configured SAML on the initial node in your cluster, the Client File Service will distribute certificate and key files to the other nodes.
- **Login URL:** For users to be able to sign in, your IdP must be configured with SAML Login endpoint that sends a POST request to the following URL:

```
https://<tableauserver>/wg/saml/SSO/index.html.
```

- **Logout URL:** To enable users to sign out after signing in with SAML (single logout, or SLO), your IdP must be configured with a SAML Logout endpoint that sends a POST request to the following URL:

```
https://<tableauserver>/wg/saml/SingleLogout/index.html.
```

SLO is only supported for server-wide SAML. SLO is not supported for site-specific SAML.

- **Post-logout redirect URL:** By default, when a user signs out of Tableau Server, the sign-in page is displayed.

To display a different page after sign-out, use the `tsm authentication saml configure` command with the `-su` or `--signout-url` option.

- To specify an absolute URL, use a fully-qualified URL starting with `http://` or `https://`, as in this example:

```
tsm authentication saml configure -su https://example.-
com
```

- To specify a URL relative to the Tableau Server host, use a page starting with a / (slash):

```
tsm authentication saml configure -su /our-
logoutpage.html
```

- **Active Directory Federation Service (AD FS):** You must configure AD FS to return additional attributes for Tableau authentication with SAML. The **Name ID** and **username** attributes can be mapped to the same AD attribute: **SAM-Account-Name**.
- **AuthNContextClassRef** : AuthNContextClassRef is an optional SAML attribute that enforces validation of certain authentication "contexts" in IdP initiated flows. You can set comma-separated values for this attribute with TSM. When this attribute is set, Tableau Server validates that the SAML response contains at least one of the values listed. If the SAML response does not contain one of the configured values, authentication will be rejected, even if the user has successfully authenticated with the IdP.

Leaving this optional attribute blank will result in default behavior: any successfully authenticated SAML response will result in a user being granted a session within Tableau Server.

This value is only evaluated for server-wide SAML. If site-SAML is configured, the `AuthNContextClassRef` attribute will be ignored.

To set this value with TSM web interface, see [Configure Server-Wide SAML](#).

To set this value with tsm configuration set, use the key, `wgserver.saml.authcontexts`, to set a comma-separated list of values.

To set this value with a JSON configuration file, see [samlSettings Entity](#).

Using SAML SSO with Tableau client applications

Tableau Server users with SAML credentials can sign in to the server from Tableau Desktop or the Tableau Mobile app. For full compatibility, we recommend that the Tableau client application version matches that of the server. To connect using site-specific SAML, users must run version 10.0 or later of the Tableau client application.

Connecting to Tableau Server from Tableau Desktop or Tableau Mobile uses a service provider (SP) initiated connection.

Redirecting authenticated users back to Tableau clients

When a user signs in to Tableau Server, Tableau Server sends a SAML request (`AuthnRequest`) to the IdP, which includes the Tableau application's **RelayState** value. If the user has signed in to Tableau Server from a Tableau client such as Tableau Desktop or Tableau Mobile, it's important that the RelayState value is returned within the IdP's SAML response back to Tableau.

When the RelayState value is not returned properly in this scenario, the user is taken to their Tableau Server home page in the web browser, rather than being redirected back to the application they signed in from.

Work with your Identity Provider and internal IT team to confirm that this value will be included as part of the IdP's SAML response, and then preserved by any network appliance (such as a proxy or load balancer) that resides between your IdP and Tableau Server.

XML data requirements

As part of SAML configuration, you exchange XML metadata between Tableau Server and the IdP. This XML metadata is used to verify a user's authentication information when the user initiates the Tableau Server sign-in process.

Tableau Server and the IdP each generates its own metadata. Each set of metadata must contain the information described in the following list. If either set is missing information, errors can occur when you configure SAML or when users try to sign in.

- **HTTP POST:** Tableau Server only supports HTTP POST requests for SAML communications. HTTP Redirect is not supported.
- With the `Binding` attribute set to `HTTP-POST`, the SAML metadata that Tableau Server and the IdP each export must contain the following elements.
 - The element that specifies the URL that the IdP redirects to after successful authentication. This is required in the Service Provider metadata, not the Identity Provider metadata.

```
<md:AssertionConsumerService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
  Location="https://<tableau-server-
>/wg/saml/SSO/index.html index="0" isDefault="true"/>
```

For Site SAML, the `Location` endpoint is `/samlservice/public/sp/metadata?alias=<site alias>`.

- The logout endpoint element appears in Tableau Server metadata and specifies the URL that the IdP will use for Tableau Server's logout endpoint. If this element is not in the IdP metadata, Tableau Server cannot negotiate a logout endpoint with the IdP and the SAML Logout feature will not be available within the Tableau Server:

```
<md:SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
  Location="https://SERVER-NAME:9031/idp/slo"
```

SLO is only supported for server-wide SAML. SLO is not supported for site-specific SAML.

- Verify that the metadata XML from the IdP includes a **SingleSignOnService** element, in which the binding is set to `HTTP-POST`, as in the following

example:

```
<md:SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="https://SERVER-NAME:9031/idp/SSO.saml2"/>
```

- This element should appear in IdP metadata and specifies the URL that Tableau Server will use for the IdP's logout endpoint.

```
<md:SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="https://SERVER-NAME:9031/idp/slo"/>
```

- **Attribute named *username*:** You must configure the IdP to return an assertion that includes the `username` attribute in the `saml:AttributeStatement` element. The assertion's attribute type must be `xs:string` (it should *not* be typed as `xs:any`).

The following example shows what this might look like.

```
<saml:Assertion assertion-element-attributes>
  <saml:Issuer>issuer-information</saml:Issuer>
  <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
    ...
  </Signature>
  <saml:Subject>
    ...
  </saml:Subject>
  <saml:Conditions condition-attributes >
    ...
  </saml:Conditions>
  <saml:AuthnStatement authn-statement-attributes >
    ...
</saml:Assertion>
```

```

</saml:AuthnStatement>

<saml:AttributeStatement>
  <saml:Attribute Name="username" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic">
    <saml:AttributeValue xmlns:x-
s="http://www.w3.org/2001/XMLSchema" xmlns:x-
si="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="xs:string">
      user-name
    </saml:AttributeValue>
  </saml:Attribute>
</saml:AttributeStatement>
</saml:Assertion>

```

By default, Tableau Server will read the `username` attribute in the `AuthNResponse` returned from the IdP. However, some IdPs may return a different attribute that is intended to identify the user. In this case, you may need to change the attribute that Tableau Server reads for the username. To authenticate successfully, the attribute's value that is returned from the IdP must match an actual username value of a Tableau Server user.

To change the SAML attribute that passes the `username` value, run the following TSM command:

```
tsm authentication saml map-assertions --user-name <USER-NAME>.
```

See `tsm authentication`.

Configure Server-Wide SAML

Configure server-wide SAML when you want all single sign-on (SSO) users on Tableau Server to authenticate through a single SAML identity provider (IdP), or as the first step to configuring site-specific SAML in a multi-site environment.

Tableau Server on Linux Administrator Guide

If you have configured server-wide SAML and are ready to configure a site, see [Configure Site-Specific SAML](#).

The SAML configuration steps we provide make the following assumptions:

- You are familiar with the options for configuring SAML authentication on Tableau Server, as described in the [SAML](#) topic.
- You have verified that your environment meets the [SAML Requirements](#), and obtained the SAML certificate files described in those requirements.

Before you begin

Gather the certificate files and place them on the Tableau Server.

In the Tableau Server folder, create a new folder named `SAML`, and place copies of the SAML certificate files in that folder. For example:

```
/var/opt/tableau/tableau_server/data/saml
```

As part of your disaster recovery plan, we recommend keeping a backup of certificate and IdP files in a safe location off of the Tableau Server. The SAML asset files that you upload to Tableau Server will be stored and distributed to other nodes by the Client File Service.

However, these files are not stored in a recoverable format. See [Tableau Server Client File Service](#).

Note: If you use the same certificate files for SSL, you could alternatively use the existing certificate location for configuring SAML, and add the IdP metadata file to that directory when you download it later in this procedure. For more information, see [Using SSL certificate and key files for SAML in the SAML requirements](#).

If you are running Tableau Server in a cluster, then the SAML certificates, keys, and metadata file will be automatically distributed across the nodes when you enable SAML.

Use the TSM web interface

This procedure requires that you upload the SAML certificates to TSM so that they are properly stored and distributed in the server configuration. The SAML files must be available to the browser on the local computer where you are running the TSM web interface in this procedure.

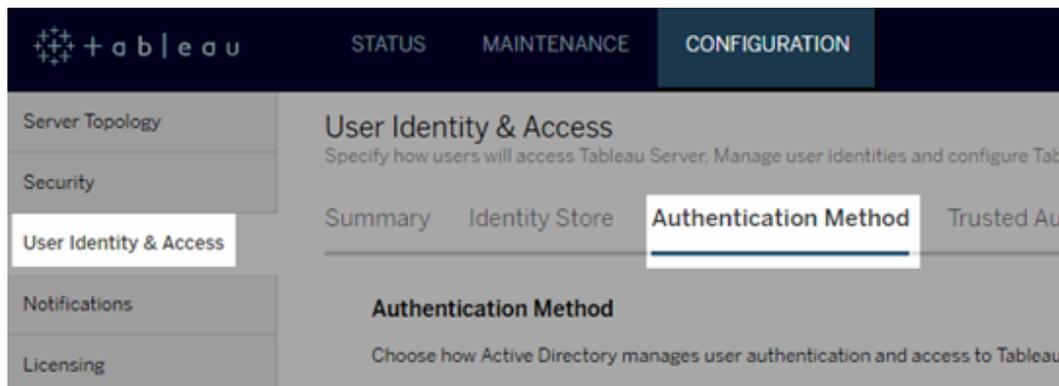
If you have gathered and saved the SAML files to the Tableau Server as recommended in the previous section, then run the TSM web interface from the Tableau Server computer where you copied the files.

If you are running the TSM web interface from a different computer, then you will need to copy all SAML files locally before proceeding. As you follow the procedure below, browse to the files on the local computer to upload them to TSM.

1. Open TSM in a browser:

`https://<tsm-computer-name>:8850`. For more information, see [Sign in to Tableau Services Manager Web UI](#).

2. On the **Configuration** tab, select **User Identity & Access**, and then select the **Authentication Method** tab.



3. For **Authentication Method**, select **SAML**.
4. In the SAML section that appears, complete Step 1 in the GUI, entering the following settings (do not yet select the check box to enable SAML for the server):

- **Tableau Server return URL**—The URL that Tableau Server users will access, such as `https://tableau-server`.

Using `https://localhost` or a URL with a trailing slash (for example, `http://tableau_server/`) is not supported.

- **SAML entity ID**—The entity ID uniquely identifies your Tableau Server installation to the IdP.

You can enter your Tableau Server URL again here. If you plan to enable site-specific SAML later, this URL also serves as the base for each site's unique ID.

- **SAML certificate and key files**— Click **Select File** to upload each of these files.

If you are using a PKCS#8 passphrase-protected key file, you must enter the passphrase with TSM CLI:

```
tsm configuration set -k wgserver.saml.key.passphrase -v <passphrase>
```

After you provide the information required in Step 1 in the GUI, the **Download XML Metadata File** button in Step 2 in the GUI becomes available.

5. Now select the **Enable SAML authentication for the server** check box above Step 1 in the GUI.
6. Complete the remaining SAML settings.
 - a. For Steps 2 and 3 in the GUI, exchange metadata between Tableau Server and the IdP. (Here's where you might need to check in with the IdP's documentation.)

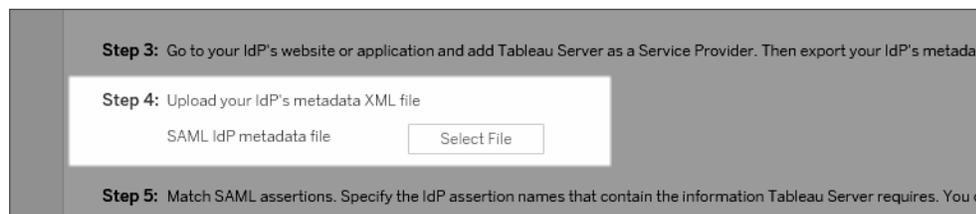
Select **Download XML Metadata File**, and specify the file location.

For other IdPs, go to your IdP account to add Tableau Server to its applications (as a service provider), providing the Tableau metadata as appropriate.

Follow the instructions in the IdP's website or documentation to download the IdP's metadata. Save the .xml file to the same location that holds your SAML certificate and key files. For example:

```
/var/opt/tableau/tableau_server/data/saml/idp-  
metadata.xml
```

- b. Return to the TSM web UI. For Step 4 in the GUI, enter the path to the IdP metadata file, and then click **Select File**.



- c. For Step 5 in the GUI: In some cases, you may need to change the assertion values in the Tableau Server configuration to match the assertion names that are passed by your IdP.

You can find assertion names in the IdP's SAML configuration. If different assertion names are passed from your IdP, then you must update Tableau Server to use the same assertion value.

Tip: “Assertions” are a key SAML component, and the concept of mapping assertions can be tricky at first. It might help to put this in a tabular-data context, in which the assertion (attribute) name is equivalent to a column heading in the table. You enter that “heading” name, rather than an example of a value that might appear in that column.

- d. For Step 6 in the GUI, select the Tableau applications in which you want to give

users a single sign-on experience.

Note: The option to disable mobile access is ignored by devices running Tableau Mobile app version 19.225.1731 and higher. To disable SAML for devices running these versions you must disable SAML as a client login option on Tableau Server.

- e. For the SAML sign-out redirect, if your IdP supports single logout (SLO), enter the page you want to redirect users to after they sign out, relative to the path you entered for the Tableau Server return URL.
 - f. For Step 7 in the GUI, optionally add a comma-separated value for the `AuthNContextClassRef` attribute. For more information about how this attribute is used, see SAML compatibility notes and requirements.
7. Click **Save Pending Changes** after you've entered your configuration information.
 8. Click **Pending Changes** at the top of the page:



9. Click **Apply Changes and Restart**.

Use the TSM CLI

Before you begin

Go to your IdP's website or application, and export the IdP's metadata XML file.

Confirm that the metadata XML from the IdP includes a **SingleSignOnService** element, in which the binding is set to `HTTP-POST`, as in the following example:

```
<md:SingleSignOnService Bind-
ing="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST" Loca-
tion="https://SERVER-NAME:9031/idp/SSO.saml2"/>
```

Step 1: Configure return URL, SAML entity ID, and specify certificate and key files

1. Open the command prompt shell and configure the SAML settings for the server (replacing placeholder values with your environment path and file names).

```
tsm authentication saml configure --idp-entity-id
https://tableau-server --idp-metadata /var/-
opt/tableau/tableau_server/data/saml/<metadata-file.xml> -
-idp-return-url https://tableau-server --cert-file /var/-
opt/tableau/tableau_server/data/saml/<file.crt> --key-file
/var/opt/tableau/tableau_server/data/saml/<file.key>
```

For more information, see `tsm authentication saml configure`.

2. If you are using a PKCS#8 key that is protected with a passphrase, enter the passphrase as follows:

```
tsm configuration set -k wgserver.saml.key.passphrase -v
<passphrase>
```

3. If SAML is not already enabled on Tableau Server; for example, you're configuring it for the first time, or you have disabled it, enable it now:

```
tsm authentication saml enable
```

4. Apply the changes:

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

Step 2: Generate Tableau Server metadata and configure the IdP

1. Run the following command to generate the required XML metadata file for Tableau server.

```
tsm authentication saml export-metadata -f <file-name.xml>
```

You can specify a file name, or omit the `-f` parameter to create a default file named `samlmetadata.xml`.

2. On your IdP's website or in its application:

- Add Tableau Server as a Service Provider.

Refer to your IdP's documentation for information about how to do this. As part of the process of configuring Tableau Server as a Service Provider, you will import the Tableau Server metadata file you generated from the `export-metadata` command.

- Confirm that your IdP uses **username** as the attribute to verify users.

Step 3: Match assertions

In some cases, you may need to change the assertion values in the Tableau Server configuration to match the assertion names that are passed by your IdP.

You can find assertion names in the IdP's SAML configuration. If different assertion names are passed from your IdP, then you must update Tableau Server to use the same assertion value.

Tip: “Assertions” are a key SAML component, and the concept of mapping assertions can be tricky at first. It might help to put this in a tabular-data context, in which the assertion (attribute) name is equivalent to a column heading in the table. You enter that “heading” name, rather than an example of a value that might appear in that column.

The following table shows the default assertion values and the configuration key that stores the value.

Assertion	Default value	Key
Username	username	wgserver.saml.idpattribute.username
Display name	displayName	Tableau does not support this attribute type.
Email	email	Tableau does not support this attribute type.
Domain	(not mapped by default)	wgserver.saml.idpattribute.domain

To change a given value, run the `tsm configuration set` command with the appropriate key:value pair.

For example, to change the `username` assertion to the value, `name`, run the following commands:

```
tsm configuration set -k wgserver.saml.idpattribute.username -v
name
```

```
tsm pending-changes apply
```

Alternatively, you can use the `tsm authentication saml map-assertions` command to change a given value.

Tableau Server on Linux Administrator Guide

For example, to set the domain assertion to a value called `domain` and specify its value as "example.myco.com," run the following commands:

```
tsm authentication saml map-assertions --domain example.myco.com  
  
tsm pending-changes apply
```

Optional: Disable client types from using SAML

By default, both Tableau Desktop and the Tableau Mobile app allow SAML authentication.

If your IdP does not support this functionality, you can disable SAML sign-in for Tableau clients using the following commands:

```
tsm authentication saml configure --desktop-access disable  
  
tsm authentication saml configure --mobile-access disable
```

Note: The `--mobile-access disable` option is ignored by devices running Tableau Mobile app version 19.225.1731 and higher. To disable SAML for devices running these versions you must disable SAML as a client login option on Tableau Server.

```
tsm pending-changes apply
```

Optional: Add AuthNContextClassRef value

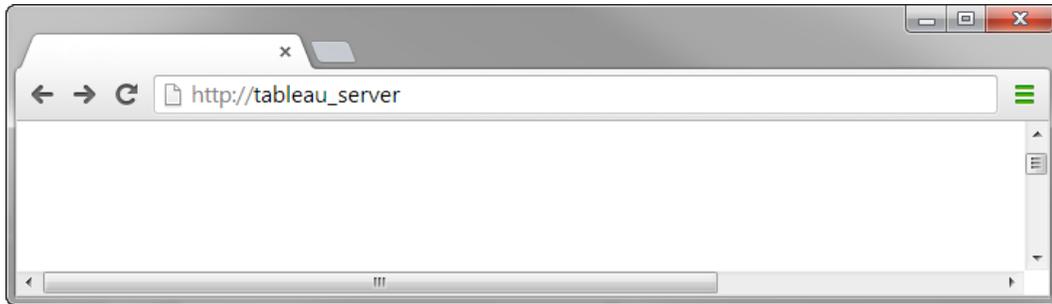
Add a comma-separated value for the `AuthNContextClassRef` attribute. For more information about how this attribute is used, see [SAML compatibility notes and requirements](#).

To set this attribute run the following commands:

```
tsm configuration set -k wgserver.saml.authcontexts -v <value>  
  
tsm pending-changes apply
```

Test the configuration

1. In your web browser, open a new page or tab, and enter the Tableau Server URL.



The browser redirects you to the IdP's sign-in form.

2. Enter your single sign-on user name and password.

A screenshot of a sign-in form. At the top, there is a green checkmark icon followed by the text "Please sign on first and we'll send you right along." Below this, there are two input fields: "Username" with the text "jsmith" entered, and "Password" with a series of black dots representing a masked password. At the bottom of the form, there are two buttons: a teal "Sign On" button and a dark blue "Cancel" button.

The IdP verifies your credentials and redirects you back to your Tableau Server start page.

Configure SAML with Salesforce IdP on Tableau Server

This topic provides road-map steps that describe how to configure SAML on Tableau Server with the Salesforce IdP.

This topic also explains how to enable Lightning Web Component (LWC). The LWC allows Salesforce administrators to embed a Tableau visualization within a Lightning page. When SSO is configured for Tableau Viz LWC on Tableau Server, the user experience is

Tableau Server on Linux Administrator Guide

seamless: after the user signs into Salesforce, embedded Tableau views will work without further authentication to Tableau Server.

Enable Salesforce as a SAML Identity Provider

If you have not yet configured Salesforce as an IdP, then follow the procedure, [Enable Salesforce as a SAML Identity Provider](#), on the Salesforce Help site.

During the process to enable Salesforce as a SAML IdP, you will either specify a certificate or Salesforce will generate a self-signed certificate for use with SAML. Download this certificate (.crt file) and the associated provider metadata file (.xml). You will need these assets in the next step.

Configure SAML on Tableau Server

Using the certificate and metadata files that you downloaded in the previous step, follow the procedure in [Configure Server-Wide SAML](#).

As part of the configuration process, you will generate a SAML Entity ID and a return URL (also referred to as the “ACS”) for Tableau Server. You will need these assets for the next step.

(Optional) After you have configured server-wide SAML, you can configure site-specific SAML on Tableau Server. See [Configure Site-Specific SAML](#).

Add Tableau Server as a Connected App in Salesforce

Follow the procedure, [Integrate Service Providers as Connected Apps with SAML 2.0](#), on the Salesforce Help site.

In this process, you will create a new connected app (Tableau Server). Some important details follow:

- Select "Enable SAML"
- Enter the Entity ID and the ACS (return URL) that you generated when configuring Tableau in the previous section.
- For the IdP certificate, be sure to select the same certificate that you specified or generated when you enabled Salesforce as a SAML provider.

- To allow users to sign in to Tableau Server from your org, manage access to your connected app by assigning the appropriate profiles or permission sets.

Enable Lightning Web Component

To enable LWC for SAML SSO on Tableau Server, you must enable in-frame authentication. Before you enable LWC, upgrade to the latest maintenance release of Tableau Server. Two additional version details are important:

- If you are not running the latest maintenance release, and your users are running Chrome browsers to access Salesforce Lightning, then review the Tableau KB article, [Embedded Views Fail to Load After Updating to Chrome 80](#).
- If you are using LWC with site-specific SAML on Tableau Server, then you must be running the Tableau Server 2020.4 or later.

After you have configured SAML on Tableau Server, run the following TSM commands to enable in-frame authentication:

```
tsm configuration set -k wgserver.saml.iframe_idp.enabled -v true
```

```
tsm pending-changes apply
```

Embed Tableau Views into Salesforce

After you have configured Tableau Server for SSO, you can then install the LWC in your Salesforce org and embed Tableau views. See [Embed Tableau Views into Salesforce](#).

Configure SAML for Tableau Viz Lightning Web Component

Tableau provides a Lightning Web Component (LWC) for embedding a Tableau visualization within a Salesforce Lightning page.

This topic describes how to enable a SSO experience for embedded Tableau visualizations in a Salesforce Lightning page. SSO for the Tableau Viz LWC scenario requires SAML configuration. The SAML IdP used for Tableau authentication must be either the Salesforce IdP or same IdP that is used for your Salesforce instance.

Tableau Server on Linux Administrator Guide

In this scenario, Salesforce administrators can drag-and-drop Tableau Viz LWC into the Lightning page to embed a visualization. Any view that is available to them on Tableau Server can be displayed in the dashboard by entering the embedded URL to the view.

When single sign-on (SSO) is configured for Tableau Viz LWC on Tableau Server, the user experience is seamless: after the user signs into Salesforce, embedded Tableau views will work without further authentication to Tableau Server.

When SSO is not configured, then users will need to reauthenticate with Tableau Server to view embedded visualizations from Tableau Server.

Requirements

- The SAML IdP used for Tableau authentication must be either the Salesforce IdP or same IdP that is used for your Salesforce instance. See [Configure SAML with Salesforce IdP on Tableau Server](#).
- SAML must be configured on Tableau Server. See [Configure Server-Wide SAML](#), or [Configure Site-Specific SAML](#).
- SAML must be configured for Salesforce.
- Install the Tableau Viz Lightning Web Component. See [Embed Tableau Views into Salesforce](#).

Configuring the authentication workflow

You may need to make additional configurations to optimize the sign-in experience for users who access Lightning with embedded Tableau views.

If a seamless authentication user experience is important, then you will need to make some additional configurations. In this context, “seamless” means that users who access the Salesforce Lightning page where Tableau Viz LWC SSO has been enabled, will not be required to perform any action to view the embedded Tableau view. In the seamless scenario, if the user is logged into Salesforce, then embedded Tableau views will be displayed with no additional user action. This scenario is enabled by *in-frame authentication*.

For a seamless user experience you will need to enable in-frame authentication on Tableau Server and at your IdP. The sections below describe how to configure in-frame authentication.

On the other hand, there are scenarios where users are interacting with the Lightning page that will require them to click a “Sign in” button to view the embedded Tableau view. This scenario, where a user must take another action to view the embedded Tableau view, is called pop-up authentication.

Pop-up authentication is the default user experience if you do not enable in-frame authentication.

Enable in-frame authentication on Tableau Server

Before you enable in-frame authentication on Tableau Server, you must have already configured and enabled SAML on Tableau Server.

Run the following TSM commands to enable in-frame authentication:

```
tsm configuration set -k wgserver.saml.iframe_idp.enabled -v true
```

```
tsm pending-changes apply
```

Note: Clickjack protection is enabled by default on Tableau Server. When you enable in-frame authentication, clickjack protection is temporarily disabled for the in-frame authentication session. You should evaluate the risk of disabling clickjack protection. See Clickjack Protection.

Tableau Server Versioning

For the best user experience, run the latest maintenance release of Tableau Server.

If you are not running the latest maintenance release, and your users are running Chrome browsers to access Salesforce Lightning, then review the Tableau KB article, [Embedded Views Fail to Load After Updating to Chrome 80](#).

Enable in-frame authentication with your SAML IdP

As described above, a seamless authentication user experience with Salesforce Mobile requires IdP support for in-frame authentication. This functionality may also be referred to as “iframe embedding” or “framing protection” at IdPs.

Salesforce safelist domains

In some cases, IdPs only allow enabling in-frame authentication by domain. In those cases, set the following Salesforce wildcard domains when you enable in-frame authentication:

```
*.force
```

```
*.visualforce
```

Salesforce IdP

Salesforce IdP supports in-frame authentication by default. You do not need to enable or configure in-frame authentication in the Salesforce configuration. However, you must run the TSM command on Tableau Server as described above.

Okta IdP

See *Embed Okta in an iframe*, in the Okta Help Center topic, [General customization options](#).

Ping IdP

See the Ping support topic, [How to Disable the "X-Frame-Options=SAMEORIGIN" Header in PingFederate](#).

OneLogin IdP

See *Framing protection*, in the OneLogin Knowledge Base article, [Account Settings for Account Owners](#).

ADFS and Azure AD IdP

Microsoft has blocked all in-frame authentication and it cannot be enabled. Instead, Microsoft only supports pop-up authentication in a second window. As a result, pop up behavior can be blocked by some browsers, which will require users to accept pop ups for the `force.com` and `visualforce.com` sites.

Salesforce Mobile App

If your users primarily interact with Lightning on the Salesforce Mobile App, then you should be aware of the following scenarios:

- The Salesforce Mobile App requires that you configure SSO/SAML to view embedded Tableau.
- The Salesforce Mobile App requires in-frame authentication. Pop-up authentication does not work. Instead, users on the Salesforce Mobile App will see the Tableau sign-in button but will not be able to sign to Tableau.
- Mobile App will not work on ADFS and Azure AD IdP.
- The Mobile App uses OAuth tokens to enable SSO. There are scenarios where the OAuth token refreshes and logs users out, requiring users to log back in. To learn more, see the Tableau KB article, [Tableau Viz Lightning Web Component On Salesforce Mobile App Prompts for Sign-in](#).
- The SSO behavior differs according to the version of Salesforce Mobile App (iOS vs Android) and the the IdP:

IdP	Mobile OS	SSO behavior
Salesforce IdP	Android	SSO works initially, but users will need to sign-in after some time.
	iOS	
External IdP	Android	SSO does not work. Users will need to manually sign-in. (SSO must still be configured to enable users access to embedded Tableau views).
	iOS	SSO works initially, but users will need to sign-in after some time.

Tableau Server on Linux Administrator Guide

Configure SAML with Azure AD IdP on Tableau Server

You can configure Azure AD as a SAML identity provider (IdP), and add Tableau Server to your supported single sign-on applications. When you integrate Azure AD with SAML and Tableau Server, your users can sign in to Tableau Server using their standard network credentials.

Prerequisites

Before you can configure Tableau Server and SAML with Azure AD, your environment must have the following:

- SSL certificate encrypted using SHA-2 (256 or 512 bit) encryption, and that meets the additional requirements listed in the following sections:
 - SSL certificate requirements
 - SAML Certificate and identity provider (IdP) requirements

Step 1: Verify SSL connection to Azure AD

Azure AD requires an SSL connection. If you haven't done so yet, complete the steps in [Configure SSL for External HTTP Traffic to and from Tableau Server](#), using a certificate that meets the requirements as specified above.

Alternatively, if Tableau Server is configured to work with a reverse proxy or load balancer where SSL is being terminated (commonly referred to as SSL off-loading), then you do not need to configure external SSL.

If your organization uses Azure AD App proxy, see the section below, [Azure AD App Proxy](#).

Step 2: Configure SAML on Tableau Server

Complete the steps in [Configure Server-Wide SAML](#) through downloading the Tableau Server metadata to an XML file. At that point, return here and continue to the next section.

Step 3: Configure Azure AD claim rules

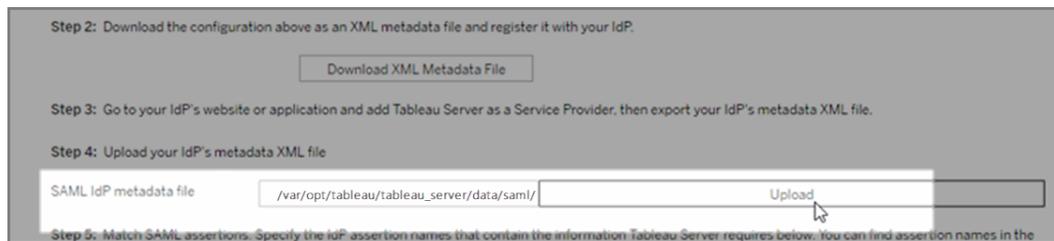
The mapping is case sensitive and requires exact spelling, so double-check your entries. The table here shows common attributes and claim mappings. Verify attributes with your specific Azure AD configuration.

LDAP Attribute	Outgoing Claim Type
SAM-Account-Name	Name ID
SAM-Account-Name	username
Given-Name	firstName
Surname	lastName

In some organizations, Azure AD as a SAML IdP is used in with Active Directory as the identity store for Tableau Server. In this case, `username` is usually the `sAMAccountName` name. See Microsoft's documentation for identifying the `sAMAccountName` attribute within Azure AD to map to the `username` attribute.

Step 4: Provide Azure AD metadata to Tableau Server

1. Return to the TSM web UI, and navigate to **Configuration > User Identity & Access > Authentication Method** tab.
2. In Step 4 of the SAML configuration window, enter the location of the XML file you exported from Azure AD, and select **Upload**.



3. Complete the remaining steps (matching assertions and specifying client type)

access) as specified in [Configure Server-Wide SAML](#). Save and apply changes.

4. Stop Tableau Server, open TSM CLI, and run the following commands.

The first command assures that Azure AD can properly redirect to Tableau Server after it accepts users' SAML credentials. The second command configures Tableau Server with the same "Refresh Token Max Inactive Time" that is the default on Azure AD and Azure AD Azure AD. Setting the maximum inactive time to match Azure AD avoids a common error state as described in [Tableau Knowledge Base article, Intermittent Error "Unable to Sign In" with SAML SSO on Tableau Server](#).

```
tsm configuration set -k wgserver.saml.sha256 -v true
```

```
tsm authentication saml configure -a 7776000
```

5. Apply the changes:

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

Azure AD App Proxy

If you are running Azure AD App Proxy in front of Tableau Server and SAML is enabled, then you will need to make an additional configuration to Azure AD App Proxy.

Tableau Server can only accept traffic from one URL when SAML is enabled. However, by default, Azure AD App Proxy sets an external URL and an internal URL.

You must set both of these values to the same URL in your custom domain. For more information, see the Microsoft documentation, [Configure custom domains with Azure AD Application Proxy](#).

Troubleshooting

Azure AD App Proxy

In some cases, links to views render internally but fail externally when traffic is crossing an Azure AD App Proxy. The issue arises when there is a pound sign (#) in the URL and users are accessing the link with a browser. The Tableau Mobile app is able to access URLs with a pound sign.

User session timeouts appear to be ignored

When Tableau Server is configured for SAML, users might experience sign in errors because the IdP maximum authentication age setting is set to a value greater than Tableau's maximum authentication age setting. To resolve this issue, you can use the tsm configuration set option `wgserver.saml.forceauthn` to require the IdP to re-authenticate the user each time Tableau redirects the authentication request, even if the IdP session for the user is still active.

For example, when the Azure AD setting `maxInactiveTime` is greater than Tableau Server's setting `maxAuthenticationAge`, Tableau redirects the authentication request to the IdP who subsequently sends Tableau an assertion that the user is already authenticated. However, because the user was authenticated outside of Tableau Server's `maxAuthenticationAge`, Tableau rejects the user authentication. In cases like this, you can do one or both of the following:

- Enable the `wgserver.saml.forceauthn` option to require the IdP to re-authenticate the user every time Tableau redirects the authentication request. For more information, see `wgserver.saml.forceauthn`.
- Increase Tableau Server's `maxAuthenticationAge` setting. For more information, see “a, --max-auth-age <max-auth-age>” in the tsm authentication topic.

Configure SAML with AD FS on Tableau Server

You can configure Active Directory Federation Services (AD FS) as a SAML identity provider, and add Tableau Server to your supported single sign-on applications. When you

Tableau Server on Linux Administrator Guide

integrate AD FS with SAML and Tableau Server, your users can sign in to Tableau Server using their standard network credentials.

Prerequisites

Before you can configure Tableau Server and SAML with AD FS, your environment must have the following:

- A server running Microsoft Windows Server 2008 R2 (or later) with AD FS 2.0 (or later) and IIS installed.
- We recommend that you secure your AD FS server (for example, using a reverse proxy). When your AD FS server is accessible from outside your firewall, Tableau Server can redirect users to the sign in page hosted by AD FS.
- SSL certificate encrypted using SHA-2 (256 or 512 bit) encryption, and that meets the additional requirements listed in the following sections:
 - SSL certificate requirements
 - SAML Certificate and identity provider (IdP) requirements

Step 1: Verify SSL connection to AD FS

AD FS requires an SSL connection. If you haven't done so yet, complete the steps in [Configure SSL for External HTTP Traffic to and from Tableau Server](#), using a certificate that meets the requirements as specified above.

Alternatively, if Tableau Server is configured to work with a reverse proxy or load balancer where SSL is being terminated (commonly referred to as SSL off-loading), then you do not need to configure external SSL.

Step 2: Configure SAML on Tableau Server

Complete the steps in [Configure Server-Wide SAML](#) through downloading the Tableau Server metadata to an XML file. At that point, return here and continue to the next section.

Step 3: Configure AD FS to accept sign-in requests from Tableau Server

Note: These steps reflect a third-party application and are subject to change without our knowledge.

Configuring AD FS to accept Tableau Server sign-in requests is a multi-step process, starting with importing the Tableau Server XML metadata file to AD FS.

1. Do one of the following to open the **Add Relying Party Trust Wizard**:

Windows Server 2008 R2:

- a. Select **Start menu > to Administrative Tools > AD FS 2.0**.
- b. In **AD FS 2.0**, under **Trust Relationships**, right-click the **Relying Party Trusts** folder, and then click **Add Relying Party Trust**.

Windows Server 2012 R2:

- a. Open **Server Manager**, and then on the **Tools** menu, click **AD FS Management**.
 - b. In **AD FS Management**, on the **Action** menu, click **Add Relying Party Trust**.
2. In the **Add Relying Party Trust Wizard**, click **Start**.
 3. On the **Select Data Source** page, select **Import data about the relying party from a file**, and then click **Browse** to locate your Tableau Server XML metadata file. By default, this file is named **samlspmetadata.xml**.
 4. Click **Next**, and on the **Specify Display Name** page, type a name and description for the relying party trust in the **Display name** and **Notes** boxes.
 5. Click **Next** to skip the **Configure Multi-factor Authentication Now** page.
 6. Click **Next** to skip the **Choose Issuance Authorization Rules** page.

7. Click Next to skip the **Ready to Add Trust** page.
8. On the **Finish** page, select the **Open the Edit Claim Rules dialog for this relying party trust when the wizard closes** check box, and then click **Close**.

Next, you'll work in the **Edit Claim Rules** dialog, to add a rule that makes sure the assertions sent by AD FS match the assertions Tableau Server expects. At a minimum, Tableau Server needs an email address. However, including first and last names in addition to email will ensure the user names displayed in Tableau Server are the same as those in your AD account.

1. In the **Edit Claim Rules** dialog box, click **Add Rule**.
2. On the **Choose Rule Type** page, for **Claim rule template**, select **Send LDAP Attributes as Claims**, and then click **Next**.
3. On the **Configure Claim Rule** page, for **Claim rule name**, enter a name for the rule that makes sense to you.
4. For **Attribute store**, select **Active Directory**, complete the mapping as shown below, and then click **Finish**.

The mapping is case sensitive and requires exact spelling, so double-check your entries. The table here shows common attributes and claim mappings. Verify attributes with your specific Active Directory configuration.

LDAP Attribute	Outgoing Claim Type
SAM-Account-Name	Name ID
SAM-Account-Name	username
Given-Name	firstName
Surname	lastName

If you are running AD FS 2016 or later, then you must add a rule to pass through all claim values. If you are running an older version of AD FS, skip to the next procedure to export AD FS metadata.

1. Click **Add Rule**.
2. Under **Claim rule template**, choose **Pass Through or Filter an Incoming Claim**.
3. Under **Claim rule name**, enter Windows.
4. On the **Edit Rule - Windows** pop-up:
 - Under **Incoming claim type**, select **Windows account name**.
 - Select **Pass through all claim values**.
 - Click **OK**.

Now you will export AD FS metadata that you'll import to Tableau Server later. You will also make sure the metadata is configured and encoded properly for Tableau Server, and verify other AD FS requirements for your SAML configuration.

1. Export AD FS Federation metadata to an XML file, and then download the file from **<https://<adfs server name>/FederationMetadata/2007-06/FederationMetadata.xml>**.
2. Open the metadata file in a text editor like Sublime Text or Notepad++, and verify that it is correctly encoded as UTF-8 without BOM.

If the file shows some other encoding type, save it from the text editor with the correct encoding.

3. Verify that AD FS uses forms-based authentication. Sign-ins are performed in a browser window, so you need AD FS to default to this type of authentication.

Edit **c:\inetpub\adfs\ls\web.config**, search for the tag , and move the line so it appears first in the list. Save the file so that IIS can automatically reload it.

Note: If you don't see the **c:\inetpub\adfs\ls\web.config** file, IIS is not installed and configured on your AD FS server.

4. (Optional) This step is required only if AD FS is configured as an IDP for site-specific SAML. This step is not required if AD FS is configured as the IDP for server-wide SAML.

Configure an additional AD FS relying party identifier. This allows your system to work around any AD FS issues with SAML logout.

Do one of the following:

Windows Server 2008 R2:

- a. In **AD FS 2.0**, right-click on the relying party you created for Tableau Server earlier, and click **Properties**.
- b. On the **Identifiers** tab, in the **Relying party identifier** box, enter **https://<tableauservername>/public/sp/metadata** and then click **Add**.

Windows Server 2012 R2:

- a. In **AD FS Management**, in the **Relying Party Trusts** list, right-click on the relying party you created for Tableau Server earlier, and click **Properties**.
- b. On the **Identifiers** tab, in the **Relying party identifier** box, enter **https://<tableauservername>/public/sp/metadata** and then click **Add**.

Note: AD FS can be used with Tableau Server for a single relying party to the same instance. AD FS cannot be used for multiple relying parties to the same instance, for example, multiple site-SAML sites or server-wide and site SAML configurations.

Step 4: Provide AD FS metadata to Tableau Server

1. Return to the TSM web UI, and navigate to **Configuration > User Identity & Access > Authentication Method** tab.
2. In Step 4 of the SAML configuration window, enter the location of the XML file you

exported from AD FS, and select **Upload**.



3. Complete the remaining steps (matching assertions and specifying client type access) as specified in Configure Server-Wide SAML. Save and apply changes.
4. Stop Tableau Server, open TSM CLI, and run the following commands.

The first command assures that AD FS can properly redirect to Tableau Server after it accepts users' SAML credentials. The second command configures Tableau Server with the same "Refresh Token Max Inactive Time" that is the default on AD FS and Azure AD FS. Setting the maximum inactive time to match AD FS avoids a common error state as described in Tableau Knowledge Base article, [Intermittent Error "Unable to Sign In" with SAML SSO on Tableau Server](#).

```
tsm configuration set -k wgserver.saml.sha256 -v true
tsm authentication saml configure -a 7776000
```

5. Apply the changes:

```
tsm pending-changes apply
```

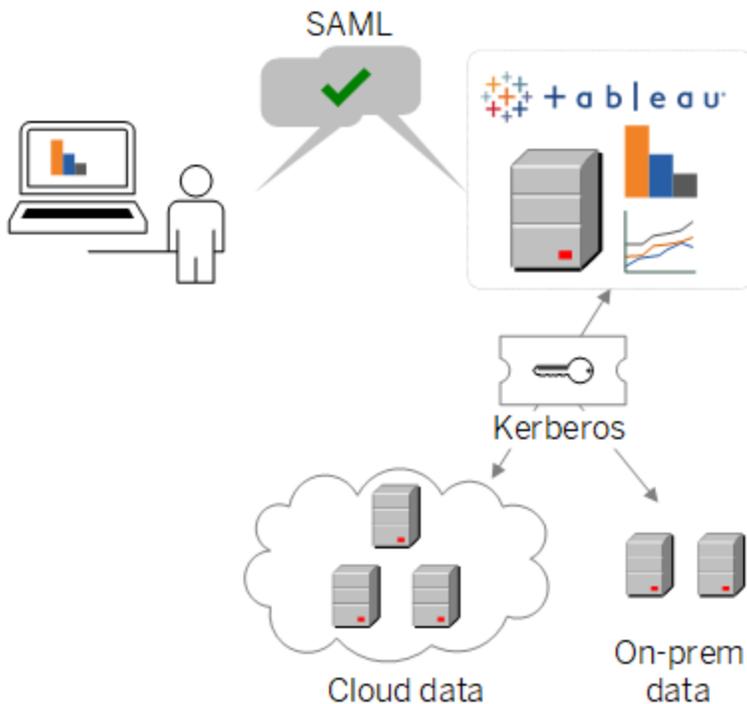
If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

Tableau Server on Linux Administrator Guide

Use SAML SSO with Kerberos Database Delegation

In a Windows Active Directory (AD) environment, you can enable SAML single sign-on (SSO) to Tableau Server, along with Kerberos database delegation. This provides authorized users direct access to Tableau Server, as well as to the underlying data defined in their published workbooks and data sources.

Overview of the process



In a typical scenario:

1. One of your Tableau analysts publishes a dashboard to Tableau Server. That dashboard contains a connection to a Hadoop cluster, for example, that is configured to accept Kerberos credentials.

Then the workbook publisher sends a link to colleagues for review.

2. When a colleague clicks the link, Tableau Server authenticates the user through the

SAML SSO process. Then it looks at the user's authorization scheme, and if allowed, uses the Tableau Server keytab to access the underlying database on behalf of the user. This populates the dashboard with the Hadoop data that the user is authorized to see.

Configure Tableau Server for SAML with Kerberos

Using SAML with Kerberos works inherently when you complete the processes to enable each separately:

1. Configure Tableau Server for SAML, as described in [Configure Server-Wide SAML](#).
2. Configure Tableau Server and your underlying databases to accept Kerberos credentials, as described in [Enable Kerberos Delegation](#) and related articles.

Configure Site-Specific SAML

Use site-specific SAML in a multi-site environment when you want to enable single sign-on, and you also use multiple SAML identity providers (IdPs) or IdP applications. When you enable site SAML, you can specify the IdP or IdP application for each site, or configure some sites to use SAML and the others to use the default server-wide authentication method.

If you want all server users to use SAML and sign in through the same IdP application, see [Configure Server-Wide SAML](#).

Prerequisites for enabling site-specific SAML

Before you can enable SAML single sign-on at the site level, complete the following requirements:

- The Tableau Server identity store must be configured for local identity store.

You cannot configure site-specific SAML if Tableau Server is configured with an external identity store such as Active Directory or OpenLDAP.
- Make sure your environment and your IdP meet the general SAML Requirements.

Some features are supported only in server-wide SAML deployments, including but limited to:

- Single logout (SLO) is only supported for server-wide SAML. SLO is not supported for site-specific SAML.
- Password-protected key files are not supported in site-specific SAML deployments.
- You must configure server-wide SAML before you configure site-specific SAML. You do not need to enable server-wide SAML, but site-specific SAML requires the server-wide configuration. See [Configure Server-Wide SAML](#).
- Note the location of the SAML certificate files. You will provide this when you Configure the server to support site-specific SAML.

For more information, see [Put metadata and certificate files in place](#) in the topic on configuring server-wide SAML.

- Add Tableau Server as a service provider to your IdP. You can find this information in the documentation the IdP provides.
- Confirm that the system clocks of the computer hosting the site-SAML IdP and the computer hosting Tableau Server are within 59 seconds of each other. Tableau Server does not have a configuration option to adjust the response skew (time difference) between the Tableau Server computer and the IdP.

Server-wide settings related to site-specific SAML

In the settings for configuring site-specific SAML, Tableau provides a site-specific return URL and entity ID based on these settings. The site-specific return URL and entity ID cannot be modified. These configurations are set by TSM as described in [Configure Server-Wide SAML](#).

Server-wide settings, maximum authentication age and response skew, do not apply to site-specific SAML. These configurations are hard-coded:

- The maximum authentication age refers to how long an authentication token from the IdP is valid after it is issued. The hard-coded maximum authentication age site-specific SAML is 24 days.
- The response skew is the maximum number of seconds difference between Tableau Server time and the time of the assertion creation (based on the IdP server time) that still allows the message to be processed. The hard-coded site-specific value for this is 59 seconds.

Configure the server to support site-specific SAML

After you complete the prerequisites listed above, you can run the following commands to configure the server to support site-specific SAML.

1. **Configure Server-Wide SAML.** At a minimum, you must run the following TSM command (if you have already configured server-wide SAML, skip to Step 2):

```
tsm authentication saml configure --idp-entity-id
<tableau-server-entity-id> --idp-return-url <tableau-
server-return-url> --cert-file <path-to-saml-cer-
tificate.crt> --key-file <path-to-saml-keyfile.key>
```

2. **Enable site SAML.** Run the following commands:

```
tsm authentication sitesaml enable

tsm pending-changes apply
```

About the commands

The `sitesaml enable` command exposes the **Authentication** tab on each site's **Settings** page in the Tableau Server web UI. After you configure the server to support site SAML, you can continue to Configure SAML for a site to work through the settings on the **Authentication** tab.

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt

using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

If you want to review the commands and settings that will be carried out when you run `pending-changes apply`, you can run the following command first:

```
tsm pending-changes list --config-only
```

Configure SAML for a site

This section takes you through the configuration steps that appear on the **Authentication** page in the Tableau Server web UI. In a self-hosted Tableau Server installation, this page appears only when support for site-specific SAML is enabled at the server level. It is enabled by default in Tableau Online.

Note: To complete this process, you will also need the documentation your IdP provides. Look for topics that refer to configuring or defining a service provider for a SAML connection, or adding an application.

Step 1: Export metadata from Tableau

To create the SAML connection between Tableau Server and your IdP, you need to exchange required metadata between the two services. To get metadata from Tableau Server, do either of the following steps. See the IdP's SAML configuration documentation to confirm the correct option.

- Select **Export metadata** to download an XML file that contains the Tableau Server SAML entity ID, Assertion Consumer Service (ACS) URL, and X.509 certificate.

The entity ID is site-specific, and based on the server-wide entity ID that you specified

when you enabled site SAML on the server. For example, if you specified `https://tableau_server`, you might see the following entity ID for the site:

```
https://tableau_server-
/samlservice/public/sp/metadata?alias=48957410-9396-430a-
967c-75bdb6e002a0
```

You cannot modify the site-specific entity ID or ACS URL that Tableau generates.

- Select **Download signing and encryption certificate** if your IdP expects the required information in a different way. For example, if it wants you to enter the Tableau Server entity ID, ACS URL, and X.509 certificate in separate locations.

The following image has been edited to show that these settings are the same in Tableau Online and Tableau Server.

1 Export metadata from Tableau Online | Server

Select an option for obtaining metadata required by the Identity Provider (IdP):

- Export an XML file that contains the metadata.

or

- Copy the Tableau Online entity ID and ACS URL individually, and download the X.509 certificate and save it as a CER file.

Tableau Online entity ID

Assertion Consumer Service URL (ACS)

Steps 2 and 3: External steps

For Step 2, to import the metadata you exported in step 1, sign in to your IdP account, and use the instructions provided by the IdP's documentation to submit the Tableau Server metadata.

For Step 3, the IdP's documentation will guide you also in how to provide metadata to a service provider. It will instruct you to download a metadata file, or it will display XML code. If it

displays XML code, copy and paste the code into a new text file, and save the file with a .xml extension.

Step 4: Import IdP metadata to the Tableau site

On the **Authentication** page in Tableau Server, import the metadata file that you downloaded from the IdP or configured manually from XML it provided.

Step 5: Match attributes

Attributes contain authentication, authorization, and other information about a user. In the **Identity Provider (IdP) Assertion Name** column, provide the attributes that contain the information Tableau Server requires.

- **Username or Email:** (Required) Enter the name of the attribute that stores user names or email addresses.
- **Display name:** (Optional but recommended) Some IdPs use separate attributes for first and last names, and others store the full name in one attribute.

Select the button that corresponds to the way your IdP stores the names. For example, if the IdP combines first and last name in one attribute, select **Display**

name, and then enter the attribute name.

5 Match attributes

Match the attribute names (assertions) in the IdP's SAML configuration to the corresponding attribute names on Tableau Server. Click Test Connection to fetch available attributes.

Tableau Server Attribute	Identity Provider (IdP) Assertion Name
<p>Username or Email</p> <p>Enter the username or email address attribute that the IdP sends during the authentication process. This must match the attribute name in Tableau</p>	<input type="text" value="NameID"/>
<p>Display Name</p> <p>Enter an assertion name for either the first name and last name, or for the full name, depending on how the IdP stores this information. Tableau Server uses these attributes to set the display name.</p> <p> <input checked="" type="radio"/> First name <input type="text" value="firstName"/> <input type="radio"/> Last name <input type="text" value="lastName"/> <input type="radio"/> Full name <input type="text" value="FullName"/> </p>	

Step 6: Manage users

Select existing Tableau Server users, or add new users you want to approve for single sign-on.

When you add or import users, you also specify their authentication type. On the Users page, you can change users' authentication type any time after adding them.

For more information, see [Add Users to a Site or Import Users and Set the User Authentication Type for SAML](#).

Important: Users that authenticate with site-specific SAML can belong only to one site. If a user needs to access multiple sites, set their authentication type to the server default. Depending on how site-specific SAML was configured by the server administrator, the server default is either local authentication or server-wide SAML.

Update SAML Certificate

After you have configured SAML authentication, you may need to periodically update the certificate. In some cases, you may need to change the certificate for operational changes in your IT environment. In either case, you must use TSM or the Site Authentication page to update the SAML certificate that has already been configured.

Below are the steps to update the certificate and key files for server-wide and site-specific SAML implementations.

Update certificate for server-wide SAML

To change or update the certificate (and the corresponding key file if required) for server-wide SAML, follow the steps below:

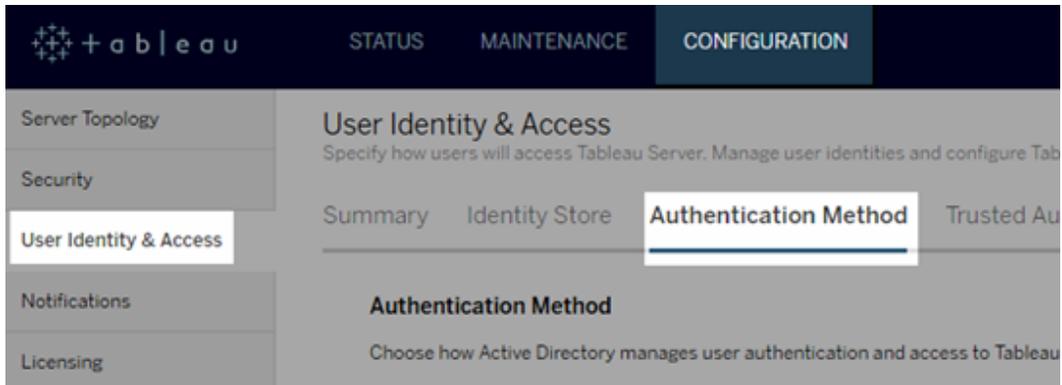
1. Open TSM in a browser:

`https://<tsm-computer-name>:8850`. For more information, see [Sign in to Tableau Services Manager Web UI](#).

2. Stop Tableau Server.

You can do this either from the TSM web UI, by clicking **Tableau Server is running**, and selecting **Stop Tableau Server**, or from the command line, using the `tsm stop` command.

3. On the **Configuration** tab, select **User Identity & Access**, and then select the **Authentication Method** tab.



4. For **Authentication Method**, select **SAML**.
5. Complete Step 1 - Step 4 in the GUI to update the SAML certificate file and exchange metadata between Tableau Server and your IdP.

Step 1: Provide the location for the following SAML attributes and files.

Tableau Server return URL	<input type="text" value="Required"/>
SAML entity ID	<input type="text" value="Required"/>
SAML certificate file	<input type="button" value="Select File"/>
SAML key file	<input type="button" value="Select File"/>

Step 2: Download XML metadata file, and register it with your IdP.

Step 3: Go to your IdP's website or application and add Tableau Server as a Service Provider. Then export your IdP's metadata XML file.

Step 4: Upload your IdP's metadata XML file

SAML IdP metadata file	<input type="button" value="Select File"/>
------------------------	--

6. Click **Save Pending Changes** after you've entered your configuration information.
7. Click **Pending Changes** at the top of the page:



8. Click **Apply Changes and Restart**.

After you change the certificate, you must run `tsm pending-changes apply` to restart Tableau Server services. We also recommend restarting any other services on the

Tableau Server on Linux Administrator Guide

computer that use the SAML certificate. If you are changing a root certificate on the operating system, you must reboot the computer.

Update certificate for site-specific SAML

The certificate used for Tableau site metadata is provided by Tableau and not configurable. To update the certificate for site-specific SAML, you must upload a new certificate to your IdP and re-exchange the metadata with Tableau Server.

1. Sign in to the site as a server or site administrator, and select **Settings > Authentication**.
2. Under Authentication types, select **Edit connection** to expand the Authentication GUI.
3. Open a new tab or window, and sign in to your IdP account.
4. Use the instructions provided by the IdP's documentation to upload a new SAML certificate.
5. Download the new XML metadata file to provide to Tableau Server.
6. Return to the **Authentication** page in Tableau Server, and in Step 4 of the GUI, import the metadata file that you downloaded from the IdP.
7. Click **Save** after you've updated your configuration information.

Troubleshoot SAML

This topic provides information about resolving issues that can occur when you configure SAML authentication.

SAML and Enable Automatic Logon

If you are using SAML and if Tableau Server is also configured to use Active Directory, do not also select **Enable automatic logon**. **Enable automatic logon** and SAML cannot both be used on the same server installation.

HTTP Status 500 error when configuring SAML

Under some circumstances you might get an HTTP status 500 error and see the following error after enabling SAML and navigating to the Tableau Server URL in a browser:

```
org.opensaml.saml2.metadata.provider.MetadataProviderException:  
User specified binding is not supported by the Identity Pro-  
vider using profile urn:oas-  
is:names:tc:SAML:2.0:profiles:SSO:browser
```

To help resolve this error, make sure of the following:

- The IdP URL for the SSO profile specified in the SAML tab is correct.
- The IdP URL for the SSO profile provided while creating the service provider in the IdP is correct.
- The IdP is configured to use `HTTP-POST` requests. (Redirect and SOAP are not supported.)

If any of these settings were not correct, make appropriate updates and then perform the SAML configuration steps again, starting with generating and exporting the XML metadata document from Tableau Server.

If these settings are correct, but you still see the error, examine the metadata XML that is produced by Tableau Server and by the IdP, as described in [SAML Requirements](#).

Signing In from the Command Line

SAML is not used for authentication when you sign in to Tableau Server using `tabcmd` or the [Tableau Data Extract command line utility](#) (provided with Tableau Desktop), even if Tableau Server is configured to use SAML. These tools require the authentication configured when Tableau Server was originally installed (either local authentication or AD).

Login fails: failed to find the user

Login fails with the following message:

Tableau Server on Linux Administrator Guide

```
>Login failure: Identity Provider authentication successful for user <username from IdP>. Failed to find the user in Tableau Server.
```

This error typically means that there is a mismatch between the usernames stored in Tableau Server and provided by the IdP. To fix this, make sure that they match. For example, if Jane Smith's username is stored in the IdP as `jsmith` it must be stored in Tableau Server as `jsmith`.

Login fails: SSL offloading

Logon fails with the following message:

```
Unable to Sign In - Invalid username or password.
```

Additionally, the vizportal logs (set to debug mode) contain the following message:

```
DEBUG com.tableau.core.util.RemoteIP - Found header null in X-FORWARDED-PROTO
```

Note: To log SAML-related events, `vizportal.log.level` must be set to `debug`. For more information, see [Change Logging Levels](#).

This combination of messages indicates a misconfiguration of an external proxy server that is offloading SSL for the connection to Tableau Server. To resolve this issue, see the KB article, ["Unable to Sign In" and "Invalid username or password" Error With SAML After Upgrading](#).

SAML Error Log

SAML authentication takes place outside Tableau Server, so troubleshooting authentication issues can be difficult. However, login attempts are logged by Tableau Server. You can create a snapshot of log files and use them to troubleshoot problems. For more information, see [Log File Snapshots \(Archive Logs\)](#).

Note: To log SAML-related events, `vizportal.log.level` must be set to `debug`. For more information, see [Change Logging Levels](#).

Check for SAML errors in the following files in the unzipped log file snapshot:

```
\vizportal\vizportal-<n>.log
```

In Tableau Server 9.0 and later, the application process (`vizportal.exe`) handles authentication, so SAML responses are logged by that process.

Trailing Slash

On the SAML tab, confirm that the **Tableau Server return URL** does not end with a trailing slash

Correct: **`http://tableau_server`**

Incorrect: `http://tableau_server/`

Confirm Connectivity

Confirm that the Tableau Server you are configuring has either a routeable IP address or a NAT at the firewall that allows two-way traffic directly to the server.

You can test your connectivity by running `telnet` on Tableau Server and attempting to connect with the SAML IdP. For example: `C:\telnet 12.360.325.10 80`

The above test should connect you to the HTTP port (80) on the IdP and you should receive an HTTP header.

Kerberos

Kerberos is a three-way authentication protocol that relies on the use of a trusted third-party network service called the Key Distribution Center (KDC) to verify the identity of computers and provide for secure connections between the computers through the exchange of *tickets*. These tickets provide mutual authentication between computers or services, verifying that one has permission to access the other.

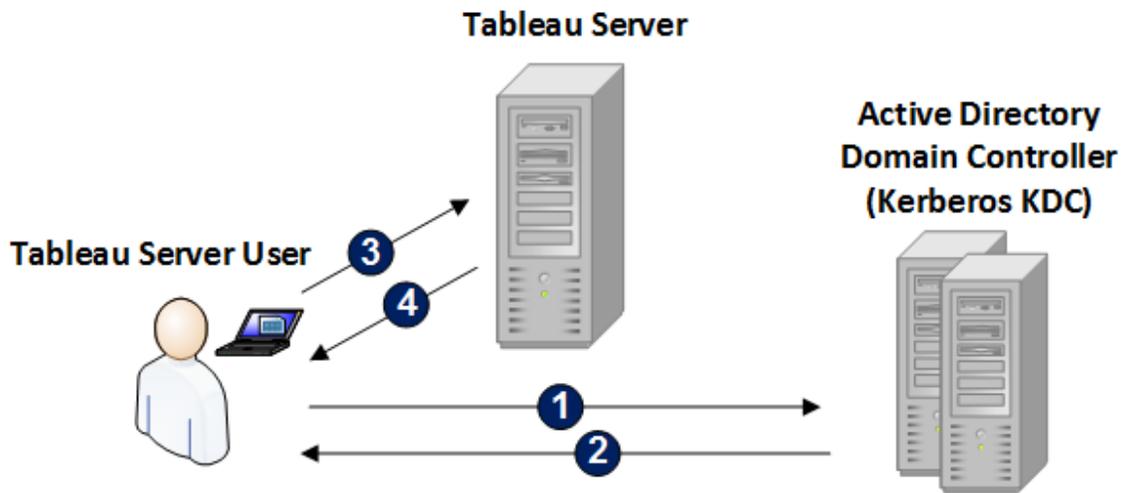
Tableau Server supports Kerberos authentication in an Active Directory Kerberos environment, with authentication to Tableau Server being handled by Kerberos.

Note: The Kerberos support in Tableau Server is for user authentication. It does not handle internal permissions and authorization related to Tableau Server content, such as workbooks.

How Kerberos works

When you configure Tableau Server for Kerberos in an Active Directory (AD) environment, the AD domain controller also serves as the Kerberos Key Distribution Center (KDC) and issues Ticket Granting Tickets to the other nodes in the domain. Users authenticated by the KDC do not have to authenticate further when connecting to Tableau Server.

The following is a diagram of the authentication workflow.



1 User logs into their Active Directory domain.

2 The Kerberos KDC authenticates the user and sends a Ticket Granting Ticket (TGT) to the

user's computer.

- 3 The user connects to Tableau Server in Tableau Desktop or in a web browser.
- 4 Tableau Server authenticates the user.

Kerberos Requirements

You can configure Kerberos authentication for Tableau Server running in Active Directory environments.

General requirements

- **External Load Balancer/Proxy Server:** If you are going to use Tableau Server with Kerberos in an environment that has external load balancers (ELBs) or proxy server, you need to set these up before you configure Kerberos in the Tableau Server Configuration utility. See [Configuring Proxies for Tableau Server](#).
- **iOS Browser Support:** An iOS user can use Kerberos authentication with mobile Safari if a Configuration Profile specifying the user's Kerberos identity is installed. See [Configuring an iOS Device for Kerberos Support](#) in the Tableau Mobile Help. For more information about browser support for Kerberos SSO, see [Tableau Client Support for Kerberos SSO](#).
- Tableau Server supports constrained delegation for authentication to datasources. In this scenario, the Tableau data access account is specifically granted rights to the target database SPNs. Unconstrained delegation is not supported.
- The supported data sources (SQL Server, MSAS, PostgreSQL, Hive/Impala, and Teradata) must be configured for Kerberos authentication.
- A keytab file that is configured with the service provider name for the Tableau Server for user authentication. For more information, see [Understanding Keytab Requirements](#).

Active Directory requirements

You must meet the following requirements to run Tableau Server with Kerberos in an Active Directory environment:

- Tableau Server must use Active Directory (AD) for authentication.
- The domain must be an AD 2003 or later domain for Kerberos connections to Tableau Server.
- Smart Card Support: Smart cards are supported when users sign into their workstations with a smartcard and this results in a Kerberos TGT being granted to the user from Active Directory.
- Single-Sign On (SSO): Users must be granted a Kerberos Ticket Granting Ticket (TGT) from Active Directory when they sign into their computers. This is standard behavior for domain-joined Windows computers and standard for Mac computers that use AD as their network account server. For more information on using Mac computers and Active Directory, see [Join your Mac to a network account server](#) in the Apple Knowledge Base.

Kerberos delegation

For Kerberos delegation scenarios the following are required:

- If the domain is AD 2003 or later, single domain Kerberos delegation is supported. The users, Tableau Server, and backend database must be on the same domain.
- If the domain is AD 2008, there is limited cross domain support. Users from other domains can be delegated if the following conditions are met. Tableau Server and the backend database must be on the same domain, and a two way trust is required between the domain where Tableau Server resides and the user's domain.
- If the domain is 2012 or later, full cross-domain delegation is supported. AD 2012 R2

is preferred because it has a dialog for configuring constrained delegation, while 2012 non-R2 requires manual configuration.

Understanding Keytab Requirements

Kerberos authentication relies on credentials that are stored in specially formatted files called keytab files. You may need to generate keytab files for your Tableau Server deployment. This topic describes the keytab files that Tableau Server uses to access various services in a typical organization. You may need to generate keytabs for Tableau Server to integrate into the following services:

- User authentication (SSO) in Windows Active Directory
- Datasource delegation
- Operating system
- Directory service

If your organization includes IT professionals who handle identity, authentication, and/or security, then you should work with them to create a plan for generating appropriate keytabs for your Tableau Server deployment.

User authentication (SSO) in Windows Active Directory

If you will be using Active Directory as the identity store for Tableau Server, and you want users to authenticate with Kerberos SSO, then you will need to generate a keytab file for Tableau Server.

Tableau is running on...	Need to manually generate a keytab?
Windows in Active Directory domain	Yes
Linux in Active Directory domain	Yes
Windows or Linux in non-Active Directory environment	Kerberos SSO is not a supported scenario.

Follow these recommendations (for Windows and Linux versions of Tableau Server):

- Create a service account in your directory for Tableau Server.
- Create a keytab specifically for the Tableau Server service account. Do not reuse the keytab file that the computer account/OS uses to authenticate. You may use the same keytab for Kerberos SSO as you use for the directory authentication in the scenario above.
- You must create service principal names (SPN) in Active Directory for the Tableau Server service.
- Use the batch file in the next section to create the SPNs and the keytab file.
- After you have created the SPNs, upload the keytab file as described in Configure Kerberos.

Batch file: Set SPN and create keytab in Active Directory

You can use a batch file to set the service principal names (SPN) and create a keytab file. These operations are a part of the process to enable Kerberos SSO for Tableau Server (on Windows or Linux) running in Active Directory.

In previous versions of Tableau Server (before 2018.2), the configuration script was generated from the Tableau Server Configuration utility.

To generate a configuration script, copy and paste the following batch file contents into a text file. The batch file creates service principal names (SPN) for Tableau Server and will create a keytab file for the user you specify in the file.

Follow the directions in the file contents. After you have finished customizing the file, save it as a .bat file.

This file must be run in an Active Directory domain by a Domain admin, who will be prompted for the service account password of the account you specify in the file.

The batch file uses the Windows `set`, `setspn`, and `ktpass` commands.

Note: The batch file below is self-documented. However, if you do not have experience with Kerberos and generating keytab files, we recommend that you read the Microsoft blog post, [All you need to know about Keytab files](#), before proceeding. Environmental details in your organization may require additional configuration of the `ktpass` command. For example, you must determine what to set for the `/crypto` parameter. We recommend specifying a single `/crypto` value that is required by your KDC. See the Microsoft article, [ktpass](#) for the full list of supported values for the `/crypto` parameter.

Creating a keytab file for user authentication in Active Directory must be performed on a Windows computer as specified here. Creating this keytab file on a Linux computer is not supported.

SPN and keytab batch file contents

```
@echo off
setlocal EnableDelayedExpansion

REM *****

REM This script generates the Service Principal Names (SPNs)
and keytab files required for
REM Kerberos SSO with Apache.
REM This script executes set, setspn, and ktpass commands
included in any Windows Server
REM Operating System from 2003 on.
REM Before running this script you must enter configuration
information for the setspn and
REM ktpass commands.
REM Elements that require your configuration information are
enclosed in as such:
REM ! -- and --!.
REM After you customize this file, save it as a .bat file, and
```

Tableau Server on Linux Administrator Guide

```
run on a domain-joined
REM computer.
REM This script must be run by a Domain admin.

REM *****

REM The following set command will prompt the domain admin for
credentials of the
REM Tableau Server service account.
REM This account must be a valid domain user account.
REM If the password contains a literal \" (backslash - double
quote), all backslashes
REM immediately before the double quote must be
REM duplicated when typed for the password to work, e.g. if pass-
word contains
REM \" replace with \\\", if passwords contains \\\\" replace with
\\\\\"

set /p adpass= "Enter password for the Tableau Server service
account."
set adpass=!adpass:="\"!

REM *****

REM The following setspn commands create the SPN in the domain.
REM More information on setspn can be found here:
REM http://technet.microsoft.com/en-us/library/cc731241
\(WS.10\).aspx
REM Enter the canonical FQDN and the host names for Tableau
Server followed by the
REM Tableau Server service account name.
REM Use this syntax: HTTP/hostname domain\service_account_name.
REM The example below shows syntax for a computer named
"tableau01" in the "example.lan"
```

```

REM domain, with service account, "tab-serv-account":
REM setspn -s HTTP/tableau01 example\tab-serv-account
REM setspn -s HTTP/tableau01.example.lan example\tab-serv-
account
REM DNS and AD are not case sensitive, but the keytab files
are.  Verify that host names
REM match letter case as stored in DNS.
REM Use Windows Server's DNS Manager utility to verify host
name case.

REM *****

echo Creating SPNs...
setspn -s HTTP/!--replace with canonical host name and service
account --!
setspn -s HTTP/!--replace with canonical FQDN and service
account --!

REM *****

REM The following commands create the keytab file in the same
directory where the
REM bat file is run. More information on ktpass can be found
here:
REM https://docs.microsoft.com/en-us/windows-server-
/administration/windows-commands/ktpass
REM Note: keytab files are case-sensitive.
REM The realm following the FQDN should be all uppercase.
REM Syntax is:
REM ktpass /princ HTTP/!--FQDN--!@!--Kerberos_Realm--! /pass
!adpass!
REM /ptype KRB5_NT_PRINCIPAL /crypto !--cipher--! /out key-
abs\kerberos.keytab
REM Best practice: specify the /crypto value that is required

```

Tableau Server on Linux Administrator Guide

by your KDC.

```
REM Options for /crypto = {DES-CBC-CRC|DES-CBC-MD5|RC4-HMAC-NT|AES256-SHA1|AES128-SHA1|All}
```

```
REM Specifying /crypto All will result in passwords stored with RC4 cipher, which is
```

```
REM no longer considered secure.
```

```
REM When using AES256-SHA1 OR AES128-SHA1, the /mapuser option must be included
```

```
REM in the ktpass command to ensure the keytab file is mapped properly to the user. For example:
```

```
REM ktpass /princ HTTP/!--FQDN--!@!--Kerberos_Realm--! /pass !ad-pass! /ptype KRB5_NT_PRINCIPAL /mapuser <domain\username> /crypto AES256-SHA1 /out keytabs\kerberos.keytab
```

```
REM The following example shows the ktpass syntax with the example.lan configuration from above:
```

```
REM ktpass /princ HTTP/!--FQDN--!@!--Kerberos_Realm--! /pass !ad-pass! /ptype KRB5_NT_PRINCIPAL /crypto DES-CBC-CRC /out keytabs\kerberos.keytab
```

```
REM *****
```

```
echo Creating Keytab files in %CD%\keytabs
```

```
mkdir keytabs
```

```
ktpass /princ HTTP/!--FQDN--!@!--Kerberos_Realm--! /pass !ad-pass! /ptype KRB5_NT_PRINCIPAL /crypto DES-CBC-CRC /out keytabs\kerberos.keytab
```

Operating system

If your organization uses Kerberos for authentication, then the computer where Tableau Server is running must be authenticated with the Kerberos realm in which it's running.

Tableau is running on...	Need to manually generate a keytab?
---------------------------------	--

Windows in Active Directory domain	No
Linux in Active Directory domain	Yes
Windows or Linux in non-Active Directory environment	Yes

If you are running Tableau Server on Windows, and the computer is joined to the Active Directory, then you do not need to manage or generate a keytab file for the operating system.

If you are running Tableau Server on Linux in a Kerberos realm (MIT KDC or Active Directory), then you will need to generate a keytab file specifically for the computer operating system. The keytab you create for the computer should be specifically for OS authentication. Do not use the same keytab file for OS authentication that you will be using for the other services described later in this topic.

Directory service

If your organization uses a directory service, such as LDAP or Active Directory, to manage user identity, then Tableau Server requires read-only access to the directory.

Alternatively, you can configure Tableau Server to manage all accounts by installing with a local identity store. In this case, you do not need a keytab.

The following table summarizes keytab requirements:

Tableau is running on...	Directory service	Need to manually generate a keytab?
Windows in AD domain	Active Directory	No
Windows	LDAP (GSSAPI bind)	Yes
Linux	Active Directory or LDAP (GSSAPI bind)	Yes
Windows or Linux	Active Directory or LDAP	No

	(Simple bind)	
Windows or Linux	Local identity store	No keytab required.

If you need to manually generate a keytab for this scenario, then you will use it for GSSAPI bind to the directory. Follow these recommendations:

- Create a service account in your directory for Tableau Server.
- Create a keytab specifically for the Tableau Server service account. Do not reuse the keytab file that the computer account/OS uses to authenticate.
- Upload the keytab file as part of the json configuration of the Tableau Server identity store. See identityStore Entity.

As part of your disaster recovery plan, we recommend keeping a backup of the keytab and conf files in a safe location off of the Tableau Server. The keytab and conf files that you add to Tableau Server will be stored and distributed to other nodes by the Client File Service. However, the files are not stored in a recoverable format. See Tableau Server Client File Service.

Datasource delegation

You can also use Kerberos delegation to access data sources in an Active Directory. In this scenario, users can be authenticated to Tableau Server with any supported authentication mechanism (SAML, local authentication, Kerberos, etc), but can access datasources that are enabled by Kerberos.

Follow these recommendations:

- The computer account for Tableau Server (Windows or Linux) must be in Active Directory domain.
- The keytab file that you use for Kerberos delegation can be the same keytab that you use for Kerberos user authentication (SSO).
- The keytab must be mapped to the service principal for Kerberos delegation in Active

Directory.

- You may use the same keytab for multiple data sources.

For more information, see the following configuration topics:

- Tableau Server on Linux: [Enable Kerberos Delegation](#)
- Tableau Server on Windows: [Enabling Kerberos Delegation](#)

Configure Kerberos

You can configure Tableau Server to use Kerberos. This allows you to provide a single sign-on (SSO) experience across all the applications in your organization. Before you configure Tableau Server for Kerberos make sure your environment meets the Kerberos Requirements.

Note: Kerberos constrained delegation for SSO to Tableau Server is not supported. (Constrained delegation for data sources is supported.) For more information, see Single-Sign On (SSO) in Kerberos Requirements.

To configure Kerberos, you must first enable Kerberos, and then specify a keytab file for user authentication. The keytab file you specify must be configured with the service provider name for the Tableau Server for user authentication. If you are using Kerberos authentication for data sources, those credentials should be included in the single keytab file that you will specify during Kerberos configuration on Tableau Server.

As part of your disaster recovery plan, we recommend keeping a backup of the keytab file in a safe location off of the Tableau Server. The keytab file that you add to Tableau Server will be stored and distributed to other nodes by the Client File Service. However, the file is not stored in a recoverable format. See Tableau Server Client File Service.

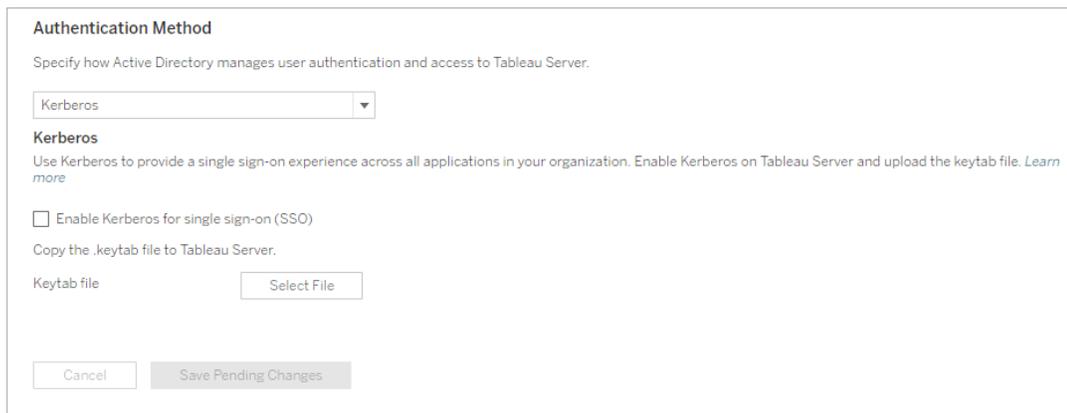
Use the TSM web interface

Tableau Server on Linux Administrator Guide

1. Open TSM in a browser:

`https://<tsm-computer-name>:8850`. For more information, see [Sign in to Tableau Services Manager Web UI](#).

2. Click **User Identity & Access** on the **Configuration** tab and then click **Authentication Method**.
3. Under **Authentication Method**, select **Kerberos** in the drop-down menu.
4. Under Kerberos, select **Enable Kerberos for single sign-on (SSO)**.
5. To copy the keytab file to the server, click **Select File**, and then browse to the file on your computer.



The screenshot shows the 'Authentication Method' configuration dialog box. It has a title bar 'Authentication Method' and a subtitle 'Specify how Active Directory manages user authentication and access to Tableau Server.' Below the subtitle is a dropdown menu with 'Kerberos' selected. Under the 'Kerberos' section, there is a checkbox labeled 'Enable Kerberos for single sign-on (SSO)' which is currently unchecked. Below the checkbox is the text 'Copy the .keytab file to Tableau Server.' and a 'Keytab file' label next to a 'Select File' button. At the bottom of the dialog are 'Cancel' and 'Save Pending Changes' buttons.

6. Click **Save Pending Changes** after you've entered your configuration information.
7. Click **Pending Changes** at the top of the page:



8. Click **Apply Changes and Restart**.

Use the TSM CLI

1. Copy the keytab file to the computer running Tableau Server and run the following command to set permissions on the file:

```
chmod 644 "/path/keytab_file"
```

If you are running Tableau Server on in a distributed cluster deployment, then you will need to manually distribute the keytab file to each node and then set the permissions. Copy the keytab file to the same directory on each node in the cluster. After you have copied the keytab file to each node and set permissions on the file, then run the following TSM commands on one node. The configuration will propagate to each node.

2. Type the following command to specify the location and name of the keytab file:

```
tsm authentication kerberos configure --keytab-file <path-to-keytab_file>
```

3. Type the following command to enable Kerberos:

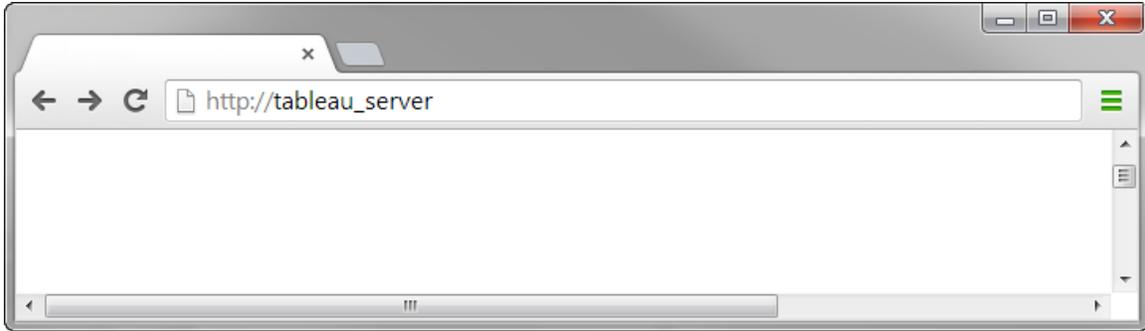
```
tsm authentication kerberos enable
```

4. Run `tsm pending-changes apply` to apply changes.

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

Confirm your SSO configuration

Once Tableau Server has restarted, test your Kerberos configuration from a web browser on a different computer by typing the Tableau Server name in the URL window:



You should be automatically authenticated to Tableau Server.

Tableau Client Support for Kerberos SSO

This article describes some requirements for and nuances with using Kerberos single sign-on (SSO) with Tableau Server, depending on the particular Tableau client and operating system. Tableau clients covered in this article include common web browsers, Tableau Desktop, and the Tableau Mobile app.

General browser client support

To use browser-based Kerberos Single Sign-on (SSO), the following must be true:

- Kerberos must be enabled on Tableau Server.
- The user must have a user name and password to sign in to Tableau Server.

Note: If Kerberos SSO fails, users can fall back on their user name and passwords credentials.

- The user must be authenticated to Active Directory through Kerberos on the client computer or mobile device. Specifically, this means that they have a Kerberos Ticket Granting Ticket (TGT).

Tableau Desktop and browser clients

On Windows or the Mac, you can use Kerberos SSO to sign in to Tableau Server from the following versions of Tableau Desktop or browser. Where noted, additional configuration is required.

Windows

- Tableau Desktop 10.3 or later supported.
- Internet Explorer - supported, may require configuration - see [Note 1](#)
- Chrome - supported, may require configuration -see [Note 1](#)
- Firefox - requires configuration - see [Note 2](#)
- Safari - not supported

Mac OS X

- Tableau Desktop 10.3 or newer
- Safari - supported
- Chrome - see [Note 3](#)
- Firefox - see [Note 2](#)
- Internet Explorer - not supported

Tableau Mobile app clients

On a iOS or Andoid device, you can use the following Tableau Mobile or mobile browser versions to use Kerberos authentication to Tableau Server:

iOS

- Tableau Mobile app- see [Note 4](#)
- Safari - see [Note 4](#)
- Chrome - not supported

Android - see [Note 5](#)

- Tableau Mobile app
- Chrome

Operating system and browser-specific notes

The following notes describe configuration requirements or issues with specific operating system and client combinations.

Note 1: Internet Explorer or Chrome on Windows desktop

Kerberos SSO is supported in both Internet Explorer and Chrome, but it requires configuration in **Windows Internet Options**:

1. Enable **Integrated Windows Authentication**.
2. Verify that Tableau Server URL is in the local intranet zone.

Internet Explorer can sometimes detect intranet zones and configure this setting. If it has not detected and configured the Tableau Server URL, you must manually add the URL to the local intranet zone.

To enable Integrated Windows Authentication:

1. In Windows Control Panel, open **Internet Options**.
2. On the **Advanced** tab scroll down to the **Security** section.
3. Select **Enable Integrated Windows Authentication**.
4. Click **Apply**.

To verify or add the Tableau Server URL to the local intranet zone:

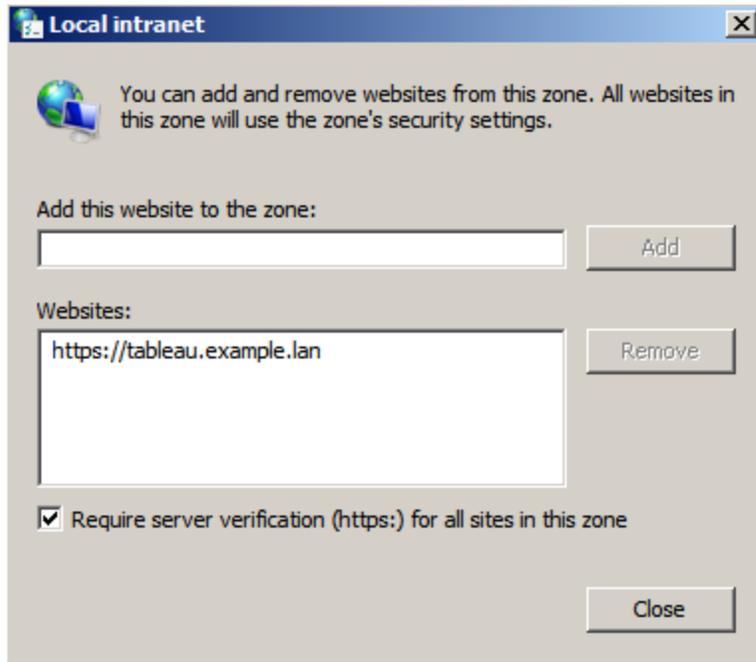
1. In Windows Control Panel, open **Internet Options**.
2. On the **Security** tab, select **Local intranet**, and then click **Sites**.
3. On the **Local intranet** dialog box, click **Advanced**.

In the **Websites** field, look for the internal Tableau Server URL.

In some organizations, IT administrators will use a wildcard (*) to specify internal URLs. For example, the following URL includes all servers in the internal `example.lan` namespace in the local intranet zone:

```
https://*.example.lan
```

The following image shows a specific URL of `https://tableau.example.lan`.



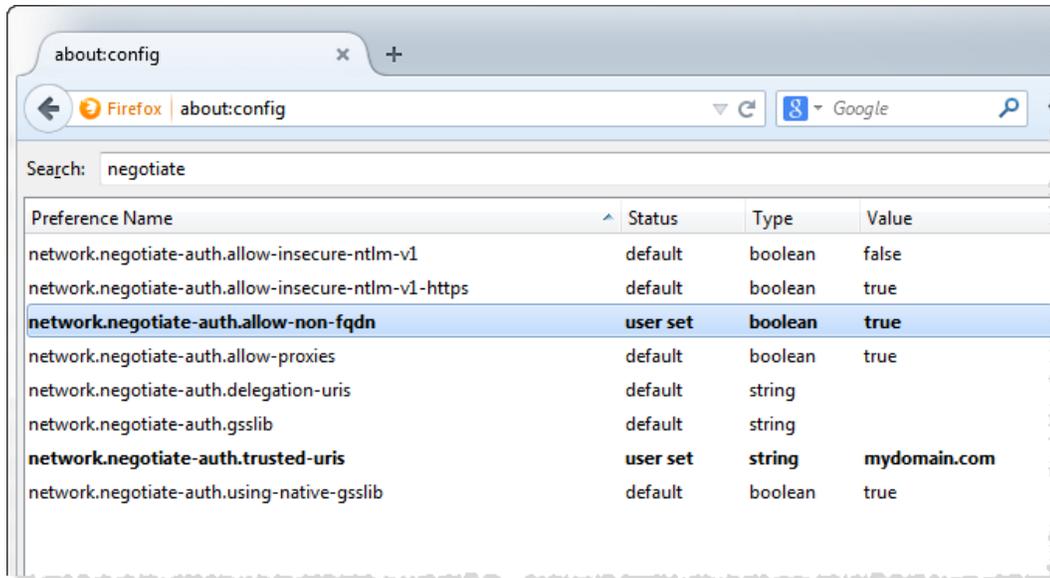
4. If the Tableau Server URL or a wildcard URL is not specified in the **Websites** field, enter the Tableau Server URL in the **Add the website to the zone** field, click **Add**, and then click **OK**.

If the Tableau Server URL is already listed in **Websites**, you can simply close the dialog.

Note 2: Firefox on Windows or Mac OS X desktop

You can use Firefox with Kerberos SSO on either Windows or Mac to sign in to Tableau Server. To do this, you must complete the following steps to configure Firefox to support Kerberos:

1. In Firefox, enter `about:config` in the address bar.
2. Click **I'll be careful, I promise** when warned about changing advanced settings.
3. Enter `negotiate` in the **Search** box.



4. Double-click **network.negotiate-auth.allow-non-fqdn**, and then set the value to **true**.
5. Double-click **network.negotiate-auth.trusted-uris** and enter the Tableau Server fully qualified domain name (FQDN). For example, `tableau.example.com`.

Note 3: Chrome on Mac OS X desktop

According to Chrome documentation, Kerberos SSO works on a Mac when you launch Chrome from a terminal window with the following command:

```
open -a "Google Chrome.app" --args --auth-server-whitel-
ist="tableauserver.example.com"
```

where `tableauserver.example.com` is the URL for Tableau Server in your environment.

However, we have found inconsistent results in our testing. Therefore, if you want to use Kerberos SSO on a Mac, we recommend that you use Safari or Firefox. For more information, see the *Integrated Authentication* section at [HTTP authentication](#) on The Chromium Projects site.

Note: Users can still use Chrome on Mac OS X to sign in to Tableau Server, but they might be prompted to enter their user name and password (single sign-on may not work).

Note 4: Mobile Safari or Tableau Mobile on iOS

Kerberos SSO is supported if iOS is configured for Kerberos. The iOS device must have a Kerberos authentication configuration profile installed. This is usually done by an enterprise IT group. Tableau Support cannot assist with configuring iOS devices for Kerberos. See the [authentication topic](#) in the *Tableau Mobile Deployment Guide*.

Note 5: Android platform

Kerberos SSO is not supported on the Tableau Mobile app on the Android operating system. You can still use your Android device and the Tableau Mobile app or a supported mobile browser to connect to Tableau Server. In this scenario, rather than authenticating with Kerberos, users will be prompted to enter their credentials when accessing Tableau Server.

More information

- *Tableau Mobile Deployment Guide: [Control authentication and access for Tableau Mobile](#)*
- See *Web Browsers* under [Tableau Server Tech Specs](#)

Troubleshoot Kerberos

The troubleshooting suggestions in this topic are divided into issues related to single sign-on (SSO) on the server and issues with the delegated data sources.

Also see the Tableau Community wiki page, [Testing Database Kerberos Configuration On Linux](#).

Tableau Server on Linux Administrator Guide

Single sign-on to Tableau Server

In a Kerberos SSO environment, a user signing in to Tableau Server from a web browser or Tableau Desktop might see a message indicating that Tableau Server can't sign them in automatically (using single sign-on). It suggests that they provide a Tableau Server user name and password instead.



Troubleshooting sign-in errors on the client computer

- **Enter the user name and password**—To check the user's general access to Tableau Server, sign in by entering the user's name and password.

If these credentials fail, the user might not be a user on Tableau Server. For Kerberos SSO to work, the user must be able to access Tableau Server, and they must be granted a Ticket Granting Ticket (TGT) by Active Directory (as described in the **TGT** item later in this list).

- **Check other users' SSO credentials**—Try to connect with SSO to Tableau Server using other user accounts. If all users are affected, the problem might be in the

Kerberos configuration.

- **Use a computer other than the server computer**—Kerberos SSO does not work when you sign in to Tableau Server on localhost. Clients must connect from a computer other than the Tableau Server computer.
- **Use a server name, not IP address**—Kerberos SSO does not work if you enter an IP address as the Tableau Server name. In addition, the server name you use to access Tableau Server must match the name used in the Kerberos configuration (see [Key table entry](#), below).
- **Confirm that the client has TGT**—The client computer must have a TGT (Ticket Granting Ticket) from the Active Directory domain. Constrained delegation, with the proxy granting a ticket, is not supported.

To confirm the client computer has a TGT, do the following:

- On Windows, open a command prompt and type the following: `klist tgt`
- On the Mac, open a terminal window and type the following: `klist`

The output should show a TGT for the user/domain trying to authenticate to Tableau Server.

The client computer might not have a TGT in the following circumstances:

- The client computer is using a VPN connection.
 - The client computer is not joined to the domain (for example, it is a non-work computer being used at work).
 - The user signed in to the computer with a local (non-domain) account.
 - The computer is a Mac that is not using Active Directory as a network account server.
- **Confirm browser version and settings**—For web browser sign-in, make sure the

browser is supported for Kerberos and, if necessary, is configured correctly.

- Internet Explorer (IE) and Chrome work “out of the box” on Windows.
- Safari works “out of the box” on the Mac.
- Firefox requires additional configuration.

For more information, see Tableau Client Support for Kerberos SSO.

Troubleshooting sign-in errors on the server

If you cannot solve the problem from the client computer, your next steps are to troubleshoot on the computer running Tableau Server. The administrator can use the request ID to locate the sign-in attempt in the Apache logs on Tableau Server.

- **Log files**—Check the Apache error.log for an error with the exact time/date of the failed sign-in attempt.
- **Key table entry**—If the error.log entry includes the message, “No key table entry matching HTTP/<servername>.<domain>.<org>@”, for example:

```
[Fri Oct 24 10:58:46.087683 2014] [[:error] [pid 2104:tid 4776] [client 10.10.1.62:56789] gss_acquire_cred() failed: Unspecified GSS failure. Minor code may provide more information (, No key table entry found matching HTTP/server-name.domain.com@)
```

This error is a result of a mismatch of any of the following:

- **Tableau Server URL** - The URL used by the client computer to access the server.

This is the name that you type into Tableau Desktop or a browser address bar. It could be a **shortname** (`http://servername`) or a **fully-qualified domain name** (`http://servername.domain.com`)



- **DNS reverse lookup** for the server IP address.

This looks up a DNS name using an IP address.

At a command prompt type:

```
ping servername
```

with the IP address returned by pinging the server, do a reverse DNS lookup type:

```
nslookup <ip address>
```

The nslookup command will return network information for the IP address. In the *Non-authoritative answer* portion of the response, verify that the fully qualified domain name (FQDN) matches the following configured values:

- The Kerberos .keytab file
- Service Principal Name (SPN) for the server

For more information about configuring these values, see [Understanding Keytab Requirements](#).

Verify Kerberos configuration script

You may need to modify the `ktpass` command that you used to generate the keytab file for environmental variables. Review the troubleshooting steps in the Knowledge Base article, [Unable to Generate Kerberos Script Configuration for Tableau Server](#).

Data source SSO

Delegated data source access failures

Check the `vizqlserver` log files for "workgroup-auth-mode."

Look for "workgroup-auth-mode" in the log files. It should say "kerberos-impersonate" not "as-is".

Kerberos delegation multi-domain configuration

Tableau Server can delegate users from other Active Directory domains. If your database uses MIT Kerberos, you may need to adjust your Kerberos principal to database user mapping. Specifically, you will need to update `krb5.conf` with rules for each Kerberos realm that users will connect from. Use the `auth_to_local` tag in the `[realms]` section to map principal names to local user names.

For example, consider a user, `EXAMPLE\jsmith`, whose Kerberos Principal is `jsmith@EXAMPLE.LAN`. In this case, Tableau Server will specify a delegated user, `jsmith@EXAMPLE`. Tableau Server will use the Active Directory legacy domain alias as the Kerberos Realm.

The target database may already have a rule such as the following to map the user, `jsmith@EXAMPLE.LAN` to the database user, `jsmith`.

```
EXAMPLE.LAN = {
    RULE: [1:$1@$0] (. *@EXAMPLE.LAN) s/@. *//
    DEFAULT
}
```

To support delegation, you must add another rule to map `jsmith@EXAMPLE` to a database user:

```
EXAMPLE.LAN = {
    RULE: [1:$1@$0] (. *@EXAMPLE.LAN) s/@.*//
    RULE: [1:$1@$0] (. *@EXAMPLE) s/@.*//
    DEFAULT
}
```

See the MIT Kerberos Documentation topic, [krb5.conf](#), for more information.

Cross-domain constrained delegation

In some cross-domain scenarios where the KDC is running on a Windows Server prior to Windows 2012, delegation may fail. Errors you may see include:

- SQL Server Network Interfaces: The system cannot contact a domain controller to service the authentication request. Please try again later.
- SQL Server Native Client: Cannot generate SSPI context
- The Domain Controller returns: `KRB-ERR-POLICY` error with a status `STATUS_CROSSREALM_DELEGATION_FAILURE (0xc000040b)`.

Cross-domain refers to a scenario where Tableau Server is running in a different domain than the data source with different service accounts. For example:

- Tableau Server runs on DomainA with DomainA service account.
- SQL Server runs on DomainB with DomainB service account.

Traditional constrained delegation only works if both servers are in the same domain. The user can come from other domains.

If you are seeing the errors noted above, then to enable this scenario, your Active Directory administrator should remove any traditional constrained delegation which is configured on the delegating account. Removing delegation can be achieved with Active Directory management tools or by removing the values associated with the Active Directory property, `msDS-AllowedToDelegateTo`.

If you want to preserve an existing single domain delegation alongside cross-domain delegation, you must configure both using resource-based constrained delegation.

For more information about Kerberos and constrained delegation, see the Microsoft topic, [Kerberos Constrained Delegation Overview](#).

Web authoring

There are two web authoring scenarios that do not support Kerberos delegation: "Connect to data on the web" and "Create datasource on the web." feature does not support delegation yet. Specifically, if you create a datasource that uses Kerberos on the with web authoring, the data source will use Run As service account authentication. If you want to use Kerberos delegation to create a datasource, then you must publish with Tableau Desktop. For more information on Run As service account, see [Enable Kerberos Service Account Access](#).

Configure Mutual SSL Authentication

Using mutual SSL, you can provide users of Tableau Desktop, Tableau Mobile, and other approved Tableau clients a secure, direct-access experience to Tableau Server. With mutual SSL, when a client with a valid SSL certificate connects to Tableau Server, Tableau Server confirms the existence of the client certificate and authenticates the user, based on the user name in the client certificate. If the client does not have a valid SSL certificate, Tableau Server can refuse the connection.

You can also configure Tableau Server to fall back to username/password authentication if mutual SSL fails. Additionally, a user can log in using the REST API with a username and password (if one exists) whether or not fallback authentication is configured.

User authentication session time limits

When users log in with mutual SSL, the authentication session is governed by the same method that governs the Tableau Server global authentication session configuration.

For clients that connect to Tableau Server using a web browser, configuration of the global authentication session is described in the *Security Hardening Checklist*, see 9. Verify session lifetime configuration.

Sessions for connected clients (Tableau Desktop, Tableau Mobile, Tableau Prep Builder, and Bridge) use OAuth tokens to keep users logged in by re-establishing a session. By default, OAuth client tokens reset after a year. If a client token has not been used in 14 days, then it will expire. You can change these values by setting the `refresh_token.absolute_expiry_in_seconds` and `refresh_token.idle_expiry_in_seconds` options. See `tsm configuration set Options`.

Certificate usage

Before you enable and configure mutual SSL, you must configure external SSL. External SSL authenticates Tableau Server to the client and encrypts the session using the certificate and key that is required when you configure external SSL.

For mutual SSL, an additional certificate file is required. The file is a concatenation of CA certificate files. The file type must be `.crt`. A "CA" is a *certificate authority* that issues certificates to the client computers that will connect to Tableau Server. The action of uploading the CA certificate file establishes a trust, which enables Tableau Server to authenticate the individual certificates that are presented by the client computers.

As part of your disaster recovery plan, we recommend keeping a backup of the certificate and revocation (if applicable) files in a safe location off of the Tableau Server. The certificate and revocation files that you add to Tableau Server will be stored and distributed to other nodes by the Client File Service. However, the files are not stored in a recoverable format. See `Tableau Server Client File Service`.

RSA key and ECDSA curve sizes

The CA certificate used for mutual SSL must either have an RSA key strength of 2048, or ECDSA curve size of 256.

.You can configure Tableau Server to accept the less-secure sizes by setting the respective configuration keys:

- `ssl.client_certificate_login.min_allowed.rsa_key_size`
- `ssl.client_certificate_login.min_allowed.elliptic_curve_size`

See `tsm configuration set Options`.

Client certificate requirements

Users authenticating to Tableau Server with mutual SSL must present a client certificate that meets minimum security requirements.

Signing algorithm

Client certificates must use a SHA-256 or greater signing algorithm.

Tableau Server configured for mutual SSL authentication will block authentication of users with client certificates that use the SHA-1 signing algorithm.

Users who attempt to log in with SHA-1 client certificates encounter an "Unable to sign in" error, and the following error will be visible in the VizPortal logs:

```
Unsupported client certificate signature detected: [certificate  
Signature Algorithm name]
```

You can configure Tableau Server to accept the less secure SHA-1 signing algorithm by setting the `ssl.client_certificate_login.blocklisted_signature_algorithms` `tsm configuration` option.

RSA key and ECDSA curve sizes

The client certificate used for mutual SSL must either have an RSA key strength of 2048, or ECDSA curve size of 256.

Tableau Server will fail mutual authentication requests from client certificates that do not meet these requirements. You can configure Tableau Server to accept the less-secure sizes by setting the respective configuration keys:

- `ssl.client_certificate_login.min_allowed.rsa_key_size`
- `ssl.client_certificate_login.min_allowed.elliptic_curve_size`

See tsm configuration set Options.

Use the TSM web interface

1. Configure SSL for External HTTP Traffic to and from Tableau Server.
2. Open TSM in a browser:

`https://<tsm-computer-name>:8850`. For more information, see Sign in to Tableau Services Manager Web UI.
3. On the **Configuration** tab, select **User Identity & Access > Authentication Method**.
4. Under **Authentication Method**, select **Mutual SSL** in the drop-down menu.
5. Under Mutual SSL, select **Use mutual SSL and automatic sign in with client certificates**.
6. Click **Select File** and upload your certificate authority (CA) certificate file to the server.

The file (.crt) is an all-in-one file that includes certificates of CAs that are used for client authentication. The file you upload must be a concatenation of the various PEM-encoded certificate files, in order of preference.

7. Enter remaining SSL configuration information for your organization.

Username format: When Tableau Server is configured for mutual SSL, the server gets the user name from the client certificate, so it can establish a direct sign-in for the client user. The name that Tableau Server uses depends on how Tableau Server is configured for user authentication:

- Local Authentication—Tableau Server uses the UPN (User Principal Name) from the certificate.

Tableau Server on Linux Administrator Guide

- Active Directory (AD)—Tableau Server uses LDAP (Lightweight Directory Access Protocol) to get the user name.

Alternatively, you can set Tableau Server to use the CN (Common Name) from the client certificate.

The screenshot shows the 'Authentication Method' configuration window. At the top, it says 'Specify how Active Directory manages user authentication and access to Tableau Server.' Below this is a dropdown menu set to 'Mutual SSL'. Under the 'Mutual SSL' section, there are two checkboxes: 'Use mutual SSL and automatic sign in with client certificates' (unchecked) and 'Use username and password if SSL authentication fails' (unchecked). The second checkbox has a sub-section for 'Specify a method for retrieving the username from the certificate.' with three radio button options: 'LDAP (Lightweight Directory Access Protocol)', 'UPN (User Principal Name)' (which is selected), and 'CN (Common Name)'. There is also a 'Select File' button for the 'SSL CA certificate file'. At the bottom, there are 'Cancel' and 'Save Pending Changes' buttons.

8. Click **Save Pending Changes** after you've entered your configuration information.
9. Click **Pending Changes** at the top of the page:



10. Click **Apply Changes and Restart**.

Use the TSM CLI

Step 1: Require SSL for external server communication

To configure Tableau Server to use SSL for external communication between Tableau Server and web clients, run the `external-ssl enable` command as follows, providing the names for the server certificate's `.crt` and `.key` files:

```
tsm security external-ssl enable --cert-file <file.crt> --key-
file <file.key>
```

- For `--cert-file` and `--key-file`, specify the location and file name where you saved the server's CA-issued SSL certificate (.crt) and key (.key) files.
- The above command assumes the you are signed in as a user that has the **Server Administrator** site role on Tableau Server. You can instead use the `-u` and `-p` parameters to specify an administrator user and password.
- If the certificate key file requires a passphrase, include the `--passphrase` parameter and value.

Step 2: Configure and enable mutual SSL

Add mutual authentication between the server and each client, and allow for Tableau client users to be authenticated directly after the first time they provide their credentials.

1. Run the following command:

```
tsm authentication mutual-ssl configure --cert-file
<file.crt>
```

For `--cert-file`, specify the location and file name of the certificate authority (CA) certificate file.

The file (.crt) is an all-in-one file that includes certificates of CAs that are used for client authentication. The file you upload must be a concatenation of the various PEM-encoded certificate files, in order of preference.

2. Run the following commands to enable mutal SSL and apply the changes:

```
tsm authentication mutual-ssl enable

tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

Additional options for mutual SSL

You can use `mutual-ssl configure` to configure Tableau Server to support the following options.

For more information, see [tsm authentication mutual-ssl <commands>](#).

Fallback authentication

When Tableau Server is configured for mutual SSL, authentication is automatic and clients must have a valid certificate. You can configure Tableau Server to allow a fallback option, to accept user name and password authentication.

```
tsm authentication mutual-ssl configure -fb true
```

Tableau Server accepts username and password authentication from REST API clients, even if the above option is set to `false`.

User name mapping

When Tableau Server is configured for mutual SSL, the server authenticates the user directly by getting the user name from their client certificate. The name that Tableau Server uses depends on how the server is configured for user authentication:

- **Local Authentication**—uses the UPN (User Principal Name) from the certificate.
- **Active Directory (AD)**—uses LDAP (Lightweight Directory Access Protocol) to get the user name.

You can override either of these defaults to set Tableau Server to use the common name.

```
tsm authentication mutual-ssl configure -m cn
```

For more information, see [Mapping a Client Certificate to a User During Mutual Authentication](#)

Certificate Revocation List (CRL)

You might need to specify a CRL if you suspect that a private key has been compromised, or if a certificate authority (CA) did not issue a certificate properly.

```
tsm authentication mutual-ssl configure -rf <revoke-file.pem>
```

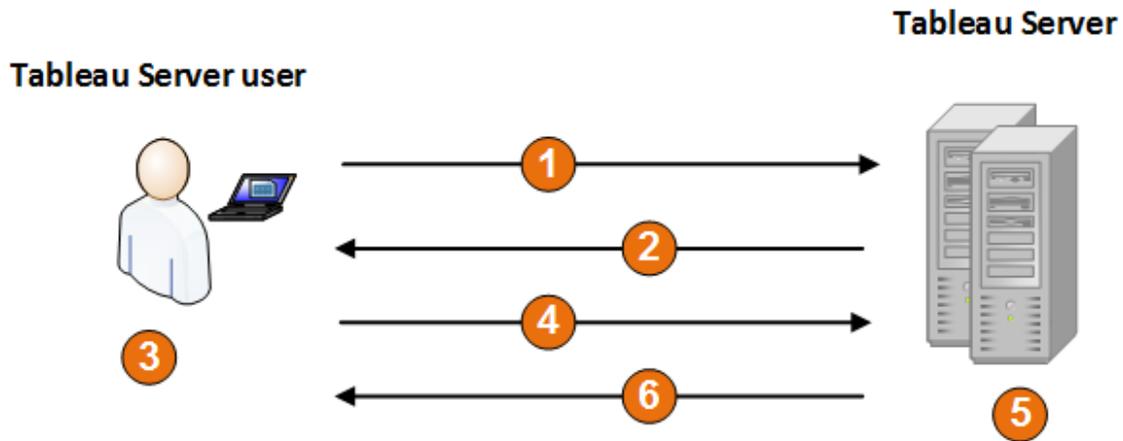
How Mutual SSL Authentication Works

Mutual (or two-way) SSL authentication provides a combination of an encrypted data stream, mutual authentication of both server and client, and direct access convenience. To use mutual SSL with Tableau Server, you need the following:

- External SSL configured on Tableau Server.
- A trusted CA-issued SSL certificate for Tableau Server. The file is a concatenation of CA certificate files. A "CA" is a *certificate authority* that issues certificates to the client computers that will connect to Tableau Server. The action of uploading the CA certificate file establishes a trust, which enables Tableau Server to authenticate the individual certificates that are presented by the client computers.
- A certificate on each client that will connect to Tableau Server.
- A Tableau Server configured to use mutual SSL.

Tableau Server and the client verify that each other has a valid certificate, and Tableau Server authenticates the user, based on the user name in the client certificate.

The following image shows a little more detail about the sequence of events that occurs with mutual SSL.



1. The user navigates to Tableau Server.
2. Tableau Server sends its SSL certificate to the client computer.
3. The client computer verifies the Tableau Server certificate.
4. The client computer sends its certificate to Tableau Server.
5. Tableau Server verifies the client certificate.
6. Tableau Server references the user name in the client certificate to authenticate the user.

Mapping a Client Certificate to a User During Mutual Authentication

When you use mutual (two-way) SSL authentication, the client presents its certificate to Tableau Server as part of the authentication process. Tableau Server then maps user information in the client certificate to a known user identity. The strategy that Tableau Server uses to perform client mapping depends on the content of your organization's client certificates.

This topic discusses the ways information in a client certificate can map to a user identity and how to change the way Tableau Server performs that mapping. To understand how the mapping happens and whether you need to change it, you must know how client certificates are structured in your organization.

- [User-name mapping options](#)
- [Change the certificate mapping](#)
- [Address user-name ambiguity in multi-domain organizations](#)

User-name mapping options

Tableau Server uses one of the following approaches to map a client certificate to a user identity:

- **Active Directory.** If Tableau Server is configured to use Active Directory for user authentication, when Tableau Server receives a client certificate, it passes the certificate to Active Directory, which maps the certificate to an Active Directory identity. Any explicit user name information in the certificate is ignored.

Note: This approach requires client certificates to be published for the user accounts in Active Directory.

- **User principal name (UPN).** A client certificate can be configured to store the user name in the user principal name field. Tableau Server reads the UPN value and maps it to a user in Active Directory or to a local user.
- **Common name (CN).** A client certificate can be configured to store the user name in the common name field of the certificate. Tableau Server reads the CN value and maps it to a user in Active Directory or to a local user.

If you configure the server for Active Directory authentication and UPN or CN user-name mapping, put the user name in one of the following formats:

`username, domain/username, or username@domain.`

For example: `jsmith, example.org/jsmith, or jsmith@example.org.`

If the server uses local authentication, the format of the name in the UPN or CN fields is not predetermined, but the name in the field must match a user name on the server.

Tableau Server on Linux Administrator Guide

Change the certificate mapping

You use the `tsm authentication mutual-ssl <commands>` commands to map a client certificate to a user identity in Tableau Server:

```
tsm authentication mutual-ssl configure -m <value>
```

Possible values are `ldap` for Active Directory mapping, `upn` for UPN mapping, or `cn` for CN mapping.

When you first install and configure Tableau Server, the server sets the default user-name mapping to match the server's authentication type:

- If the server is configured to use Active Directory, it also uses Active Directory for mapping the certificate to the user identity.
- If the server is configured to use local authentication, the server gets the user-name value from the UPN field in the certificate.

If the default behavior for how Tableau Server maps a user name to an identity is not correct for your server configuration, run the following set of commands to change the mapping to use the CN value:

```
tsm authentication mutual-ssl configure -m cn
```

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

Address user-name mapping ambiguity in multi-domain organizations

Under some circumstances, the user name in a certificate's UPN or CN field can be ambiguous. This ambiguity can lead to unexpected results when the user name is mapped to a user identity on the server.

For example, if Tableau Server is presented with a user name that does not include a domain, the server maps the user name to an identity using the default domain. This can cause an incorrect user-name mapping, potentially assigning a user a different user's identity and permissions.

This can occur particularly in environments where the following conditions apply:

- Your organization supports multiple Active Directory domains.
- The server is configured to use Active Directory authentication.
- The server is configured to use UPN or CN mapping.
- Some users have the same user name but different domains. For example, `jsmith@example.org` and `jsmith@example.com`.
- The user name in the certificate's UPN or CN fields does not include the domain as part of the user name—for example, it shows `jsmith`.

To avoid incorrect user-name mapping, make sure the client certificates include fully qualified user names with the domain, using the format `jsmith@example.org` or `example.org/jsmith`.

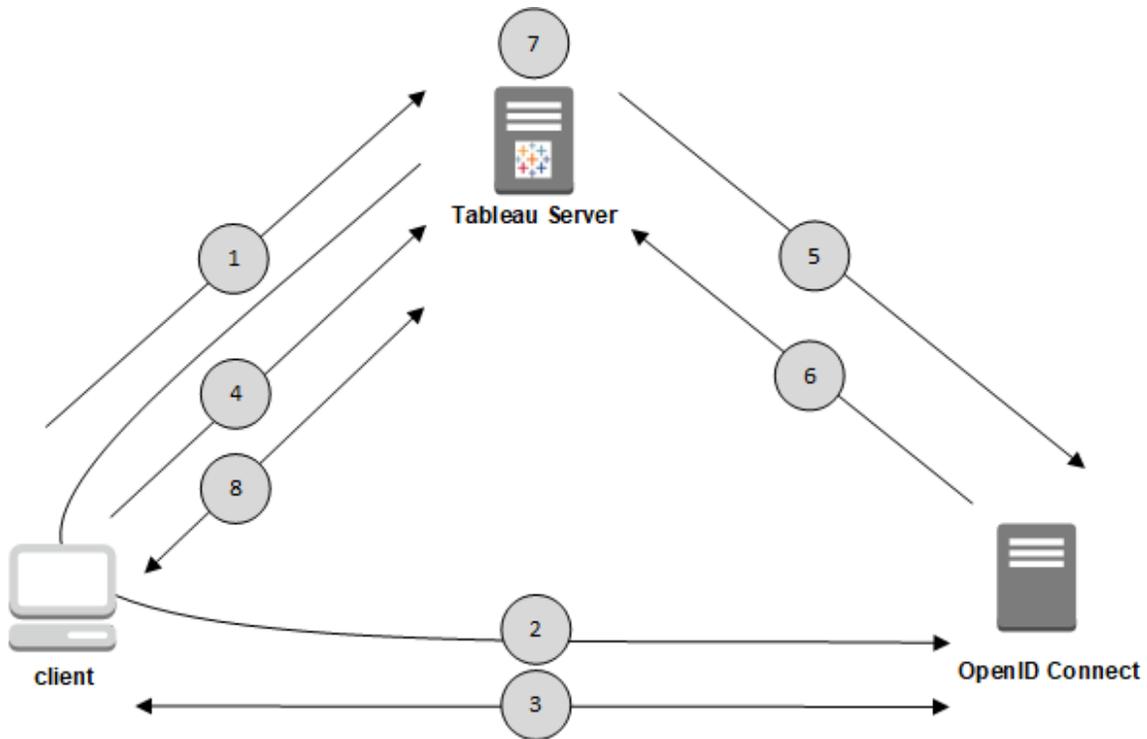
OpenID Connect

You can configure Tableau Server to support OpenID Connect for single sign-in (SSO). OpenID Connect is a standard authentication protocol that lets users sign in to an identity provider (IdP) such as Google. After they've successfully signed in to their IdP, they are automatically signed in to Tableau Server.

Configuring OpenID Connect involves several steps. The topics in this section provide general information about using Tableau Server with OpenID Connect, and provide a sequence for configuring the IdP and Tableau Server.

Authentication overview

This section describes the OpenID Connect authentication process with Tableau Server.



Step 1: A user attempts to log in to Tableau Server from a client computer.

Step 2: Tableau Server redirects the request for authentication to the IdP gateway.

Step 3: The user is prompted for credentials and successfully authenticates to the IdP. The IdP responds with a redirect URL back to Tableau Server. The redirect URL includes an authorization code for the user.

Step 4: The client is redirected to Tableau Server and presents the authorization code.

Step 5: Tableau Server presents the client's authorization code to the IdP along with its own client credentials. Tableau Server is also client of the IdP. This step is intended to prevent spoofing or man-in-the-middle attacks.

Step 6: The IdP returns an access token and an ID token to Tableau Server.

- JSON Web Token (JWT) validation: By default Tableau Server performs a validation of the IdP JWT. During discovery, Tableau Server retrieves the public keys specified by the `jwtks_uri` in the IdP configuration discovery document. Tableau Server validates the ID token for expiry and then verifies the JSON web signature (JWS), the issuer (IdP), and the client ID. You can learn more about the JWT process in the OpenID documentation, [10. Signatures and Encryption](#), and the IETF proposed standard, [JSON Web Token](#). We recommend leaving JWT validation enabled, unless your Idp does not support it.
- The ID token is a set of attribute key-pairs for the user. The key-pairs are called *claims*. Here is an example IdP claim for a user:

```
"sub" :
"7gYhRR3HiRRCaRcgvY50ubrtjGQBMJW4rXbpPFp-
g2cptHP62m2sqowM7G1LwjN5"
"email" : "alice@tableau.com",
"email_verified" : true,
"name" : "Alice Adams",
"given_name" : "Alice",
"family_name" : "Adams",
"phone_number" : "+359 (99) 100200305",
"profile" : "https://tableau.com/users/alice"
```

Step 7: Tableau Server identifies the user from the IdP claims and completes the authentication request from Step 1. Tableau Server searches for the user's account record stored in the repository by matching the "sub" (subject identifier) to identify the correct user account. If no user account is stored with the sub claim value, then Tableau Server searches for a username in the repository that matches the "email" claim from the IdP.

When a username match succeeds, Tableau Server will store the corresponding sub claim to the user's record in the repository. Tableau Server can be configured to use different claims for this process. See [Requirements for Using OpenID Connect](#).

Step 8: Tableau Server authorizes the user.

How Tableau Server Works with OpenID Connect

OpenID Connect is a flexible protocol that supports many options for the information that's exchanged between a service provider (here, Tableau Server) and an IdP. The following list provides details about the Tableau Server implementation of OpenID Connect. These details can help you understand what types of information Tableau Server sends and expects, and how to configure an IdP.

- Tableau Server supports only the OpenID Authorization Code Flow as described in the [OpenID Connect final specification](#).
- Tableau Server relies on using discovery or a provider URL to retrieve the OpenID Provider metadata. Alternatively, you can host a static discovery document on Tableau Server. For more information see [Configure Tableau Server for OpenID Connect](#).
- Tableau Server supports only the `client_secret_jwt` Client Authentication method specified in the OpenID Connect specification. In addition, Tableau Server supports only RSA Asymmetric Encryption for handling the JWT. However, you can turn off JWT validation. See `tsm authentication openid <commands>`.
- Tableau Server expects a `kid` value in the `id_token` attribute's JOSE Header. This value is matched with one of the keys found in the JWK Set document, whose URI is specified by the `jwtks_uri` value in the OpenID discovery document. A `kid` value must be present even if there is only one key in the JWK Set document.
- Tableau Server does include OpenID support for the JWK `x5c` parameter or for using X.509 certificates.
- Tableau Server ignores the proxy settings configured on the operating system and

sends all OpenID requests directly to the IdP. You cannot configure OpenID to honor local operating system proxy settings. Tableau Server must be able to communicate with the internet.

Requirements for Using OpenID Connect

This topic describes the requirements to use OpenID Connect with Tableau Server.

IdP account

You must have access to an identity provider (IdP) that supports the OpenID Connect (OIDC) protocol. You must also have an account with the IdP. OpenID Connect is supported by many identity providers. The OIDC protocol is an open and flexible standard, and as such, not all implementations of the standard are identical. As you configure Tableau Server for OIDC, work with your IdP.

The Google IdP implementation has been extensively tested with Tableau Server and is the model IdP for the configuration documented in these topics.

Local identity store

To use OpenID Connect on Tableau Server, the server must be configured to use local identity store. The server must be configured so that you explicitly create users on the Tableau Server, rather than importing them from an external directory such as Active Directory. Managing users with an external identity store is not supported with OpenID.

IdP claims: mapping users

To sign in successfully to Tableau Server, a given user must be provisioned in OpenID and then mapped to a user account on Tableau Server. OpenID uses a method that relies on *claims* to share user account attributes with other applications. Claims include user account attributes such as email, phone number, given name, etc. To understand how Tableau Server maps IdP claims to user accounts, see [Authentication overview](#).

Tableau Server relies on the IdP claims to map user accounts from the IdP to those hosted on Tableau Server. By default, Tableau Server expects the IdP to pass the email claim.

Depending on your IdP, you may need to configure Tableau Server to use a different IdP claim.

If you are using Google as an IdP, then use the default, `email` claim to map IdP identities to Tableau Server user accounts. If you are not using Google as an IdP, then work with your IdP to determine the claim for which you should configure Tableau Server.

Default: using email claim to map users

By default, the user's user name in Tableau Server must match the `email` claim in the IdP ID token. Therefore, in the default configuration, you must use email addresses (also referred to as UPN) as the username in Tableau Server. If you use Google as the IdP, the user name in Tableau Server must be the user's Gmail address (`alice@gmail.com`). Using a complete email address helps to guarantee the uniqueness of the user name in Tableau Server, even when two users have the same email but are on different email hosts.

Note: When you create a user identity in Tableau Server, you specify a user name, password, and optionally an email address. For using OpenID Connect in the default configuration, the user name (expressed as an email address) is the value that must match the user's name in the IdP. The optional email address in the Tableau Server user identity is not used for OpenID authentication.

Ignoring the domain name

You can configure Tableau to ignore the domain portion of an email address when matching the IdP `email` claim to a user account on Tableau Server. In this scenario, the `email` claim in the IdP might be `alice@example.com`, but this will match a user named `alice` in Tableau Server. Ignoring the domain name might be useful if you already have users defined in Tableau Server that match the user names portion of the `email` claim, but not the domain portions.

Important: We do not recommend ignoring the user domain name without taking precautions. Specifically, verify that user names are unique across the configured domains that you've created in your IdP.

Setting Tableau Server to ignore the user domain name has the potential to result in unintended user log on. Consider the case where your IdP has been configured for multiple domains (`example.com` and `tableau.com`). If two users with the same first name, but different user accounts (`alice@tableau.com` and `alice@example.com`) are in your organization, then the first one to complete the OpenID provisioning sequence will claim the `sub` mapping in the IdP. If the wrong user is mapped, then the other user will be unable to log on until the associated `sub` value is reset.

To configure Tableau Server to ignore domain names in user names from the IdP, set `tsm authentication openid configure --ignore-domain` to `true`. For more information, see `tsm authentication openid <commands>`.

When you change the `tsm authentication openid configure --ignore-domain` option to ignore the domain in user names, all user names in Tableau Server must have a domain name.

Using custom claims to map users

As referenced in *Authentication overview*, the `sub` claim is often included in IdP claims. Typically, the `sub` claim is a unique string that identifies a given user account. The benefit of using a `sub` claim is that it will not change, even if you or another admin updates other user attributes or IdP claims (email, phone number, etc) associated with that account. By default, Tableau Server identifies and verifies OpenID users according to the `sub` claim in the IdP ID token.

The OpenID `sub` claim value must be mapped to the corresponding user in Tableau Server. Since the `sub` claim is an arbitrary string, a different claim is used to associate accounts during the first sign-in session. The first time a user signs in to Tableau Server with OpenID, Tableau will match the OpenID user account to a corresponding user account on

Tableau Server on Linux Administrator Guide

Tableau. By default, Tableau will use the IdP claim, `email`, to identify the Tableau user. Tableau will then update that user's record with the `sub` claim from OpenID. Since the ID token always includes the `sub` claim along with other claims, on subsequent sessions, Tableau will identify that user with the `sub` claim only.

For some organizations, mapping user names with the email address is not reliable or not supported by the IdP. Beginning with Tableau Server 10.2, you can map user accounts from any arbitrary IdP claim to the Tableau Server username.

The IdP claim you are using must map exactly to a corresponding Tableau Server username. In the example below, the username is `kwilliams`.

The screenshot shows the Tableau Server interface with the 'New User' dialog box open. The dialog contains the following fields and options:

- Username:** (Status: Username available)
- Display name:**
- Password:**
- Confirm password:**
- Email (optional):**
- Site:** (Search sites:)
- Site role table:**

Site	Site role
<input checked="" type="checkbox"/> Customer Support	<input type="text" value="Interactor"/>
<input checked="" type="checkbox"/> Default	<input type="text" value="Publisher"/>
<input type="checkbox"/> Development	
- Selected users are Server Administrators
- Buttons:** Cancel, Create

To change the IdP claim that is used to map identity on Tableau Server, use the `tsm authentication openid map-claims --user-name` command. For more information, see `tsm authentication openid <commands>`.

Changing the `sub` claim

As described above, the `sub` claim is the identifier that Tableau Server uses to identify users after the initial mapping session. The `sub` claim is written to the corresponding user account in Tableau Server. If your IdP does not provide a `sub` claim, then you can specify an arbitrary claim to use instead. Like `sub`, the claim value you specify must be unique and should not change when other user claims are updated.

To specify a different IdP claim for default sub claim, use the use the `tsm authentication openid map-claims --id` command. For more information, see `tsm authentication openid <commands>`.

Where `arbitraryClaim` is the name of the IdP claim that you want to use as the replacement for the `sub` claim.

Authentication context

If your OpenID Connect IdP requires a specific authentication context, you can specify a list of essential and voluntary ACR values using the `vizportal.openid.essential_acr_values` and `vizportal.openid.voluntary_acr_values` configuration keys. For more information, see `tsm configuration set Options`.

Configure the Identity Provider for OpenID Connect

This topic provides information about configuring an identity provider (IdP) to use OpenID Connect with Tableau Server. This is one step in a multi-step process. The following topics provide information about configuring and using OpenID Connect with Tableau Server.

- OpenID Connect
- Configure the Identity Provider for OpenID Connect (you are here)
- Configure Tableau Server for OpenID Connect
- Signing In to Tableau Server Using OpenID Connect

Configure the IdP

Before you can use OpenID Connect with Tableau Server, you must have an account with an identity provider (IdP) and a project or application with the IdP. When you configure Tableau Server, you will need to be able to provide the following information:

- Provider client ID. This is the identifier that the IdP assigned to your application.
- Provider client secret. This is a token that is used by Tableau to verify the authenticity of the response from the IdP. This value is a secret and should be kept securely.
- Provider configuration URL. This is the URL at the provider's site that Tableau Server should send authentication requests to.

Redirect URL

Some IdPs will require a redirect URL for your Tableau Server. You can manually construct your URL for the IdP using the following syntax:

```
<protocol>://<host>/vizportal/api/web/v1/auth/openIdLogin
```

For example, `https://tableau.example.com/vizportal/api/web/v1/auth/openIdLogin`.

Example IdP process

The following procedure provides an outline of the steps that you follow with the provider. As an example, the procedure discusses using Google as a provider. However, each provider has a somewhat different flow, so the specifics of the steps (and their order) might vary depending on your provider.

1. Register at the provider's developer site and sign in. For example, for Google, you can go to the Developers Console at this URL: <https://console.developers.google.com>
2. Create a new project, application, or relying party account.
3. In the developer dashboard, follow the steps for getting an OAuth 2.0 client ID and client secret. Record these values for later.

Note: Keep the client secret in a secure place.

4. On the developer site, find the URL of the endpoint that the IdP uses for OpenID Connect discovery. For example, Google uses the URL <https://accounts.google.com/.well-known/openid-configuration>. Record this URL for later.

Alternatively, if your IdP has provided you with a static discovery document, copy that file to a local directory on the Tableau Server for later.

Configure Tableau Server for OpenID Connect

This topic describes how to configure Tableau Server to use OpenID Connect for single-sign on (SSO). This is one step in a multi-step process. The following topics provide information about configuring and using OpenID Connect with Tableau Server.

- OpenID Connect
- Configure the Identity Provider for OpenID Connect
- Configure Tableau Server for OpenID Connect (you are here)
- Signing In to Tableau Server Using OpenID Connect

Note: Before you perform the steps described here, you must configure the OpenID identity provider (IdP) as described in [Configure the Identity Provider for OpenID Connect](#).

Use the TSM web interface

1. Open TSM in a browser:

`https://<tsm-computer-name>:8850`. For more information, see [Sign in to Tableau Services Manager Web UI](#).

2. Click User **Identity & Access** on the **Configuration** tab and then click

Authentication Method.

3. Under **Authentication Method**, select **OpenID Connect** in the drop-down menu.
4. Under OpenID Connect, select **Enable OpenID authentication for the server**.
5. Enter the OpenID configuration information for your organization:

The screenshot shows the 'Authentication Method' configuration page. At the top, it says 'Specify how Active Directory manages user authentication and access to Tableau Server.' A dropdown menu is set to 'OpenID Connect'. Below this, the 'OpenID Connect' section is active, with a checkbox 'Enable OpenID authentication for the server' checked. Step 1 instructions lead to three input fields: 'Provider client ID' (laakajwdlnaoilasdfoadjkdwha), 'Provider client secret' (masked with dots), and 'Provider configuration URL' (https://example.lan/openid-configuration). Step 2 instructions lead to a 'Tableau Server external URL' field (http://tableau.example.com). Step 3 instructions lead to a URL field (http://tableau.example.com/vizportal/api/web/v1/openIdLogin). At the bottom are 'Cancel' and 'Save Pending Changes' buttons.

Note: If your provider relies on a configuration file hosted on the local computer (rather than a file hosted at a public URL), you can specify the file with the `tsm authentication openid <commands>`. Use the `--metadata-file <file_path>` option to specify a local IdP configuration file.

6. Click **Save Pending Changes** after you've entered your configuration information.
7. Click **Pending Changes** at the top of the page:



8. Click **Apply Changes and Restart**.

Use the TSM CLI

The procedure in this section describes how to use TSM command line interface to configure OpenID Connect. You can also use a configuration file for the initial configuration of OpenID Connect. See `openIDSettings` Entity.

1. Use the `configure` command of `tsm authentication openid <commands>` to set the following required options:

`--client-id <id>`: Specifies the provider client ID that your IdP has assigned to your application. For example, "laakjwdlnaoiloadjkwaha".

`--client-secret <secret>`: Specifies the provider client secret. This is a token that is used by Tableau to verify the authenticity of the response from the IdP. This value is a secret and should be kept securely. For example, "fwahfkjaw72123=".

`--config-url <url>` or `--metadata-file <file_path>`: Specifies location of provider configuration json file. If the provider hosts a public json discovery file, then use `--config-url`. Otherwise, specify a path on the local computer and file name for `--metadata-file` instead.

`--return-url <url>`: The URL of your server. This is typically is the public name of your server, such as "http://example.tableau.com".

For example, run the command:

```
tsm authentication openid configure --client-id
"laakjwdlnaoiloadjkwaha" --client-secret "fwahfkjaw72123="
--config-url "https://example.com/openid-configuration" --
return-url "http://tableau.example.com"
```

Tableau Server on Linux Administrator Guide

There are additional, optional configurations that you can set for Open ID Connect using either `openIDSettings` Entity or `tsm authentication openid <commands>`. In addition, if you need to configure IdP claim mapping, see [Options for openid map-claims](#).

2. Type the following command to enable Open ID Connect:

```
tsm authentication openid enable
```

3. Run `tsm pending-changes apply` to apply changes.

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see [tsm pending-changes apply](#).

Signing In to Tableau Server Using OpenID Connect

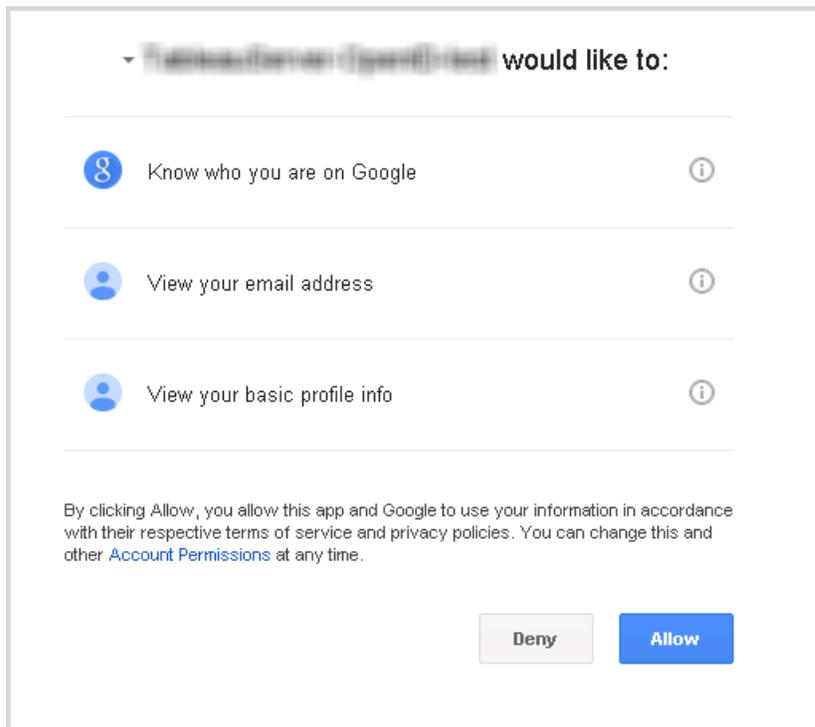
This topic provides information about signing in to Tableau Server using OpenID Connect. The following topics provide information about configuring and using OpenID Connect with Tableau Server.

- [OpenID Connect](#)
- [Configure the Identity Provider for OpenID Connect](#)
- [Configure Tableau Server for OpenID Connect](#)
- [Signing In to Tableau Server Using OpenID Connect \(you are here\)](#)

Signing in using OpenID Connect

After Tableau Server has been configured to use OpenID Connect, users who access the server and aren't already signed in are redirected to the IdP site, where they are prompted to sign in. Users enter the credentials that they have with the IdP. In many cases, the user is

also asked to authorize the IdP to share information with Tableau Server, as in the following example:



When a user signs in using OpenID Connect, the IdP sends a unique user identifier (known in OpenID as the sub value) as part of the information that's redirected to Tableau Server. This sub value is associated with the user's Tableau user identity.

Restricting sign-in to server administrators for command-line tools

Command-line tools for working with Tableau Server (`tabcmd`, TSM, and `tableau.com`) do not support sign-in using OpenID Connect. When OpenID Connect is enabled for the server, these tools still require sign-in using a Tableau Server username and password.

Even if users normally authenticate using OpenID Connect, each user has a Tableau Server username and password. This means that users could use command-line tools like `tabcmd`. As a security measure, you can make sure that *only* server administrators can use command-line tools. To do this, use `tsm configuration set to set wgserv-er.authentication.restricted to true`. When this setting is `true`, only server

Tableau Server on Linux Administrator Guide

administrators can sign in to Tableau Server using a username and password; all other users *must* sign in to the server using a single sign-on (SSO) option like OpenID Connect. The effect is that users who are not administrators also cannot then use command-line tools. To make this change, run the following sequence of TSM commands:

```
tsm configuration set -k wgserver.authentication.restricted -v true
```

```
tsm pending-changes apply
```

OpenID Connect Authentication Request Parameters

The OpenID authentication request sent from Tableau Server passes information using a limited set of parameters, as listed in this topic. If your OpenID IdP requires parameters that are not in the following list, it is not compatible for use with Tableau Server.

- `scope`. This value specifies a profile that tells the IdP what user information claims to return. This value can be configured by a Tableau Server administrator. The default value is "openid email profile". For more information, see [Configure the scope value](#) later in this document.
- `response_type`. OpenID Connect supports multiple flows. This value tells the IdP which flow Tableau Server expects. Tableau supports only the authorization code flow, and the value is always set to "code".
- `client_id`. This value specifies the server's ID (**Provider client ID** in the Tableau Server Configuration dialog box), which lets the IdP know where the request came from. It is provided by the IdP when the service is registered. The value is configurable by a Tableau Server administrator.
- `redirect_uri`. This value specifies the URL that the IdP redirects to after the user has authenticated using OpenID Connect. The URL must include the host and protocol (for example, `http://example.tableau.com`), but Tableau provides the URL endpoint.

- `nonce`. Tableau Server generates a nonce value to verify that the client that it redirected to matches the entity that comes back from the IdP.

Configure the scope value

The `scope` value indicates to the IdP the information that Tableau Server requests about the user. By default, Tableau Server sends the value "openid profile email". This indicates that Tableau uses OpenID to authenticate (this part of the `scope` attribute value must always be included) and that Tableau Server is requesting the user profile and email information during the exchange of the user authorization code.

If this default scope is not appropriate for your scenario, you can have Tableau Server request custom information about the user. To do so, you configure the IdP with a custom profile (for example, something like "tableau-scope"). You can then configure Tableau Server to send the scope request using the name of the custom profile.

To change the scope value that Tableau Server requests, use the following TSM CLI command:

```
tsm authentication openid configure --custom-scope-name custom-scope-name
```

Note: Tableau Server always includes "openid" as part of the scope value (even if you don't include it in the `custom_scope` setting).

Changing IdPs in Tableau Server for OpenID Connect

This topic provides information about changing an identity provider (IdP) if you have configured Tableau Server to use OpenID Connect.

Changing providers

You might decide to change the IdP that Tableau Server is configured to use. To do so, you follow the procedure that you used to configure the first IdP: establish an account, get a customer ID and secret, configure Tableau Server with that information, and provide the IdP

Tableau Server on Linux Administrator Guide

with the redirect URL for Tableau Server. For more information, see [Configure Tableau Server for OpenID Connect](#).

However, you also need to perform an additional step: you must clear any user identifiers (`sub` values) that have already been associated with Tableau Server users. The new IdP will have different `sub` values for each user, and you must clear the existing ones so that Tableau Server can store a new `sub` value when the user signs in using the new IdP.

To clear `sub` values for users, use the `tabcmd reset_openid_sub` command. You can reset (that is, clear) `sub` values for an individual user, as in the following example:

```
tabcmd reset_openid_sub --target-username jsmith
```

You can also clear the `sub` value for all users using this command:

```
tabcmd reset_openid_sub --all
```

Troubleshoot OpenID Connect

Use the following topics to troubleshoot OpenID Connect (OIDC) issues in Tableau Server.

OpenID Connect protocol is supported by many identity providers. The OIDC protocol is an open and flexible standard, and as such, not all implementations of the standard are identical. Most issues that administrators encounter when configuring Tableau Server for OIDC are the result of how different identity providers implement OIDC. If you encounter errors as you set up OIDC with Tableau Server, we recommend that you work with your IdP to resolve them.

Signing In from the Command Line

Even if Tableau Server is configured to use OpenID, it is not used if you sign in to Tableau Server using `tabcmd`, the [REST API](#), or the [Tableau Data Extract command line utility](#) (provided with Tableau Desktop).

Login Failed

Login can fail with the following message:

Login failure: Identity Provider authentication successful for user <username from IdP>. Failed to find the user in Tableau Server.

This error typically means that there is a mismatch between a username stored in Tableau Server and the username provided by the IdP. To fix this, make sure that they match. For example, if Jane Smith's username is stored in the IdP as `j.smith` it must be stored in Tableau Server as `j.smith` as well.

Error 69: "Unable to Sign In"

An error 69 may be returned when you attempt to sign in to Tableau Server with a web browser and receive an error, "Unable to Sign In. Sign in failed. Contact your Tableau Server administrator." The URL that returns this message is `https://example.-com/#/error/signin/69?redirectPath=%2.`

If you receive this error, check with your IDP provider to verify if the IdP is expecting `client_secret_post` instead of `client_secret_basic` (the Tableau default).

If the IdP is expecting `client_secret_post`, then you must set the `vizportal.openid.client_authentication` parameter to `client_secret_post`.

For example; If you receive this error and you have configured OIDC for the Salesforce IdP, then you must set the `vizportal.openid.client_authentication` parameter.

See `tsm configuration set Options` for more information.

OpenID Error Log

OpenID authentication takes place outside Tableau Server, so troubleshooting authentication issues can be difficult. However, sign-in attempts are logged by Tableau Server. You can create a snapshot of log files and use them to troubleshoot problems. For more information, see [Tableau Server Logs and Log File Locations](#).

Note: To log OpenID-related events, `vizportal.log.level` must be set to `debug` with tsm configuration set Options.

Check for OpenID errors in the following files in the unzipped log file snapshot:

```
\vizportal\vizportal-<n>.log
```

Trusted Authentication

When you embed Tableau Server views into webpages, everyone who visits the page must be a licensed user on Tableau Server. When users visit the page they are prompted to sign in to Tableau Server before they can see the view. If you already have a way of authenticating users on the webpage or within your web application, you can avoid this prompt and save your users from having to sign in twice by setting up trusted authentication.

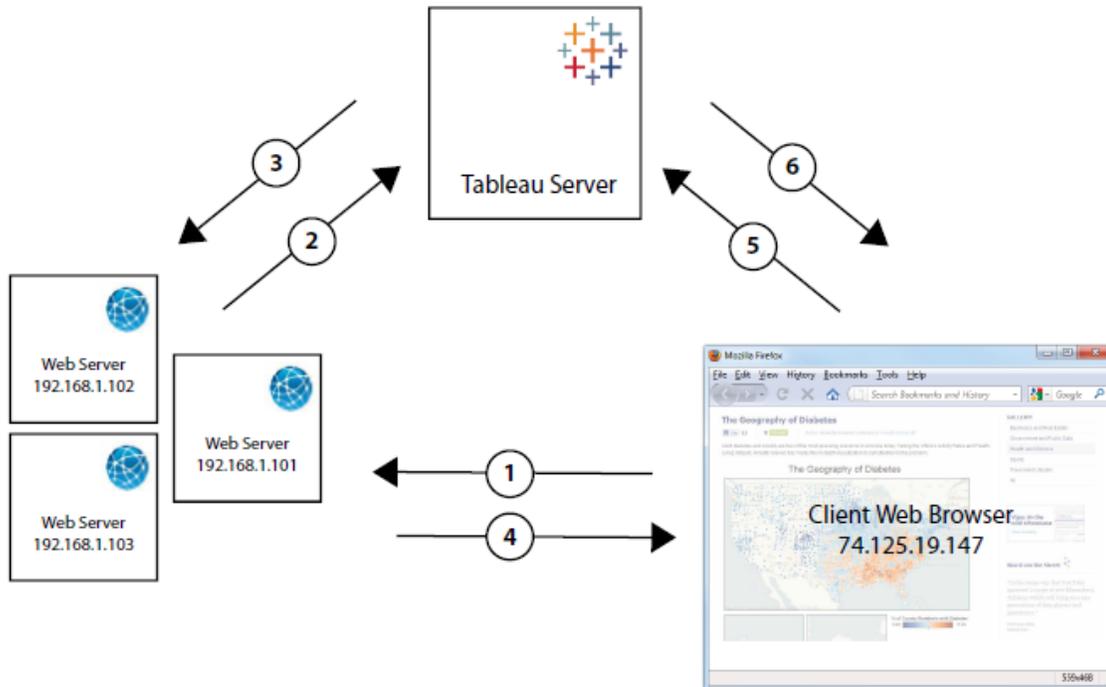
Trusted authentication simply means that you have set up a trusted relationship between Tableau Server and one or more web servers. When Tableau Server receives requests from these trusted web servers it assumes that your web server has handled whatever authentication is necessary.

If your web server uses SSPI (Security Support Provider Interface), you do not need to set up trusted authentication. You can embed views and your users will have secure access to them as long as they are licensed Tableau Server users and members of your Active Directory.

Note: Client browsers must be configured to [allow third-party cookies](#) if you want to use trusted authentication with embedded views.

How Trusted Authentication Works

The diagram below describes how trusted authentication works between the client's web browser, your web server(s) and Tableau Server.



- 1 User visits the webpage:** When a user visits the webpage with the embedded Tableau Server view, the webpage sends a GET request to your web server for the HTML for that page.
- 2 Web server POSTS to Tableau Server:** The web server sends a POST request to the trusted Tableau Server (for example, `https://tabaserver/trusted`, not `https://tabserver`). That POST request must have a `username` parameter. The `username` value must be the username for a licensed Tableau Server user. If Tableau Server is hosting multiple sites and the view is on a site other than the Default site, then the POST request must also include a `target_site` parameter.
- 3 Tableau Server creates a ticket:** Tableau Server checks the IP address or host name of the web server (192.168.1.XXX in the above diagram) that sent the POST request. If the web server is listed as a trusted host then Tableau Server creates a ticket in the form of a unique string. Tickets must be redeemed within three minutes after they are issued.

Tableau Server responds to the POST request with that ticket. Or if there is an error and the ticket cannot be created, then Tableau Server responds with a value of `-1`. The server must have an IPv4 address. IPv6 addresses are not supported. For more information, see [Ticket Value of -1 Returned from Tableau Server](#).

4 Web server passes the URL to the browser: The web server constructs the URL for the view and inserts it into the HTML for the page. The ticket is included (for example, `https://tabserver/trusted/<ticket>/views/requested_view_name`). The web server passes the HTML back to the client's web browser.

5 Browser requests view from Tableau Server: The client web browser sends a GET request to Tableau Server that includes the URL with the ticket.

6 Tableau Server redeems the ticket: Tableau Server redeems the ticket, creates a session, logs the user in, removes the ticket from the URL, and then sends the final URL for the embedded view to the client.

The session allows the user to access any of the views that the user would have if they logged onto the server. In the default configuration, users authenticated with trusted tickets have restricted access such that only views are available. They cannot access workbooks, project pages, or other content hosted on the server.

To change this behavior, see the `wgserver.unrestricted_ticket` option at [tsm configuration set Options](#).

How is a trusted ticket stored?

Tableau Server stores trusted tickets in the Tableau Server repository using the following process:

1. Tableau Server generates a two-part ticket: the first part is a Base64-encoded unique ID (UUID) and the second part is a 24-character random secret string.
2. Tableau Server hashes the secret string and stores it with the unique ID in the repository. Hashing takes the secret string as input, and uses an algorithm to compute a

unique string. This unique string protects the security of the secret string from unauthorized users.

3. Tableau Server sends the Base64 UUID and the original 24-character random string to the client.
4. The client returns the Base64 UUID and the original 24-character secret string to Tableau Server as part of the request for a view.
5. Tableau Server locates the string pair with the Base64 UUID, and then hashes the secret string to verify that it matches the hash stored in the repository.

This process ensures that any trusted ticket content stored on Tableau Server cannot be used to impersonate users or access content protected by authentication. However, because the full trusted ticket is sent over HTTP between Tableau Server and the client, the process relies on secure and encrypted transmission of HTTP data. Therefore, we recommend that you only deploy trusted tickets over SSL/TLS or another layer of network encryption.

Add Trusted IP Addresses or Host Names to Tableau Server

The first step in setting up trusted authentication is to configure Tableau Server to recognize and trust requests from one or more web servers:

Use the TSM web interface

1. Open TSM in a browser:

https://<tsm-computer-name>:8850. For more information, see Sign in to Tableau Services Manager Web UI.
2. Click User **Identity & Access** on the **Configuration** tab and then click **Trusted Authentication**.
3. Under **Trusted Authentication**, for each trusted host, enter the name or IP address and then click **Add**:

Tableau Server on Linux Administrator Guide

Trusted Authentication

Use trusted authentication to allow single sign-on to view Tableau Server content embedded in webpages. Establish a trusted relationship between Tableau Server and one or more web server by adding trusted hosts and specifying token length for each trusted ticket. Do not set up trusted authentication if your web server uses SSPI. [Learn more](#)

Trusted hosts	<input type="text" value="10.32.139.6"/>	<input type="button" value="Delete"/>
	<input type="text" value="webserv1"/>	<input type="button" value="Delete"/>
	<input type="text" value="webserv2"/>	<input type="button" value="Add"/>
Token Length	<input type="text" value="24"/>	<input type="button" value="x"/>

Notes:

The values you specify completely overwrite any previous setting. Therefore, you must include the full list of hosts if you want to amend an existing list.

Static IP addresses are required: The web servers you specify must use static IP addresses, even if you use host names.

If you have one or more proxy servers between the computer that is requesting the trusted ticket (one of those configured in Step 2 as shown at Trusted Authentication) and Tableau Server, you also need to add them as trusted gateways using the `tsm configuration set gateway.trusted` option. See [Configuring Proxies for Tableau Server](#) for steps.

4. Enter a value in **Token Length** (Optional).

The token length determines the number of characters in each trusted ticket. The default setting of 24 characters provides 144 bits of randomness. The value can be set to any integer between 9 and 255, inclusive.

5. Click **Save Pending Changes** after you've entered your configuration information.
6. Click **Pending Changes** at the top of the page:



7. Click **Apply Changes and Restart**.

Use the TSM CLI

1. Enter the following command:

```
tsm authentication trusted configure -th <trusted IP
address or host name>
```

In the command above, <trusted IP address> should be a comma-separated list of the IPv4 addresses or host names of your web server(s), with each host name or IP address in quotes.

Note: The values you specify completely overwrite any previous setting. Therefore, you must include the full list of hosts in the `tsm authentication trusted configure -th` command. (You cannot amend the list of hosts by running the `tsm authentication trusted configure -th` command repeatedly.)

For example:

```
tsm authentication trusted configure -th "192.168.1.101",
"192.168.1.102", "192.168.1.103"
```

or

```
tsm authentication trusted configure -th "webserv1", "web-
serv2", "webserv3"
```

Notes:

Each host name or IP address in the list must be in double-quotes, followed by a

comma and one space after each comma.

The web servers you specify must use static IP addresses, even if you use host names.

2. If you have one or more proxy servers between the computer that is requesting the trusted ticket (one of those configured in Step 2 as shown at Trusted Authentication) and Tableau Server, you also need to add them as trusted gateways using the `tsm configuration set gateway.trusted` option. See [Configuring Proxies for Tableau Server](#) for steps.

3. Type the following command to save the changes to all the server configuration files:

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

There are other optional trusted authentication configurations (legacy support, logging, and timeout settings) that you can make by passing a json file to Tableau Server. See `trustedAuthenticationSettings` Entity.

Next, you need to [configure your web server to receive tickets from Tableau Server](#).

Get a Ticket from Tableau Server

After you've [added trusted IP addresses](#) to Tableau Server, you're ready to configure your web server to get tickets from Tableau Server via POST requests ([Step 2 in the diagram](#)).

The POST request must be sent to `http://<server name>/trusted`, not `http://t-abserv`. For example `http://tabserv/trusted`.

Note: If SSL is enabled you must use https instead of http. For example: `https://tabserver/trusted`.

Here's the data you can use in a POST request to Tableau Server:

- **username=<username>** (required): The username for a licensed Tableau Server user. If you are using Local Authentication the username can be a simple string (for example, `username=jsmith`). If you are using Active Directory with multiple domains you must include the domain name with the user name (for example, `username=MyCo\jsmith`).
- **target_site=<site id>** (required if view not on Default site): Specifies the site containing the view if Tableau Server is running **multiple sites** and the view is on a site other than the Default site (for example, `target_site=Sales`). The value you use for `<site id>` should be the **Site ID** that was provided when the site was created. This value is case sensitive. If the **Site ID** is `SAles`, then the `target_site=SAles`.
- **client_ip=<IP address>** (optional): Used to specify the IP address of the computer whose web browser is accessing the view (for example, `client_ip=123.45.67.891`). It is not the IP address of the web server making the POST request of Tableau Server. If you decide to use this parameter, see [Optional: Configure Client IP Matching](#) for more information.

Tableau Server's response to the POST request will be a unique string (the ticket). If Tableau Server isn't able to process the request, the return will be `-1`. See [Ticket Value of -1 Returned from Tableau Server](#) for tips on how to correct this. Also, in order for users to successfully authenticate when they click an embedded view, their browsers must be configured to [allow third-party cookies](#).

The ticket format changed in Tableau Server 10.2. The ticket format is now a string composed of two parts. Each part is a 128 bit string that is encoded before it is returned to the client. The first part is a universally unique ID (UUID v4) that is Base64-encoded. The second

Tableau Server on Linux Administrator Guide

part is a 24-character secure random string. The concatenation of these parts can be expressed as Base64(UUIDv4):SecureRandomString. An example of a ticket might look like this: 9D10byqDQmSIOyQpKdy4Sw==:dg62gCsSE0QRArXNTOp6m1J5.

Next, you need to add code that allows the web server to **construct an URL** for the view that includes the view's location and the ticket.

Display the View with the Ticket

After you **create the POST request**, you need to write code that provides the web server with the view's location and the ticket from Tableau Server. It will use this information to display the view. How you specify it depends on whether the view is embedded, and if Tableau Server is running multiple sites.

Tableau Server View Examples

Here's an example of how to specify a view that users only access via Tableau Server (the view is not embedded):

```
http://tabserver/trusted/<ticket>/views/<workbook>/<view>
```

If Tableau Server is running **multiple sites** and the view is on a site other than the Default site, you need to add `t/<site ID>` to the path. For example:

```
http://t-  
abserver/trusted/<ticket>/t/Sales/views/<workbook>/<view>
```

Use the same capitalization that you see in the Tableau Server URL.

Embedded View Examples

Here are some examples of how to specify embedded views. Because there are two approaches you can take with embed code, both ways are provided below. Regardless of which you use, there is some information unique to trusted authentication that you must provide. For more information, search for "Writing Embed Code" in the Tableau Server Help.

Note: The examples below use embed code parameters. For more information, see [Embed Code Parameters](#) in the Tableau Help.

Script Tag Examples

This example uses the `ticket` object parameter:

```
<script type="text/javascript" src-
c="http://myserver/javascripts/api/viz_v1.js"></script>
<object class="tableauViz" width="800" height="600" style-
e="display:none;">
  <param name="name" value="MyCoSales/SalesScoreCard" />
  <param name="ticket" value-
e="9D10byqDQmSIOyQpKdy4Sw==:dg62gCsSE0QRARXNTOp6mlJ5" />
</object>
```

Here's what the above example looks like for a multi-site Tableau Server, where the view is published on the `Sales` site:

```
<script type="text/javascript" src-
c="http://myserver/javascripts/api/viz_v1.js"></script>
<object class="tableauViz" width="800" height="600" style-
e="display:none;">
  <param name="site_root" value="/t/Sales" />
  <param name="name" value="MyCoSales/SalesScoreCard" />
  <param name="ticket" value-
e="9D10byqDQmSIOyQpKdy4Sw==:dg62gCsSE0QRARXNTOp6mlJ5" />
</object>
```

Instead of using `ticket`, you can use the `path` parameter to state the full path of the view explicitly. When `path` is used, you do not also need the `name` parameter, which is usually a required parameter in Tableau JavaScript embed code:

Tableau Server on Linux Administrator Guide

```
<script type="text/javascript" src-  
c="http://myserver/javascripts/api/viz_v1.js"></script>  
<object class="tableauViz" width="900" height="700" style-  
e="display:none;">  
    <param name="path" value-  
="tru-  
sted/9D1ObyqDQmSIOyQpKdy4Sw-  
w==:dg62gCsSE0QRARXNTOp6mlJ5/views/MyCoSales/SalesScoreCard" />  
</object>
```

Here's the same example, but for a multi-site server. Note that `/t/<site ID>` is used here:

```
<script type="text/javascript" src-  
c="http://myserver/javascripts/api/viz_v1.js"></script>  
<object class="tableauViz" width="900" height="700" style-  
e="display:none;">  
    <param name="path" value-  
="tru-  
sted/9D1ObyqDQmSIOyQpKdy4Sw-  
==:dg62gCsSE0QRARXNTOp-  
p6mlJ5/t/Sales/views/MyCoSales/SalesScoreCard" />  
</object>
```

Iframe Tag Example

```
<iframe src-  
="h-  
ttp://t-  
abserver-  
/trus-  
ted/9D1ObyqDQmSIOyQpKdy4Sw-  
==:dg62gCsSE0QRARXNTOp-  
p6mlJ5/views/workbookQ4/SalesQ4?:embed=yes" width="800"  
height="600"></iframe>
```

Optional: Configure Client IP Matching

By default, Tableau Server does not consider the client web browser IP address when it creates or redeems tickets. To change this, you need to do two things: specify an IP address using the `client_ip` parameter in the POST request that obtains the ticket, and follow the steps below to configure Tableau Server to enforce client IP address matching.

1. Open TSM CLI and type the following command:

```
tsm configuration set -k wgserver.extended_trusted_ip_
checking -v true
```

2. Then type the following command:

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

Test Trusted Authentication

The steps below provide a method you can use to test retrieving a trusted ticket from your web server. This simple test can help evaluate connectivity between the web server and Tableau Server, and whether or not trusted authentication has been configured correctly.

Important: The test code provided in this topic runs client-side in the browser to provide a quick, visual validation that trusted authentication is configured correctly on Tableau Server. If the client browser loading the html page is not on the trusted web server, you may have to temporarily trust the client IP. In practice, you should never trust client IP addresses as part of your trusted authentication configuration. All trusted IP addresses are able to request tickets as any user including your administrator. In a

production environment, all trusted authentication flows should run server-side, between Tableau Server and the trusted web server(s) only.

Because the test is run on a client browser, the test isn't an exact replica of the communication path in a production environment. After successfully running the test, we recommend that you then test requesting tickets for your users with a server-side post request for final verification.

An alternative testing method is to run a trusted ticket generator to test your configuration. The following URL references a trusted ticket generator that is not supported by Tableau. However the generator has been used by many customers to test their trusted ticket configuration: <https://github.com/mkannan-tsi/Trusted-Ticket-Generator>.

Step 1: Add a test user

Create a user on the Tableau Server that you can use to test trusted ticket functionality. See [Add Users to Tableau Server](#). Add that user to a site on the server, and set the user's site role to **Explorer**.

Step 2: Create a test HTML page

Paste the following code into a new .html file that you save on the web server you specified in Step 1. You can change the labels and style attributes as you prefer.

```
<html>
<head>
<title>Trusted Ticket Requester</title>
<script type="text/javascript">
  function submitForm(){
    document.getElementById('form1').action =
    document.getElementById('server').value + "/trusted";
  }
</script>
```

```

<style type="text/css">
  .style1 {width: 100%;}
  .style2 {width: 429px;}
  #server {width: 254px;}
</style>
</head>
<body>
<h3>Trusted Ticketer</h3>
<form method="POST" id="form1" onSubmit="submitForm()">
  <table class="style1">
    <tr>
      <td class="style2">Username</td>
      <td><input type="text" name="username" value="" /></td>
    </tr>
    <tr>
      <td class="style2">Server</td>
      <td><input type="text" id="server" name="server" value="https://" /></td>
    </tr>
    <tr>
      <td class="style2">Client IP (optional)</td>
      <td><input type="text" id="client_ip" name="client_ip" value="" /></td>
    </tr>
    <tr>
      <td class="style2">Site (leave blank for Default site; otherwise enter the site name)</td>
      <td><input type="text" id="target_site" name="target_site" value="" /></td>
    </tr>
    <tr>

```

Tableau Server on Linux Administrator Guide

```
<td class="style2"><input type="submit" name="submittable"
value="Get Ticket" /></td>
<td>&#160;</td>
</tr>
</table>
</form>
<h4>Be sure to add your IP as a Trusted IP address to the server-
</h4>
</body>
</html>
```

Step 3: Retrieve a trusted ticket from Tableau Server

The following procedure will return a trusted ticket from Tableau Server.

1. Open the web page that you created in the previous step.

Trusted Ticketer

Username

Server

Client IP (optional)

Site (leave blank for Default site; otherwise enter the site name)

Be sure to add your IP as a Trusted IP address to the server

This operation requires JavaScript, so the web browser might prompt you to allow scripts to run.

2. In the text boxes, enter the following:
 - **Username:** The test user that was created in Step 1.
 - **Server:** the address of your Tableau Server, e.g., `https://server_name`.
 - **Client IP (optional):** The IP address of the user's computer, if it's configured for client trusted IP matching.
 - **Site:** The name of the Tableau Server site that the test user is a member of.

3. Click **Get Ticket**. One of the following will be returned:

- **A unique ticket:** A trusted ticket is a string composed of a base64-encoded UUID and a 24-character random string, for example,
9D101xmDQmSIOyQpKdy4Sw== : dg62gCsSE0QRARXNTOp6mlJ5.
- **-1:** If the value, -1 is returned, the configuration contains an error. See Ticket Value of -1 Returned from Tableau Server.

Step 4: Test access with trusted ticket

Now that you have a ticket, you can use it to access content on Tableau Server.

Construct a URL with the unique ticket that you generated in the previous step to verify access with the trusted ticket. The URL syntax is different if you are accessing a Tableau Server with a single site vs a server that hosts multiple sites.

Single (Default) site server url

```
https://<server-name>/trusted/<unique-ticket>/views/<workbook-name>/<view-name>
```

Non-default site server url

```
https://<server-name>/trusted/<unique-ticket>/t/<site-name>/views/<workbook-name>/<view-name>
```

Variables in the URLs are indicated by angle brackets (< and >). All other syntax is literal.

Troubleshoot Trusted Authentication

This section includes some common issues and errors you might encounter when you're configuring trusted authentication.

Trusted authentication information is written to `/var/opt/tableau/tableau_server/data/tabsvc/logs/vizqlserver/vizql-*.log`.

To increase the logging level from `info` to `debug`, run the following commands:

Tableau Server on Linux Administrator Guide

```
tsm configuration set -k vizqlserver.trustedticket.log_level -v
debug
tsm pending-changes apply
```

To test your trusted authentication deployment, see [Test Trusted Authentication](#).

Ticket Value of -1 Returned from Tableau Server

Tableau Server returns -1 for the ticket value if it cannot issue the ticket as part of the trusted authentication process. Before troubleshooting this scenario, be sure to set the log level for trusted authentication to `debug` as specified in [Troubleshoot Trusted Authentication](#).

The exact reason for this message is written to the `vizqlserver_node*-* .log.*` files in the following folder:

```
/var/opt/tableau/tableau_server/data/tabsvc/logs/vizqlserver
```

Here are some things to confirm:

- **All web server host names or IP addresses are added to trusted hosts**

The log error, "Invalid request host: <IP address>" may indicate that the IP address or host name for the computer sending the POST request is not in the list of trusted hosts on Tableau Server. See [Add Trusted IP Addresses or Host Names to Tableau Server](#) to learn how to add IP addresses or host names to this list.

- **IP addresses are IPv4**

If you are using IP addresses to specify trusted hosts, they must be in Internet Protocol version 4 (IPv4) format. An IPv4 address looks like this: 123.456.7.890. IPv6 addresses (for example, fe12::3c4a:5eab:6789:01c%34) are not supported as a way of inputting trusted hosts.

- **Username in POST request is a valid Tableau Server user**

The username you send in the POST request must be a licensed Tableau Server user. You can see a list of users by signing in to Tableau Server as an administrator.

The following log errors indicate a user POST issue:

- "Missing username and/or client_ip"
- "Invalid user: <username>"
- "Unlicensed user is not allowed: <username>"

Username in POST request includes domain

If Tableau Server is configured to use Local Authentication, the username that you send in the POST can be a simple string. However, if the server is configured for Active Directory you must include the domain name with the user name (domain\username). For example, the username parameter might be: `username=dev\jsmith`. A common error log for this scenario is "Invalid user: <username>."

- **Content-Type is specified**

If you are designing an ASP.NET or C# application, you need to declare the content type in your HTTP request. For example:

```
http.setRequestHeader("Content-Type", "application/x-www-form-urlencoded; charset=UTF-8")
```

If you do not specify content type and Tableau Server returns a -1, the log files contain the error: "missing username and/or client_ip".

HTTP 401 - Not Authorized

If you receive a 401- Not Authorized error, you may have configured Tableau Server to use Active Directory with SSPI. If your web server uses SSPI, you do not need to set up trusted authentication. You can embed views and your users will have access to them as long as they are licensed Tableau server users and members of your Active Directory.

See `tsm authentication sspi <commands>`.

Tableau Server on Linux Administrator Guide

If you see a 401 error (or a 302 - Redirect error) after you have deployed Tableau Server 2021.2, then it's likely the trusted ticket code you have written to construct the URL for the client has not been updated to account for the two-part ticket URL format.

See [Get a Ticket from Tableau Server](#).

HTTP 404 - File Not Found

You may receive this error if your program code references a Tableau Server URL that does not exist. For example, your web server may construct an invalid URL that cannot be found when the webpage tries to retrieve it.

Another cause for this error is if you did not enter the trusted web servers as specified in [Add Trusted IP Addresses or Host Names to Tableau Server](#). If you have already entered the trusted web servers, verify that the IP addresses or host names are correct.

Invalid User (SharePoint or C#)

You may encounter this error if you've configured Tableau Server for trusted authentication.

The example code for the SharePoint .dll references the following GET request:

```
SPContext.Current.Web.CurrentUser.Name
```

The above request will return the display name of the current Windows Active Directory user. If you want to use the login ID, then you will need to change the code to:

```
SPContext.Current.Web.CurrentUser.LoginName
```

After you make the change, recompile the SharePoint .dll.

Attempting to Retrieve the Ticket from the Wrong IP Address

You may encounter this error if you've configured Tableau Server for trusted authentication.

The client web browser IP address is not considered by default when redeeming the ticket. If Tableau Server is configured to enforce client IP address matching, make sure that the client's web browser IP address that is sent in the POST to Tableau Server is the same as when the browser tries to retrieve the embedded view. For example, in the [Trusted](#)

Authentication diagram, if the **POST request in step 3** sends the parameter `client_ip=74.125.19.147`, then the **GET request in step 5** must come from that same IP address.

See [Optional: Configure Client IP Matching](#) to learn how to configure Tableau Server to enforce client IP address matching.

Cookie Restriction Error

When a user signs in to Tableau Server, a session cookie is stored in their local browser. The stored cookie is how Tableau Server maintains that the signed in user has been authenticated and can access the server. Because the cookie is set with the same domain or sub-domain as the browser's address bar, it is considered a first-party cookie. If a user's browser is configured to block first-party cookies, they will be unable to sign in to Tableau Server.

When a user signs in to Tableau Server via an embedded view, or in an environment where trusted authentication has been configured, the same thing happens: a cookie is stored. In this case, however, the browser treats the cookie as a third-party cookie. This is because the cookie is set with a domain that's different from the one shown in the browser's address bar. If a user's web browser is set to block third-party cookies, authentication to Tableau Server will fail. To prevent this from occurring, web browsers must be configured to allow third-party cookies.

An error occurred communicating with the server (403)

If Tableau Server is configured for trusted authentication, you may receive this error after opening a new view in a browser and attempting to navigate back to views you'd opened earlier. Tableau Server provides protection against unauthorized reuse of VizQL sessions through the tsm configuration set `Optionsvizqlserver.protect_sessions`, which is set to `true` by default. Because Tableau Server is configured for trusted authentication, you may not also need to enable `vizqlserver.protect_sessions`. To disable it, use tsm configuration set Options to change it to `false`.

Personal Access Tokens

Personal access tokens (PATs) provide Tableau Server users the ability to create long-

lived authentication tokens. The tokens allow users to run automation with Tableau REST APIs without requiring hard-coded credentials or interactive login. More information about using personal access tokens with Tableau REST APIs is at [Signing In and Out \(Authentication\)](#).

Personal access tokens are not used for generic client access to the Tableau Server web interface, TSM, or tabcmd interfaces.

We recommend creating personal access tokens for automated scripts and tasks that are created with Tableau REST API:

- **Improve security:** Personal access tokens reduce risk in the event credentials are compromised. In the case where Tableau Server uses Active Directory or LDAP as an identity store, you can reduce the scope of credential compromise by using a personal access token for automated tasks. In this case, using an application-specific token doesn't expose the broader system in the event that automation or script files are compromised. If a token gets compromised or is used in automation that is failing or posing a risk, you can just revoke the token. You do not need to rotate or revoke the user's credentials.
- **Audit and track:** As an administrator, you can review Tableau Server logs to track when a token is used, what sessions are created from that token, and the actions that are performed in those sessions. You can also determine if a session and the related tasks were performed from a session that was generated from a token or from an interactive login.
- **Manage automation:** A token can be created for each script or task that is run. This allows you to silo and review automation tasks across your organization. Additionally, by using tokens then password resets or metadata changes (username, email, etc.) on user accounts will not disrupt automation as it would when credentials are hard-coded into the scripts.

Understand personal access tokens

When a token is created, it is hashed then stored in the repository. After the token is hashed and stored, the original token is deleted. Users are instructed to copy the token to a safe place and to handle it as they would a password. When the token is used at run-time,

Tableau Server hashes the token presented by the user and compares it to the hashed value stored in the repository. If a match is made, then an authenticated session is started.

In the context of authorization, Tableau Server handles the authenticated session with same permissions and rights that the user has as an interactive user.

Server administrator impersonation

Beginning with version 2021.1, you can enable Tableau Server personal access token impersonation. In this scenario, access tokens that are created by server administrators can be used for [user impersonation](#) when using the Tableau Server REST API. Impersonation is useful in scenarios where you are embedding end-user-specific Tableau content within your application. Specifically, impersonation access tokens allow you to build applications that query as a given user and retrieve content that the user is authorized for within Tableau Server, without hard-coding any credentials.

For more information, see the Tableau REST API Help section, [Impersonating a User](#).

Enable Tableau Server to accept personal access tokens during impersonation sign-in requests

By default, Tableau Server does not allow impersonation for server administrator personal access tokens. You must enable the server-wide setting by running the following commands.

```
tsm authentication pat-impersonation enable [global options]
tsm pending-changes apply
```

Important: After you have run the commands, all personal access tokens created by server administrators (including preexisting tokens) can be used for impersonation. To bulk-revoke all existing server administrator personal access tokens, you can post the

DELETE /api/{api-version}/auth/serverAdminAccessTokens URI. See the Tableau REST API Help section, [Impersonating a User](#).

Create tokens

Users with accounts on Tableau Server can create, manage, and revoke personal access tokens on the **My Account Settings** page. See [Manage Your Account Settings](#) in the *Tableau Desktop and Web Authoring Help* for more information.

Users must create their own personal access tokens. Administrators cannot create tokens for users.

Token expiry

Personal access tokens will expire if they are not used after 15 consecutive days. If they are used more frequently than every 15 days, an access token will expire after 1 year. After a year, you must create a new token. Expired personal access tokens will not display on the **My Account Settings** page.

You can change refresh token expiry time span using the `refresh_token.absolute_expiry_in_seconds` option with the `tsm configuration set` command.

Revoke users' tokens

Users are able to revoke their own tokens on the **My Account Settings** page. As an administrator, you can also revoke personal access tokens.

1. Sign in to the Tableau Server Admin Area as a site administrator or server administrator.
2. Locate the user whose token you want to revoke. For more information about navigating Server Admin pages and locating users, see [View, Manage, or Remove Users](#).
3. Click the user's name to open their profile page.
4. On the user's profile page, click the **Settings** tab.
5. In the **Personal Access Tokens** section, identify the token that you want to revoke

and then click **Revoke**.

6. On the verification pop-up, click **Delete**.

Track and monitor usage

All token-related actions are logged in the Tableau Server Application Server (vizportal) service.

To locate token-related activities, filter log entries containing the string, `RefreshTokenService`.

Tokens are stored in this format: `:Token Guid: <TokenID(Guid)>`, where the `TokenID` is a base64 encoded string. The token secret is not included in the logs. For example:

```
Token Guid: 49P+CxmARY6A2GHxyvHHAA== (e3d3fe0b-1980-458e-80d8-61f1caf1c700).
```

The following is an example snippet of two log entries. The first shows how a user is mapped to a token. The second shows a refresh event for the same token:

```
RefreshTokenService - Issued refresh token to the following user: jsmith. Token Guid: 49P+CxmARY6A2GHxyvHHAA== (e3d3fe0b-1980-458e-80d8-61f1caf1c700)
RefreshTokenService - Redeemed refresh token. Token Guid: 49P+CxmARY6A2GHxyvHHAA== (e3d3fe0b-1980-458e-80d8-61f1caf1c700)
```

To locate key operations, filter log entries containing the string, `OAuthController`.

Data Connection Authentication

You can configure data connection authentication using Kerberos, OAuth, and single sign-on.

Tableau Server on Linux Administrator Guide

More information

- [Tableau Server on Linux - Connecting to a Windows Shared Directory](#) (Tableau Community)
- [Setting an Oracle Connection to Use TNSNames.ora or LDAP.ora](#) (Tableau Support)

Enable Kerberos Delegation

Kerberos delegation enables Tableau Server to use the Kerberos credentials of the viewer of a workbook or view to execute a query on behalf of the viewer. This is useful in the following situations:

- You need to know who is accessing the data (the viewer's name will appear in the access logs for the data source).
- Your data source has row-level security, where different users have access to different rows.

Supported data sources

Tableau supports Kerberos delegation with the following data sources:

- Cloudera: Hive/Impala
- Denodo
- Hortonworks
- Oracle
- PostgreSQL
- Spark
- SQL Server
- Teradata
- Vertica

MSAS is not supported on Linux platforms.

Requirements

Kerberos delegation requires Active Directory.

- The Tableau Server information store must be configured to use LDAP - Active Directory.
- The computer where Tableau Server is installed must be joined to Active Directory domain.
- MIT KDC is not supported.

Web authoring

There are two web authoring scenarios that do not support Kerberos delegation: "Connect to data on the web" and "Create datasource on the web." feature does not support delegation yet. Specifically, if you create a datasource that uses Kerberos on the with web authoring, the data source will use Run As service account authentication. If you want to use Kerberos delegation to create a datasource, then you must publish with Tableau Desktop. For more information on Run As service account, see [Enable Kerberos Service Account Access](#).

Configuration process

This section provides an example of the process to enable Kerberos delegation. The scenario also includes example names to help describe the relationships between the configuration elements.

1. Tableau Server will need a Kerberos service ticket to delegate on behalf of the user that is initiating the call to the database. You must create a domain account that will be used to delegate to the given database. This account is referred to as the Run As service account. In this topic, the example user configured as the delegation/Run As account is `tabsrv@example.com`.

The account must be configured with Active Directory User and Computers on a Windows Server that is connected to the user domain:

- Open the **Properties** page for the Run As service account, click the **Delegation** tab and select **Trust this user for delegation to specified services only** and **Use any authentication protocol**.
2. Create a keytab file for the Run As service account.

Tableau Server on Linux Administrator Guide

For example, the following commands create a keytab (`tabsrv-runas.keytab`) using the `ktutil` tool:

```
sudo ktutil
```

```
ktutil: addent -password -p tabsrv@EXAMPLE.COM -k 2 -e  
<encryption scheme>
```

Encryption schemes for this command include `RC4-HMAC`, `aes128-cts-hmac-sha1-96`, and `aes256-cts-hmac-sha1-96`. Consult your IT team for the correct encryption scheme for your environment and data source.

```
ktutil: wkt tabsrv-runas.keytab
```

Tableau Server will use the Run As service account and the associated keytab to authenticate and make a direct connection to the database.

3. Copy the keytab into the Tableau Server data directory and set proper ownership and permissions.

```
mkdir /var/opt/keytab
```

```
sudo cp -p tabsrv-runas.keytab /var/opt/keytab
```

```
sudo chown $USER /var/opt/keytab/tabsrv-runas.keytab
```

```
chgrp tableau /var/opt/keytab/tabsrv-runas.keytab
```

```
chmod g+r /var/opt/keytab/tabsrv-runas.keytab
```

4. Run the following TSM commands to enable Kerberos delegation, set the delegation service account, and associate the keytab file with the service account:

```
tsm configuration set -k wgserver.delegation.enabled -v  
true
```

```

tsm configuration set -k native_api.datasource_impersonation_runas_principal -v tabsrv@EXAMPLE.COM
tsm configuration set -k native_api.datasource_impersonation_runas_keytab_path -v /var/opt/keytab/tabsrv-runas.keytab
tsm configuration set -k native_api.protocol_transition_ad_short_domain -v false
tsm configuration set -k native_api.protocol_transition_uppercase_realm -v true

```

In some cases, TSM may return an error mentioning `--force-keys`. If you get this error, run the command again with the `--force-keys` parameter appended to the argument.

5. Run the following TSM command apply the changes to Tableau Server:

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

6. Enable delegation for data connections:

- **SQL Server**—See [Enabling Kerberos Delegation for SQL Server](#) in the Tableau Community.
- **PostgreSQL**—See [Enabling Kerberos Delegation for PostgreSQL](#) in the Tableau Community.
- **Teradata**—See [Enabling Kerberos Delegation for Teradata](#) in the Tableau

Community.

- **Denodo**—See [Enabling Kerberos Delegation for Denodo on Linux](#) in the Tableau Community.
- **Oracle**—See [Enable Kerberos Delegation for Oracle](#) in the Tableau Community.
- **Cloudera**—See [Enable Kerberos Delegation for Hive/Impala](#) in the Tableau Community.
- **Vertica**—See [Enabling Kerberos Delegation for Vertica](#) in the Tableau Community.

See also

Troubleshoot Kerberos

Enable Kerberos Delegation for JDBC Connectors

As of version 2020.2, Tableau Server supports Kerberos delegation for JDBC connectors.

Kerberos delegation enables Tableau Server to use the Kerberos credentials of the viewer of a workbook or view to execute a query on behalf of the viewer. This is useful in the following situations:

- You need to know who is accessing the data (the viewer's name will appear in the access logs for the data source).
- Your data source has row-level security, where different users have access to different rows.

Supported data sources

Tableau supports JDBC Kerberos RunAs authentication with the following data sources:

- Oracle
- PostgreSQL

Both native and JDBC-based connectors use the same configuration on Tableau Server on Linux. See [Enable Kerberos Delegation](#).

Enable Kerberos Run As Authentication for JDBC Connectors

As of version 2020.2, Tableau Server supports Kerberos authentication for JDBC connectors.

You can configure Tableau Server to use a Kerberos service account to access a database. In this scenario, Tableau Server connects to databases with a service account, also referred to as a "Run As service account". This scenario is referred to as "Run As authentication"

To use Run As authentication on Tableau Server you must first create a workbook or data-source in Tableau Desktop that uses integrated authentication. When you publish to Tableau Server you will get the option to use Run As authentication. When creating a data-source with Web Authoring, Run As authentication is the default operation if you select integrated authentication.

Supported data sources

Tableau supports JDBC Kerberos delegation with the following data source:

- Oracle

Both native and JDBC-based connectors use the same configuration on Tableau Server on Linux. To configure Run As authentication see [Enable Kerberos Service Account Access](#).

OAuth Connections

Tableau Server supports OAuth for a number of different connectors. In many cases, OAuth functionality does not require additional configuration on Tableau Server.

From Tableau, when users sign in to data with a connector that uses OAuth, users are redirected to the authentication provider's sign-in page. After user provide their credentials and authorize Tableau to access their data, the authentication provider sends Tableau an **access token** that uniquely identifies Tableau and the users. This access token is used to

access data on users' behalf. For more information, see [Overview of the OAuth process](#) below.

Using OAuth-based connections provides the following benefits:

- **Security:** Your database credentials are never known to or stored in Tableau Server, and the access token can be used only by Tableau on behalf of users.
- **Convenience:** Instead of having to embed your data source ID and password in multiple places, you can use the token provided for a particular data provider for all published workbooks and data sources that access that data provider.

In addition, for live connections to Google BigQuery data, each workbook viewer can have a unique access token that identifies the user, rather than sharing a single user name and password credential.

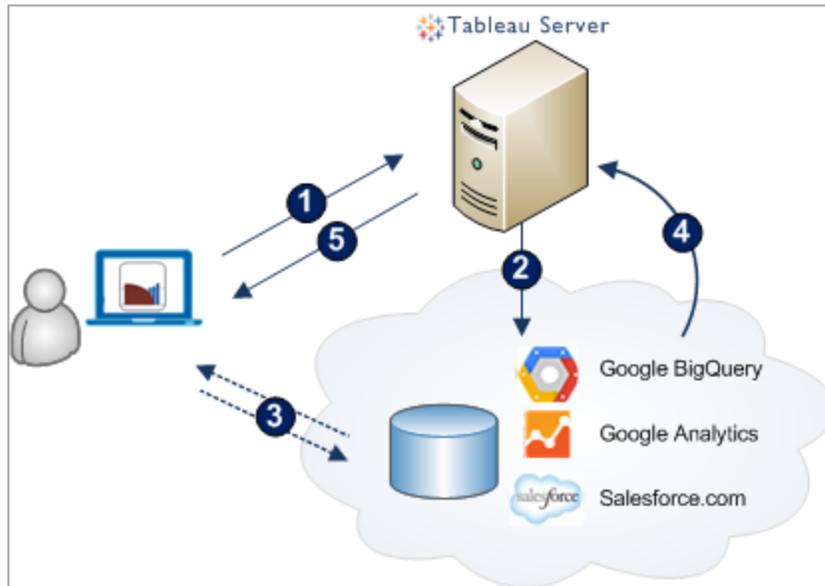
Overview of the OAuth process

The following steps describe a workflow in the Tableau environment that calls the OAuth process.

1. A user takes an action that requires access to a cloud-based data source.

For example, you open a workbook that's published to Tableau Server.
2. Tableau directs the user to the cloud data provider's sign-in page. The information that is sent to the data provider identifies Tableau as the requesting site.
3. When the user signs in to the data, the provider prompts the user to confirm their authorization for Tableau Server to access the data.
4. Upon the user's confirmation, the data provider sends an access token back to Tableau Server.

5. Tableau Server presents the workbook and data to the user.



The following user workflows can use the OAuth process:

- Creating a workbook and connecting to the data source from Tableau Desktop or from Tableau Server.
- Publishing a data source from Tableau Desktop.
- Signing in to Tableau Server from an approved *client*, such as Tableau Mobile or Tableau Desktop.

Default saved credential connectors

Saved credentials refers to the functionality where Tableau Server stores user tokens for OAuth connections. This allows users to save their OAuth credentials to their user profile on Tableau Server. After they have saved the credentials, they will not be prompted with them subsequently publish, edit, or refresh when accessing the connector.

The following connectors use saved credentials by default and do not require additional configuration on Tableau Server.

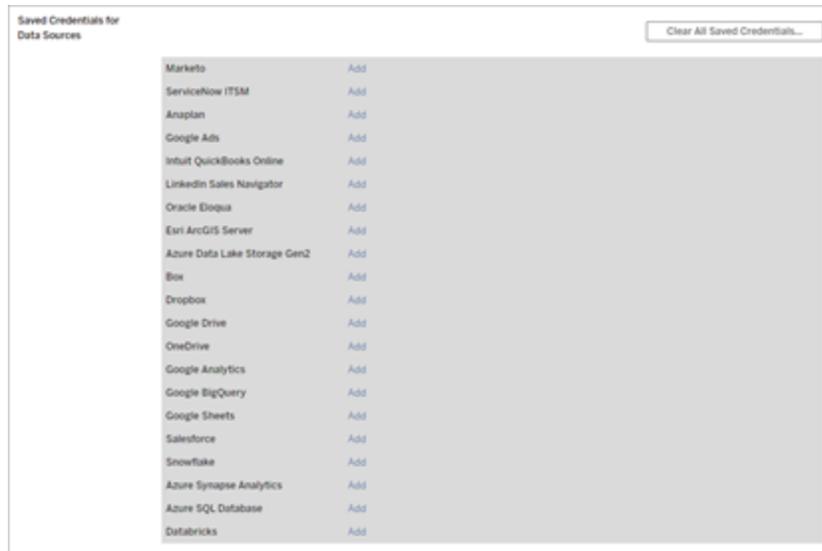
Tableau Server on Linux Administrator Guide

- Anaplan
- Box
- Dropbox
- Esri ArcGIS Server
- Google Ads, Google Drive
- LinkedIn Sales Navigator
- Marketo
- OneDrive
- Oracle Eloqua
- ServiceNow ITSM
- Snowflake - To use "private link" requires additional configuration. For more information, see [Configure Snowflake OAuth for Partner Applications](#) on the Snowflake website and [Change Snowflake OAuth to Private Link with Saved Credentials](#).

The following connectors can use saved credentials with additional configuration by the server administrator.

- Azure Data Lake Storage Gen2, Azure Synapse, Azure SQL Database, Databricks
For more information, see [Configure Azure AD for OAuth and Modern Authentication](#).
- Google Analytics, Google BigQuery, Google Sheets
For more information, see [Change Google OAuth to Saved Credentials](#).
- Intuit QuickBooks Online
For more information, see [Set up OAuth for Intuit QuickBooks Online](#).
- Salesforce
For more information, see [Change Salesforce.com OAuth to Saved Credentials](#).

All supported connectors are listed under **Saved Credentials for Data Sources** on users' **My Account Settings** page on Tableau Server. Users manage their saved credentials for each connector.



Access tokens for data connections

You can embed credentials based on access tokens with data connections, to enable direct access after the initial authentication process. An access token is valid until a Tableau Server user deletes it, or the data provider revokes it.

It is possible to exceed the number of access tokens your data source provider allows. If that's the case, when a user creates a new token, the data provider uses length of time since last access to decide which token to invalidate to make room for the new one.

Access tokens for authentication from approved clients

By default, Tableau Server allows users to access their sites directly from approved Tableau clients, after users provide their credentials the first time they sign in. This type of authentication also uses OAuth access tokens to store the users' credentials securely.

For more information, see [Disable Automatic Client Authentication](#).

Default managed keychain connectors

Managed keychain refers to the functionality where OAuth tokens are generated for Tableau Server by the provider and shared by all users in the same site. When a user first publishes a data source, Tableau Server prompts the user for the data source credentials. Tableau Server submits the credentials to the data source provider which returns OAuth tokens for Tableau Server to use on behalf of the user. On subsequent publishing operations, the OAuth token stored by Tableau Server for the same class and user name is used so that the user is not prompted for the OAuth credentials. Should the data source password change, then the above process is repeated and the old token is replaced by a new token on Tableau Server.

Additional OAuth configuration on Tableau Server is not required for the default managed keychain connectors:

- Google Analytics, Google BigQuery, and Google Sheets
- Salesforce

Token limit and storage

Google has 50 token limit per user per client application (in this scenario, Tableau Server is the client application). Because the OAuth token is stored on Tableau Server and reused by the user, the user is unlikely to exceed the token limit.

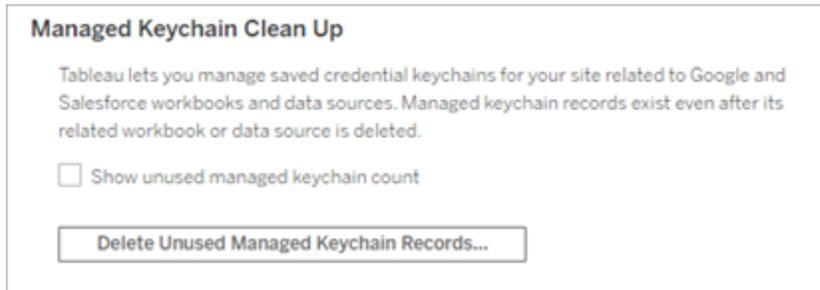
All user tokens are encrypted at rest when stored on Tableau Server. See [Manage Server Secrets](#) for more information.

Removing unused keychain records

A managed keychain record contains connection attributes like dbClass, username, and OAuth secret attributes. All managed keychain records for a given site are merged, encrypted, and stored in PostgreSQL.

Records are persisted even for workbooks and data sources that have been removed. Over time, these records can grow to large sizes which may cause issues.

We recommend purging the unused keychain records periodically as a regular maintenance task. You can view the number of records and unused records stored on each site. You can also delete unused records.



To access **Managed Keychain Clean Up**, sign in to the Tableau Server admin pages, navigate to the site where you want to delete unused records, and click **Settings**.

Scenario limitations with managed keychain

Three scenarios are not supported when using managed keychain OAuth with Tableau Server:

- Prompting for OAuth credentials on live connections. Users must embed credentials on live connections with managed-keychain OAuth.
- Editing the OAuth data source connection on Tableau Server
- Web authoring

Convert managed keychain to saved credentials

You can convert the connectors that use managed keychain to use saved credentials by configuring Tableau Server with an OAuth client ID and secret for each connector. By converting these connectors to saved credentials, users will be able to manage their credentials

for each connector type on **My Account Settings** page on Tableau Server. Additionally, live connection prompts, editing connections, and web authoring are also supported.

Configure custom OAuth for a site

For a subset of connectors, you can configure site-level OAuth by configuring custom OAuth clients. For more information, see one of the following topics:

- For Azure Data Lake Storage Gen2, Azure SQL Database, Azure Synapse, and Databricks, see [Configure custom OAuth for a site](#).
- For Google Analytics, Google BigQuery, Google Sheets, see [Configure custom OAuth for a site](#).
- For Salesforce, see [Configure custom OAuth for a site](#).
- For Snowflake, see [Configure custom OAuth for a site](#).

Change Salesforce.com OAuth to Saved Credentials

By default, the Salesforce.com connector uses a managed keychain for OAuth tokens that are generated for Tableau Server by the provider and shared by all users in the same site. You can configure Tableau Server with saved client ID and client secret. There are two scenarios where you might want to do this:

- **Salesforce data connector**—If you're using the Salesforce data connector, you can configure Tableau Server with an OAuth client ID and secret, so the connector can use saved credentials.
- **Einstein Discovery**—If you are integrating Einstein Discovery extensions with Tableau Server, you need to do this OAuth client ID and secret configuration. The ability to integrate Einstein Discovery and Tableau Server was added in version 2021.1.0. For more information, see [Configure Einstein Discovery Integration](#).

For more information about managed keychain and saved credentials, see [OAuth Connections](#).

This topic describes how to set up your Salesforce.com data sources and Einstein Discovery extensions for OAuth saved credentials. Complete these steps for each Tableau Server instance.

Set up OAuth by following these two procedures:

- Create a Connected App in Salesforce.
- Use the information you obtained to configure your server.

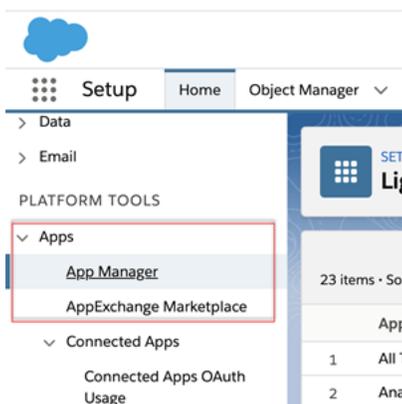
Create a Connected Salesforce App

Note: This procedure documents the process in Salesforce Lightning. If you are using the traditional interface, the navigation may be different but the configuration is the same.

1. Sign in to your Salesforce.com developer account, click your user name in the upper-right, and then select **Setup**.

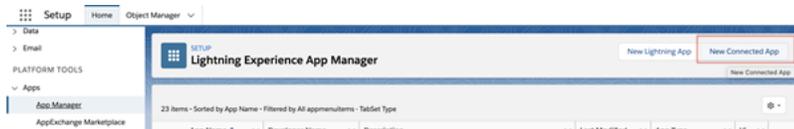


2. In the left navigation column, under **Apps**, select **App Manager**.



3. In the Connected Apps section, click **New Connected App**.

Tableau Server on Linux Administrator Guide



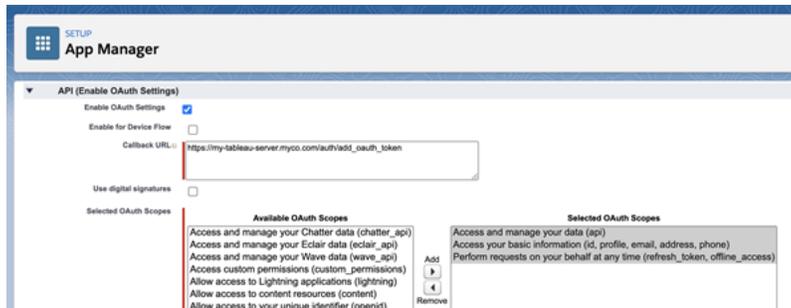
4. In **Basic Information**, give the app a name, tab through the api field so it will self-populate in the correct format, and enter a contact email for the app.
5. In the **API [Enable OAuth Settings]** section, select **Enable OAuth Settings**.
6. In the new OAuth settings that appear, for **Callback URL**, type the fully qualified domain name of your server, using the https protocol, and append the following text to the URL: `auth/add_oauth_token`.

For example:

```
https://www.your_tableau_server.com/auth/add_oauth_token
```

7. Move the following items from **Available OAuth Scopes** to **Selected OAuth Scopes**:

- **Access and manage your data (api)**
- **Access your basic information (id)**
- **Perform requests on your behalf at any time (refresh_token)**



8. Click **Save**.

After you save the app, Salesforce populates the API section with the following IDs that you will use to configure Tableau Server:

- **Consumer Key**
- **Consumer Secret**
- **Callback URL**



Configure Tableau Server for Salesforce.com OAuth

Once your connected app is created in Salesforce and you have the Customer Key, Customer Secret, and the Callback URL, you can configure Tableau Server for Salesforce data connections and Einstein Discovery.

1. On the Tableau Server computer, at a command prompt, run the following commands:

```
tsm configuration set -k oauth.salesforce.client_id -v
<your_customer_key>
```

```
tsm configuration set -k oauth.salesforce.client_secret -v
<your_customer_secret>
```

```
tsm configuration set -k oauth.salesforce.redirect_uri -v
<your_redirect_URL>
```

2. (Optional) To change the default login server, type the following command:

```
tsm configuration set -k oauth.salesforce.server_base_url
-v <URL>
```

3. Enter the following command to apply changes:

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

Configure custom OAuth for a site

You can configure a custom Salesforce OAuth client for a site.

Consider configuring a custom OAuth client to 1) override an OAuth client if configured for the server or 2) enable support for securely connecting to data that requires unique OAuth clients.

When a custom OAuth client is configured, the site-level configuration takes precedence over any server-side configuration and all new OAuth credentials created use the site-level OAuth client by default. No Tableau Server restart is required for the configurations to take effect.

Important: Existing OAuth credentials established before the custom OAuth client is configured are temporarily usable but both server administrators and users must update their saved credentials to help ensure uninterrupted data access.

Step 1: Prepare the OAuth client ID, client secret, and redirect URL

Before you can configure the custom OAuth client, you need the information listed below. After you have this information prepared, you can register the custom OAuth client for the site.

- **OAuth client ID and client secret:** First register the OAuth client with the data provider (connector) to retrieve the client ID and secret generated for Tableau Server.
- **Redirect URL:** Note the correct redirect URL. You will need this during the registration process in **Step 2:** below.

`https://<your_server_name>.com/auth/add_oauth_token`

For example, `https://myco.com/auth/add_oauth_token`

Step 2: Register the OAuth client ID and client secret

Follow the procedure described below to register the custom OAuth client to the site.

1. Sign in to your Tableau Server site using your admin credentials and navigate to the **Settings** page.
2. Under OAuth Clients Registry, click the **Add OAuth Client** button.
3. Enter the required information, including the information from **Step 1** above:
 - a. For **Connection Type**, enter one of the following values depending on the connector whose custom OAuth client you want to configure:
 - `azuredatalakestorageegen2`
 - `azure_sqldb`
 - `azure_sql_dw`
 - `bigquery`
 - `databricks`
 - `google-analytics`
 - `google-sheets`
 - `salesforce`
 - b. For **Client ID**, **Client Secret**, and **Redirect URL**, enter the information you prepared in **Step 1** above.
 - c. Click the **Add OAuth Client** button to complete the registration process.

Create OAuth Client

Connection Type
⊗ Enter a valid connection type.

OAuth Instance URL

Client ID
⊗ Enter the client ID from your OAuth provider.

Client Secret
⊗ Enter the client secret from your OAuth provider.

Redirect URL
⊗ Enter a valid URL ending with "/auth/add_oauth_token". Domain name can only be "localhost" when using HTTP schema.

4. (Optional) Repeat step 3 for all supported connectors.
5. Click the **Save** button at the bottom or top of the Settings page to save changes.

Step 3: Validate and update saved credentials

To help ensure uninterrupted data access, you (and your site users) must delete the previous saved credentials and add it again to use the custom OAuth client for the site.

1. Navigate to your **My Account Settings** page.
2. Under **Saved Credentials for Data Sources**, do the following:
 1. Click **Delete** next to the existing saved credentials for the connector whose custom OAuth client you configured in **Step 2** above.
 2. Next to connector name, click **Add** and follow the prompts to 1) connect to the

custom OAuth client configured in **Step 2** above and 2) save the latest credentials.

Step 4: Notify users to update their saved credentials

Make sure you notify your site users to update their saved credentials for the connector whose custom OAuth client you configured in **Step 2** above. Site users can use the procedure described in Update saved credentials to update their saved credentials.

Manage access tokens

After you configure the server for OAuth, you can allow users to manage their own access tokens in their profile settings, or you can manage the tokens centrally. For more information, see Allow Saved Access Tokens.

Change Google OAuth to Saved Credentials

By default, the Google Analytics, Google BigQuery, and Google Sheets connectors use a managed keychain for OAuth tokens that are generated for Tableau Server by the provider and shared by all users in the same site.

You can convert the connectors that use managed keychain to use saved credentials by configuring Tableau Server with an OAuth client ID and secret for each connector.

For more information about managed keychain and saved credentials, see OAuth Connections.

This topic describes how to set up your Google BigQuery, Google Sheets, and Google Analytics data sources for OAuth with saved credentials.

Complete these steps for each Tableau Server instance.

Set up OAuth by following these three procedures:

- Enable API access and create access token from Google.
- Configure OAuth on Tableau Server.

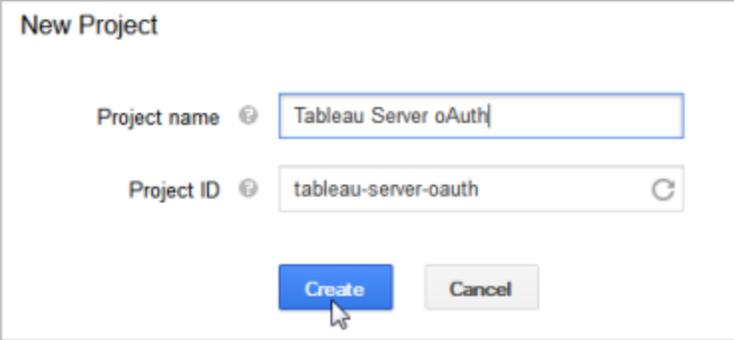
- Create and edit Google data source.

Obtain a Client ID and Enable Google APIs

Note These steps reflect the settings in the Google Cloud Platform console at the time of this writing. For more information, see [Using OAuth 2.0 for Web Server Applications](#) in the Google Developers Console Help.

1. Sign in to [Google Cloud Platform](#), and then click **Go to my console**.
2. On the drop-down menu, **Select a Project**, select **Create project**.
3. In the new project form that appears, complete the following:
 - Give the project a meaningful name that reflects the Tableau Server instance for which you'll use this project.
 - Determine whether you want to change the project ID.

Note After you create the project, you will not be able to change the project ID. For information, click the question mark icons.



The screenshot shows a 'New Project' form with the following details:

- Project name:** Tableau Server OAuth
- Project ID:** tableau-server-oauth
- Buttons:** Create (active), Cancel

4. Open the new project, navigate to **APIs & Services > OAuth consent screen**, and

select the User Type.

5. Click the **OAuth consent screen** tab and then enter a meaningful name for **Product name** shown to users.
6. Click **Credentials** and click the **Create Credentials** tab, then click **OAuth client ID**.
7. On the **Create OAuth client ID** screen, fill out the required fields. Follow the steps to authorize your OAuth tokens:
 - Select **Web Application**.
 - Enter a client **Name**.
 - For **Authorized JavaScript Origins**, click **ADD URI** and enter the local computer name of your Tableau Server.
 - For **Authorized redirect URIs**, click **ADD URI** and replace the example text with the Internet address for your Tableau Server, and add the following text to the end of it: **auth/add_oauth_token**. For example:

```
https://your_server_url.com/auth/add_oauth_token
```
8. Copy the Authorized Redirect URI, and paste it in a location that you can access from your Tableau Server computer.
9. Click **Create** .
10. Copy the following values that Google returns, and paste them in a location that you can access from your Tableau Server computer:
 - Client ID
 - Client secret
11. In **APIs Manager > Dashboards**, verify that **BigQuery API**, **Google Drive API** (to enable Google Sheets) or **Analytics API** is enabled. To enable APIs, click

ENABLE API at the top of the page.

Configure Tableau Server for Google OAuth

Using the information you obtained by completing the steps in Obtain a Client ID and Enable Google APIs, configure your Tableau Server:

- On the Tableau Server computer, open the shell and run the following commands to specify the access token and URI:

```
tsm configuration set -k oauth.google.client_id -v <your_client_ID>
```

```
tsm configuration set -k oauth.google.client_secret -v <your_client_secret>
```

```
tsm configuration set -k oauth.google.redirect_uri -v <your_authorized_redirect_URI>
```

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

Configure custom OAuth for a site

You can configure a custom Google OAuth client for a site.

Consider configuring a custom OAuth client to 1) override an OAuth client if configured for the server or 2) enable support for securely connecting to data that requires unique OAuth

clients.

When a custom OAuth client is configured, the site-level configuration takes precedence over any server-side configuration and all new OAuth credentials created use the site-level OAuth client by default. No Tableau Server restart is required for the configurations to take effect.

Important: Existing OAuth credentials established before the custom OAuth client is configured are temporarily usable but both server administrators and users must update their saved credentials to help ensure uninterrupted data access.

Step 1: Prepare the OAuth client ID, client secret, and redirect URL

Before you can configure the custom OAuth client, you need the information listed below. After you have this information prepared, you can register the custom OAuth client for the site.

- **OAuth client ID and client secret:** First register the OAuth client with the data provider (connector) to retrieve the client ID and secret generated for Tableau Server.
- **Redirect URL:** Note the correct redirect URL. You will need this during the registration process in **Step 2:** below.

`https://<your_server_name>.com/auth/add_oauth_token`

For example, `https://myco.com/auth/add_oauth_token`

Step 2: Register the OAuth client ID and client secret

Follow the procedure described below to register the custom OAuth client to the site.

1. Sign in to your Tableau Server site using your admin credentials and navigate to the **Settings** page.

2. Under OAuth Clients Registry, click the **Add OAuth Client** button.
3. Enter the required information, including the information from **Step 1** above:
 - a. For **Connection Type**, enter one of the following values depending on the connector whose custom OAuth client you want to configure:
 - azuredatalakestorageegen2
 - azure_sqldb
 - azure_sql_dw
 - bigquery
 - databricks
 - google-analytics
 - google-sheets
 - salesforce
 - b. For **Client ID**, **Client Secret**, and **Redirect URL**, enter the information you prepared in **Step 1** above.
 - c. Click the **Add OAuth Client** button to complete the registration process.

Create OAuth Client

Connection Type
❌ Enter a valid connection type.

OAuth Instance URL

Client ID
❌ Enter the client ID from your OAuth provider.

Client Secret
❌ Enter the client secret from your OAuth provider.

Redirect URL
❌ Enter a valid URL ending with "/auth/add_oauth_token". Domain name can only be "localhost" when using HTTP schema.

4. (Optional) Repeat step 3 for all supported connectors.
5. Click the **Save** button at the bottom or top of the Settings page to save changes.

Step 3: Validate and update saved credentials

To help ensure uninterrupted data access, you (and your site users) must delete the previous saved credentials and add it again to use the custom OAuth client for the site.

1. Navigate to your **My Account Settings** page.
2. Under **Saved Credentials for Data Sources**, do the following:
 1. Click **Delete** next to the existing saved credentials for the connector whose custom OAuth client you configured in **Step 2** above.
 2. Next to connector name, click **Add** and follow the prompts to 1) connect to the custom OAuth client configured in **Step 2** above and 2) save the latest credentials.

Step 4: Notify users to update their saved credentials

Make sure you notify your site users to update their saved credentials for the connector whose custom OAuth client you configured in **Step 2** above. Site users can use the procedure described in Update saved credentials to update their saved credentials.

Create and edit Google data source

Next, you must publish the Google data sources to the server. See the Tableau Desktop topic, [Google BigQuery](#).

After you've published the data sources, the final step is to edit the data source connection to use the embedded access token that you configured earlier. See Edit Connections on Tableau Server.

Managing access tokens

After you configure the server for OAuth, you can allow users to manage their own access tokens in their profile settings, or you can manage the tokens centrally. For more information, see [Allow Saved Access Tokens](#).

[Configure Azure AD for OAuth and Modern Authentication](#)

Starting in Tableau 2021.1, the Azure Synapse, Azure SQL Database, Azure Databricks, and Azure Data Lake Gen2 connectors support authentication through Azure AD by configuring an OAuth client for Tableau Server.

Note: OAuth support for Azure AD is only supported with [Microsoft SQL Server driver 17.3](#) and later.

Step 1. Register OAuth application for Azure

See the Tableau Community post, [Azure Application Registration for On-Prem Server OAuth](#).

Step 2: Configure Tableau Server for Azure

Configuring Tableau Server requires running a TSM command. Azure Data Lake Storage Gen2 requires a different set of commands than the common command that is run for Azure Synapse, Azure SQL Database, or Databricks.

Configure default OAuth client for Azure Data Lake Storage Gen2

To configure Tableau Server for Data Lake Storage Gen2, you must have the following configuration parameters:

- Azure OAuth client ID: this is generated from the procedure in Step 1. Copy this value for `[your_client_id]` in the first tsm command below.
- Azure OAuthClient secret: this is generated from the procedure in Step 1. Copy this value for `[your_client_secret]` in the second tsm command below.
- The Tableau Server url, such as `https://myserver.com`. Copy this value for `[your_server_url]` in the third tsm command below.

Run the following tsm commands to configure Tableau Server OAuth for Azure Data Lake Storage Gen2:

- `tsm configuration set -k oauth.azuredatalake_storage_gen2.client_id -v [your_client_id] --force-keys`
- `tsm configuration set -k oauth.azuredatalake_storage_gen2.client_secret -v [your_client_secret] --force-keys`
- `tsm configuration set -k oauth.azuredatalake_storage_gen2.redirect_uri -v http://[your_server_url]/auth/add_oauth_token --force-keys`
- `tsm pending-changes apply`

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

Configure default client for Azure Synapse, Azure SQL Database, or Databricks

To configure Tableau Server, you must have the following configuration parameters:

- Azure OAuth client ID: this is generated from the procedure in Step 1. Copy this value for `[your_client_id]` in the tsm command that follows.
- Azure OAuthClient secret: this is generated from the procedure in Step 1. Copy this value for `[your_client_secret]` in the tsm command that follows.

Tableau Server on Linux Administrator Guide

- The Tableau Server url, such as `https://myserver.com`. Copy this value for `[your_server_url]` in the `tsm` command that follows.
- Configuration ID: this is the value for the `oauth.config.id` parameter in the `tsm` command that follows. Valid values:
 - Azure Synapse: `azure_sql_dw`
 - Azure SQL Database: `azure_sqldb`
 - Databricks: `databricks`

Run the following `tsm` commands to configure Azure AD for Azure Synapse, Azure SQL Database, or Databricks:

- ```
tsm configuration set -k oauth.config.clients -v "[{\\"oauth.config.id\\":\\"azure_sql_dw\\", \\"oauth.config.client_id\\":\\"[your_client_id]\\", \\"oauth.config.client_secret\\":\\"[your_client_secret]\\", \\"oauth.config.redirect_uri\\":\\"[your_server_url]/auth/add_oauth_token\\"}]" --force-keys
```
- ```
tsm pending-changes apply
```

Setting multiple connectors

If you have multiple connectors to set, you must include all of them in a single command:

- ```
tsm configuration set -k oauth.config.clients -v "[{\\"oauth.config.id\\":\\"azure_sql_dw\\", \\"oauth.config.client_id\\":\\"[your_client_id]\\", \\"oauth.config.client_secret\\":\\"[your_client_secret]\\", \\"oauth.config.redirect_uri\\":\\"[your_server_url]/auth/add_oauth_token\\"}, {\\"oauth.config.id\\":\\"azure_sqldb\\", \\"oauth.config.client_id\\":\\"[your_client_id]\\", \\"oauth.config.client_secret\\":\\"[your_client_secret]\\", \\"oauth.config.redirect_uri\\":\\"[your_server_url]/auth/add_oauth_token\\"}, {\\"oauth.config.id\\":\\"databricks\\", \\"oauth.config.client_id\\":\\"[your_client_id]\\", \\"oauth.config.client_secret\\":\\"[your_client_secret]\\", \\"oauth.config.redirect_uri\\":\\"[your_server_url]/auth/add_oauth_token\\"}]" --
```

- ```
force-keys
```
- `tsm pending-changes apply`

Configure custom OAuth for a site

You can configure custom Azure Data Lake Storage Gen2, Azure Synapse, Azure SQL Database, and Databricks OAuth clients for a site.

Consider configuring a custom OAuth client to 1) override an OAuth client if configured for the server or 2) enable support for securely connecting to data that requires unique OAuth clients.

When a custom OAuth client is configured, the site-level configuration takes precedence over any server-side configuration and all new OAuth credentials created use the site-level OAuth client by default. No Tableau Server restart is required for the configurations to take effect.

Important: Existing OAuth credentials established before the custom OAuth client is configured are temporarily usable but both server administrators and users must update their saved credentials to help ensure uninterrupted data access.

Step 1: Prepare the OAuth client ID, client secret, and redirect URL

Before you can configure the custom OAuth client, you need the information listed below. After you have this information prepared, you can register the custom OAuth client for the site.

- **OAuth client ID and client secret:** First register the OAuth client with the data provider (connector) to retrieve the client ID and secret generated for Tableau Server.
- **Redirect URL:** Note the correct redirect URL. You will need this during the registration process in **Step 2:** below.

`https://<your_server_name>.com/auth/add_oauth_token`

For example, `https://myco.com/auth/add_oauth_token`

Step 2: Register the OAuth client ID and client secret

Follow the procedure described below to register the custom OAuth client to the site.

1. Sign in to your Tableau Server site using your admin credentials and navigate to the **Settings** page.
2. Under OAuth Clients Registry, click the **Add OAuth Client** button.
3. Enter the required information, including the information from **Step 1** above:
 - a. For **Connection Type**, enter one of the following values depending on the connector whose custom OAuth client you want to configure:
 - azuredatalakestoragegen2
 - azure_sqldb
 - azure_sql_dw
 - bigquery
 - databricks
 - google-analytics
 - google-sheets
 - salesforce
 - b. For **Client ID**, **Client Secret**, and **Redirect URL**, enter the information you prepared in **Step 1** above.
 - c. Click the **Add OAuth Client** button to complete the registration process.

Create OAuth Client

Connection Type
⊘ Enter a valid connection type.

OAuth Instance URL

Client ID
⊘ Enter the client ID from your OAuth provider.

Client Secret
⊘ Enter the client secret from your OAuth provider.

Redirect URL
⊘ Enter a valid URL ending with "/auth/add_oauth_token". Domain name can only be "localhost" when using HTTP schema.

4. (Optional) Repeat step 3 for all supported connectors.
5. Click the **Save** button at the bottom or top of the Settings page to save changes.

Step 3: Validate and update saved credentials

To help ensure uninterrupted data access, you (and your site users) must delete the previous saved credentials and add it again to use the custom OAuth client for the site.

1. Navigate to your **My Account Settings** page.
2. Under **Saved Credentials for Data Sources**, do the following:
 1. Click **Delete** next to the existing saved credentials for the connector whose custom OAuth client you configured in **Step 2** above.
 2. Next to connector name, click **Add** and follow the prompts to 1) connect to the

custom OAuth client configured in **Step 2** above and 2) save the latest credentials.

Step 4: Notify users to update their saved credentials

Make sure you notify your site users to update their saved credentials for the connector whose custom OAuth client you configured in **Step 2** above. Site users can use the procedure described in Update saved credentials to update their saved credentials.

Change Snowflake OAuth to Private Link with Saved Credentials

By default, the Tableau Snowflake connector uses a managed keychain for OAuth tokens that are generated for Tableau Server by the provider and shared by all users in the same site. Starting with Tableau 2020.4, you can configure Tableau Server to use a new OAuth service. In this scenario, you do not have to safelist the IP addresses to run the OAuth flow in AWS PrivateLink or Azure Private Link VPCs.

You can convert the Tableau Server to support “private link” environments by configuring the Snowflake connector to use Saved Credentials with a new OAuth Service.

Step 1: Obtain a client ID with Snowflake

To register a custom OAuth client with Snowflake, follow the procedure at [Configure Snowflake OAuth for Custom Clients](#).

After you register, you will use the following Snowflake parameters to configure Tableau Server:

- Account instance URL
- Client ID
- Client secret
- Redirect URL

Step 2: Configure Tableau Server for Snowflake OAuth

1. On the Tableau Server computer, run the following command to enable the Snowflake OAuth service:

```
tsm configuration set -k native_api.enable_snowflake_
privatelink_on_server -v true
```

2. Copy, paste, and customize the following command in a text editor:

```
tsm configuration set -k oauth.snowflake.clients -v "[{\"oauth.snowflake.instance_url\": \"https://account.snowflakecomputing.com\",
\"oauth.snowflake.client_id\": \"client_id_string\",
\"oauth.snowflake.client_secret\": \"client_secret_string\",
\"oauth.snowflake.redirect_uri\": \"http://your_server_url.-
com/auth/add_oauth_token\" }]"
```

The `oauth.snowflake.clients` key takes an array of key pairs. Each element in the key pair must be encapsulated by double quotes. Double quotes must be escaped as `\`.

Replace the values for each key as listed below:

- Account instance URL: `oauth.snowflake.instance_url`
- Client ID: `oauth.snowflake.client_id`
- Client secret: `oauth.snowflake.client_secret`
- Redirect URL: `oauth.snowflake.redirect_uri`

Note: Before running the command, verify the syntax carefully. TSM will not validate this input.

Copy the command into TSM CLI and run the command.

3. Enter the following command to apply changes:

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

Configure custom OAuth for a site

You can configure a custom Snowflake OAuth client for a site.

Consider configuring a custom OAuth client to 1) override an OAuth client if configured for the server or 2) enable support for securely connecting to data that requires unique OAuth clients.

When a custom OAuth client is configured, the site-level configuration takes precedence over any server-side configuration and all new OAuth credentials created use the site-level OAuth client by default. No Tableau Server restart is required for the configurations to take effect.

Important: Existing OAuth credentials established before the custom OAuth client is configured are temporarily usable but both server administrators and users must update their saved credentials to help ensure uninterrupted data access.

Step 1: Prepare the OAuth client ID, client secret, and redirect URL

Before you can configure the custom OAuth client, you need the information listed below. After you have this information prepared, you can register the custom OAuth client for the site.

- **OAuth client ID and client secret:** First register the OAuth client with the data provider (connector) to retrieve the client ID and secret generated for Tableau Server.
- **Redirect URL:** Note the correct redirect URL. You will need this during the registration process in **Step 2:** below.

`https://<your_server_name>.com/auth/add_oauth_token`

For example, `https://myco.com/auth/add_oauth_token`

Step 2: Register the OAuth client ID and client secret

Follow the procedure described below to register the custom OAuth client to the site.

1. Sign in to your Tableau Server site using your admin credentials and navigate to the **Settings** page.
2. Under OAuth Clients Registry, click the **Add OAuth Client** button.
3. Enter the required information, including the information from **Step 1** above:
 - a. For **Connection Type**, enter one of the following values depending on the connector whose custom OAuth client you want to configure:
 - `azuredatalakestorageegen2`
 - `azure_sqldb`
 - `azure_sql_dw`
 - `bigquery`
 - `databricks`

- google-analytics
 - google-sheets
 - salesforce
- For **Client ID**, **Client Secret**, and **Redirect URL**, enter the information you prepared in **Step 1** above.
 - Click the **Add OAuth Client** button to complete the registration process.

Create OAuth Client

Connection Type
⊗ Enter a valid connection type.

OAuth Instance URL

Client ID
⊗ Enter the client ID from your OAuth provider.

Client Secret
⊗ Enter the client secret from your OAuth provider.

Redirect URL
⊗ Enter a valid URL ending with "/auth/add_oauth_token". Domain name can only be "localhost" when using HTTP schema.

- (Optional) Repeat step 3 for all supported connectors.
- Click the **Save** button at the bottom or top of the Settings page to save changes.

Step 3: Validate and update saved credentials

To help ensure uninterrupted data access, you (and your site users) must delete the previous saved credentials and add it again to use the custom OAuth client for the site.

- Navigate to your **My Account Settings** page.
- Under **Saved Credentials for Data Sources**, do the following:

1. Click **Delete** next to the existing saved credentials for the connector whose custom OAuth client you configured in **Step 2** above.
2. Next to connector name, click **Add** and follow the prompts to 1) connect to the custom OAuth client configured in **Step 2** above and 2) save the latest credentials.

Step 4: Notify users to update their saved credentials

Make sure you notify your site users to update their saved credentials for the connector whose custom OAuth client you configured in **Step 2** above. Site users can use the procedure described in Update saved credentials to update their saved credentials.

Set up OAuth for Intuit QuickBooks Online

This topic describes how to set up your Intuit QuickBooks Online data sources for OAuth authentication. Complete these steps for each Tableau Server instance.

Setting up OAuth for QuickBooks Online consists of two tasks:

- Create a Connected App on the Intuit developer platform.
- Use the information you get as part of the Connected App to configure your server.

Create a Connected Intuit App

1. Sign in to your Intuit developer account, and then click **My Apps**.
2. In the **Just start coding** section, click **Select APIs**.
3. Select **Accounting** and click **Create App**.
4. In the **Get your app ready for submission** section, click the link to get your production keys.

Important: You must use production keys rather than development keys.

5. Copy the app token, OAuth consumer key, and OAuth consumer secret.

Configure Tableau Server for Intuit QuickBooks Online

- On the Tableau Server computer, open the bash shell and run the following commands:

```
tsm configuration set -k oauth.quickbooks.oauth_callback_uri -v http://YOUR-SERVER/auth/add_oauth_token
```

```
tsm configuration set -k oauth.quickbooks.consumer_key -v <your_consumer_key>
```

```
tsm configuration set -k oauth.quickbooks.consumer_secret -v <your_consumer_secret>
```

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

Managing access tokens

If you run an extract refresh job for your QuickBooks Online data source, Tableau Server attempts to renew access tokens for you. To help ensure that your access tokens do not

expire, run your extract refresh jobs more than once a month. Otherwise, the access tokens from QuickBooks Online expire and your extract refresh jobs fail. If your access tokens do expire, you can edit your saved credentials from the **Settings** page.

The saved credentials can be managed centrally or by your users. For more information, see [Allow Saved Access Tokens](#).

Allow Saved Access Tokens

After you configure Tableau Server for OAuth, you can decide to allow users to manage their own OAuth credentials, or you want to manage them centrally. If you want users to manage their own, you need to enable user profile settings from the server.

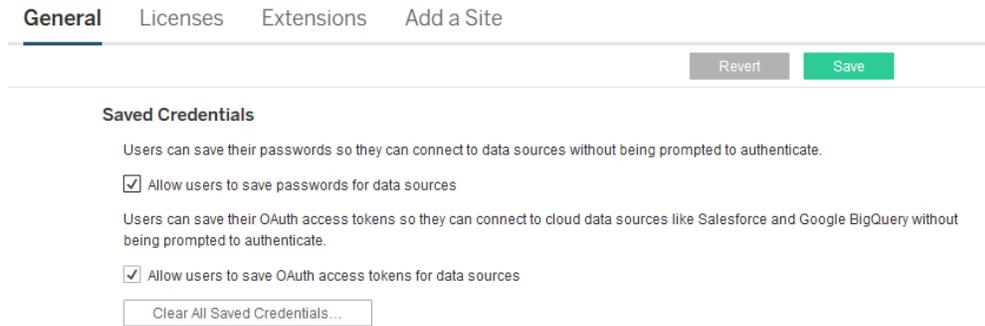
Note: If you have not yet configured your server to enable OAuth data connections, see the related topics listed below.

1. Sign in to Tableau Server as a server administrator.
2. **Single-site:** Click **Settings > General**.

Multisite: In the site menu, click **Manage All Sites** and then click **Settings > General**.

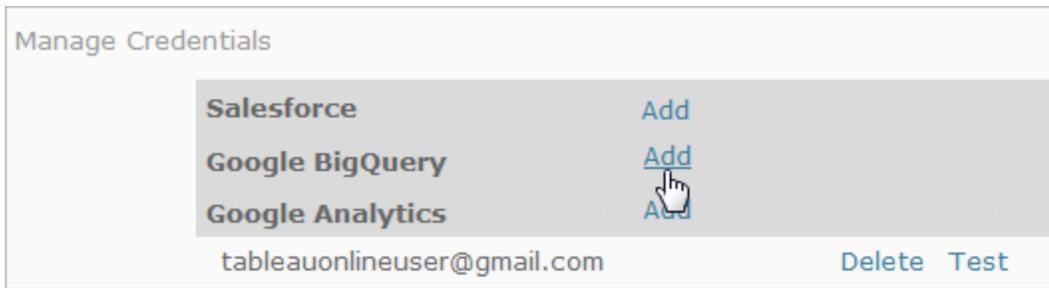
3. In the **Saved Credentials** section, select the following:
 - **Allow users to save passwords for data sources** (allows users to save their individual credentials with data sources).

- **Allow users to save OAuth access tokens for data sources**



4. Click **Save**.

After you select these check boxes, users will see a **Manage Credentials** section in their profile settings, where they can add access tokens for OAuth data connections.



Managing credentials centrally

Server administrators alternatively can manage OAuth credentials centrally. This can work well, for example, if multiple users work from the same data, and you have a dedicated user account for your data provider.

To manage credentials centrally, you do the following:

- Clear the check boxes described in the preceding procedure.
- Edit connection information as data sources are published.

When you edit the connection, you embed credentials that use an OAuth access token instead of an individual's user name and password.

When the settings for saving passwords and access tokens are not enabled, the Manage Credentials section is excluded from users' profile settings.

See also

[Change Google OAuth to Saved Credentials](#)

[Change Salesforce.com OAuth to Saved Credentials](#)

[Set up OAuth for Intuit QuickBooks Online](#)

[Troubleshoot OAuth Connections](#)

This topic provides information about resolving issues that can occur when you configure OAuth data connections.

Conflict error

In some cases, users may receive an error when attempting to connect with OAuth. The first sentence of the error message is:

The server encountered an internal error or misconfiguration and was unable to complete your request.

This error indicates that the fully qualified domain name (FQDN) of the Tableau Server needs to be added to the allowlist redirect key on Tableau Server.

When users are accessing a Tableau Server by the local host name (<https://tableau>) and the OAuth data provider is responding to the public DNS name (<https://data.example.com>), Tableau Server must associate the external FQDN with the local server name. The local host name is the server name in the URL that users enter when accessing Tableau Server from the internal network.

Tableau Server on Linux Administrator Guide

To fix this error, run `tsm configuration set` with the `oauth.whitelisted.redirect_to_origin_host` key option. This key takes a value pair, "internal_host, FQDN1, FQDN2". For example, the following commands set the local host name to *tableau* and the FQDN to *tableau.example.com*:

```
tsm configuration set -k oauth.whitelisted.redirect_to_origin_
host -v "tableau,tableau.example.com"

tsm pending-changes apply
```

In the case where multiple public URLs are used to access the internal Tableau Server, add additional FQDNs to the command, separated by commas, for example:

```
tsm configuration set -k oauth.whitelisted.redirect_to_origin_
host -v "tableau,tableau.example.com,tableau2.example.com"
```

If you need to edit an existing allowlist redirect configuration, you must enter the full mapping set. You cannot truncate or append existing configuration keys.

Configure SAP HANA SSO

You can configure Tableau Server to use SAML delegation to provide a single sign-on (SSO) experience for SAP HANA. This scenario is not dependent on SAML authentication to Tableau Server. You do not need to use SAML sign on with Tableau Server in order to use HANA SAML delegation. You can sign in to Tableau Server using whatever method you choose.

With SAML delegation for SAP HANA, Tableau Server functions as an identity provider (IdP).

Before you begin

Configuring SAML delegation with SAP HANA requires configuration on both Tableau Server and on SAP HANA. This topic provides configuration information about configuring Tableau Server. Before you configure Tableau Server, you must complete the following:

- Acquire a SAML certificate and key file for Tableau Server. The certificate file must be a PEM-encoded x509 certificate with the file extension `.crt` or `.cert`. This file is used by Tableau Server, and must also be installed on HANA. The private key must be a DER-encoded private key file that is not password protected, and that has the file extension `.der`. This file is only used by Tableau Server.
- Install the certificate in HANA. To avoid `libxmlsec` errors in HANA, we recommend configuring in-memory certificate store on SAP HANA. For more information, see this [SAP support topic](#).
- Install the latest version of SAP HANA driver (minimum version is 1.00.9) on Tableau Server.
- Configure network encryption from Tableau Server to SAP HANA (recommended).

For more information about generating the certificate/key pair, encrypting the SAML connection, and configuring SAP HANA, see [How to Configure SAP HANA for SAML SSO with Tableau Server](#) in the Tableau Community.

Configure Tableau Server SAML for SAP HANA

The following procedure describes how to configure SAML for SAP HANA on Tableau Server using `tsm data-access`. You can also configure SAML for SAP HANA using the `sapHanaSettings` Entity.

If you are running Tableau Server in a distributed deployment, run the following procedure on the initial node.

1. Place certificate files in a folder named `saml`. For example:

```
/var/opt/saml
```

2. Run the following commands to specify the location of the certificate and key files:

```
tsm data-access set-saml-delegation configure --cert-key
<cert-key> --cert-file <cert-file>
```

Where `<cert-key>` and `<cert-file>` are file paths to the private key and certificate file, respectively.

For example,

```
tsm data-access set-saml-delegation configure --cert-key
/var/opt/saml/hana_pkey_pkcs8.der --cert-file /var/-
opt/saml/hana_cert.pem
```

You can specify other options. For example, you can specify user name format and how credentials are normalized. See `tsm data-access`.

3. Run the following commands to enable delegation:

```
tsm data-access set-saml-delegation enable

tsm configuration set -k wgserver.sap_hana_sso.enabled -v
true

tsm configuration set -k wgserver.delegation.enabled -v
true
```

4. When you have finished, run `tsm pending-changes apply`.

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

Enable Kerberos Service Account Access

You can configure Tableau Server to use a Kerberos service account to access a database. In this scenario, Tableau Server connects to databases with a service account, also referred to as a "RunAs account".

To use RunAs authentication on Tableau Server you must first create a workbook or data-source that uses integrated authentication. When users publish to Tableau Server they will get the option for RunAs authentication. If you create a datasource with Tableau Server web

authoring that uses integrated authentication, the datasource will use RunAs authentication by default.

Note: *Integrated authentication* is also referred to as *Windows Authentication* on some connectors. In both cases, Tableau Server uses Kerberos authentication.

Data Access with the RunAs Service Account

To use RunAs authentication, the RunAs account requires read and query permissions to external databases. As designed, Tableau Server users with the *Creator* role or the *Explorer (Can Publish)* role have full access to the RunAs account for queries to external databases.

For example, a user with the Creator role can view all databases that have been granted access to the RunAs service account. They can also list tables and run Custom SQL.

If the Creator-user specifies the database host name and selects Integrated Authentication when creating a new data source with web authoring, then databases that have been granted RunAs access will be displayed to the user.

View access to database assets are not restricted to users who connect to Tableau Server with web authoring. Sophisticated users, who have the same roles noted above and who have knowledge of database server names, could create workbooks with Tableau Desktop that display databases that have been granted RunAs access.

Recommendations

Whether user access to databases in these scenarios is acceptable must be assessed by your organization. Generally, reducing the usage and scope of the RunAs service account will reduce the likelihood of inadvertent user access to database content. However, reducing the usage and scope of the RunAs service account may also impose more credential management to you and your users.

Evaluate the following recommendations in context of your business needs and data access policies.

- Firstly, be sure that you trust all users who have Creator roles or Explorer (Can Publish) roles. You will rely on these users to perform actions in Tableau with integrity.
- If you cannot trust all of your users who have publishing rights on data sources that are accessed by the RunAs service account, then you should consider embedding credentials for those data sources.
- If a data source is not set up for automated extract refreshes, that is, the data source is primarily accessed as a live connection, then you may be able to use Kerberos Delegation. For requirements, see [Enable Kerberos Delegation](#).

Requirements

- MIT Kerberos is not supported.
- The RunAs service account must have read access to the target database.

Configuration process

This section provides an example of the process to enable Kerberos service account access.

1. Create a domain user account to act as the RunAs service account. This account must have read access to the target database.

In the example here, the RunAs service account is User principal named `tab-srv@example.com`.

2. Create a keytab file for the RunAs service account.

For example, the following commands create a keytab (`tabsrv-runas.keytab`) using the `ktutil` tool:

```
ktutil
ktutil: addent -password -p tabsrv@EXAMPLE.COM -k 2 -e
<encryption scheme>
```

Encryption schemes for this command include `RC4-HMAC`, `aes128-cts-hmac-sha1-96`, and `aes256-cts-hmac-sha1-96`. Consult your IT team for the correct encryption scheme for your environment and data source.

```
ktutil: wkt tabsrv-runas.keytab
```

Tableau Server will use the RunAs service account and the associated keytab to authenticate and make a direct connection to the database.

3. Copy the keytab into the Tableau Server data directory and set proper ownership and permissions. The keytab should be readable by the unprivileged user. The default unprivileged user created by Tableau Setup is `tableau`.

If you are running a multi-node deployment, then you must run the following commands on each node in the cluster:

```
mkdir /var/opt/tableau/tableau_server/keytab
```

```
sudo cp -p tabsrv-runas.keytab /var/opt/tableau/tableau_server/keytab
```

```
sudo chown $USER /var/opt/tableau/tableau_server-  
/keytab/tabsrv-runas.keytab
```

```
chgrp tableau /var/opt/tableau/tableau_server-  
/keytab/tabsrv-runas.keytab
```

```
chmod g+r /var/opt/tableau/tableau_server/keytab/tabsrv-  
runas.keytab
```

4. Run the following TSM commands to enable RunAs access, set the RunAs service account, and associate the keytab file with the service account.

```
tsm configuration set -k features.RunAsAuthLinux -v true -  
-force-keys
```

```
tsm configuration set -k native_api.datasource_runas_prin-  
cipal -v tabsrv@EXAMPLE.COM --force-keys
```

```
tsm configuration set -k native_api.datasource_runas_
```

```
keytab_path -v /var/opt/tableau/tableau_server-  
/keytab/tabsrv-runas.keytab --force-keys
```

5. Run the following TSM command apply the changes to Tableau Server deployment:

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

Enable Kerberos Run As Authentication for JDBC Connectors

As of version 2020.2, Tableau Server supports Kerberos authentication for JDBC connectors.

You can configure Tableau Server to use a Kerberos service account to access a database. In this scenario, Tableau Server connects to databases with a service account, also referred to as a "Run As service account". This scenario is referred to as "Run As authentication"

To use Run As authentication on Tableau Server you must first create a workbook or data-source in Tableau Desktop that uses integrated authentication. When you publish to Tableau Server you will get the option to use Run As authentication. When creating a datasource with Web Authoring, Run As authentication is the default operation if you select integrated authentication.

Supported data sources

Tableau supports JDBC Kerberos delegation with the following data source:

- Oracle

Both native and JDBC-based connectors use the same configuration on Tableau Server on Linux. To configure Run As authentication see [Enable Kerberos Service Account Access](#).

SQL Server Impersonation

Impersonation in the context of Tableau Server means allowing one user account to act on behalf of another user account. You can configure Tableau and Microsoft SQL Server to perform database user impersonation, so that the SQL Server database account used by Tableau Server queries on behalf of SQL Server database users, who are also Tableau users.

The main benefit of this feature is it allows administrators to implement and control their data security policy in one place: their databases. When Tableau users access a view with a live connection to a SQL Server database, the view only displays what the users' database permissions authorize them to see. An additional benefit is that the users don't have to respond to a database sign-in prompt when they open the view. Also, workbook publishers don't have to rely on user-specific filters to restrict what's seen in views.

Impersonation Requirements

Here's what you need to use feature:

- **Live connections to SQL Server only:** Impersonation can only be used for views that have a live connection to a SQL Server database, version 2005 or newer.
- **Individual database accounts:** Each person who'll be accessing the view must have an explicit, individual account in the SQL Server database to which the view connects. Members of an Active Directory (AD) group cannot be impersonated. For example, if Jane Smith is a member of the AD group Sales, and her database administrator adds the Sales AD group to the SQL Server database, Jane cannot be impersonated.
- **Matching credentials and authentication type:** The credentials of each Tableau user's account and their Tableau user authentication type must match their credentials and authentication type in the SQL Server database. For example, if Jane

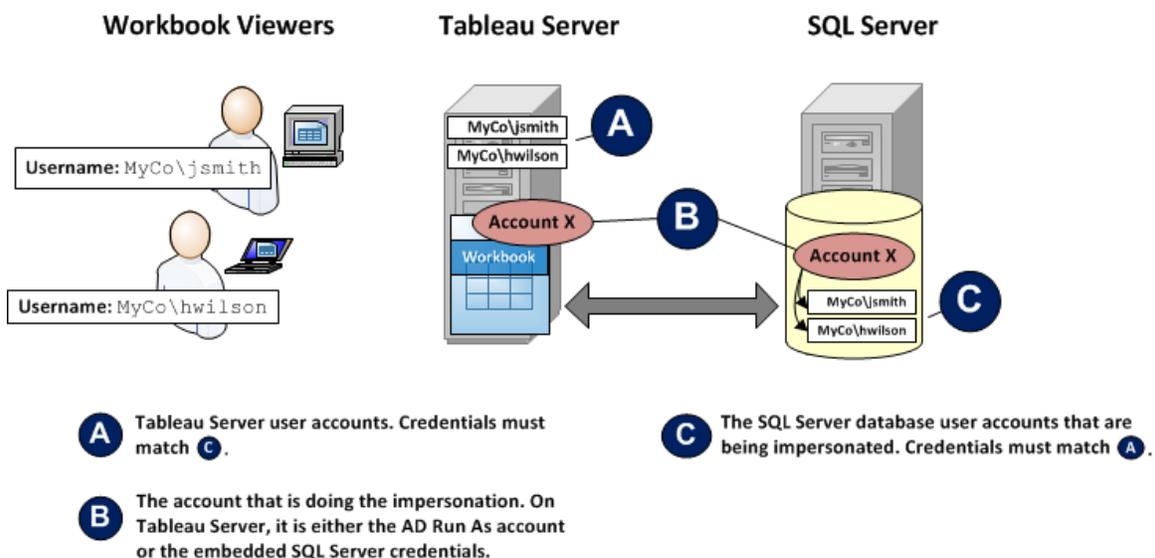
Tableau Server on Linux Administrator Guide

Smith's Tableau Server user account is `MyCo\jsmith`, the username on the SQL Server database must also be `MyCo\jsmith`. SQL Server must be using Windows Integrated Authentication.

- **SQL Server prerequisites:** In SQL Server you should have a data security table, a view that enforces data security, and you should require that your database users use the view.
- **SQL IMPERSONATE account:** You need a SQL Server database account that has IMPERSONATE permission for the above database users. This is either an account with the sysadmin role or one that has been granted IMPERSONATE permission for each individual user account (see the [MSDN article on EXECUTE AS](#)). The SQL Server account must be one of the following:
 - The Tableau Server Run As service account. See Enable Kerberos Service Account Access.
 - The workbook publisher's account. See Impersonate with Embedded SQL Credentials.

How Impersonation Works

Here's an illustration of how database user impersonation works:



In the above illustration, Jane Smith (MyCo\jsmith) is a West Coast sales representative and Henry Wilson (MyCo\hwilson) covers the East. In the SQL Server database, the account permissions for Jane's account, MyCo\jsmith, only give her access to West Coast data. Henry's account, MyCo\hwilson, can only access data for the East Coast.

A view has been created that displays data for the entire country. It has a live connection to a SQL Server database. Both users sign in to Tableau Server and click the view. Tableau Server connects to SQL Server using a database account with IMPERSONATE permission for each user's database account. This account acts on behalf of each user's database account.

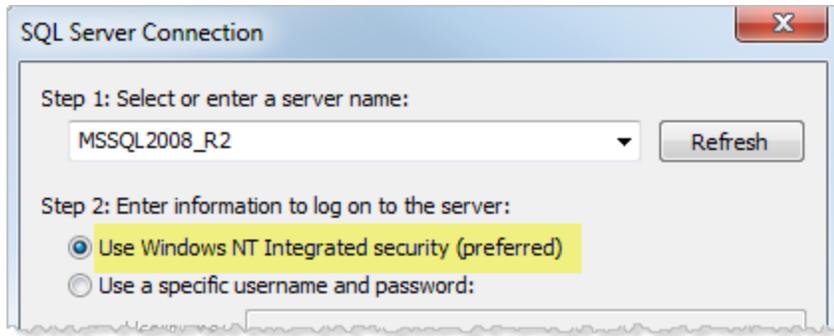
When the view displays, it is restricted by each user's individual database permissions: Jane sees only the West Coast sales data, Henry sees only the East Coast data.

Impersonate with a Run As Service Account

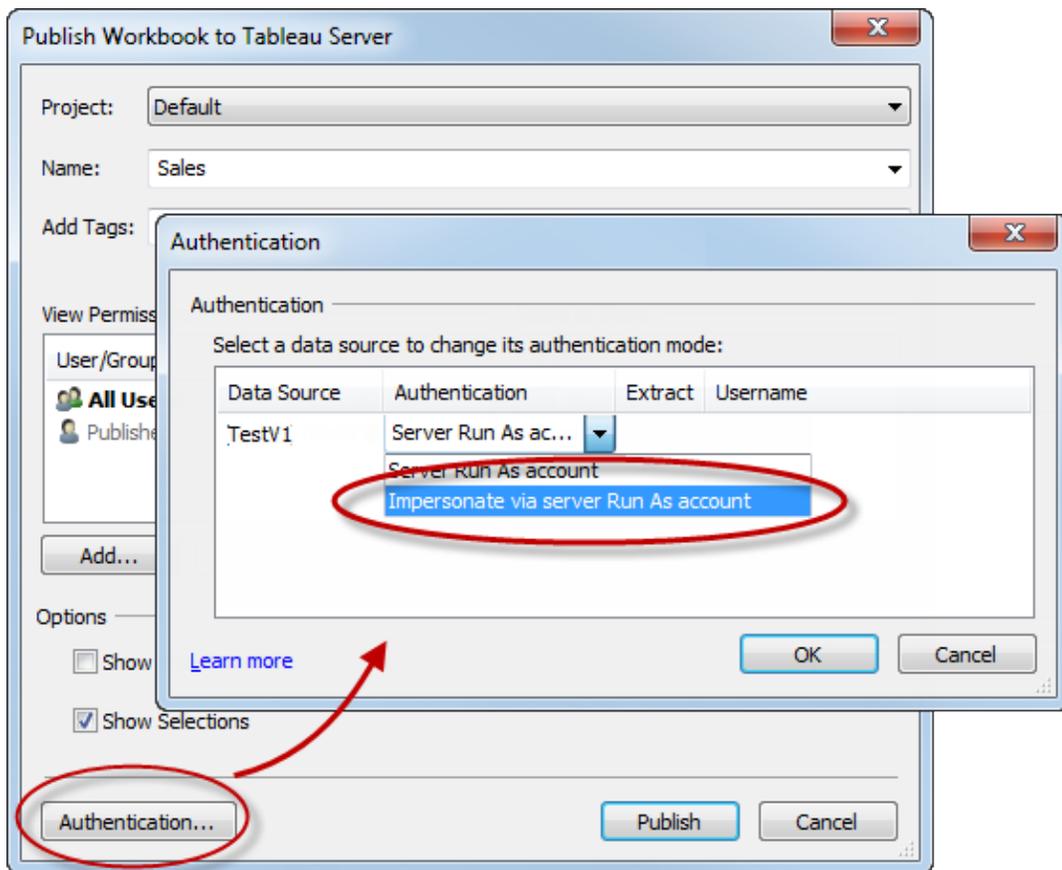
Impersonating via a Run As service account is the recommended way to perform impersonation. The Run As service account is an Active Directory user account the Tableau Server service can run under on the machine hosting Tableau Server. This same account must have IMPERSONATE permission for the database user accounts in SQL Server. From a data security standpoint, using the Tableau Server Run As service account for impersonation gives the administrator the most control.

To set up impersonation with a Run As User account:

1. Enable Kerberos Service Account Access.
2. Create a workbook in Tableau Desktop. When you create the data connection, select **Use Windows NT Integrated security** for the workbook's live connection to a SQL Server database:



3. In Tableau Desktop, publish the workbook to Tableau Server (**Server > Publish Workbook**).
4. In the Publish dialog box, click Authentication, then in the Authentication dialog box, select **Impersonate via server Run As account** from the drop-down list:



5. Click **OK**.
6. Test the connection by signing into Tableau Server as a user. When you click a view, you should not be prompted for database credentials and you should only see the data the user is authorized to see.

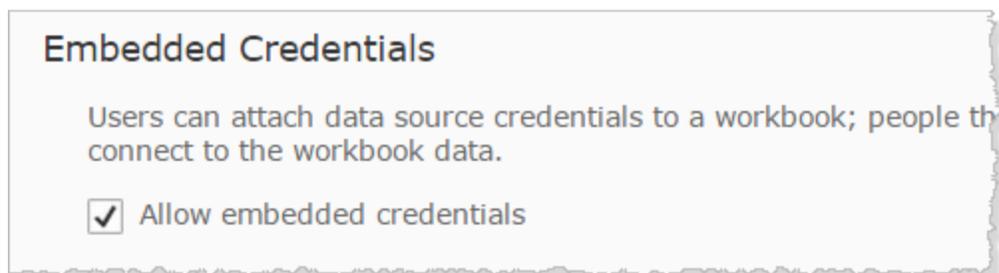
Impersonate with Embedded SQL Credentials

You can also perform impersonation by having the person who publishes a view embed their SQL Server account credentials in the view. Tableau Server can be running under any type of account, but it will use these credentials, supplied by the publisher, to connect to the database.

This may be the right choice for your site if the account that handles the impersonation cannot be an Active Directory (AD) account and if you're comfortable giving workbook publishers an account with a potentially high permission level on SQL Server.

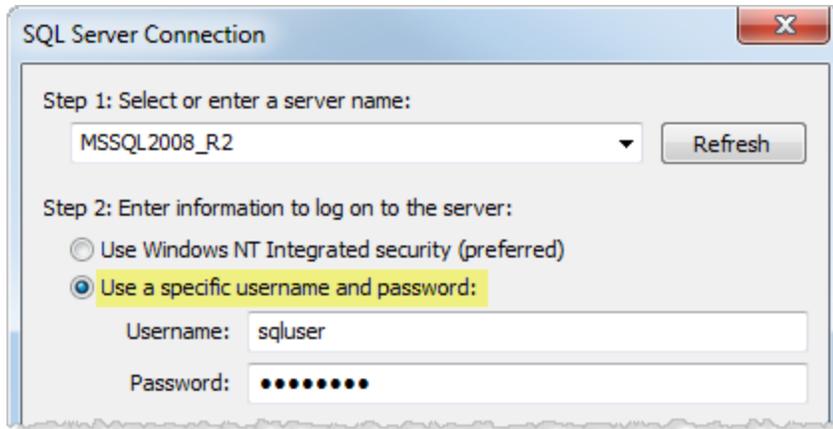
Note:

To use this approach, **Embedded Credentials** must be enabled on the server Settings page in Tableau Server:

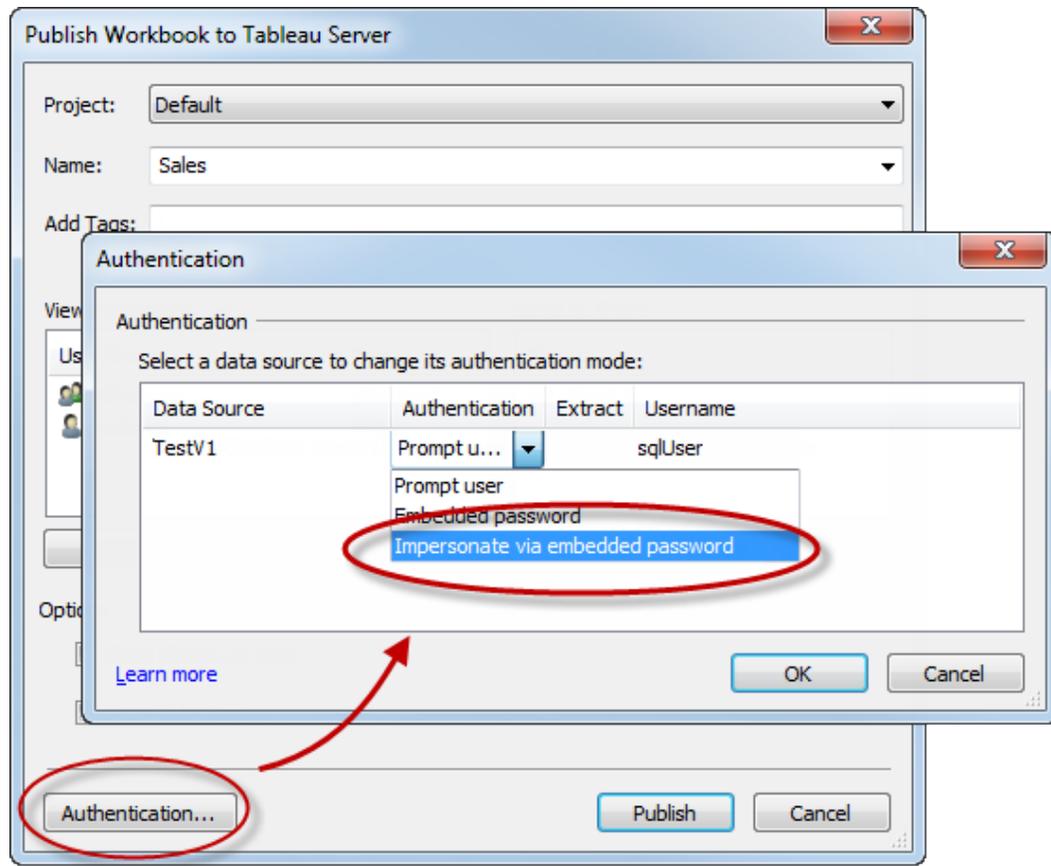


To impersonate with the workbook publisher's SQL account:

1. In Tableau Desktop, create a workbook. When you create the data connection, select Use a specific username and password for the workbook's live connection to a SQL Server database:



2. Publish the workbook to Tableau Server (**Server > Publish Workbook**).
3. In the Publish dialog box, click Authentication, then in the Authentication dialog box, select **Impersonate via embedded password** from the drop-down list:



4. Click **OK**.
5. Test the connection by signing in to Tableau Server as a user. When you click a view, you should not be prompted for database credentials and you should only see the data the user is authorized to see.

Configure a Custom TSM Administration Group

This topic describes how to configure a custom TSM administration group.

By default, the Tableau Server installation process creates a group called `tmsadmin`. Users in this group are authorized as administrators of TSM. You can change this default group during installation. See Help Output for `initialize-tsm` Script.

Tableau Server on Linux Administrator Guide

If you have already installed Tableau Server and wish to change the group used for TSM administration, follow the procedure in this topic.

You can configure Tableau Server to use an arbitrary group as a custom TSM administration group. Once configured, any user that is a member of the custom group will be able to administer TSM.

Step 1: Create the new group

Create a new group on your computer. Do not change the permissions on the default group as created by Linux.

Step 2: Configure Tableau Server

The custom TSM administration group name is stored in the `tsm.authorized.groups` configuration key. If you want to specify a group name (other than `tsmadmin`) then you will need to update the `tsm.authorized.groups` configuration key and then restart Tableau Server.

Use the `tsm configuration set` command to set the group name value. For example, to change the TSM administrative group name to `myadmingroup`, run the following commands:

```
tsm configuration set -k tsm.authorized.groups -v myadmingroup
tsm pending-changes apply
```

Notes

- Setting the `tsm.authorized.groups` configuration key overwrites any existing value stored on that key.
- If you have users in the existing `tsmadmin` group, and you overwrite the `tsm.authorized.groups` configuration key with a new value, then the users in the existing `tsmadmin` group will no longer be authorized for TSM.
- You can specify multiple groups by entering a comma-separated list of group names as the value.

Step 3: Add users to the new group

After you finished setting `tsm.authorized.groups`, any user in the new group(s) that you have specified will have full TSM administrative rights on Tableau Server.

Authorization

Authorization refers to how and what users can access on Tableau Server after authentication has been verified. Authorization includes:

- What users are allowed to do with content hosted on Tableau Server, including projects, sites, workbooks, and views.
- What users are allowed to do with the data sources that are managed by Tableau Server.
- What tasks users are allowed to perform to administer Tableau Server, such as configuring server settings, running command line tools, creating sites, and other tasks.

Authorization for these actions is managed by Tableau Server and determined by a combination of the user's site role and permissions associated with specific entities such as workbooks and data sources.

Site roles

Site roles define who is an administrator. Administrators can be assigned at the site or server level. For non-admins, site roles indicate the maximum level of access a user can have on a given site, subject to permissions set on content assets. For example, if one user is assigned the Viewer site role, and another the CreatorCreator

For more information about site roles, see [Set Users' Site Roles](#).

Permissions

Permissions determine whether a given user is allowed or denied to perform a specific action on a specific content asset.

As an administrator setting up Tableau Server, it's important that you understand how permissions are evaluated. Understanding the Tableau permissions process will enable you to

set up and configure permissions on sites, projects, and other assets so that you can control how content and data is shared, published, viewed, extracted, and imported.

Four important concepts to understand about permissions in Tableau are:

- **Permissions are asset-based.** Permissions are assigned to individual content assets (projects, data sources, workbooks) and are granted to users or groups.
- **Permissions are implicitly denied, and non-admin users must explicitly be allowed to access content.** The process by which Tableau Server determines the “allow” or “deny” permission is explained in detail in Permissions.
- **Permissions inheritance exists only in locked projects and in workbooks with tabbed views.** When content permissions are locked to the top-level project, workbooks, views, and data sources in the entire project hierarchy will use the default permissions set at the top-level project. In workbooks saved with the option **Show sheets as tabs**, views inside those workbooks use the workbook permissions. For more information, see Permissions.
- **In a project that is not locked, initial permissions are a one-time copy of the container item's permissions.** A data source or workbook starts with the default permissions, but authorized users can subsequently edit permissions on those assets. For more information on default permissions and projects, see Permissions.

Tableau Server provides a flexible permissions infrastructure that allows you to manage access to all content for countless scenarios. For more detailed information, see Permissions.

Data access and external authorization

There are scenarios where Tableau Server and Desktop rely on external authorization to enable access to data. For example:

- Users connecting to external data sources might require authorization that is outside the scope of Tableau Server's authority. If users publish an external data source, Tableau Server will manage access and capabilities of that data source. But if users embed an external data source in a workbook, it's up to the user who publishes the workbook to determine how other users who open the workbook will authenticate with the underlying data that the workbook connects to.

- Running Tableau Server in an organization with Active Directory, where Tableau has been configured with a Run As user account, results in a dependency on Active Directory and NTFS for authorization. For example, if you configure Tableau Server to use the Run As account to impersonate users connecting to SQL, then object-level authorization is reliant on NTFS and Active Directory.
- How users authenticate and are authorized by specific database solutions can differ. As noted, Tableau Server can be configured to provide access authorization when a data source is configured, but some databases will authorize access according to their own authentication scheme.

Data Security

Tableau provides several ways for you to control which users can see which data. For data sources that connect to live databases, you can also control whether users are prompted to provide database credentials when they click a published view. The following three options work together to achieve different results:

- **Database login account:** When you create a data source that connects to a live database, you choose between authenticating to the database through Windows NT or through the database's built-in security mechanism.
- **Authentication mode:** When you publish a data source or a workbook with a live database connection, you can choose an **Authentication mode**. Which modes are available depends on what you choose above.
- **User filters:** You can set filters in a workbook or data source that control which data a person sees in a published view, based on their Tableau Server login account.

The table below outlines some dependencies with the above options:

Database Connection Options		Data Security Questions		
Database login account uses...	Authentication mode	Is database security possible per Tableau Server user?	Are user filters the only way to restrict which data each user sees?	Are web caches shared among users?
Active Directory credentials (Windows Authentication)	Kerberos service account	No	Yes	Yes
	Impersonate via server Kerberos service account	Yes	No*	No
	Viewer enters their credentials	Yes	No*	No
User name and password	Prompt user: Viewers are prompted for their database credentials when they click a view. Credentials can be saved.	Yes	No	No
	Embedded credentials: The workbook or data source publisher can embed their database credentials.	No	Yes	Yes
	Impersonate via embedded pass-	Yes	No*	No

Database Connection Options		Data Security Questions		
Database login account uses...	Authentication mode	Is database security possible per Tableau Server user?	Are user filters the only way to restrict which data each user sees?	Are web caches shared among users?
	word: Database credentials with impersonate permission are embedded.			

* Because it can create unexpected results, Tableau recommends that you not use this authentication mode with user filters.

User filters, the embedded credentials option and the impersonation modes have similar effects—when users click a view, they are not prompted for database credentials and they see only the data that pertains to them. However, user filters are applied in the workbook by authors, and the impersonation authentication modes rely on security policies defined by administrators in the database itself.

Best Practices for Row Level Security

Row Level Security (RLS) in Tableau refers to restricting the rows of data a certain user can see in a workbook. This differs from Tableau permissions, which are used to control access to content and feature functionality. For example, permissions control whether a user can comment on a workbook, while Row Level Security allows two users viewing the same dashboard to see only the data they are allowed to see.

For live connections and multi-table extracts, the basic RLS workflow is:

Tableau Server on Linux Administrator Guide

1. The user is identified by logging into Tableau Server or Tableau Online
 - This requires a distinct username per user and secure single sign-on (SSO)
 - Active Directory, LDAP, or the Tableau REST API can be used to synchronize user names and establish permissions
2. The set of data entitlements for the user is retrieved from all possible data entitlements
 - This requires a data structure that can link entitlements to the Tableau user-name
3. The data is filtered by the entitlements for that user
 - This often requires using user functions in a calculated field
4. The published, filtered data is used to build content
 - Using a published (rather than embedded) data source with a data source filter ensures the RLS cannot be modified by downloading or web editing the workbook

How the joins, calculated fields, and filters are set up depends on the structure of the data and how users are managed.

Note: For more in-depth examples of the concepts outlined in this topic, refer to the whitepaper [Best Practices for Row Level Security with Entitlement Tables](#) or [How to Set Up Your Database for Row Level Security in Tableau](#) on the blog *Tableau and Behold*.

Entitlement tables

Any unique combination of attributes that the data can be filtered on is an entitlement. Most commonly, there are separate tables for specifying the entitlements themselves and mapping those entitlements to users or user roles. Denormalizing is recommended from a performance standpoint because joins are expensive operations.

The entitlements view, consisting of the entitlements mapped to users or roles, is joined with the data. A user-based data source filter is then applied, acting as a WHERE clause that brings in only the entitlements—and therefore the appropriate data rows—for the relevant user. (Query optimization should ensure the filtering occurs before joining when the query is

processed to minimize data duplication. For more information, see Performance and processing order of operations.)

Entitlement table models

Generally, there are two models for representing entitlements:

Full mapping to the deepest level of granularity

- Entitlements are defined fully for every column.
- There is one row in the mapping table for every possible entitlement the user has.
- This model requires fewer join clauses.

Sparse entitlements

- Entitlements are defined for every level of hierarchy, with NULL used to represent an “all” state.
- There is a single row in the mapping table for a particular level in the entitlement hierarchy, which vastly reduces the number of entitlement rows for users at high levels in a hierarchy.
- This model requires more complex joins and filters.

Users and roles

Combinations of entitlements are commonly represented as *roles*, which are then linked to users in a many-to-many mapping table. This allows for easily changing or removing a user from the role, while still maintaining a record of the role and its entitlements.

Alternatively, a many-to-many mapping table can be created that instead assigns users directly to entitlements as opposed to going through joining a role table. It will require managing the values more directly in the table but does eliminate a join.

Note: The user values associated with a role or entitlement need to match the username or full name on the Tableau site in order to take advantage of the user functions in Tableau Desktop.

Joins

Regardless of the model used to represent the entitlements, it is advisable to join all entitlements and mapping tables together into a single denormalized entitlements view. While at first this will cause a “blowup” (highly duplicative) version of the entitlements, the data source filter on the user will reduce it back down. You will also want this view if you plan on using an extract.

The deepest granularity method can have a performance benefit when everything is hierarchical—you only need to do a single join on the deepest level of the hierarchy. This only works if all of the attributes at the lowest level are distinct. If there is a chance for duplication (for example, a Central sub-region in more than one region), then you’ll need to join on all the columns to achieve the effect of a distinct key value.

The actual details and their performance characteristics depend on the data system and require testing. For example, using a single key could potentially improve the performance because the join is then only executing on one column, but correctly indexing all of the columns may give equal performance when other factors are taken into consideration.

Implement Row Level Security

Deepest granularity

Once the denormalized view of mapped entitlements is created, an inner join is set up between the view and the data in the Tableau data connection dialog. The data can remain in a traditional star schema. Alternatively, the dimension and fact tables can be materialized together into two views. Multi-table extracts will build extract tables to match the joins, so creating the two views will simplify the resulting extract. The SQL will follow this basic pattern:

```
SELECT *
FROM data d INNER JOIN entitlements e ON
d.attribute_a = e.attribute_a AND
d.attribute_b = e.attribute_b AND ...
WHERE e.username = USERNAME()
```

Sparse entitlements

If your entitlements more closely resemble the sparse entitlements model, then the custom SQL to join the data to the entitlements would be a little more complex because of the NULL values. Conceptually, it would look like the following:

```
SELECT *
FROM data d
INNER JOIN entitlements e ON
(e.region_id = d.region_id OR ISNULL(e.region_id) AND
(e.sub_region_id = d.sub_region_id OR ISNULL(e.sub_region_id)
AND
(e.country_id = d.country_id OR ISNULL(e.country_id)
```

Without using custom SQL, this can be done with a cross join and additional filters in Tableau Desktop. Create a join calculation on both sides of the join dialog that simply consists of the integer 1 and set them equal. This joins every row from the data table with every row in the entitlements table.

Then you need a calculation (or individual calculations) to account for the levels in the hierarchy. For example, you could have several calculations that follow this format: `[region_id] = [region_id (Entitlements View)] OR ISNULL([region_id (Entitlements View)])`

Or you could have a combined calculation for all levels in one:

```
([region_id] = [region_id (Entitlements View)] OR ISNULL
([region_id (Entitlements View)])
AND
([sub_region_id] = [sub_region_id (Entitlements View)] OR
ISNULL([sub_region_id (Entitlements View)])
AND
([country_id] = [country_id (Entitlements View)] OR ISNULL
([country_id (Entitlements View)])
```

The ISNULL function matches any entitlement column to all items in the other column. As always with RLS, these calculations should be added as data source filters.

Data source filter

For both approaches, once the entitlements are correctly joined with the data, a filter needs to be set up to limit the data for a specific user. A calculated field should be created with a user function. For example, a simple Boolean comparison of whether the user listed in the Username field is the same as the username of the person logged into the Tableau site:

```
[Username] = USERNAME ()
```

This calculation should be used as a data source filter (with TRUE selected).

If the data source is embedded and a user has permissions to web edit or download the workbook, then the RLS is nonexistent since the filters enforcing it can be easily removed. The Tableau data source should be published separately as opposed to being left embedded in the workbook.

All access with deepest granularity

There is also a common scenario in which there are two access levels within the organization: people who can see everything (“all access”) or people with some reasonably definable subset of entitlements. This is most commonly seen for embedded applications—the organization hosting the data can see everything, but each client can only see their own data. In this case, you need a way to return the full data for the “all access” users, while maintaining the deepest granularity joins for all other users.

For this technique, you will use Tableau groups to create an override using a calculation in the join condition.

1. Create a group for users who should see all the data (here called All Access)
2. From the fact view, create a left join with two join conditions
 - The first join condition should be on the column that represents the deepest level of granularity

- The second join condition should be two calculations:
 - On the left side (the fact view), for the calculation, enter `True`
 - On the right side (the entitlements view), the calculation should be: `IF ISMEMBEROF('All Access') THEN False ELSE True END`
- 3. On a sheet, create a calculation structured as: `[Username] = USERNAME() OR ISMEMBEROF(['All Access'] (Entitlements View))`
- 4. Create a data source filter on the username calculation

If a user is a member of the All Access group, then the join becomes a left join on `True = False`. This means there are no matches at all in the entitlements view, so the entire fact view is returned with NULLs for the columns from the entitlements view (zero duplication). In the case where the user is not part of the All Access group, the `True = True` join condition doesn't change anything and the join will function as expected.

The user calculation used as a data source filter is true for all rows when the group override is working, or it will filter down to only the user's deepest granularity in the hierarchy.

Performance and processing order of operations

When a visualization is viewed in Tableau (Desktop, Server, or Online), Tableau sends an optimized query to the RDBMS which then processes the query and sends results back to Tableau to render the visualization with the resulting data. The order of operations for when joins, calculations, and filters are carried out depends on the query optimizer and how the query is executed.

Live connections

When using a live connection to a data source in Tableau, the performance of the query execution is dependent on the query optimizer which translates the incoming SQL into an efficient plan for retrieving the data.

There are two ways the query can be processed:

1. Filter the entitlement rows to the user then join to the fact table
2. Join the entitlements to the fact table then filter to the user's rows

In an ideal situation, the query optimizer will ensure the database processes the query by *filtering then joining*. If a user is entitled to everything, this means the maximum number of rows processed will be the number of rows in the data table.

If the database processes the query by *joining then filtering*, there may be duplication of data. The maximum number of rows processed will be the number of users entitled to see that particular row times each row in the data table.

It will be clear if this second scenario happens: your queries take a long time to finish, you get errors, or there is an indication of performance issues in the database. Your total data volume will expand exponentially, which could cause inordinate system strain on the backend.

Extracts

When the data source in Tableau is a live connection, Tableau sends every query that is necessary to render a particular viz or dashboard to the RDBMS. When the data source is an extract, the process of querying data from the underlying data source only happens at extract creation and refresh. All of the individual queries for visualizations are answered by the extract engine from the extract file.

The same order of operations issue is present when building single table extracts. However, the “blowup” will happen both on the underlying data source and within the resulting extract itself.

Considerations with extracts

Starting in version 2018.3, the data engine can create a multi-table extract and RLS can be implemented as described above. Using multiple table extracts reduces the time it takes to generate an extract with many-to-many relationships by not materializing the join.

The extract should be built with a *data object* and an *entitlements object*. This is the simplest storage in the extract and results in the best performance.

- The *data object* is the table, view or custom SQL query that represents the denormalized combination of the fact and necessary dimension tables

- The *entitlements object* is a denormalized table, view or custom SQL query of whatever entitlements are necessary to filter the data at the most granular level, which requires:
 - A column for username matching the exact usernames in Tableau Server or Tableau Online
 - A row for each of the most granular entitlements to the data object

This format is laid out in the deepest granularity method above. Multi-table extracts use the same method, with the caveat that only two data objects are being joined and any field-specific filtering is already applied within the object.

Because multiple table extracts have extract filters disabled, you can filter either in the views or tables you connect to in the data source, or define the filters in custom SQL objects in the Tableau data connection dialog.

Note: As with live connections, if the data source is embedded and a user has permissions to web edit or download the workbook, then the RLS is nonexistent since the filters enforcing it can be easily removed. The extract should be published separately as opposed to being left embedded in the workbook.

Single table extracts

The following method is only recommended when using a version of Tableau prior to 2018.3—multiple table extracts are preferable if available.

Single table extracts materialize any joins you build when constructing the Tableau data source and stores everything as a single table through one query, the results of which are transformed in a single table in the extract file. This denormalization carries the risk of causing massive data duplication, as every row that was allocated to more than one entitlement or user would be duplicated as a result of the many-to-many relationship.

To prevent this duplication:

1. Create a Security Users Field that contains the usernames for that entitlement
 - for example, a value may be “bhowell|mosterheld|rdugger”

2. Use the CONTAINS() function within Tableau to correctly identify individual users
 - For example, `CONTAINS([Security Users Field], USERNAME())`

This method obviously has some caveats. It requires that you go from your entitlements in rows to a single column separated correctly using SQL, and that column can only contain so many characters. Partial matches can be trouble, and you need to use separators that will never be valid in the IDs themselves. Although it is performant within the Tableau Data Engine, as a string calculation it will be very slow for most databases. This limits your ability to switch back to a live connection.

Alternatively, you can take different extracts per “role” or entitlement level, so that only the data appropriate to that person or level is contained within the extract, but this will require processes to appropriately permission and leverage template publication within Tableau Server, generally via the APIs.

Use built-in Row Level Security in a data source

Many data sources have mechanisms for RLS built in. If your organization has already put effort into building Row Level Security in a data source, you may be able to take advantage of your existing RLS. It is not necessarily easier or better to implement a built-in RLS model vs. building it with Tableau in mind; these techniques are generally leveraged when an organization has already invested in these technologies and they want to take advantage of the investment. The main benefit of using built-in RLS is that administrators can implement and control their data security policy in one place: their databases. For more information, see [Row Level Security in the Data Source](#).

Row Level Security in the Data Source

If your organization has already put effort into building out Row Level Security (RLS) in a data source, you may be able to use one of the following techniques to take advantage of your existing RLS. In order to leverage the data source’s security models, live connections are required. Additionally, these techniques are likely not available in Tableau Online; the Tableau username for Online is a unique email address which is not typically the user identity on the data source side.

It is not necessarily easier or better to implement a built-in RLS model vs. building it with Tableau in mind; these techniques are generally leveraged when an organization has already invested in these technologies and they want to take advantage of the investment.

Impersonation (Microsoft SQL Server)

Microsoft SQL Server (and a few related systems) can be configured so that users of the database only have access to views with RLS filters built in, either utilizing Security Junction Tables or views built by the DBA. Tableau can take advantage of this using a concept called “impersonation.”

When publishing a Tableau Data Source containing an MS SQL Server connection to Tableau Server, there are two authentication options available to take advantage of impersonation. The menu you see will depend on whether you logged into the SQL Server using network authentication or by entering username/password credentials.

To enable RLS filtering for any user who can access the Published Data Source in Tableau Server, either the AD Run-As Account or the embedded SQL server credentials must have permission to EXECUTE AS for all of the Tableau users in the database that will be accessing the dashboard or data source. All Tableau users must exist in the database server as users, with SELECT rights for the Views you are trying to connect to (and have RLS applied to). See Impersonation Requirements for the comprehensive list of requirements.

Kerberos & constrained delegation

Constrained delegation within Tableau Server using Kerberos operates similarly to impersonation in that it allows Tableau Server to use the Kerberos credentials of the view of a workbook or view to execute a query on behalf of the viewer, so if RLS is set up on the data source, the viewer of the workbook will only see their data.

To see the comprehensive list of data sources where Kerberos delegation is supported, see Enable Kerberos Delegation. Active Directory is required; the computer where Tableau Server is installed must be joined to the Active Directory domain. The **authentication method** specified when publishing the data source must be **viewer credentials**.

Note that Kerberos can be leveraged for RLS when using Microsoft Analysis Services.

OLAP Cubes

OLAP Cube connections in Tableau do not have the equivalent of a data source filter, which is required for the entitlements table-based RLS method in Tableau, or access to the `USERNAME()` function. For these reasons, Kerberos and constrained delegation is a recommended approach to RLS with OLAP data sources, which allows Tableau to leverage user filtering that has already been implemented on the OLAP Server side.

If the users viewing the dashboard will not be part of the domain, then the manual approach to creating user filters is possible. However, because the User Filter Set generated cannot be added as a data source filter, and will instead exist on the filters shelf, it is important that Web Editing and Download Workbook functionality is not permissible for any published views utilizing this method.

SAML delegation & SAP HANA

If Tableau Server is configured to use Configure SAP HANA SSO to provide a single sign-on experience, the viewer credentials are used to execute the query as that user, which will operate within whatever security is applied on the user level. The **authentication method** specified when publishing the data source must be **viewer credentials**.

Initial SQL to force a user-specific session (Oracle VPD)

Initial SQL allows you to specify a SQL command that is run when the connection is made to the database for the purpose of setting up temporary tables to use during the session or to set up a custom data environment.

For Oracle VPD, you can set up a session specific to a user by running a particular stored procedure or function to set the context of the database connection to match the Tableau user's username:

```
begin
DBMS_SESSION.SET_IDENTIFIER([TableauServerUser]);
end;
```

The same high-level requirements hold true for using this for RLS as with impersonation; the DBA must set up VPD and all of the associated users to exist on the database.

On MS SQL Server, you could force an EXECUTE AS command (however, this is similar to what Tableau does with impersonation already) :

```
EXECUTE AS USER = [TableauServerUser] WITH NO REVERT;
```

Note: If the data source is embedded and a user has permissions to web edit or download the workbook, then the RLS is nonexistent since the initial SQL enforcing it can be easily removed. The data source should be published separately as opposed to being left embedded in the workbook.

Comparison matrix for Row Level Security Methods

Method	Useful when	Pros	Cons
Entitlements table (Recommended)	<ul style="list-style-type: none"> There is an existing concept of entitlements in the database The organization is setting up Row Level Security for the first time 	<ul style="list-style-type: none"> Easy to test, update, maintain, and scale Works for both live connections and extracts in version 2018.3+ 	<ul style="list-style-type: none"> Requires creating and maintaining entitlements table Could require selecting and creating appropriate keys to optimize for performance
CONTAINS() with extracts	<ul style="list-style-type: none"> Implementing RLS in extracts prior to version 2018.3 	<ul style="list-style-type: none"> Allows you to take advantage of extract efficiencies 	<ul style="list-style-type: none"> Requires mapping all users to a single column Difficult to switch back to

			live connections because of string calculation
Impersonation	<ul style="list-style-type: none"> • Every user accessing the data will exist as a user in your SQL server (Usually, internal deployments) 	<ul style="list-style-type: none"> • Security is handled and maintained in one place—the database 	<ul style="list-style-type: none"> • Requires every person accessing the view to exist as a user within your SQL Server • Only works for Microsoft SQL Server
Kerberos	<ul style="list-style-type: none"> • All necessary data sources are set up for Kerberos delegation and RLS is set up on the data source (usually internal deployments) 	<ul style="list-style-type: none"> • The viewer's name appears on the access logs for the data source • Security is handled and maintained in the data source 	<ul style="list-style-type: none"> • Tableau must be configured to use LDAP-Active Directory • Tableau Server must be joined to the AD domain • Every user must exist within your AD domain
Initial SQL	<ul style="list-style-type: none"> • The database supports Initial SQL and RLS is set up on the data 	<ul style="list-style-type: none"> • Allows the passing of Tableau parameters at load time 	<ul style="list-style-type: none"> • Not all data sources support Initial SQL • Potential performance

source side

- Dedicated connection that can't be shared with other users
 - Users must exist within data source to execute query as user
- implications because of restricted cache sharing

Manage Server Secrets

Tableau Server needs to store a number of secrets it uses to perform various functions, typically securing internal communication, communicating with other applications or the operating system, or providing secure communication with clients. In this context, the term *secret* may refer to a password, a token, or other string that is used to authenticate one entity to another.

There are two categories of secrets that are required to run Tableau Server. They differ according to how the secrets are generated:

- **Secrets that are generated by administrators.** These include credentials and associated secrets for the Run As User account and the SMTP credentials used by Tableau Server.
- **Secrets that are automatically generated by various processes in the system.** For example, a secret is required to protect communication between the Cluster Controller and ZooKeeper processes. And a number of different passwords are required for each service and programmatic user that communicates with Postgres.

Most secrets are encrypted while at rest. When a secret is needed, it is decrypted at run time.

This topic describes how secrets storage works and what you need to do to properly manage storage of secrets on Tableau Server.

Tableau Server on Linux Administrator Guide

Understanding how secrets storage works

During installation Tableau Server generates and stores a master key in a Java keystore. The master key is used to encrypt a configuration encryption key that is used across the system.

Whenever a new secret is created or updated, the secret is encrypted with the configuration encryption key. The encrypted value is then stored with its corresponding configuration parameter in a YAML file on the server. Parameters that hold an encrypted value use the format, `ENC(<encrypted string>)`, where `<encrypted string>` is a Base64-encoded encrypted string.

At run time, when a given secret needs to be accessed, the encrypted values are read into memory and decrypted with the configuration encryption key.

In the case of pending changes, where secrets are entered during a configuration change, the entire transaction is encrypted. In this scenario, after you enter a secret and then save the pending change, the secret is transmitted to the Coordination Service (over encrypted SSL). The Coordination Service encrypts the secret and stores it until the pending changes are applied. When changes are applied, the secret (still encrypted) is promoted to the current configuration version.

Tableau Server encrypts secrets using 256-bit AES in GCM mode. The keys used for secure storage are different than the asset keys that are used to encrypt embedded database credentials before they are stored in the repository.

Who has access to the master key?

In a default installation, the Java keystore for Tableau Server will be replicated into the `/tabsvc/keystores` folder for each service on that node under `/var/opt/tableau/tableau_server/data/tabsvc/config`.

For example,

```
/var/opt/tableau/tableau_server-
/data/tabsvc/config/tabadminagent_<version_num-
ber>/tabsvc/keystores/tableauserver.jks.
```

If you use a custom install directory, then the keystore files will be found under

```
<install_directory>/tableau_server/data/tabsvc/config/<service
name_#.version_number>/tabsvc/keystores
```

By default, the following users and groups have access to this directory:

- root
- tableau (user)
- members of the 'tableau' group

Import and export configuration information

Tableau Services Manager introduces the capability to import and export configuration information using tsm configuration.

Note: This version of Tableau Server does not support restoring configuration information from a backup. Instead, we recommend using the export and import configuration commands to backup and restore configuration information.

While configuration secrets are encrypted when stored on disk internally, when the configuration is exported to a file, secrets are written into the file in plain text. It is up to the administrator to take measures to protect this file. There are a variety of options available:

- Write the file to an encrypted file system.
- Write the file to a directory that is restricted to specific users or groups by file system permissions.
- Encrypt the output file.

Securing secrets for import and export operations

This section describes how to PGP encrypt the backup output. With this method, you will

create a named pipe and provide it as the file argument, then use that as input to gpg encrypt it. The advantage is the secrets are never written to disk in plain text. gpg must be available and a private key available. An example of this method is shown in the following section.

The example in this section describes one way of handing secrets to store them on a separate computer.

See the following external references for more information:

- [File encryption on the command line](#)
- [mkfifo\(1\) - Linux man page](#)

Example: encrypt and export

The following is an example of how to secure the file when exporting the configuration.

```
mkfifo -m 600 /tmp/secure1 && (gpg --symmetric --batch --yes --  
passphrase-file ~/.secrets/pgppassphrase.txt --cipher-algo  
AES256 --output encrypted.enc < /tmp/secure1 &) && tsm settings  
export -f /tmp/secure1 && rm /tmp/secure1
```

The details of this operation are:

- Create a named pipe with access limited by file permissions to rw for current user.

```
mkfifo -m 600 /tmp/secure1
```

- Call gpg to encrypt the data sent to the named pipe, backgrounding it to a separate process. It will block waiting for data. The result will be a file containing the encrypted data.

```
gpg --symmetric --batch --yes --passphrase-file ~/.secret-  
s/pgppassphrase.txt --cipher-algo AES256 --output encryp-  
ted.enc < /tmp/secure1 &
```

- Call tsm to export the configuration, providing the named pipe as the file argument.

```
tsm settings export -f /tmp/secure1
```

- Delete the named pipe.

```
rm /tmp/secure1
```

The encrypted data is in the file "encrypted.enc."

Example: decrypt and import

The following is an example of how to decrypt and import the configuration.

```
mkfifo -m 600 /tmp/secret2 && (gpg --decrypt --batch --yes --
passphrase-file ~/.secrets/pgppassphrase.txt encrypted.enc >
/tmp/secret2 &) && tsm settings import -f /tmp/secret2 && rm
/tmp/secret2
```

The details of this operation are:

- Create a named pipe with access limited by file permissions to rw for current user.

```
mkfifo -m 600 /tmp/secure2
```

- Decrypt the configuration and send it to the named pipe. Background this to a separate process, it will block waiting to be read.

```
gpg --decrypt --batch --yes --passphrase-file ~/.secret-
s/pgppassphrase.txt encrypted.enc > /tmp/secret2 &
```

- Execute the tsm configuration import command, logging in as needed.

```
tsm settings import -f /tmp/secret2
```

- Delete the named pipe.

```
rm /tmp/secure1
```

The pending configuration contains the imported configuration.

Run `tsm pending-changes apply` to commit changes. If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

Cluster nodes

When adding a new node to your Tableau Server cluster, you will first need to generate the node configuration file (tsm topology). The node configuration file contains a copy of the master keystore file used for encrypting the configuration secrets.

Important: We strongly recommend that you take additional measures to secure the node configuration file when exporting a configuration file with secrets.

When [installing and configuring Tableau Server on the new node](#), you will need to provide the node configuration file to the `initialize-tsm` command. You can use a similar technique as described above to decrypt the contents of the file that was previously encrypted and send it via a named pipe to the `initialize-tsm` command.

Secrets storage event logging

The following events related to secrets storage are logged:

- Generating new encryption keys
- Encryption key is rolled or changed
- Encrypting a new value in the configuration file

For more information about log files and where they are stored, see [Work with Log Files](#).

Managing secrets

As a Tableau Server administrator the most important task related to secrets storage is to periodically update secrets. In some cases (server troubleshooting or auditing), you may need to retrieve a password.

For other operations, such as upgrading versions, backing up and restoring, or adding new nodes to a cluster—as noted above—Tableau Server manages secrets storage and related processes automatically.

Updating secrets

You should update secrets periodically, according to your company's security policy.

To update the master key and automatically generated secrets, run `tsm security regenerate-internal-tokens`.

Retrieving passwords

In some cases, you may need to retrieve a password for troubleshooting or other operations. For example, you may need the Postgres readonly user credentials that are generated and encrypted by Tableau Server. In these cases, you can run a `tsm` command that will retrieve and decrypt the password for you.

To retrieve a password, open Command Prompt and issue a `tsm configuration get` command for one of the parameters listed in the table below.

For example, to retrieve a password for the readonly Postgres user, type the following command:

```
tsm configuration get -k postgres.readonly_password
```

The command will return the password in clear text:

```
$ tsm configuration get -k postgres.readonly_password
```

```
password
```

Configuration Parameter	Description
<code>clustercontroller.zookeeper.password</code>	Password for cluster controller to connect to zookeeper.
<code>elasticsearch.client.password</code>	Password for logging into Elastic search

Tableau Server on Linux Administrator Guide

	service.
<code>elasticsearch.ssl.admin.cert.bytes</code>	Admin certificate that is used for administrative access to the Elastic search service. The admin certificate is used to generate the node certificate.
<code>elasticsearch.ssl.admin.key.file_bytes</code>	Certificate key for administrative access to the Elastic search service.
<code>elasticsearch.ssl.node.cert.bytes</code>	Certificate that is used for Elastic node-to-node communication.
<code>elasticsearch.ssl.node.key.file_bytes</code>	Certificate key that is used for Elastic node-to-node communication.
<code>elasticsearch.ssl.root.cert.bytes</code>	Certificate that is used to sign the admin and node certificates . This certificate is used by TSM for health check and by NLP to connect to Elasticsearch.
<code>elasticsearch.ssl.root.key.file_bytes</code>	Certificate key for root certificate.
<code>filestore.zookeeper.password</code>	Password for filestore to connect to zookeeper.
<code>hyper.connection.init_password</code>	Password used to initialize the Hyper database for user <code>tableau_internal_user</code> and is then used for connecting to Hyper.
<code>jdbc.password</code>	Password for the rails Postgres user.
<code>kms.persistent_store</code>	A collection of master encryption keys (MEKs) used by the Key Management System.
<code>maestro.rserve.password</code>	Password for connecting to an external Rserve instance used by Tableau Prep

	Conductor for running flows that have nodes with R scripts.
maestro.tabpy.password	Password for connecting to an external TabPy (Python server) instance used by Tableau Prep Conductor for running flows that have nodes with Python scripts.
oauth.google.client_secret	Client secret of the Google Cloud Platform account.
oauth.quickbooks.consumer_secret	Consumer secret of the Intuit developer account.
oauth.salesforce.client_secret	Client secret of the Salesforce developer account.
pgsql.adminpassword	<p>Password for the tblwgadmin Postgres user.</p> <div style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;"> <p>Note: Although the configuration parameter is encrypted in Tableau's configuration files (tabsvc.yml, workgroup.yml), this password is stored in plain text in files used by SAML.</p> </div>
pgsql.readonly_password	Password for the readonly Postgres user.
pgsql.remote_password	Password for the tableau Postgres user.
redis.password	Password for Redis.

	<p>Note: Although the configuration parameter is encrypted in Tableau's configuration files (tabsvc.yml, workgroup.yml), the configuration will still be in plain text in the redis.conf file that is consumed by the Redis application. Redis does not support encrypted/secured passwords.</p>
searchserver.client.password	
servercrashupload.proxy_server_password	Password for custom proxy server used to upload crash reports.
service.runas.password	Password of the Run As users. Stored temporarily.
ssl.cert.file_bytes	The content of one of the three SSL certificate files uploaded by the administrator. The certificate files are required to enable secure external connections to Tableau Server.
ssl.chain.file_bytes	The chain file(s) for the certificates uploaded by the administrator for external SSL.
ssl.key.file_bytes	Key file(s) for the certificates uploaded by the administrator for external SSL.
ssl.key.passphrase	Optional passphrase used to protect the external SSL key.
svcmonitor.notification.smtp.password	SMTP Server password supplied by the administrator through TabConfig.exe.

tabadminservice.password	Password for the service that allows server admins to download log files through the web interface.
vizportal.openid.client_secret	This is the password ("provider client secret") used for OpenID Connect SSO.
vizqlserver.external_proxy_password	Password used to authenticate to an external proxy.
wgserver.domain.password	Password used to bind to Active Directory.
wgserver.saml.key.passphrase	Passphrase used to access the PKCS#8 SAML key file.
zookeeper.tsm.password	Password that TSM uses to connect to Zookeeper coordination service

Extension Security - Best Practices for Deployment

The following information is for IT officers and administrators, Tableau server and site administrators, and anyone who is interested in managing dashboard extensions and the security of their data and business. The suggestions for deployment are intended for companies that have a mix of users who are on Tableau Desktop and Tableau Server or Tableau Online.

Security for extensions in Tableau

Extensions are web applications that could be hosted inside your network, or outside on a third-party server, or in a secure *sandboxed* environment hosted by Tableau. Extensions can interact with other components in the dashboard and potentially have access to the visible and underlying data in the workbook (through a well-defined API). Tableau supports two types of extensions:

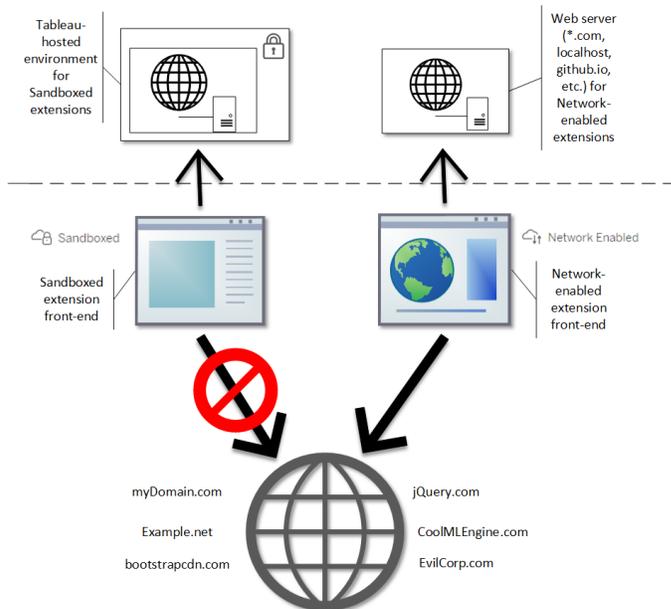
Tableau Server on Linux Administrator Guide

Network-enabled extensions

Network-enabled extensions are hosted on web servers that are located inside or outside of your local network and have full access to the web. Network-enabled extensions can connect with other applications and services, offering new capabilities to Tableau inside the dashboard, such as custom data visualizations, natural language generation, and write-back to data source scenarios. Network-enabled extensions have full access to the web, which means that while they can offer rich features and experiences by being able to connect to outside resources, they should be carefully evaluated before deploying or adopting.

Sandboxed extensions

Sandboxed extensions run in a protected environment without access to any other resource or service on the web. Sandboxed extensions are hosted by Tableau and provide the most security and eliminate the risk of data exfiltration. To safeguard against cyber-attacks, the Sandboxed extensions environment and hosting service has undergone extensive penetration testing by a 3rd-party consultant.



You can use Sandboxed and Network-enabled extensions in Tableau Desktop, Tableau Server, and Tableau Online. Tableau Server and Tableau Online provide the most control over the extensions your users can run.

Potential security risks with Network-enabled extensions

Because extensions are web applications there is the potential that a Network-enabled extension could be vulnerable to certain types of malicious attacks, which in turn could present a risk to your computer or data. The [Open Web Application Security Project](#) (OWASP) annually identifies the most critical web application security risks. These risks include the following:

- SQL injection
- Cross-site scripting (XSS)
- Sensitive data exposure

These risks could compromise the extension if the developers of the extension do not properly validate and handle user inputs, or if they generate dynamic queries to access sensitive databases. As you evaluate the extensions that you want to allow in Tableau, be sure to consider how they manage authentication, data access, or user input, and how they mitigate security risks.

Mitigating the security threats with Network-enabled extensions

Understanding what an extension does is a first step to identifying the risks for your enterprise. In many cases, a dashboard extension does not access underlying data in the workbook and all the JavaScript code runs in the context of the browser running on the user's computer. In these cases, no data leaves the computer even though the extension might be hosted on a third-party site outside of your domain. Some extensions allow you to connect Tableau with other applications you have already deployed in your domain.

Tableau provides security measures and security requirements for extensions. These are enabled for Tableau Desktop, Tableau Server, and Tableau Online.

- All extensions must use the HTTP Secure (HTTPS) protocol.

Tableau Server on Linux Administrator Guide

- By default, anyone using a dashboard with a Network-enabled extension will be prompted and asked to allow or deny the extension permission to run. The extension must request permission if it will access underlying data.
- To run on Tableau Server or Tableau Online, the URL of the Network-enabled extension must be added to a safe list. The server administrator manages this list for Tableau Server; the site administrator manages this list for Tableau Online.
- On Tableau Server and Tableau Online, the server or site administrator (respectively) can control whether the prompt appears for each Network-enabled extension.

For more information, see [Manage Dashboard Extensions in Tableau Server](#).

Manage extensions using Tableau

Extensions provide a way to add unique features to dashboards. You can use extensions to directly integrate the dashboard with applications outside of Tableau. While extensions open up a world of possibilities, there are instances where you need or want to maintain control of how extensions are deployed in your company or enterprise. In this respect, extensions are no different from any other software that you intend to use. Before you deploy software applications in your company you should thoroughly test and verify that the software works as expected and is secure. The same is true for extensions.

After you determine what level of access your users should have, and identify the extensions you want to use (or conversely, the extensions you don't want used), you can use the controls and features within Tableau to restrict and curate the dashboard extensions users have access to.

- Do you need to restrict who can add or use extensions in Tableau Desktop? See [Recommendations for Tableau Desktop](#)
- Do you need to restrict or control the extensions your users have access to? See [Recommendations for Tableau Server and Tableau Online](#).

Recommendations for Tableau Desktop

You have a range of options for deploying Tableau Desktop in your company. You can allow unrestricted access to Sandboxed and Network-enabled extensions, or you can put limits and restrictions on who has access to extensions and under what circumstances.

By default, Tableau Desktop users have unrestricted access to Sandboxed and Network-enabled extensions. You can use two options during installation to change the default settings.

- Turn off all extensions (`DISABLEEXTENSIONS`)
- Turn off Network-enabled extensions (`DISABLENETWORKEXTENSIONS`).

Note: You can change these settings after Tableau Desktop installation by editing the Registry (Windows) or running a script (Mac) on each Desktop. See [Turn off dashboard extensions](#).

Deployment scenarios

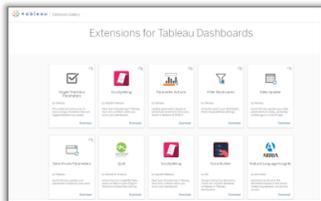
Using the installation settings, you can deploy Tableau Desktop in several ways.

- **Allow all extensions** - In this deployment scenario, you choose to trust Tableau dashboard authors to select the Sandboxed and Network-enabled extensions they want to use. If you want to empower your Tableau Desktop users with the greatest flexibility, use the default installation settings. Using the default settings, Tableau Desktop users have unrestricted access to Sandboxed and Network-enabled extensions. The default settings are: `DISABLEEXTENSIONS=0` and `DISABLENETWORKEXTENSIONS=0`. See [Install Tableau Desktop from the Command Line](#).
- **Only allow Sandboxed extensions** - In this scenario, you know that Sandboxed extensions are safe and you want to allow them, but you aren't sure about Network-enabled extensions and want to prevent their use. To turn off support for Network-enabled extensions, set the `DISABLENETWORKEXTENSIONS` property (`DISABLENETWORKEXTENSIONS=1`). Keep the default setting for enabling extensions (`DISABLEEXTENSIONS=0`). See [Install Tableau Desktop from the Command Line](#).
- **No extensions allowed** - In this scenario, you don't want to allow users to use

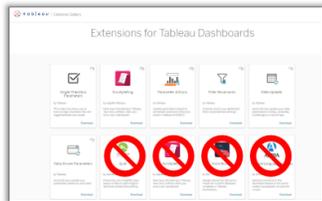
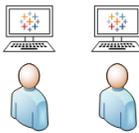
extensions of either type, Network-enabled or Sandboxed. In this case, turn support for all extensions off by using the `DISABLEEXTENSIONS` property (`DISABLEEXTENSIONS=1`). See [Install Tableau Desktop from the Command Line](#).

Use a combination of settings You might have some users who need and should have unrestricted access to all extensions, and others for whom access to Sandboxed extensions is sufficient, and then finally a set of users who need no access to extensions at all. Because the extension options are set per desktop, you can configure your deployment for specific users and their use cases.

Web authoring - If Tableau Server or Tableau Online are available for your users, they can use web authoring to access extensions. In web authoring, the server or site settings for extensions apply. In this scenario, the server and site administrators can determine which extensions to allow users access to. Administrators can use the server and site settings to restrict access to Sandboxed extensions only, or to restrict access to Sandboxed extensions and the Network-enabled extensions that have been added to a safe list.



If extensions are enabled on Tableau Desktop, users have unrestricted access to extensions.

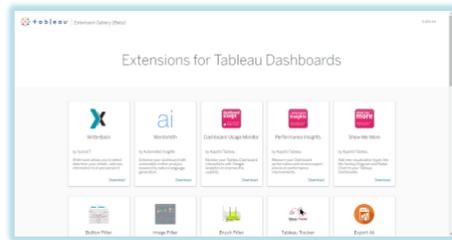


You can turn off extensions on Tableau Desktop to restrict access per desktop. And restrict access to Sandboxed extensions only.

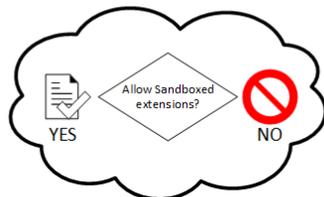


Recommendations for Tableau Server and Tableau Online

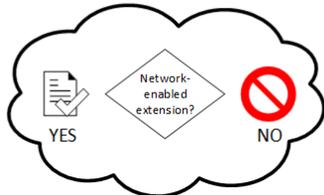
If your users have access to Tableau Server or Tableau Online, you can use the built-in security controls to put limits and restrictions on the extensions that can be used and under what circumstances. If you have turned off extensions on Tableau Desktop, you can still allow users to add extensions in web authoring, but you can limit the number of extensions that can be used to just ones you approve of.



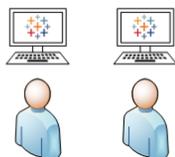
Use Tableau Server/Online settings to control access to Sandboxed extensions per site.



Use Tableau Server/Online settings to control access to Network-enabled extensions that you approve.



Creators can use web authoring to add Sandboxed extensions and approved Network-enabled extensions. All users can view and use Sandboxed and approved Network-enabled extensions.



Trust Sandboxed extensions and the Network-enabled extensions on the safe list

Starting with Tableau 2019.4, only Sandboxed extensions are allowed to run by default. Network-enabled extensions are not allowed unless they have been added to the safe list. Administrators can add Network-enabled extensions to the settings page for the site (**Settings > Extensions > Enable specific Extensions**).

Note To make the safe list the default behavior for extensions in Tableau 2018.2 and Tableau 2018.3, you need to change the settings for the site. On the Extensions settings page, under **Default behavior for Extensions**, clear the **Enable unknown extensions...** option. In Tableau Server 2019.1, Tableau 2019.2, and Tableau 2019.3, by default, no extensions are allowed to run unless they have been added to the safe list.

Checklist for the safe list:

- Does the extension come from a source that you know and trust?
- Check the URL of the extension. Does the URL look suspicious or contain dubious domain names?
- Does the extension require access to full (underlying data) or summary data? See [Understand data access](#).
- Test the extensions before allowing broad use. See [Test extensions for security](#). See [Test Network-enabled extensions for security](#).

Add extensions to the safe list:

- See [Add extensions to the safe list and configure user prompts](#).

Block specific extensions from running on Tableau Server

On Tableau Server, you can block specific extensions by adding their URL to the block list. This is useful if you have multiple sites that are configured differently for extensions. For example, if you have a test site where you want to be able to test internal or third-party extensions, you might have enabled the default behavior for extensions, where unlisted extensions are allowed to run provided they do not access the underlying data in the workbook. Adding an extension to the block list will prevent it from inadvertently being used on the test site.

- Add the URL of the extensions that you do not want to allow to the blocked list. This option is only available on Tableau Server. See [Block specific extensions](#).

Turn off extensions for a site

By default, extensions are enabled on Tableau Server and Tableau Online. On Tableau Server, the server administrator can turn off extensions for a site. On Tableau Online, the site administrator can turn off extensions for the site. On Tableau Server, the server administrator can turn off extensions completely, which overrides the site settings. You should not have to change this setting on the server or for the site, as you can control the Network-enabled extensions that you want to allow on the safe list, and you can control the settings for Sandboxed extensions, which are allowed by default.

- To disable extensions on a site (Tableau Server, Tableau Online), change the site settings that enables users to run extensions on the site. See [Control dashboard extensions and access to data](#).

Show or hide user prompts to run Network-enabled extensions

When you add a Network-enabled extension to the safe list, you can configure whether users see prompts by default when they are adding the extension to a dashboard, or when they are interacting with a view that has the extension. The prompt tells users details about the Network-enabled extension and whether the extension has access to full data. The prompt gives users the ability to allow or deny the extension from running. You can hide this prompt from users, allowing the extension to run immediately. When enabled for a site, Sandboxed extensions are allowed by default and do not prompt users.

Turn off Sandboxed extensions

Starting in Tableau 2019.4, Sandboxed extensions are enabled for Tableau Server and Tableau Online by default. Sandboxed extensions run in a protected environment and are hosted by Tableau. Administrators can control whether to let users run Sandboxed extensions on a site. Sandboxed extensions don't need to be added to the safe list. When Sandboxed extensions are allowed, users are able to freely add Sandboxed extensions to dashboards and are able to open and use dashboards that contain Sandboxed-extensions. If you need to block a Sandboxed extension, a server administrator can add the Sandboxed extension to a global block list. If you need to turn off Sandboxed extensions completely, you can change the default setting for the site. If you change the default setting for Sandboxed

extensions, only the extensions (including Sandboxed extensions) that are on the safe list will be allowed to run.

Tableau Server Key Management System

Tableau Server has three Key Management System (KMS) options that allow you to enable encryption at rest. One is a local option that is available with all installations of Tableau Server. Two additional options require the Server Management add-on, but allow you to use a different KMS.

Beginning in version 2019.3, Tableau Server added these KMS options:

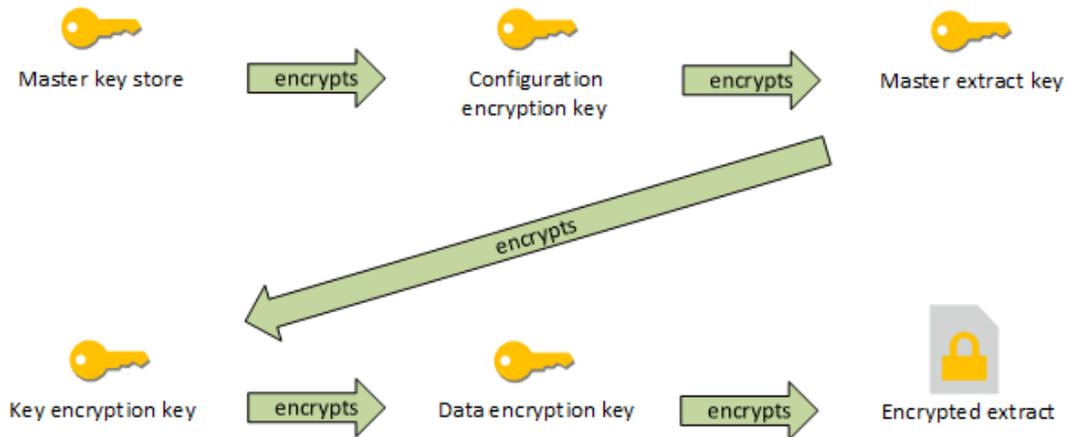
- A local KMS that is available with all installations. This is described below.
- An AWS-based KMS that comes as part of the Server Management add-on. For details, see [AWS Key Management System](#).

Beginning in version 2021.1, Tableau Server added another KMS option:

- An Azure-based KMS that comes as part of the Server Management add-on. For details, see [Azure Key Vault](#).

Tableau Server local KMS

The Tableau Server local KMS uses the secret storage capability described in [Manage Server Secrets](#) to encrypt and store the master extract key. In this scenario, the Java key-store serves as the root of the key hierarchy. The Java keystore is installed with Tableau Server. Access to the master key is managed by native file system authorization mechanisms by the operating system. In the default configuration, the Tableau Server local KMS is used for encrypted extracts. The key hierarchy for local KMS and encrypted extracts is illustrated here:



Troubleshoot configuration

Multi-node misconfiguration

In a multi-node setup for AWS KMS, the `tsm security kms status` command may report healthy (OK) status, even if another node in the cluster is misconfigured. The KMS status check only reports on the node where the Tableau Server Administration Controller process is running and does not report on the other nodes in the cluster. By default the Tableau Server Administration Controller process runs on the initial node in the cluster.

Therefore, if another node is misconfigured such that Tableau Server is unable to access the AWS CMK, those nodes may report Error states for various services, which will fail to start.

If some services fail to start after you have set KMS to the AWS mode, then run the following command to revert to local mode: `tsm security kms set-mode local`.

Regenerate RMK and MEK on Tableau Server

To regenerate the root master key and the master encryption keys on Tableau Server, run the `tsm security regenerate-internal-tokens` command.

Extract Encryption at Rest

Extract encryption at rest is a data security feature that allows you to encrypt .hyper extracts while they are stored on Tableau Server.

Tableau Server administrators can enforce encryption of all extracts on their site or allow users to specify to encrypt all extracts associated with particular published workbooks or data sources.

Limitations

Before they can be encrypted, older .tde file extracts must be upgraded to .hyper file extracts. This happens automatically as a part of the encryption job. For more information about the impacts of extract upgrade, see [Extract Upgrade to .hyper Format](#).

Temporary files and cache files are not encrypted at rest with this feature.

Workbooks (.twb) and data source files (.tds) are not encrypted with this feature. These files will contain metadata such as database table column names and formatting instructions. In certain cases, they may contain some row-level data if it is included in filters.

Other data files, such as Excel or JSON files, are not encrypted with this feature unless they are converted to extracts before being published.

When extracts are downloaded from the server they are decrypted.

Performance Overview

Increase in Background Load

You may see a slight to moderate increase in background load when you turn on encryption at rest. Encryption and decryption are computationally intensive operations. Encryption at rest alters existing backgrounder jobs and introduces new jobs to run on backgrounder. The overall increase in backgrounder load depends on the number and size of affected extracts and how often the scenarios below apply.

- **Initial publishing:** When publishing workbooks or data sources using extracts that

should be encrypted, the encryption happens on the server's backgrounders.

- **Extract refreshes from Tableau Server:** Full and incremental refreshes of encrypted extracts on Tableau Server will consume slightly more CPU.
- **Extract refreshes from Tableau Bridge and third-party applications (e.g., Informatica, Alteryx):** These flows will require new encryption jobs, scheduled on the backgrounders for any refreshed extract, resulting in a slight to moderate increase in backgrounder load.
- **Encrypting and decrypting extracts in already published workbooks and data sources:** If the site setting for encryption at rest is set to **Enable**, users might choose to encrypt or decrypt extracts in already published workbooks and data sources on Tableau server. Depending on the number and size of extracts, this will add slight to moderate load on the backgrounders.
- **Changing a site's encryption mode:** When switching a site's setting for encryption at rest to **Disable** or **Enforce**, the backgrounder will, respectively, decrypt or encrypt all existing extracts on the site. Depending on the number and size of extracts, this may significantly increase the load on backgrounders until all extracts are unencrypted or encrypted.
- **Rotating encryption keys:** Rotating encryption keys results in the backgrounders re-encrypting all existing extracts published on that site, using fresh encryption keys. Depending on the number and size of extracts, this may significantly increase the load on backgrounders until all extracts are re-encrypted.

If running at or over capacity, consider:

- Adding additional backgrounder processes and resources.
- Letting users encrypt individual workbooks and data sources instead of enforcing encryption for the whole site or disable encryption at rest for sites where it isn't necessary. Note that scheduled and ad hoc extract refreshes will take precedence over encryption and decryption jobs.

Increase in Viz Load Time and Worker Load

Query performance, for example, when loading or interacting with a viz or dashboard, will require the data being decrypted once, when loaded from disk to memory. This will result in

Tableau Server on Linux Administrator Guide

a slight increase in viz load time and CPU consumption on worker nodes for the first user loading a workbook. This will not affect other users accessing those workbooks at the same time because the data will already be decrypted in memory.

Impact on Backup and Restore

Encrypted extracts in backups remain encrypted. The size of backup files (.tbks) may increase up to 50-100% due to the ineffectiveness of compression on encrypted extracts. The size increase depends, among other factors, on the number of extracts that are encrypted. The time to restore a backup that contains encrypted extracts might increase slightly due to the time to exchange encryption keys.

If your Tableau Server installation has mostly or only encrypted extracts, consider disabling compression during backups to significantly improve the time backups take. To learn more about TSM backup, see [tsm maintenance backup](#).

Enforce Encryption at Rest on a Site

Tableau Server administrators can enforce encryption of all extracts on their site.

1. In a web browser, sign in to Tableau Server as a server administrator.
2. Go to the site you want to configure.
3. Click **Settings**.
4. Scroll down to the Extract Encryption at Rest section.
Click **Enforce** to encrypt all extracts that are published and stored on the site.
Encrypting all existing extracts stored on the site may take a while.
5. Click **Save**

Enable Encryption at Rest on a Site

Tableau Server administrators can allow users to specify to encrypt all extracts associated with particular published workbooks or data sources.

1. In a web browser, sign in to Tableau Server as a server administrator.
2. Go to the site you want to configure.
3. Click **Settings**.

4. Scroll down to the Extract Encryption at Rest section.
5. Click **Enable** to allow users to optionally encrypt extracts on the site.
Changing to Enable will cancel pending decryption jobs and pending encryption jobs.
No encryption jobs are created.
6. Click **Save**

Disable Encryption at Rest on a Site

1. In a web browser, sign in to Tableau Server as a server administrator.
2. Go to the site you want to configure.
3. Click **Settings**.
4. Scroll down to the Extract Encryption at Rest section.
5. Click **Disable** to not allow encrypted extracts on the site.
Changing to Disable will decrypt all existing encrypted extracts. Decrypting all extracts stored on the site may take a while.
6. Click **Save**

View Extract Encryption Mode for All Sites

1. On a multi-site server, click **Manage all sites** on the site menu.

Note: The **Manage all sites** option only displays when you are signed in as a server administrator.

2. Click **Sites**.
3. The encryption mode of each site is displayed in the **Extract encryption at rest** column.

Encrypt or Decrypt Extracts for a Published Workbook or Data Source

Note: The option to encrypt or decrypt the extracts associated with particular published workbook or data source is only available when the site setting for encryption at rest is set to **Enable**. When a site is set to Disable, all content is not encrypted. When a site is set to Enforce, all content is encrypted.

Note: You must be the owner or administrator.

Tableau Server on Linux Administrator Guide

1. Go to the published workbook or published data source page.
2. Click the dropdown menu that says **Encrypted Extract** or **Unencrypted Extract**.
3. Select **Unencrypted**.

You will see a message that says, “Decrypting extract.”

-or-

Select **Encrypted**.

An encryption job is started.

Alternatively, you can encrypt or decrypt extracts on the card view action menu, list view action menu, and action menu in the header section.

Encrypt or Decrypt Multiple Items

1. Go to the Data Sources page.
2. Select the check box beside one or more data sources.
3. In the upper-left of the Data Sources page, click **Actions**.
4. Click **Encrypt** or **Decrypt**.

View Encryption Status for a Single Item

1. Sign in to the site.
2. Go to a single data source page.
-or-
Go to a single workbook page for a workbook containing embedded data sources.
3. The encryption status is displayed on the page.

Filter Data Sources by Encryption Status

1. In the site, click **Explore**.
2. At the top-right, click the Explore: Top-level Projects dropdown menu and select **All Data Sources**.
3. Click the filter icon.
4. Scroll down to the “Live or extract” section and select a filtering option: All, Live, Extracts, Unencrypted Extracts, Encrypted Extracts, Currently Encrypting, or Currently Decrypting.
5. Select the checkbox beside “Include .tde and .hyper files” if you want to include “Live to .tde file” and “Live to .hyper file” connections in your filter results.

Filter Workbooks by Encryption Status

1. In the site, click **Explore**.
2. At the top-right, click the Explore: Top-level Projects dropdown menu and select **All Workbooks**.
3. Click the filter icon.
4. Scroll down to the “Live or extract” section and select a filtering option: All, Live, Extracts, Published, Unencrypted Extracts, Encrypted Extracts, Currently Encrypting, or Currently Decrypting.
5. Select the checkbox beside “Include .tde and .hyper files” if you want to include “Live to .tde file” and “Live to .hyper file” connections in your filter results.
Any workbooks that have at least one connection that matches the filter selection will be displayed.

View Status of Encrypt or Decrypt Extracts Background Tasks

1. In the site, click **Site Status**.
2. Click **Background Tasks for Non Extracts** to see completed and pending background task details.
Note: **Background Tasks for Non Extracts** includes all tasks not related to extract refreshes, so it includes encryption jobs.
3. In the Task menu, select **Encrypt Extracts** or **Decrypt Extracts** and click **Apply**.
4. In the Time Range menu, select a range.
You see "Encrypt Extracts" or "Decrypt Extracts" background tasks for all of your extract-based published data sources and workbooks.

The tabcmd Utility

The tabcmd command-line utility has commands and options to control extract encryption. For more information, see the tabcmd documentation.

Specify the extract encryption mode when you create a site

```
tabcmd createsite <site-name> --extract-encryption-mode
[enforced | enabled | disabled]
```

Tableau Server on Linux Administrator Guide

Specify the extract encryption mode when you edit a site

```
tabcmd editsite <site-name> --extract-encryption-mode [enforced  
| enabled | disabled]
```

Get the extract encryption mode when you list sites

```
tabcmd listsites --get-extract-encryption-mode
```

Encrypt extracts when you publish a workbook, data source, or extract to the server

```
tabcmd publish "filename.hyper" --encrypt-extracts
```

Decrypt all extracts on a site

Note: Depending on the number and size of extracts, this operation may consume significant server resources. Consider running this command outside of normal business hours.

```
tabcmd decryptextracts <site-name>
```

Encrypt all extracts on a site

Note: Depending on the number and size of extracts, this operation may consume significant server resources. Consider running this command outside of normal business hours.

```
tabcmd encryptextracts <site-name>
```

Reencrypt all extracts on a site with new encryption keys

You must specify a site.

Note: Depending on the number and size of extracts, this operation may consume significant server resources. Consider running this command outside of normal business hours.

```
tabcmd reencryptextracts <site-name>
```

For more information, see `reencryptextracts`.

Tableau Server Rest API

With the Tableau Server REST API you can manage Tableau Server resources programmatically. You can use this access to create your own custom applications or to script interactions with Tableau Server resources.

To learn more, see [Extract Encryption Methods](#).

Network Security

There are three main network interfaces in Tableau Server:

- **Client to Tableau Server:** The client can be a web browser, Tableau Mobile, Tableau Desktop, or the `tabcmd` utility.
- **Tableau Server to your database(s):** To refresh data extracts or handle live database connections, Tableau Server needs to communicate with your database(s).
- **Server component communication:** This applies to distributed deployments only.

In most organization, Tableau Server is also configured to communicate with the internet and with a SMTP server.

Client to Tableau Server

A Tableau Server client can be a web browser, a device running Tableau Mobile, Tableau Desktop, or `tabcmd` commands. Communications between Tableau Server and its clients

use standard HTTP requests and responses. We recommend configuring Tableau Server for HTTPS for all communications. When Tableau Server is configured for SSL, all content and communications between clients are encrypted using SSL, and the HTTPS protocol is used for requests and responses.

By default, passwords are communicated from browsers and `tabcmd` to Tableau Server using 1024-bit public/private key encryption. This level of encryption is not considered robust enough for secure communications. Additionally, this method, where a public key is sent to the recipient in the clear and without network layer authentication is susceptible to man-in-the-middle attacks.

To adequately secure network traffic from clients to Tableau Server, you must configure SSL with a certificate from a trusted certificate authority.

See [Configure SSL for External HTTP Traffic to and from Tableau Server](#).

Client access from the Internet

We recommend a gateway proxy server to enable secure client access from the internet to your Tableau Server. We do not recommend running Tableau Server in a DMZ or otherwise outside your protected, internal network.

Configure a reverse proxy server, with SSL enabled, to handle all inbound traffic from the internet. In this scenario, the reverse proxy is the only external IP address (or range of addresses if multiple reverse proxies are load-balancing inbound requests) that Tableau Server will communicate with. Reverse proxies are transparent to requesting clients, thereby obfuscating Tableau Server network information and simplifying client configuration.

For configuration information, see [Configuring Proxies for Tableau Server](#).

Clickjack Protection

By default, Tableau Server has *clickjack protection* enabled. This helps prevent certain types of attacks in which the attacker overlays a transparent version of a page on top of an innocuous-looking page in order to lure a user into clicking links or entering information. With click-

jack protection enabled, Tableau Server imposes certain restrictions on embedding views. For more information, see [Clickjack Protection](#).

Tableau Server to your database

Tableau Server makes dynamic connections to databases to process result sets and refresh extracts. It uses native drivers to connect to databases whenever possible and relies on a generic ODBC adapter when native drivers are unavailable. All communications to the database are routed through these drivers. As such, configuring the driver to communicate on non-standard ports or provide transport encryption is part of the native driver installation. This type of configuration is transparent to Tableau.

When a user stores credentials for external data sources on Tableau Server, they are stored encrypted in Tableau Server's internal database. When a process uses those credentials to query the external data source, the process retrieves the encrypted credentials from the internal database and decrypts them in process.

Tableau Server to the internet

In some cases, where users connect to external data sources, such as the Tableau map servers, then Tableau Server will need to connect to the internet. We recommend that you run all components of Tableau inside your protected network. Therefore, connections to the internet may require that you configure Tableau Server to use a forward proxy.

Tableau Server to a SMTP server

You can configure Tableau Server to send event notifications to administrators and users. As of version 2019.4, Tableau Server supports TLS for the SMTP connection. See [Configure SMTP Setup](#).

Communication with the repository

You can configure Tableau Server to use Secure Sockets Layer (SSL) for encrypted communications on all traffic that is exchange with the Postgres repository to and from other

server components. By default, SSL is disabled for communications between server components and the repository.

For more information, see `tsm security repository-ssl enable`

Server component communication in a cluster

There are two aspects to communication between Tableau Server components in a distributed server installation: trust and transmission. Each server in a Tableau cluster uses a stringent trust model to ensure that it is receiving valid requests from other servers in the cluster. Computers in the cluster running a gateway process accept requests from third parties (clients), unless they are fronted by a load balancer, in which case the load balancer receives the requests. Servers not running a gateway process only accept requests from other trusted members of the cluster. Trust is established by an allowlist of IP address, port, and protocol. If any of these are invalid, the request is ignored. All members of the cluster can communicate with each other.

When a user stores credentials for external data sources on Tableau Server, they are stored encrypted in Tableau Server's internal database. When a process uses those credentials to query the external data source, the process retrieves the encrypted credentials from the internal database and decrypts them in process.

Clickjack Protection

Tableau Server includes protection against clickjack attacks. *Clickjacking* is a type of attack against web pages in which the attacker tries to lure users into clicking or entering content by displaying the page to attack in a transparent layer over an unrelated page. In the context of Tableau Server, an attacker might try to use a clickjack attack to capture user credentials or to get an authenticated user to change settings on your server. For more information about clickjack attacks, see [Clickjacking](#) on the Open Web Application Security Project website.

Note: Clickjack protection was available in previous versions of Tableau Server, but was disabled by default. New installations of Tableau Server 9.1 and later will always have clickjack protection on unless you explicitly disable it.

Effects of clickjack protection

When clickjack protection is enabled on Tableau Server, the behavior of pages loaded from Tableau Server changes in the following ways:

- Tableau Server adds the `X-Frame-Options: SAMEORIGIN` header to certain responses from the server. In the current versions of most browsers, this header prevents the content from being loaded into an `<iframe>` element, which helps prevent clickjacking attacks.
- The top-level page from Tableau Server cannot be loaded in `<iframe>` elements. This includes the sign-in page. One consequence is that you cannot host Tableau Server pages in an application that you create.
- Only views can be embedded.
- If an embedded view requires data source credentials, a message is displayed in the `<iframe>` element with a link to open the view in a secure window where the user can safely enter credentials. Users should always verify the address of the opened window before entering credentials.
- Views can be loaded only if they include the `:embed=y` parameter in the query string, as in this example:

```
http://<server>/views/Sales/CommissionModel?:embed=y
```

Note: When clickjack protection is enabled, embedded views that use the URL copied from the browser address bar might not load. These view URLs usually

contain the hash symbol (#) after the server name (for example, `http://myserver/#/views/Sales/CommissionModel?embed=y`) are blocked when clickjack protection is enabled on Tableau Server.

Disabling clickjack protection

You should leave clickjack protection enabled unless it is affecting how your users work with Tableau Server. If you want to disable clickjack protection, use the following `tsm` commands:

1. `tsm configuration set -k wgserver.clickjack_defense.enabled -v false`
2. `tsm pending-changes apply`

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

HTTP Response Headers

Tableau Server supports some of the response headers specified in the [OWASP Secure Headers Project](#).

This topic describes how to configure the following response headers for Tableau Server:

- HTTP Strict Transport Security (HSTS)
- Referrer-Policy
- X-Content-Type-Options
- X-XSS-Protection

Tableau Server also supports the Content Security Policy (CSP) standard. CSP configuration is not covered in this topic. See [Content Security Policy](#).

Configuring response headers

All response headers are configured with the `tsm configuration set` command.

When you are finished configuring response headers, run `tsm pending-changes apply`.

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

HTTP Strict Transport Security (HSTS)

HSTS forces clients connecting to Tableau Server to connect with HTTPS. For more information see the OWASP entry, [HTTP Strict Transport Security \(HSTS\)](#).

Options

`gateway.http.hsts`

Default value: `false`

The HTTP Strict Transport Security (HSTS) header forces browsers to use HTTPS on the domain where it is enabled.

`gateway.http.hsts_options`

Default value: `"max-age=31536000"`

By default, HSTS policy is set for one year (31536000 seconds). This time period specifies the amount of time in which the browser will access the server over HTTPS.

Referrer-Policy

Beginning in 2019.2, Tableau Server includes the ability to configure Referrer-Policy HTTP header behavior. This policy is enabled with a default behavior that will include the origin

URL for all "secure as" connections (policy `no-referrer-when-downgrade`). In previous versions, the Referrer-Policy header was not included in responses sent by Tableau Server. For more information about the various policy options that Referrer-Policy supports, see the OWASP entry, [Referrer-Policy](#).

Options

`gateway.http.referrer_policy_enabled`

Default value: `true`

To exclude the Referrer-Policy header from responses sent by Tableau Server, set this value to `false`.

`gateway.http.referrer_policy`

Default value: `no-referrer-when-downgrade`

This option defines the referrer policy for Tableau Server. You may specify any of the policy value strings listed in the [Referrer-Policy](#) table on the OWASP page.

X-Content-Type-Options

The X-Content-Type-Options response HTTP header specifies that the MIME type in the Content-Type header should not be changed by the browser. In some cases, where MIME type is not specified, a browser may attempt to determine the MIME type by evaluating the characteristics of the payload. The browser will then display the content accordingly. This process is referred to as "sniffing." Misinterpreting the MIME type can lead to security vulnerabilities.

For more information see the OWASP entry, [X-Content-Type-Options](#).

Option

`gateway.http.x_content_type_nosniff`

Default value: `true`

The X-Content-Type-Options HTTP header is set to 'nosniff' by default with this option.

X-XSS-Protection

The HTTP X-XSS-Protection response header is sent to the browser to enable cross-site scripting (XSS) protection. The X-XSS-Protection response header overrides configurations in cases where users have disabled XSS protection in the browser.

For more information see the OWASP entry, [X-XSS-Protection](#).

Option

gateway.http.x_xss_protection

Default value: `true`

The X-XSS-Protection response header is enabled by default with this option.

Content Security Policy

Tableau Server supports the Content Security Policy (CSP) standard. CSP is intended to be an additional layer of security against cross-site scripting and other malicious web-based attacks. CSP is implemented as a HTTP response header that allows you to specify where external resources, such as scripts and images, can be safely loaded from.

See the [Mozilla website](#) for more information about CSP.

Configure and enable CSP

CSP is configured and enabled using the `tsm configuration set Options` command. If you are running Tableau Server in a distributed deployment, run these commands on the initial node in the cluster. The configuration will be applied across the cluster after you run `tsm pending-changes apply`.

Step 1: Set default directives

Tableau Server includes the set of default directives in the table below.

Tableau Server on Linux Administrator Guide

To set a directive, use the following tsm syntax:

```
tsm configuration set -k content_security_policy.-  
directive.<directive_name> -v "<value>"
```

For example, to set the `connect_src` directive, run the following command:

```
tsm configuration set -k content_security_policy.-  
directive.connect_src -v "* unsafe-inline"
```

Option	Default value	Description
content_security_policy.- directive.default_src	'none'	Serves as a fallback for the other fetch directives. Valid values for <code>default_src</code> .
content_security_policy.- directive.connect_src	*	Restricts the URLs which can be loaded using script interfaces. Valid values for <code>connect_src</code> .
content_security_policy.directive.script_src	*	Specifies valid sources for JavaScript. Valid values for <code>script_src</code> .
content_security_policy.directive.style_src	* 'unsafe-inline'	Specifies valid sources for stylesheets. Valid values for <code>style_src</code> .

content_security_policy.directive.img_src	* data:	Specifies valid sources of images and favicons. Valid values for <code>img_src</code> .
content_security_policy.directive.font_src	* data:	Specifies valid sources for fonts loaded using <code>@font-face</code> . Valid values for <code>font_src</code> .
content_security_policy.-directive.frame_src	* data:	Specifies valid sources for nested browsing contexts loading using elements such as <code><frame></code> and <code><iframe></code> . Valid values for <code>frame_src</code> .
content_security_policy.-directive.object_src	data:	Specifies valid sources for the <code><object></code> , <code><embed></code> , and <code><applet></code> elements. Valid values for <code>object_src</code> .
content_security_policy.-directive.report_uri	/vizql/csp-report	Instructs the user agent to report attempts to violate the CSP. These violation reports consist of JSON documents sent via an HTTP POST request to

		<p>the specified URI.</p> <p>Valid values for <code>report_</code> <code>uri</code>.</p>
--	--	--

Step 2: Add additional directives (optional)

The default directives included with Tableau Server are a subset of directives that are supported by CSP.

For a full list of supported CSP directives, go to <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Security-Policy>.

You can add directives to the existing default set, by using adding the new directive in the `content_security_policy.directive` namespace. You must include the `--force-keys` parameter when adding new directives. The syntax is as follows:

```
tsm configuration set -k content_security_policy.directive.<new_
directive_name> -v "<value>" --force-keys
```

For example, to add the `worker-src` directive, run the following command:

```
tsm configuration set -k content_security_policy.-
directive.worker-src -v "none" --force-keys
```

Step 3: Specify report-only directives (optional)

You can configure CPS to report some directives and to enforce others. When you set `content_security_policy.enforce_enabled` to `true`, then all directives are enforced (even if `content_security_policy.report_only_enable` is also set to `true`).

To specify directives as "report-only" and not enforced, add the directives to the `report_only_` directive namespace. You must include the `--force-keys` parameter when adding new directives. The syntax is as follows:

```
tsm configuration set -k content_security_policy.report_only_
directive.<directive_name> -v "<value>" --force-keys
```

For example, to report only on the `script_src` directive, run the following command:

```
tsm configuration set -k content_security_policy.report_only_
directive.script_src -v " http://*.example.com" --force-keys
```

Step 4: Enable CSP on Tableau Server

After you have configured directives, enable CSP on Tableau Server.

The following options are used to enable enforcement or report only mode for the directives you have set.

Option	Default value	Description
<code>content_security_policy.enforce_enabled</code>	false	Adds a CSP header to all requests so that any violation will be enforced by the browser.
<code>content_security_policy.report_only_enabled</code>	true	Adds a CSP header to all requests so that any violation will be recorded in our vizql-client logs, but will not be enforced by the browser.

To enable enforcement of the CSP directives that you've specified, run the following command

```
tsm configuration set -k content_security_policy.enforce_
enabled -v true
```

Step 5: Run tsm pending-changes apply

When you are finished configuring CSP, run `tsm pending-changes apply`.

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

View CSP report

To view CSP violations for a given viz, load the viz in a browser that includes developer tools. This example uses the Chrome browser.

1. Load a test viz with violations that is hosted on the Tableau Server deployment where you configured CSP.
2. Enter `CTRL+Shift+I` to open the developer tools in Chrome.
3. Click the **Network** tab.
4. In the **Filter** field, enter `csp-report`, and then click **Find All**.
 - If there are no violations then the search will not return any CSP reports.
 - If there are violations, click the Headers tab in the results pane and scroll to the bottom to view **Request Payload**.

SSL

SSL (Secure Sockets Layer) is a standard security technology that establishes an encrypted link between a web server and clients. To use SSL, you need to install an SSL certificate on Tableau Server.

You can configure Tableau Server to use SSL in the following ways:

- Use SSL for external HTTP traffic.
- Use mutual (two-way) SSL between clients (Tableau Desktop, web browsers, and `tabcmd.exe`) and Tableau Server.
- Use SSL for all HTTP traffic between internal server components and the repository.

If you are using mutual SSL, each client also needs a certificate.

Note: Tableau Server uses SSL for user authentication. SSL is not used to handle permissions and authorization to content (data sources and workbooks) hosted on Tableau Server.

Configure SSL for External HTTP Traffic to and from Tableau Server

You can configure Tableau Server to use Secure Sockets Layer (SSL) encrypted communications on all external HTTP traffic. Setting up SSL ensures that access to Tableau Server is secure and that sensitive information passed between the server and Tableau clients—such as Tableau Desktop, the REST API, analytics extensions, and so on—is protected. Steps on how to configure the server for SSL are described this topic; however, you must first acquire a certificate from a trusted authority, and then import the certificate files into Tableau Server.

For a basic primer about SSL and Tableau Server, see [Using SSL to encrypt Tableau Server communication](#) in the *Everybody's Install Guide*.

Mutual SSL authentication is not supported on Tableau Mobile.

SSL certificate requirements

Acquire an Apache SSL certificate from a trusted authority (for example, Verisign, Thawte, Comodo, GoDaddy). You can also use an internal certificate issued by your company. Wild-card certificates, which allow you to use SSL with many host names within the same domain, are also supported.

When you acquire an SSL certificate for external communication to and from Tableau Server, follow these guidelines and requirements:

- All certificate files must be valid PEM-encoded X509 certificates with the extension `.cert`.
- Use a SHA-2 (256 or 512 bit) SSL certificate. Most browsers no longer connect to a server that presents an SHA-1 certificate.
- In addition to the certificate file, you must also acquire a corresponding SSL certificate key file. The key file must be a valid RSA or DSA private key file (with the extension `.key` by convention).

You can choose to passphrase-protect the key file. The passphrase you enter during configuration will be encrypted while at rest. However, if you want to use the same certificate for SSL and SAML, you must use a key file that is *not* passphrase protected.

- SSL certificate chain file: A certificate chain file is required for Tableau Desktop on the Mac and for Tableau Prep Builder on the Mac and Tableau Prep Builder on Windows. The chain file is also required for the Tableau Mobile app if the certificate chain for Tableau Server is not trusted by the iOS or Android operating system on the mobile device.

The chain file is a concatenation of all of the certificates that form the certificate chain for the server certificate. All certificates in the file must be x509 PEM-encoded and the file must have a `.cert` extension (not `.pem`).

- For multiple sub-domains, Tableau Server supports wildcard certificates.
- Verify that the domain, host name, or IP address that clients use to connect to Tableau Server is included in the Subject Alternative Names (SAN) field. Many clients (Tableau Prep, Chrome and Firefox browsers, etc) require valid entry in the SAN field to establish a secure connection.

Note: If you plan to configure Tableau Server for single-sign on using SAML, see Using SSL certificate and key files for SAML in the SAML requirements to help determine whether to use the same certificate files for both SSL and SAML.

Configuring SSL for a Cluster

You can configure a Tableau Server cluster to use SSL. If the initial node is the only one running the gateway process (which it does by default), you need to configure SSL only on that node, using the steps described in this topic.

SSL with multiple gateways

A highly available Tableau Server cluster can include multiple gateways, fronted by a load balancer. If you are configuring this type of cluster for SSL, you have the following choices:

- **Configure the load balancer for SSL:** Traffic is encrypted from the client web browsers to the load balancer. Traffic from the load balancer to the Tableau Server gateway processes is not encrypted. No SSL configuration in Tableau Server is required by you. It's all handled by the load balancer.
- **Configure Tableau Server for SSL:** Traffic is encrypted from the client web browsers to the load balancer, and from the load balancer to the Tableau Server gateway processes. For more information, continue to the following section.

Additional configuration information for Tableau Server cluster environments

When you want to use SSL on all Tableau Server nodes that run a gateway process, you complete the following steps.

Tableau Server on Linux Administrator Guide

1. Configure the external load balancer for SSL passthrough.

Or if you want to use a port other than 443, you can configure the external load balancer to terminate the non-standard port from the client. In this scenario, you would then configure the load balancer to connect to Tableau Server over port 443. For assistance, refer to the documentation provided for the load balancer.

2. Make sure the SSL certificate is issued for the load balancer's host name.
3. Configure the initial Tableau Server node for SSL.
4. If you are using mutual SSL, upload the SSL CA certificate file. See `tsm authentication mutual-ssl <commands>`.

SSL certificate and key files will be distributed to each node as part of the configuration process.

Prepare the environment

When you get the certificate files from the CA, save them to a location accessible by Tableau Server, and note the names of the certificate `.crt` and `.key` files and the location where you save them. You will need to provide this information to Tableau Server when you enable SSL.

Configure SSL on Tableau Server

Use the method you're most comfortable with.

Use the TSM web interface

1. Open TSM in a browser:

`https://<tsm-computer-name>:8850`. For more information, see [Sign in to Tableau Services Manager Web UI](#).

2. On the **Configuration** tab, select **Security > External SSL**.

Note: If you are updating or changing an existing configuration, click **Reset** to clear the existing settings before proceeding.

3. Under **External web server SSL**, select **Enable SSL for server communication**.
4. Upload the certificate and key files, and if required for your environment, upload the chain file and enter the passphrase key:

External web server SSL
Configure SSL for secure communication between Tableau Server and web clients. [Learn more about external web server SSL.](#)

Enable SSL for server communication

SSL certificate file (Required)

SSL certificate key file (Required)

SSL certificate key passphrase

SSL certificate chain file

If you are using SSL for server communication and want to configure SSL communication between Tableau Server and clients using certificates on both the server and clients, you must first enable mutual SSL. [Click here to configure the mutual SSL authentication method.](#)

If you are running Tableau Server in a distributed deployment, then these files will be automatically distributed to each appropriate node in the cluster.

5. Click **Save Pending Changes**.
6. Click **Pending Changes** at the top of the page:



7. Click **Apply Changes and Restart**.

Use the TSM CLI

After you have copied the certificate files to the local computer, run the following commands:

Tableau Server on Linux Administrator Guide

```
tsm security external-ssl enable --cert-file <path-to-file.crt>  
--key-file <path-to-file.key>
```

```
tsm pending-changes apply
```

See the command reference at `tsm security external-ssl enable` to determine whether you want to include additional options for `external-ssl enable`. Tableau has specific recommendations for the `--protocols` option.

The `external-ssl enable` command imports the information from the `.crt` and `.key` files. If you run this command on a node in a Tableau Server cluster, it also distributes the information to any other gateway node.

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

Port redirection and logging

After the server has been configured for SSL, it accepts requests to the non-SSL port (default is port 80) and automatically redirects to the SSL port 443.

Note: Tableau Server supports only port 443 as the secure port. It cannot run on a computer where another application is using port 443.

SSL errors are logged in the at the following location. Use this log to troubleshoot validation and encryption issues:

```
/var/opt/tableau/tableau_server/data/tabsvc/logs/httpd/error.log
```

Add SSL port to the local firewall

If you are running a local firewall, you must add the SSL port to the firewall on Tableau Server. The example below describes how to configure the firewall running on RHEL/CentOS distributions. The example uses [Firewalld](#), which is the default firewall on CentOS.

1. Start firewalld:

```
sudo systemctl start firewalld
```

2. Add port 443 for SSL:

```
sudo firewall-cmd --permanent --add-port=443/tcp
```

3. Reload the firewall and verify the settings:

```
sudo firewall-cmd --reload
```

```
sudo firewall-cmd --list-all
```

Change or update SSL certificate

After you have configured SSL, you may need to periodically update the certificate. In some cases, you may need change the certificate for operational changes in your IT environment. In either case, you must use TSM to replace the SSL certificate that has already been configured for external SSL.

Do not copy a new certificate to the file directory on the operating system. Rather, when you add the certificate with either the TSM web UI or the `tsm security external-ssl enable` command, the certificate file is copied to the appropriate certificate store. In a distributed deployment, the certificate is also copied across the nodes in the cluster.

To change or update the SSL certificate (and the corresponding key file if required), follow the steps in the previous section of this topic, [Configure SSL on Tableau Server](#).

After you change the certificate, you must run `tsm pending-changes apply` to restart Tableau Server services. We also recommend restarting any other services on the computer that use the SSL certificate. If you are changing a root certificate on the operating system, you must reboot the computer.

Example: SSL Certificate - Generate a Key and CSR

Important: This example is intended to provide general guidance to IT professionals who are experienced with SSL requirements and configuration. The procedure described in this article is just one of many available methods you can use to generate the required files. The process described here should be treated as an example and not as a recommendation.

When you configure Tableau Server to use Secure Sockets Layer (SSL) encryption, this helps ensure that access to the server is secure and that data sent between Tableau Server and Tableau Desktop is protected.

Looking for Tableau Server on Windows? See [Example: SSL Certificate - Generate a Key and CSR](#).

Tableau Server uses Apache, which includes [OpenSSL](#). You can use the OpenSSL toolkit to generate a key file and Certificate Signing Request (CSR) which can then be used to obtain a signed SSL certificate.

Steps to generate a key and CSR

To configure Tableau Server to use SSL, you must have an SSL certificate. To obtain the SSL certificate, complete the steps:

1. [Generate a key file](#).
2. [Create a Certificate Signing Request \(CSR\)](#).
3. [Send the CSR to a certificate authority \(CA\) to obtain an SSL certificate](#).
4. [Use the key and certificate to configure Tableau Server to use SSL](#).

You can find additional information on the [SSL FAQ page](#) on the Apache Software Foundation website.

Configure a certificate for multiple domain names

Tableau Server allows SSL for multiple domains. To set up this environment, you need to modify the OpenSSL configuration file, `openssl.conf`, and configure a Subject Alternative Name (SAN) certificate on Tableau Server. See [For SAN certificates: modify the OpenSSL configuration file](#) below.

Generate a key

Generate a key file that you will use to generate a certificate signing request.

1. Run the following command to create the key file:

```
openssl genrsa -out <yourcertname>.key 4096
```

Note: This command uses a 4096-bit length for the key. You should choose a bit length that is at least 2048 bits because communication encrypted with a shorter bit length is less secure. If a value is not provided, 512 bits is used.

Create a certificate signing request to send to a certificate authority

Use the key file you created in the procedure above to generate the certificate signing request (CSR). You send the CSR to a certificate authority (CA) to obtain a signed certificate.

Important: If you want to configure a SAN certificate to use SSL for multiple domains, first complete the steps in [For SAN certificates: modify the OpenSSL configuration file](#) below, and then return to here to generate a CSR.

1. Run the following command to create a certificate signing request (CSR) file:

```
openssl req -new -key yourcertname.key -out yourcert-  
name.csr -config ../conf/openssl.cnf
```

2. When prompted, enter the required information.

Note: For **Common Name**, type the Tableau Server name. The Tableau Server name is the URL that will be used to reach the Tableau Server. For example, if you reach Tableau Server by typing `tableau.example.com` in the address bar of your browser, then `tableau.example.com` is the common name. If the common name does not resolve to the server name, errors will occur when a browser or Tableau Desktop tries to connect to Tableau Server.

Send the CSR to a certificate authority to obtain an SSL certificate

Send the CSR to a commercial certificate authority (CA) to request the digital certificate. For information, see the Wikipedia article [Certificate authority](#) and any related articles that help you decide which CA to use.

Use the key and certificate to configure Tableau Server

When you have both the key and the certificate from the CA, you can configure Tableau Server to use SSL. For the steps, see [Configure External SSL](#).

For SAN certificates: modify the OpenSSL configuration file

In a standard installation of OpenSSL, some features are not enabled by default. To use SSL with multiple domain names, before you generate the CSR, complete these steps to modify the `openssl.cnf` file.

1. Navigate to the Apache **conf** folder for Tableau Server.

For example: `/opt/tableau/tableau_server-
/packages/apache.<version_code>/conf`

2. Open **openssl.cnf** in a text editor, and find the following line: `req_extensions = v3_req`

This line might be commented out with a hash sign (#) at the beginning of the line.

```
UTF8Strings
# so use this option with caution!
string_mask = nombstr

# req_extensions = v3_req # The extensions to add to a
certificate request

[ req_distinguished_name ]
countryName = Country Name (2 letter code)
```

If the line is commented out, uncomment it by removing the **#** and **space** characters from the beginning of the line.

3. Move to the [**v3_req**] section of the file. The first few lines contain the following text:

```
# Extensions to add to a certificate request
basicConstraints = CA:FALSE
keyUsage = nonRepudiation, digitalSignature, keyEn-
cipherment
```

After the **keyUsage** line, insert the following line:

```
subjectAltName = @alt_names
```

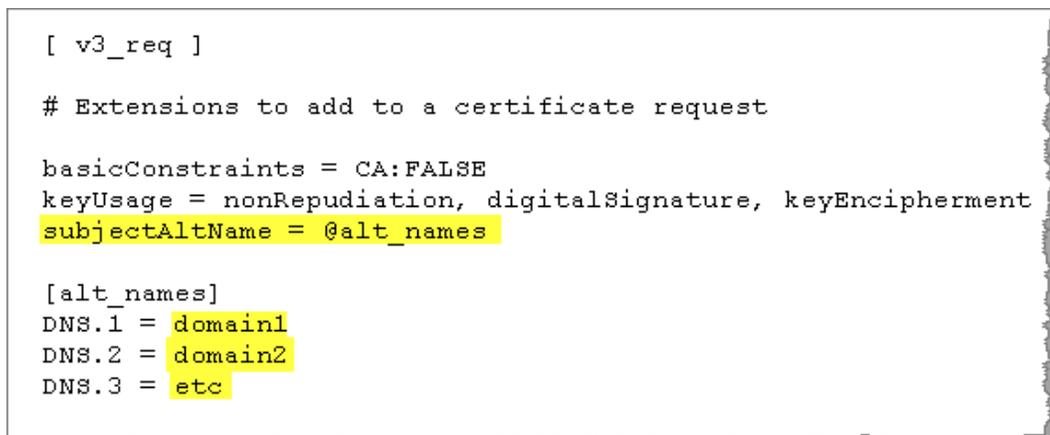
If you're creating a self-signed SAN certificate, do the following to give the certificate permission to sign the certificate:

Tableau Server on Linux Administrator Guide

- a. Add the `cRLSign` and `keyCertSign` to the **keyUsage** line so it looks like the following: `keyUsage = nonRepudiation, digitalSignature, keyEncipherment, cRLSign, keyCertSign`
 - b. After the **keyUsage** line, add the following line: `subjectAltName = @alt_names`
4. In the **[alt_names]** section, provide the domain names you want to use with SSL.

```
DNS.1 = [domain1]
DNS.2 = [domain2]
DNS.3 = [etc]
```

The following image shows the results highlighted, with placeholder text that you would replace with your domain names.



```
[ v3_req ]

# Extensions to add to a certificate request

basicConstraints = CA:FALSE
keyUsage = nonRepudiation, digitalSignature, keyEncipherment
subjectAltName = @alt_names

[alt_names]
DNS.1 = domain1
DNS.2 = domain2
DNS.3 = etc
```

5. Save and close the file.
6. Complete the steps in [Create a certificate signing request to send to a certificate authority](#) section, above.

Configure SSL for Internal Postgres Communication

You can configure Tableau Server to use SSL (TLS) for encrypted communication between the Postgres repository and other server components. By default, communication that is

internal to Tableau Server components is not encrypted.

While you enable support for internal SSL, you can also configure support for direct connections to the repository from Tableau clients, such as Tableau Desktop, Tableau Mobile, REST API, web browsers.

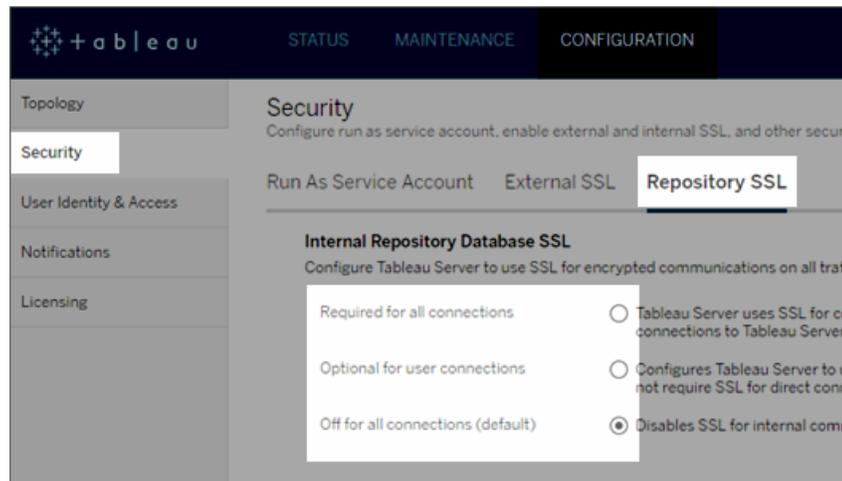
Use the TSM web interface

1. As a server administrator, open TSM in a browser:

`https://<tsm-computer-name>:8850`

For more information, see [Sign in to Tableau Services Manager Web UI](#).

2. On the **Configuration** tab, select **Security > Repository SSL**.



3. Select one of the options for using repository SSL.

- **Required for all connections**—uses SSL for internal Tableau Server communication, and requires SSL for Tableau clients and any external (non-Tableau) clients that connect directly to the postgres repository, including those using the **tableau** or **readonly** user.

Important: Unless you complete the steps in Configure Postgres SSL to Allow Direct Connections from Clients, to place the certificate files in the correct location on the client computers, Tableau clients and external postgres clients will not be able to validate the identity of the Tableau repository by comparing certificates on the client computers with the SSL certificate from the repository computer.

- **Optional for user connections**—When enabled, Tableau uses SSL for internal Tableau Server communication, and supports but does not require SSL for direct connections to the server from Tableau clients and external clients.
- **Off for all connections (default)**—Internal server communication is not encrypted, and SSL is not required for direct connections from clients.

4. Click **OK**.

The first two options generate the server's certificate files, **server.crt** and **server.key**, and place them in the following location.

```
/var/opt/tableau/tableau_server/data/tabsvc/config/pgsql_  
<version>/security
```

Use this .crt file if you need to configure clients for direct connections.

Use the TSM CLI

To enable SSL for internal traffic among the server components, run the following commands:

```
tsm security repository-ssl enable  
  
tsm pending-changes apply
```

What the command does

`repository-ssl enable` generates the server's certificate files, which it places in the following location:

```
/var/opt/tableau/tableau_server/data/tabsvc/config/pgsql_<version>/security
```

By default, this command sets Tableau Server to require SSL for traffic between the repository and other server components, as well as for direct connections from Tableau clients (including for connections through the **tableau** or **readonly** users).

To complete the configuration, you must also do the steps described in [Configure Postgres SSL to Allow Direct Connections from Clients](#), to place the certificate files in the correct location on the client computers.

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

Option for repository-ssl enable

If you want to require SSL only for internal Tableau Server communication, and not for direct connections from client apps, use the following option with the `repository-ssl enable` command:

```
--internal-only
```

Cluster environments

If you run `repository-ssl enable` on a node in a cluster, it copies the required certificate file to the same location on each other node.

For more information about downloading the public certificate for direct connections, see [Configure Postgres SSL to Allow Direct Connections from Clients](#).

Configure Postgres SSL to Allow Direct Connections from Clients

When Tableau Server is configured to use SSL for internal communication with the postgres repository, you can also require Tableau clients and external postgres clients that connect directly to the repository to verify the identity of the Tableau postgres repository by comparing the SSL certificate presented by the internal postgres instance with the certificate distributed to the Tableau or external postgres client.

Direct connections include those using the **tableau** user or the **readonly** user. Examples of Tableau clients include Tableau Desktop, Tableau Mobile, REST API, web browsers.

1. Enable internal SSL for the repository by running the following commands:

```
tsm security repository-ssl enable
```

```
tsm pending-changes apply
```

This enables internal SSL support and generates new server certificate and key files, and requires all Tableau clients to use SSL to connect to the repository. For additional repository-ssl commands and options, see [tsm security](#).

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see [tsm pending-changes apply](#).

2. (Optional) If you have configured your client computer to validate Postgres SSL connections, then you must import the certificate that is generated by Tableau Server onto the computers running Tableau Desktop. For each client computer that will connect directly to the repository, do the following:

- Copy the **server.crt** file to the client computer. You can find this file in the following directory:

```
/var/opt/tableau/tableau_server-  
/data/tabsvc/config/pgsql_<version_code>/security
```

Note: Do not copy **server.key** to the client computer. This file should reside only on the server.

- Import the certificate into the computer's certificate store.

For information, use the documentation from the operating system manufacturer.

3. (Optional) Configure any external (non-Tableau) postgres clients (PgAdmin or Dbeaver for example) to verify the identity of the Tableau Server postgres repository. Do this in the postgresql JDBC driver the client is using to connect by setting the "sslmode" directive to "verify-ca" or "verify-full". The options available may be different depending on the version of the postgres driver being used. For more information, see the drive documentation about SSL support.

Configure Mutual SSL Authentication

Using mutual SSL, you can provide users of Tableau Desktop, Tableau Mobile, and other approved Tableau clients a secure, direct-access experience to Tableau Server. With mutual SSL, when a client with a valid SSL certificate connects to Tableau Server, Tableau Server confirms the existence of the client certificate and authenticates the user, based on the user name in the client certificate. If the client does not have a valid SSL certificate, Tableau Server can refuse the connection.

You can also configure Tableau Server to fall back to username/password authentication if mutual SSL fails. Additionally, a user can log in using the REST API with a username and password (if one exists) whether or not fallback authentication is configured.

Tableau Server on Linux Administrator Guide

User authentication session time limits

When users log in with mutual SSL, the authentication session is governed by the same method that governs the Tableau Server global authentication session configuration.

For clients that connect to Tableau Server using a web browser, configuration of the global authentication session is described in the *Security Hardening Checklist*, see 9. Verify session lifetime configuration.

Sessions for connected clients (Tableau Desktop, Tableau Mobile, Tableau Prep Builder, and Bridge) use OAuth tokens to keep users logged in by re-establishing a session. By default, OAuth client tokens reset after a year. If a client token has not been used in 14 days, then it will expire. You can change these values by setting the `refresh_token.absolute_expiry_in_seconds` and `refresh_token.idle_expiry_in_seconds` options. See `tsm configuration set Options`.

Certificate usage

Before you enable and configure mutual SSL, you must configure external SSL. External SSL authenticates Tableau Server to the client and encrypts the session using the certificate and key that is required when you configure external SSL.

For mutual SSL, an additional certificate file is required. The file is a concatenation of CA certificate files. The file type must be `.cert`. A "CA" is a *certificate authority* that issues certificates to the client computers that will connect to Tableau Server. The action of uploading the CA certificate file establishes a trust, which enables Tableau Server to authenticate the individual certificates that are presented by the client computers.

As part of your disaster recovery plan, we recommend keeping a backup of the certificate and revocation (if applicable) files in a safe location off of the Tableau Server. The certificate and revocation files that you add to Tableau Server will be stored and distributed to other nodes by the Client File Service. However, the files are not stored in a recoverable format. See `Tableau Server Client File Service`.

RSA key and ECDSA curve sizes

The CA certificate used for mutual SSL must either have an RSA key strength of 2048, or ECDSA curve size of 256.

.You can configure Tableau Server to accept the less-secure sizes by setting the respective configuration keys:

- `ssl.client_certificate_login.min_allowed.rsa_key_size`
- `ssl.client_certificate_login.min_allowed.elliptic_curve_size`

See tsm configuration set Options.

Client certificate requirements

Users authenticating to Tableau Server with mutual SSL must present a client certificate that meets minimum security requirements.

Signing algorithm

Client certificates must use a SHA-256 or greater signing algorithm.

Tableau Server configured for mutual SSL authentication will block authentication of users with client certificates that use the SHA-1 signing algorithm.

Users who attempt to log in with SHA-1 client certificates encounter an "Unable to sign in" error, and the following error will be visible in the VizPortal logs:

```
Unsupported client certificate signature detected: [certificate  
Signature Algorithm name]
```

You can configure Tableau Server to accept the less secure SHA-1 signing algorithm by setting the `ssl.client_certificate_login.blocklisted_signature_algorithms` tsm configuration option.

RSA key and ECDSA curve sizes

The client certificate used for mutual SSL must either have an RSA key strength of 2048, or ECDSA curve size of 256.

Tableau Server on Linux Administrator Guide

Tableau Server will fail mutual authentication requests from client certificates that do not meet these requirements. You can configure Tableau Server to accept the less-secure sizes by setting the respective configuration keys:

- `ssl.client_certificate_login.min_allowed.rsa_key_size`
- `ssl.client_certificate_login.min_allowed.elliptic_curve_size`

See `tsm configuration set Options`.

Use the TSM web interface

1. Configure SSL for External HTTP Traffic to and from Tableau Server.
2. Open TSM in a browser:

`https://<tsm-computer-name>:8850`. For more information, see [Sign in to Tableau Services Manager Web UI](#).
3. On the **Configuration** tab, select **User Identity & Access > Authentication Method**.
4. Under **Authentication Method**, select **Mutual SSL** in the drop-down menu.
5. Under Mutual SSL, select **Use mutual SSL and automatic sign in with client certificates**.
6. Click **Select File** and upload your certificate authority (CA) certificate file to the server.

The file (.crt) is an all-in-one file that includes certificates of CAs that are used for client authentication. The file you upload must be a concatenation of the various PEM-encoded certificate files, in order of preference.

7. Enter remaining SSL configuration information for your organization.

Username format: When Tableau Server is configured for mutual SSL, the server gets the user name from the client certificate, so it can establish a direct sign-in for the

client user. The name that Tableau Server uses depends on how Tableau Server is configured for user authentication:

- Local Authentication—Tableau Server uses the UPN (User Principal Name) from the certificate.
- Active Directory (AD)—Tableau Server uses LDAP (Lightweight Directory Access Protocol) to get the user name.

Alternatively, you can set Tableau Server to use the CN (Common Name) from the client certificate.

Authentication Method

Specify how Active Directory manages user authentication and access to Tableau Server.

Mutual SSL ▼

Mutual SSL

Use mutual SSL for secure communication between Tableau Server and web clients and for automatic sign-in across all Tableau Server components. [Learn more](#)

Use mutual SSL and automatic sign in with client certificates

SSL CA certificate file

Use username and password if SSL authentication fails

Specify a method for retrieving the username from the certificate.

Username retrieval method

LDAP (Lightweight Directory Access Protocol)

UPN (User Principal Name)

CN (Common Name)

8. Click **Save Pending Changes** after you've entered your configuration information.

9. Click **Pending Changes** at the top of the page:



10. Click **Apply Changes and Restart**.

Use the TSM CLI

Step 1: Require SSL for external server communication

To configure Tableau Server to use SSL for external communication between Tableau Server and web clients, run the `external-ssl enable` command as follows, providing the names for the server certificate's `.crt` and `.key` files:

```
tsm security external-ssl enable --cert-file <file.crt> --key-file <file.key>
```

- For `--cert-file` and `--key-file`, specify the location and file name where you saved the server's CA-issued SSL certificate (`.crt`) and key (`.key`) files.
- The above command assumes the you are signed in as a user that has the **Server Administrator** site role on Tableau Server. You can instead use the `-u` and `-p` parameters to specify an administrator user and password.
- If the certificate key file requires a passphrase, include the `--passphrase` parameter and value.

Step 2: Configure and enable mutual SSL

Add mutual authentication between the server and each client, and allow for Tableau client users to be authenticated directly after the first time they provide their credentials.

1. Run the following command:

```
tsm authentication mutual-ssl configure --cert-file <file.crt>
```

For `--cert-file`, specify the location and file name of the certificate authority (CA) certificate file.

The file (.crt) is an all-in-one file that includes certificates of CAs that are used for client authentication. The file you upload must be a concatenation of the various PEM-encoded certificate files, in order of preference.

2. Run the following commands to enable mutual SSL and apply the changes:

```
tsm authentication mutual-ssl enable
```

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

Additional options for mutual SSL

You can use `mutual-ssl configure` to configure Tableau Server to support the following options.

For more information, see [tsm authentication mutual-ssl <commands>](#).

Fallback authentication

When Tableau Server is configured for mutual SSL, authentication is automatic and clients must have a valid certificate. You can configure Tableau Server to allow a fallback option, to accept user name and password authentication.

```
tsm authentication mutual-ssl configure -fb true
```

Tableau Server accepts username and password authentication from REST API clients, even if the above option is set to `false`.

User name mapping

When Tableau Server is configured for mutual SSL, the server authenticates the user directly by getting the user name from their client certificate. The name that Tableau Server uses depends on how the server is configured for user authentication:

- **Local Authentication**—uses the UPN (User Principal Name) from the certificate.
- **Active Directory (AD)**—uses LDAP (Lightweight Directory Access Protocol) to get the user name.

You can override either of these defaults to set Tableau Server to use the common name.

```
tsm authentication mutual-ssl configure -m cn
```

For more information, see [Mapping a Client Certificate to a User During Mutual Authentication](#)

Certificate Revocation List (CRL)

You might need to specify a CRL if you suspect that a private key has been compromised, or if a certificate authority (CA) did not issue a certificate properly.

```
tsm authentication mutual-ssl configure -rf <revoke-file.pem>
```

Mapping a Client Certificate to a User During Mutual Authentication

When you use mutual (two-way) SSL authentication, the client presents its certificate to Tableau Server as part of the authentication process. Tableau Server then maps user information in the client certificate to a known user identity. The strategy that Tableau Server uses to perform client mapping depends on the content of your organization's client certificates.

This topic discusses the ways information in a client certificate can map to a user identity and how to change the way Tableau Server performs that mapping. To understand how the mapping happens and whether you need to change it, you must know how client certificates are structured in your organization.

- [User-name mapping options](#)
- [Change the certificate mapping](#)
- [Address user-name ambiguity in multi-domain organizations](#)

User-name mapping options

Tableau Server uses one of the following approaches to map a client certificate to a user identity:

- **Active Directory.** If Tableau Server is configured to use Active Directory for user authentication, when Tableau Server receives a client certificate, it passes the certificate to Active Directory, which maps the certificate to an Active Directory identity. Any explicit user name information in the certificate is ignored.

Note: This approach requires client certificates to be published for the user accounts in Active Directory.

- **User principal name (UPN).** A client certificate can be configured to store the user name in the user principal name field. Tableau Server reads the UPN value and maps it to a user in Active Directory or to a local user.
- **Common name (CN).** A client certificate can be configured to store the user name in the common name field of the certificate. Tableau Server reads the CN value and maps it to a user in Active Directory or to a local user.

If you configure the server for Active Directory authentication and UPN or CN user-name mapping, put the user name in one of the following formats:

`username, domain/username, or username@domain.`

For example: `jsmith, example.org/jsmith, or jsmith@example.org.`

If the server uses local authentication, the format of the name in the UPN or CN fields is not predetermined, but the name in the field must match a user name on the server.

Tableau Server on Linux Administrator Guide

Change the certificate mapping

You use the `tsm authentication mutual-ssl <commands>` commands to map a client certificate to a user identity in Tableau Server:

```
tsm authentication mutual-ssl configure -m <value>
```

Possible values are `ldap` for Active Directory mapping, `upn` for UPN mapping, or `cn` for CN mapping.

When you first install and configure Tableau Server, the server sets the default user-name mapping to match the server's authentication type:

- If the server is configured to use Active Directory, it also uses Active Directory for mapping the certificate to the user identity.
- If the server is configured to use local authentication, the server gets the user-name value from the UPN field in the certificate.

If the default behavior for how Tableau Server maps a user name to an identity is not correct for your server configuration, run the following set of commands to change the mapping to use the CN value:

```
tsm authentication mutual-ssl configure -m cn
```

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

Address user-name mapping ambiguity in multi-domain organizations

Under some circumstances, the user name in a certificate's UPN or CN field can be ambiguous. This ambiguity can lead to unexpected results when the user name is mapped to a user identity on the server.

For example, if Tableau Server is presented with a user name that does not include a domain, the server maps the user name to an identity using the default domain. This can cause an incorrect user-name mapping, potentially assigning a user a different user's identity and permissions.

This can occur particularly in environments where the following conditions apply:

- Your organization supports multiple Active Directory domains.
- The server is configured to use Active Directory authentication.
- The server is configured to use UPN or CN mapping.
- Some users have the same user name but different domains. For example, `jsmith@example.org` and `jsmith@example.com`.
- The user name in the certificate's UPN or CN fields does not include the domain as part of the user name—for example, it shows `jsmith`.

To avoid incorrect user-name mapping, make sure the client certificates include fully qualified user names with the domain, using the format `jsmith@example.org` or `example.org/jsmith`.

Configure Encrypted Channel to LDAP External Identity Store

Tableau Server that is configured to connect to an external LDAP identity store must query the LDAP directory and establish a session. The process of establishing a session is called *binding*. There are multiple ways to bind. Tableau Server supports two methods of binding to an LDAP directory:

- **Simple bind:** Establishes a session by authenticating with a username and password. By default, Tableau Server will attempt StartTLS to encrypt sessions when connecting to Windows Active Directory. If Tableau Server has a valid TLS certificate, then the session will be encrypted. Otherwise, LDAP with simple bind is not encrypted. If you are configuring LDAP with simple bind, we strongly recommend that you enable LDAP over SSL/TLS.
- **GSSAPI bind.** GSSAPI uses Kerberos to authenticate. When configured with a keytab file, authentication is secure during GSSAPI bind. However, subsequent traffic to the LDAP server is not encrypted. We recommend configuring LDAP over SSL/TLS . StartTLS is not supported for GSSAPI bind with Active Directory.

If you are running Tableau Server on Linux on a computer that is joined to an Active Directory domain, you can configure GSAPI. See LDAP with GSSAPI (Kerberos) bind.

This topic describes how to encrypt the channel for simple LDAP bind for communications between Tableau Server and LDAP directory servers.

Certificate requirements

You must have a valid SSL/TLS certificate that can be used for encryption. The certificate must be installed on the computer(s) running Tableau Server.

The certificate you install must include `Key Encipherment` in the key usage field to be used for SSL/TLS. Tableau Server will only use this certificate for encrypting the channel to the LDAP server. The expiry, trust, and CRL and other attributes are not validated.

Note: If you are running Tableau Server in a distributed deployment, then you must manually copy the SSL certificate to each node in the cluster. Copy the certificate only to those nodes where the Tableau Server Application Server process is configured. Unlike other shared files in a cluster environment, the SSL certificate used for LDAP will not be automatically distributed by the Client File Service.

Importing certificate into the Tableau keystore

If you do not have certificates already in place on your computer that are configured for the LDAP server then you must obtain a SSL certificate for the LDAP server and import it into the Tableau system keystore.

Use the "keytool" Java tool to import certificates. In a default installation, this tool is installed with Tableau Server at `/opt/tableau/tableau_server-
/packages/repository.<installer version>/jre/bin/keytool`.

The following command (for RHEL-like distributions) imports the certificate:

```
sudo "$PROGRAMFOLDER"/packages/repository*/jre/bin/keytool -
importcert -file "$CERTSDIR" -alias "$OPENLDAPSSLSERVER" -key-
store /etc/opt/tableau/tableau_server-
/tableauservicesmanagerca.jks -storepass changeit -noprompt
```

The password for the Java keystore is `changeit`. (Do not change the password for the Java keystore).

Encryption methods

Tableau Server 2021.1 and newer supports two methods for encrypting the LDAP channel for simple bind:

- **StartTLS:** This is the default configuration for communicating with Active Directory in Tableau Server 2021.2. Beginning with Tableau Server 2021.2, TLS is enforced for simple bind LDAP connections to Active Directory. This default TLS configuration is enforced for both new installs and for upgrade scenarios.

Note: StartTLS is only supported on Tableau Server on Linux when communicating with Active Directory and simple bind. StartTLS is not supported for communication with other LDAP server types or with GSSAPI.

Tableau Server on Linux Administrator Guide

The StartTLS method works by establishing an insecure connection with the Active Directory server. After a client-server negotiation, the connection is upgraded to a TLS encrypted connection. As the default configuration, this scenario only requires a valid TLS certificate on Tableau Server. No other configuration is required.

- LDAPS: Secure LDAP, or LDAPS, is a standard encrypted channel that requires more configuration. Specifically, in addition to a TLS certificate on Tableau Server, you must set the host name and the secure LDAP port for the target LDAP server.

LDAPS is supported on any LDAP server, including Active Directory servers.

Configuring encrypted channel for simple bind

This section describes how to configure Tableau Server to use an encrypted channel for LDAP simple bind.

New installations of Tableau Server

If your organization uses an LDAP directory other than Active Directory, then you cannot use the TSM GUI Setup to configure the identity store as part of Tableau Server installation. Instead, you must use JSON entity files to configure the LDAP identity store. See [identityStore Entity](#),

Before you configure the `identityStore` entity, import a valid SSL/TLS certificate into the Tableau key store as documented earlier in this topic.

Configuring LDAPS requires setting the `hostname` and `sslPort` options in the `identityStore` JSON file.

New installations in an Active Directory environment

If you are using Active Directory as an external identity store, you must run the GUI version of Tableau Server Setup. Unlike the CLI process for installing Tableau Server, the GUI version of Setup includes logic to simplify and validate Active Directory configuration.

The Tableau Server Setup GUI where you configure Active Directory is shown here.

Identity Store

You cannot change the identity store after initializing.

Local
 Active Directory

Domain	NetBIOS (Nickname)
<input type="text" value="example.lan"/>	<input type="text" value="example"/>

Hostname	Port
<input type="text" value="Hostname"/>	<input type="text" value="Port"/>

Specify and configure the encryption method Tableau Server will use to communicate with Active Directory. Encrypted communication (TLS/SSL) requires a valid certificate in the Tableau certificate store.

To use LDAPS, you must specify a hostname and port.

LDAP over StartTLS (encrypted channel)
 LDAPS (encrypted channel)
 LDAP (unencrypted channel)

Tableau Server requires read access to Active Directory. Specify how Tableau Server will authenticate with Active Directory.

LDAP simple bind
 LDAP GSSAPI bind

Username	Password
<input type="text" value="Username"/>	<input type="text" value="Password"/>

If you are installing a new instance of Tableau Server on Linux and you have a valid SSL/TLS certificate installed in the Tableau keystore, we recommend that you leave the default option set to StartTLS.

If you want to configure for LDAPS, then enter the hostname and secure port (typically 636) for the LDAP server, before selecting the LDAPS option.

You can make changes to these configurations after you install by signing into TSM Web UI, clicking the **Configuration** tab, **User Identity & Access**, and then **Identity Store**.

Upgrade scenarios

If you are upgrading to a 2021.2 (or newer) version of Tableau Server and using Active Directory as your external identity store, then the encrypted channel is enforced for LDAP simple bind connections. If you do not have an encrypted channel configured, then upgrade will fail.

To successfully upgrade to version 2021.2 or newer, one of the following must be true:

- The existing Tableau Server installation has already been configured for LDAPS and includes a certificate in the Tableau key store.
- A valid SSL/TLS certificate is present in the Tableau key store prior to upgrading. In this scenario, the default StartTLS configuration will enable an encrypted channel.
- The encrypted LDAP channel has been disabled as described in the following section.

Disable default encrypted LDAP channel

If you are running Tableau Server on Linux and connecting to Active Directory, you can disable the encrypted channel requirement.

When disabled, user credentials that are used to establish the bind session with Active Directory are communicated in plaintext between Tableau Server and the Active Directory server.

Disable new installation

If you will be using Active Directory as your identity store, then you must use the TSM GUI to configure the Active Directory connection. See [Configure Initial Node Settings](#).

Identity Store

You cannot change the identity store after initializing.

Local
 Active Directory

Domain	NetBIOS (Nickname)
<input type="text" value="example.lan"/>	<input type="text" value="example"/>

Hostname	Port
<input type="text" value="Hostname"/>	<input type="text" value="Port"/>

Specify and configure the encryption method Tableau Server will use to communicate with Active Directory. Encrypted communication (TLS/SSL) requires a valid certificate in the Tableau certificate store.

To use LDAPS, you must specify a hostname and port.

LDAP over StartTLS (encrypted channel)
 LDAPS (encrypted channel)
 LDAP (unencrypted channel)

Tableau Server requires read access to Active Directory. Specify how Tableau Server will authenticate with Active Directory.

LDAP simple bind
 LDAP GSSAPI bind

Username	Password
<input type="text" value="Username"/>	<input type="text" value="Password"/>

Select **LDAP (unencrypted channel)** when running Setup.

Disable before upgrading

If you are upgrading to Tableau Server 2021.2 (or newer) from an earlier version, run the following commands on earlier version of Tableau Server before you upgrade:

Tableau Server on Linux Administrator Guide

```
tsm configuration set -k wgserver.domain.ldap.starttls.enabled -  
v false --force-keys  
  
tsm pending-changes apply
```

To verify that the key has been set, run the following command:

```
tsm configuration get -k wgserver.domain.ldap.starttls.enabled
```

The command should return `false`.

Error messages

The following error messages may be displayed or logged. If you see these errors

- Verify that your certificate is valid and imported to the Tableau key store as described earlier in this topic.
- (LDAPS only) - Verify that the host and port name is correct.

Setup GUI

The following error will be displayed if you have misconfigured LDAPS or StartTLS when running the Setup or Upgrade GUI.

```
TLS handshake failed. Tableau Server and the Active Directory  
server could not negotiate a compatible level of security.
```

Vizportal logs

If you are configuring LDAPS or StartTLS using CLI, the following error message will not be displayed. Rather, the error will be logged in the vizportal logs at `/var/opt/tableau/tableau_server/data/tabsvc/logs/vizportal`.

```
Authentication with LDAP server failed. The provided credentials  
or configuration are either incorrect or do not have the neces-  
sary permissions to bind.
```

System User, sudo Privileges, and systemd

This topic describes system user, systemd user service, and sudo privilege in the context of Tableau Server.

Privilege separation

Following standard security best practices, Tableau Server for Linux runs processes with the least privilege possible. During installation, the unprivileged user, *tableau*, is created in a server authorized group (*tableau*).

An example user entry in the `/etc/passwd` file is as follows:

- `tableau:x:993:991:Tableau Server:/var/opt/tableau/tableau_server:/bin/bash`

All processes run as the unprivileged *tableau* user. This means that if one of the Tableau Server processes (such as a process displaying vizviews to users) were compromised in some fashion, it would only be able to impact Tableau Server, not the rest of the Linux system. For this reason, you should not add the `tableau` unprivileged user to the `tmsadmin` group. The `tmsadmin` group should only contain accounts that require authorization to access OS-related Tableau configurations.

The `tableau` user and `tmsadmin` group are created by the Tableau Server initialization process. You can specify a different unprivileged user or TSM authorization group during installation. For more information about system users and groups, in the context of installation and LDAP configuration, see Identity Store.

sudo privileges

The first version (10.5) of Tableau Server on Linux relied on sudo privileges by updating the sudoers file. Updating the sudoers file conflicts with some system management configuration best practices and security policies. Therefore, the 2018.1 version (and later) of Tableau Server no longer creates or uses a privileged user (`tmsagent`). Nor does the current version of Tableau Server update or include a Tableau-specific sudoers file.

All privileged operations now occur during package and software installation.

systemd user service

In the 10.5 version of Tableau Server on Linux, sudo privileges were required to modify or restart the TSM services, which required systemctl commands. All TSM services were run from the normal system-wide systemd process (`process ID 1`, which runs all processes on the operating system). In this scheme, systemd process runs as root. Therefore, the 10.5 version of Tableau Server required sudo privileges.

With the current 2018.1 (and later) releases, we have removed the need for sudo privileges by making use of the systemd capability to run as a user service. The systemd user service runs as a normal user, so it does not need any special privileges once it has been enabled.

In normal use cases, you will not need to issue commands to systemd because TSM takes care of that. However, for troubleshooting scenarios, you may need to interact with the TSM services. As with the previous versions, you will issue the same systemctl commands for these scenarios. However, commands should be run as the `tableau` user, and not as root. If you specified a different unprivileged system user during Tableau Server setup, then run the commands as that user.

Running systemctl commands

Use the following syntax example to issue request to systemd with the systemctl commands.

Start a session as the unprivileged user. The `-l` flag is critical to set environment variables properly.

```
sudo su -l tableau
```

Then issue commands. For example:

```
systemctl --user status tabadmincontroller_0
```

```
systemctl --user restart tabadmincontroller_0
```

Security Hardening Checklist

The following list provides recommendations for improving the security ("hardening") of your Tableau Server installation.

Looking for Tableau Server on Windows? See [Security Hardening Checklist](#).

Installing security updates

Security updates are included in the latest versions and maintenance releases (MR) of Tableau Server. You cannot install security updates as patches. Rather, you must upgrade to a current version or MR to update Tableau Server with the latest security fixes.

Always reference the most current version of this topic after upgrading. The current version includes `/current/` in the topic URL.

For example, the US version URL is: https://help.tableau.com/current/server/en-us/security_harden.htm.

1. Update to the current version

We recommend that you always run the latest version of Tableau Server. Additionally, Tableau periodically publishes maintenance releases of Tableau Server that include fixes for known security vulnerabilities. (Information regarding known security vulnerabilities can be found on the Tableau [Security Bulletins](#) page and the [Salesforce Security Advisories](#) page.) We recommend that you review maintenance release notifications to determine whether you should install them.

To get the latest version or maintenance release of Tableau Server, visit the [Customer Portal](#) page.

2. Configure SSL/TLS with a valid, trusted certificate

Secure Sockets Layer (SSL/TLS) is essential for helping to protect the security of communications with Tableau Server. Configure Tableau Server with a valid, trusted certificate (not a self-signed certificate) so that Tableau Desktop, mobile devices, and web clients can connect to the server over a secured connection. For more information, see [SSL](#).

3. Disable older versions of TLS

Tableau Server uses TLS to authenticate and encrypt many connections between components and with external clients. External clients, such as browsers, Tableau Desktop, Tableau Mobile connect to Tableau using TLS over HTTPS. Transport layer security (TLS) is an improved version of SSL. In fact, older versions of SSL (SSL v2 and SSL v3) are no longer considered to be adequately secure communication standards. As a result, Tableau Server does not allow external clients to use SSL v2 or SSL v3 protocols to connect.

We recommend that you allow external clients to connect to Tableau Server with TLS v1.3 and TLS v1.2.

TLS v1.2 is still regarded as a secure protocol and many clients (including Tableau Desktop) do not yet support TLS v1.3.

TLS v1.3 capable clients will negotiate TLS v1.3 even if TLS v1.2 is supported by the server.

The following `tsm` command enables TLS v1.2 and v1.3 (using the "all" parameter) and disables SSL v2, SSL v3, TLS v1, and TLS v1.1 (by prepending the minus [-] character to a given protocol). TLS v1.3 is not yet supported by all components of Tableau Server.

```
tsm configuration set -k ssl.protocols -v 'all -SSLv2 -SSLv3 -  
TLSv1 -TLSv1.1'
```

```
tsm pending-changes apply
```

You can also modify the default list of cipher suites that Tableau Server uses for SSL/TLS sessions. For more information see the *ssl.ciphersuite* section at `tsm configuration set Options`.

4. Configure SSL encryption for internal traffic

Configure Tableau Server to use SSL to encrypt all traffic between the Postgres repository and other server components. By default, SSL is disabled for communications between server components and the repository. We recommend enabling internal SSL for all instances of Tableau Server, even single-server installations. Enabling internal SSL is especially important for multi-node deployments. See [Configure SSL for Internal Postgres Communication](#).

5. Enable firewall protection

Tableau Server was designed to operate inside a protected internal network.

Important: Do not run Tableau Server, or any components of Tableau Server on the internet or in a DMZ. Tableau Server must be run within the corporate network protected by an internet firewall. We recommend configuring a reverse proxy solution for internet clients that need to connect to Tableau Server. See [Configuring Proxies for Tableau Server](#).

A local firewall should be enabled on the operating system to protect Tableau Server in single and multi-node deployments. In a distributed (multi-node) installation of Tableau Server, communication between nodes does not use secure communication. Therefore, you should enable firewalls on the computers that host Tableau Server. See [Configure Local Firewall](#).

To prevent a passive attacker from observing communications between nodes, configure a segregated virtual LAN or other network layer security solution.

See [Tableau Services Manager Ports](#) to understand which ports and services Tableau Server requires.

6. Restrict access to the server computer and to important directories

Tableau Server configuration files and log files can contain information that is valuable to an attacker. Therefore, restrict physical access to the machine that is running Tableau Server. In addition, make sure that only authorized and trusted users have access to the Tableau Server files in the `/var/opt/tableau/tableau_server/` directory.

7. Generate fresh secrets and tokens

Any Tableau Server service that communicates with repository or the cache server must first authenticate with a secret token. The secret token is generated during Tableau Server setup. The encryption key that internal SSL uses to encrypt traffic to Postgres repository is also generated at during setup.

We recommend that after you install Tableau Server, you generate new encryption keys for your deployment.

These security assets can be regenerated with the `tsm security regenerate-internal-tokens` command.

Run the following commands:

```
tsm security regenerate-internal-tokens
```

```
tsm pending-changes apply
```

8. Disable services that you're not using

To minimize the attack surface of the Tableau Server, disable any connection points that are not needed.

JMX Service

JMX is disabled by default. If it's enabled but you're not using it, you should disable it by using the following:

```
tsm configuration set -k service.jmx_enabled -v false

tsm pending-changes apply
```

9. Verify session lifetime configuration

By default, Tableau Server does not have an absolute session timeout. This means that browser-based client (Web authoring) sessions can remain open indefinitely if the Tableau Server inactivity timeout is not exceeded. The default inactivity timeout is 240 minutes.

If your security policy requires it, you can set an absolute session timeout. Be sure to set your absolute session timeout in a range that allows the longest-running extract uploads or workbook publishing operations in your organization. Setting the session timeout too low may result in extract and publishing failures for long-running operations.

To set the session timeout run the following commands:

```
tsm configuration set -k wgserver.session.apply_lifetime_limit
-v true
```

```
tsm configuration set -k wgserver.session.lifetime_limit -v
value, where value is the number of minutes. The default is 1440, which is 24 hours.
```

```
tsm configuration set -k wgserver.session.idle_limit -v value,
where value is the number of minutes. The default is 240.
```

```
tsm pending-changes apply
```

Sessions for connected clients (Tableau Desktop, Tableau Mobile, Tableau Prep Builder, Bridge, and personal access tokens) use OAuth tokens to keep users logged in by re-establishing a session. You can disable this behavior if you want all Tableau client sessions to be

solely governed by the browser-based session limits controlled by the commands above.
See [Disable Automatic Client Authentication](#).

10. Configure a server allowlist for file-based data sources

By default, Tableau Server allows authorized Tableau Server users to build workbooks that use files on the server as file-based data sources (such as spreadsheets). In this scenario, files are accessed by the `tableau` system account.

To prevent unwanted access to files, we recommend that you configure allowlist functionality. This lets you limit `tableau` account access to just the directory paths where you host data files.

1. On the computer running Tableau Server, identify the directories where you will host data source files.

Important Make sure the file paths you specify in this procedure exist on the server. If the paths do not exist when the computer starts, Tableau Server will not start.

2. Run the following commands:

```
tsm configuration set -k native_api.allowed_paths -v "path"  
, where path is the directory to add to the allowlist. All subdirectories of the specified  
path will be added to the allowlist. If you want to specify multiple paths, separate them  
with a semicolon, as in this example:
```

```
tsm configuration set -k native_api.allowed_paths -v "/data-  
sources;/HR/data"
```

```
tsm pending-changes apply
```

11. Enable HTTP Strict Transport Security for web browser clients

HTTP Strict Transport Security (HSTS) is a policy configured on web application services, such as Tableau Server. When a conforming browser encounters a web application running HSTS, then all communications with the service must be over a secured (HTTPS) connection. HSTS is supported by major browsers.

For more information about how HSTS works and the browsers that support it, see The Open Web Application Security Project web page, [HTTP Strict Transport Security Cheat Sheet](#).

To enable HSTS, run the following commands on Tableau Server:

```
tsm configuration set -k gateway.http.hsts -v true
```

By default, HSTS policy is set for one year (31536000 seconds). This time period specifies the amount of time in which the browser will access the server over HTTPS. You should consider setting a short max-age during initial roll-out of HSTS. To change this time period, run `tsm configuration set -k gateway.http.hsts_options -v max-age=<seconds>`. For example, to set HSTS policy time period to 30 days, enter `tsm configuration set -k gateway.http.hsts_options -v max-age=2592000`.

```
tsm pending-changes apply
```

12. Disable Guest access

Core-based licenses of Tableau Server include a Guest user option, which allows any user in your organization to see and interact with Tableau views embedded in web pages.

Guest user access is enabled by default on Tableau Servers deployed with core-based licensing.

Guest access allows users to see embedded views. The Guest user cannot browse the Tableau Server interface or see server interface elements in the view, such as user name, account settings, comments, and so on.

If your organization has deployed Tableau Server with core licensing and Guest access is not required, then disable Guest access.

You can disable Guest access at the server or site level.

You must be a server administrator to disable the Guest account at either the server or the site level.

To disable Guest access at the server level:

1. In the site menu, click **Manage All Sites** and then click **Settings > General**.
2. For **Guest Access**, clear the **Enable Guest account** check box.
3. Click **Save**.

To disable Guest access for a site:

1. In the site menu, select a site.
2. Click **Settings**, and on the Settings page, clear the **Enable Guest account** check box.

For more information, see Guest User.

13. Set referrer-policy HTTP header to 'same-origin'

Beginning in 2019.2, Tableau Server includes the ability to configure Referrer-Policy HTTP header behavior. This policy is enabled with a default behavior that will include the origin URL for all "secure as" connections (`no-referrer-when-downgrade`), which sends origin referrer information only to like connections (HTTP to HTTP) or those that are more secure (HTTP to HTTPS).

However, we recommend setting this value to `same-origin`, which only sends referrer information to same-site origins. Requests from outside the site will not receive referrer information.

To update the `referrer-policy` to `same-origin`, run the following commands:

```
tsm configuration set -k gateway.http.referrer_policy -v same-origin
```

```
tsm pending-changes apply
```

For more information about configuring additional headers to improve security, see [HTTP Response Headers](#).

14. Configure TLS for SMTP connection

Beginning in 2019.4, Tableau Server includes the ability to configure TLS for the SMTP connection. Tableau Server only supports STARTTLS (Opportunistic or Explicit TLS).

Tableau Server can be optionally configured to connect to a mail server. After configuring SMTP, Tableau Server can be configured to email server administrators about system failures, and email server users about subscribed views and data-driven alerts.

To configure TLS for SMTP:

1. Upload a compatible certificate to Tableau Server. See `tsm security custom-cert add`.
2. Configure TLS connection using TSM CLI.

Run the following TSM commands to enable and force TLS connections to the SMTP server and to enable certificate verification.

```
tsm configuration set -k svcmonitor.notification.smtp.ssl_enabled -v true
```

```
tsm configuration set -k svcmonitor.notification.smtp.ssl_required -v true
```

Tableau Server on Linux Administrator Guide

```
tsm configuration set -k svcmonitor.notification.smtp.ssl_
check_server_identity -v true
```

By default, Tableau Server will support TLS versions 1, 1.1, and 1.2, but we recommend that you specify the highest TLS version that the SMTP server supports.

Run the following command to set the version. Valid values are `SSLv2Hello`, `SSLv3`, `TLSv1`, `TLSv1.1`, and `TLSv1.2`. The following example sets the TLS version to version 1.2.:

```
tsm configuration set -k svcmonitor.notification.smtp.ssl_
versions -v "TLSv1.2"
```

For more information about other TLS configuration options, see [Configure SMTP Setup](#).

3. Restart Tableau Server to apply changes. Run the following command:

```
tsm pending-changes apply
```

15. Configure SSL for LDAP

If your Tableau Server deployment is configured to use a generic LDAP external identity store, we recommend configuring SSL to protect authentication between Tableau Server and your LDAP server. See [Configure Encrypted Channel to LDAP External Identity Store](#).

If your Tableau Server deployment is configured to use Active Directory, we recommend enabling Kerberos to protect authentication traffic. See [Kerberos](#).

Change List

Date	Change
May 2018	Added clarification: Do not disable REST API in organizations that are running Tableau Prep.
May 2019	Added recommendation for referrer-policy HTTP header.

June 2019	Removed recommendation to disable Triple-DES. As of version 2019.3, Triple-DES is no longer a default supported cipher for SSL. See What's Changed - Things to Know Before You Upgrade .
January 2020	Added recommendation to configure TLS for SMTP.
February 2020	Added recommendation to configure SSL for LDAP server.
May 2020	Added TLS v1.3 to the disabled list of TLS ciphers. Added clarification to introduction about topic versioning.
October 2020	Added TLS v1.3 as a default supported cipher.
January 2021	Added clarification: All products enabled by the Data Management license require REST API.
February 2021	Removed recommendation to disable REST API. The API is now used internally by Tableau Server and disabling it may limit functionality.

Manage Licenses

You can manage your Tableau Server licenses and view license usage.

Licensing Overview

An important administrative role in a Tableau Server deployment is the Tableau portal administrator. The portal administrator manages licensing and the associated keys for the Tableau deployment. As the portal administrator, your first step is to purchase licenses on the [Tableau Customer Portal](#). When you purchase licenses, the portal will return corresponding product keys. To renew your license, visit the [Tableau renewal](#) web page.

Tableau has a number of products (e.g. Desktop, Server, Prep Builder, Add-ons, etc). Each of the Tableau products require that you activate licenses by updating the Tableau software with the product keys that are purchased and stored on the Tableau Customer Portal. As

the administrator who is tasked with activating Tableau licenses, it is important that you understand the relationship between licenses and keys. See [Understanding License Models and Product Keys](#).

Activation

Activation is the process of uploading and saving Tableau product keys to Tableau Server. This operation is done with Tableau Services Manager (TSM). TSM is a tool that makes changes to the local operating system and file system and therefore requires administrative access to the local computer. A TSM administrator requires different permissions and access than a Tableau Server administrator, which is the administrative role for day-to-day operation of Tableau Server tasks, such as adding users, sites, managing projects and permissions, etc. See [Administrative roles](#) for more information about various Tableau Server administrative roles.

The following topics describe how to connect to TSM:

- [Sign in to Tableau Services Manager Web UI](#)
- [tsm Command Line Reference](#)

Online activation

If your Tableau Server installation is able to communicate with the internet, then we recommend using the default online activation method.

- To understand how to activate during the installation process, see [Activate and Register Tableau Server](#).
- To understand how to activate product keys after you have refreshed your subscription, see [Refresh Expiration Date for the Product Key](#).
- To understand how to activate product keys after you have added purchased new features or user licenses, see [Add Capacity to Tableau Server](#).

Offline activation

If Tableau Server is running in an offline environment, where it is not possible to access the Tableau license servers on the internet, then you must activate licenses according to the Tableau offline activation process:

- To understand how to activate offline during the installation process, see [Offline Activation When Installing Tableau Server](#) .
- To understand how to activate product keys after you have refreshed your subscription, adding features, or adding users, see [Activate Tableau Server Offline - Adding a License](#). Tableau Server installations in an offline activation mode cannot perform a `tsm licenses refresh` command but must activate the new subscription key located in the Tableau Customer Portal.

Lost activation

In some cases license activations can fail after the license has been activated. These failures can occur due to connection failures with local processes or when a change has occurred with the VM or hardware configuration. For example, proxy changes, port blocking, network changes, or altering a machine hardware can cause the licensing activation to fail. If Tableau Server is unable to verify the license, operation may be interrupted and the server will be in an “unlicensed” state.

To view the product keys and the Tableau Server license state, run `tsm licenses list` and `tsm status -v`.

Depending on the product key that is unverified, Tableau Server may operate in a degraded state until the product key is in a valid state. See [Troubleshoot Licensing](#).

Deactivate

You can activate the same Tableau Server product key on up to three environments. This allows you to test Tableau Server (in a sandbox or QA environments, for example), as well as use Tableau in production. To maximize your activations, you should deactivate your product key when you remove Tableau Server from a computer or close down a VM, unless you will be reinstalling Tableau on the same computer. Doing this gives you the opportunity to use the activation on a different computer. For example, if you move Tableau Server from one computer to another, deactivate the product key, then remove Tableau from the original computer. When you install Tableau on the new computer, you can activate the key there without any conflict. If you are removing Tableau Server to reinstall it on the same com-

Tableau Server on Linux Administrator Guide

puter, you don't need to deactivate the key. Tableau will use the key when reinstalled unless an obliterate command was performed with the “-l” option

See Deactivate Product Key.

Tableau Server licensing and virtual machines (VMs)

If you run Tableau Server on VMs, either locally, or in the cloud, be aware of the potential for complications related to licensing. If you are simply upgrading Tableau Server on the VM, you do not need to take any extra action related to licensing. If you plan to clone the VM to create either a new production or test environment to upgrade, you need to deactivate any Tableau Server licenses before cloning. If you do not do this, the new VM environment will end up with untrusted licenses, and any attempts to upgrade or start Tableau Server will fail. You may also end up hitting the maximum number of activations for the licenses when trying to activate the product keys on the new VM.

To avoid issues with licensing on VMs, deactivate all Tableau licenses before cloning a VM or allowing it to be permanently shut down.

Login-based License Management

Login-based license management, helps you manage licensing for users with Creator roles on Tableau Server and Tableau Online. Users with Explorer or Viewer roles cannot use this feature. If you're using Role Based Subscriptions with Tableau Server or Tableau Online, you can simplify your license management using login-based license management to eliminate separate Tableau Desktop and Tableau Prep Builder product keys. You only need to manage one or more product keys for on-premises Tableau Server, or in the case of Tableau Online, you don't need to manage any product keys at all.

See Login-based License Management.

Adding users

Each user who accesses resources on Tableau Server must be licensed.

- To understand user roles and licensing, see [Understanding License Models and Product Keys](#).
- To understand how to add users, see [Add Users to Tableau Server](#).
- To understand how to activate product keys after you have added purchased new user licenses, see [Add Capacity to Tableau Server](#).

Understanding License Models and Product Keys

This topic describes the different licensing models and the product keys associated with them.

A useful visual of how product keys are represented in Tableau Server can be found in [tsm licenses list](#). The [Tableau Customer Portal](#) will also display product key information including type and seat count.

Term licensing models

Tableau's term license model is defined by the metric that permits use of Tableau Server.

- A *role-based* license metric allows you to deploy Tableau Server on a single computer or on multiple computers in a cluster. Each user that accesses Tableau Server must be licensed and assigned a role. Administrators can add users based on available licenses of each type.
- A *core-based* license metric imposes no constraints on the number of user accounts in Tableau Server. Instead, the license specifies the maximum number of computer cores on which you can run Tableau Server. You can install Tableau Server on a single computer or across multiple computers as a multi-node cluster, as long as the total number of cores in all the computers does not exceed the total number that the license allows.

Not all processes installed with Tableau Server impact the calculation of total number of cores used. A subset of processes is considered "licensed processes." Core licensing is calculated only on computers running licensed processes. If a computer has

one or more licensed processes installed on it, the cores on that computer count toward the total cores used. For more information about licensed processes, see [Licensed processes](#).

In the output returned by `tsm licenses list`, the `TYPE` field describes the user license metric.

Role-based license model

Tableau offers role-based term licenses that grant a range of capabilities at various price points. Three types of role-based term licenses are available: Viewer, Explorer and Creator.

- *Viewer licenses* let users view and interact with workbooks in Tableau Server. Viewer licenses also let users access Tableau Mobile, add comments to workbooks, export visuals in various formats, download workbook summary data, create subscriptions for themselves, and receive data-driven alerts.
- *Explorer licenses* are similar to the user-based licenses available in previous Tableau Server releases, and include the capabilities provided with Viewer licenses, and additional capabilities. An Explorer license allows access to workbook authoring capabilities using a web browser, as well as a full set of collaboration features.
- *Creator licenses* permit a wide range of capabilities when using Tableau Server, and also grant use of Tableau Desktop and Tableau Prep Builder. A Creator license allows all of the capabilities available under the Explorer license, as well as the following capabilities when using Tableau Server:
 - Create and publish new workbooks from a new data source.
 - Edit embedded data sources in the Data pane.
 - Create and publish new data connections.

Note: Tableau Server administrators will always consume the highest role available. If a Creator product key is activated, the Tableau Server Administrator(s) will take this role. If the highest role available on Tableau Server is an Explorer, the Server Administrator will take the Explorer role. If Creator licenses are added to the server, any existing Server Admin accounts using Explorer licenses will automatically convert to use Creator licenses.

TSM administrator accounts do not require licenses.

Product keys are used to activate and add licenses to Tableau Server. Administrators can activate additional product keys to add licenses, capacity, or turn on functionality. The Tableau Server should be restarted after activating any new product key in order for Tableau Server to reflect the new activation. On a new installation, the first product key activated will be used to generate the Tableau Server Administrator role. Please select a Tableau Server Creator product key from the Tableau Customer Portal. A Tableau Server Viewer product key should not be used to install Tableau Server because the Tableau Administrator role requires either a Creator or Explorer product key. Explorer and Viewer product keys can then be activated to add additional licenses. Once the product keys have been activated, administrators can add users and assign them site roles, which automatically consume available licenses.

In the output returned by `tsm licenses list`, the `CREATOR`, `EXPLORER`, `VIEWER` fields display the number of licenses for each role license type. Each role license type is governed by its own product key. Therefore, if your organization has purchased licenses for all three roles, then you must activate three product keys.

Core-based license model

In a core licensing model, the license defines the number of total computer cores the server can run on, instead of how many users can be added. This means the server can support virtually unlimited users (as Explorers or Viewers; the legacy term for these types of roles was “interactors”). Core-based license also allows a Guest User account, which is not possible with role-based licensing.

However, core licensing does not include Creator seats on the server (those site roles will be greyed out when adding users). An additional license is required to create new content and publish it to Tableau Server. For content publishing in a core-based organization, one of the following must be purchased:

- At least one role-based Creator license user (which includes Tableau Desktop and Tableau Prep Builder), or
- At least one licensed Tableau Desktop that uses a legacy Tableau Desktop product key, and an Explorer (can publish) site role. This allows the user who has a license for Tableau Desktop to be able to publish to Tableau Server.

In the output returned by `tsm licenses list`, the `TYPE` field will display the number of cores that are licensed. Additionally, the `GUEST ACCESS` field will display `true`.

Perpetual license model (legacy)

In the past, Tableau sold access to Tableau Server with perpetual licenses. Although these licenses are no longer available, some customers use this licensing arrangement.

In the perpetual license model, customers pay a maintenance subscription that is renewed annually. If maintenance expires, the software continues to work but the customer loses access to technical support and software upgrades.

Perpetual licenses were either sold for a specific number of users, called interactors, or for a specific number of cores:

- Interactor licensing is a named-user model where customers purchased licenses by the seat, similar to current role-based licensing. However, unlike role-based licensing, where different access roles are priced accordingly, in interactor licensing, licenses were unbound by role. Licensed users could be Server admins, Site admins, Publishers, Interactors, or Viewers. User roles were set by the administrator only as a means to manage access to content and server configuration.
- Perpetual core licensing has the same model as subscription core licensing, it specifies the number of computer cores the software can be run on and supports unlimited users and a guest account.

Perpetual licenses are no longer available. However, customers that purchased the legacy perpetual licenses may still purchase annual maintenance.

In the output returned by `tsm licenses list`, the `TYPE` field displays `Perpetual`. Note also that the `MAINT EXP` date is also displayed.

Embedded Analytics

Tableau's Embedded Analytics offering is a limited-use license for Tableau Server that's made available to customers who wish to embed Server functionality into an external facing solution to provide Tableau content and insights to clients outside of their organization.

For example, consider an organization that runs a service where they analyze consumer data and generate reports on behavioral patterns regarding different consumer demographics. In this scenario, Tableau Server acts in support of a specific proprietary application titled 'Demographics Analyzer' and connects with exported TXT files and a SQL database. The organization makes visualizations available to its clients in a secure portal, where clients log in to manage their account and view the results. End users are uniquely identified by the account they use to access the portal; this determines the number of User-Based Embedded Analytics Server licenses.

The Embedded Analytics license is not displayed in the output returned by tsm licenses list. To verify the license contact [Customer Success](#).

Feature licenses

Feature licenses are sold differently than other licenses. Features with independent licenses must be licensed for every user (or all cores) in the deployment. A deployment includes a licensed production Tableau Server installation and licensed non-production Tableau Server installations that support the production installation.

In the context of licensing, the availability of these features to the user base are "all or none." The features are licensed annually and must expire at the same time as subscription licenses or perpetual maintenance terms:

- Data Management Add-on
- Tableau Server Management Add-on
- Login-based License Management

Data Management Add-On

The Data Management Add-on license includes Tableau Catalog and Tableau Prep Conductor for a single Tableau Server deployment, which may be role-based or core-based. For more information, see [About Data Management Add-on](#).

The Data Management Add-on may require resource cores, which specifies the computing power that is used to run flows for Prep Conductor. Servers with core-based licensing are required to purchase at least four Resource Cores. See [License the Data Management Add-on](#).

In the output returned by `tsm licenses list`, a single product key for Data Management is indicated by the `DATA MANAGEMENT ADD-ON` field, which displays `true`.

Tableau Server Management Add-on

The Tableau Server Management Add-on is licensed on a per deployment basis, which may be role-based or core-based. For more information on Server Management Add-on and the features included in this Add-on, see [About Tableau Advanced Management](#).

In the output returned by `tsm licenses list`, a single product key for Server Management is indicated by the `SERVER MANAGEMENT ADD-ON` field, which displays `true`.

Login-based License Management

Login-based license management (LBLM) simplifies licensing for Tableau Desktop and Tableau Prep Builder. Instead of using product keys, these products are activated when a user with a Creator's license authenticates with Tableau Server. If a customer is using Tableau Desktop or Tableau Prep Builder without Tableau Server, LBLM cannot be used.

For more information, see [Login-based License Management](#).

In the output returned by `tsm licenses list`, a single product key for login-based licensed management is indicated by the `LBLM` field, which displays `true`.

View Server Licenses

Server administrators can view the license and product key information for Tableau Server.

Viewing licenses from the Tableau Server web UI

How you navigate to the Licenses page in Tableau Server depends on whether you have a single site, or multiple sites.

- On a server with a single site, click **Settings** and **Licenses**:
- On a multi-site server, click **Manage all sites** on the site menu, **Settings**, and **Licenses**:

Note: The **Manage all sites** option only displays when you are signed in as a server administrator.

This page displays information for any licenses that have been activated on your server, including any user-based (term) or core-based licenses.

Use the TSM web interface

1. Open TSM in a browser:

`http://<tsm-computer-name>:8850`

2. Click **Configuration** , and then click **Licensing** :

The table displays the product key, expiration date, and expiration of maintenance.

Note: The TSM Web UI provides a limited amount of licensing information. Use the TSM CLI or the Tableau Server Web UI to see additional licensing information, including the number of each type of user-based license (Creator, Explorer and Viewer).

Use the TSM CLI

1. Open a command prompt as administrator on the initial node (the node where TSM is installed).
2. Run the following command:

```
tsm licenses list
```

The command lists licenses that are activated on the Tableau Server deployment.

For example, a server with five Creator licenses, five Explorer licenses, 100 Viewer licenses, and a Data Management Add-on would provide command output similar to the following:

```
C:\Windows\system32>tsm licenses list
Number of product keys: 4
The following license keys will expire soon. Access renewal resources including information on how to renew your software or change your billing preferences here https://www.tableau.com/support/renew
TS90-86E2-BF8A-89EA-38FE  TSPR-3861-0888-8C5A-C79D  TS49-176C-E840-3410-5EAS  TSQ7-8988-5CFA-FD66-29AF
KEY          TYPE          CREATOR    EXPLORER   VIEWER     DATA MANAGEMENT ADD-ON  GUEST ACCESS LIC EXP   MAINT EXP  UPDATABLE  LBLM       SERVER MANAGEMENT ADD-ON
TS90-86E2-BF8A-89EA-38FE  Term         0          0          100        false         false  11/30/20  N/A        false      false      false
TSPR-3861-0888-8C5A-C79D  Term         0          0          0          true          false  11/30/20  N/A        false      false      false
TS49-176C-E840-3410-5EAS  Term         0          5          0          false         false  11/30/20  N/A        false      false      false
TSQ7-8988-5CFA-FD66-29AF  Term         5          0          0          false         false  11/30/20  N/A        false      false      false
```

The following fields are returned:

- **KEY:** A globally unique 16-character string that identifies the license.
- **TYPE:** Describes the type of license
 - Term: Term licenses map to a subscription schedule and must be renewed. The expiration date is listed under the LIC EXP field.
 - Perpetual: Perpetual licenses are purchased once and do not need to be renewed but must be refreshed to update the MAINT EXP or maintenance expiration date.
 - Cores: Core licenses are licenses that map to the number of cores on the computers running specific Tableau Server services. Core licensing allows for a guest user access to views on the server or embedded on other web servers. Core licenses also allow for unlimited Explorer and Viewer users.
- **CREATOR:** The number of Creator licenses issued to the Tableau Server deployment.
- **EXPLORER:** The number of Explorer licenses issued to the Tableau Server deployment.
- **VIEWER:** The number of Viewer licenses issued to the Tableau Server deployment.
- **DATA MANAGEMENT ADD-ON:** Tableau Server is licensed for the Data Management Add-on (True/False). See About Data Management Add-on.

- **GUEST ACCESS:** Tableau Server is licensed for a Guest User. See [Guest User](#). The ability to leverage a Guest User requires Core licensing. See `TYPE` field.
- **LIC EXP:** The date that the license expires and Tableau Server will stop working. Term licenses expire. See `TYPE` field. Visit the Tableau [Customer Portal](#) to refresh licenses.
- **MAINT EXP:** Applies only to legacy perpetual licenses (`TYPE = Perpetual`). For Term licenses, this field will output, `N/A`. `MAINT EXP` displays the date that the maintenance contract for the Tableau Server deployment expires. To update the license maintenance key see [Refresh Expiration Date for the Product Key](#). Visit the Tableau [Customer Portal](#) to view maintenance purchase history and to purchase additional maintenance.
- **UPDATABLE:** Specifies whether the license is an updatable subscription license (`True/False`).
- **LBLM:** Specifies if login-based license management (LBLM) is enabled for the Tableau Server deployment (`True/False`). When enabled, LBLM allows users to log into Tableau Server to license their instance of Tableau Desktop or Prep, rather than entering a product key. For more information about LBLM, see [Login-based License Management](#).
- **SERVER MANAGEMENT ADD-ON:** Tableau Server is licensed for the Server Management Add-on (`True/False`). For more information about the Server Management Add-on, see [About Tableau Advanced Management](#).

Note: The license terms for Creator, Explorer and Viewer users are set according to the terms of the user-based license (term license), if present. So, a server with only a core-based license will have unlimited Explorer and Viewer users and guest access, but no Creator users. To learn more, see [Use role-based licenses on a server with core-based licensing](#).

Refresh Expiration Date for the Product Key

When you purchase a new subscription, you must refresh your product key(s). Refreshing the product key(s) updates the product key with the new date that reflects the new expiration dates. If you refresh a subscription (term) product key before the expiration date

occurs, the product key will not change but the expiration date will. This can create a mismatch between the product keys listed in the Tableau Customer Portal and those listed in the Tableau Server TSM product key list. The product key will change when the expiration date occurs or shortly thereafter. If a subscription (term) product key is not refreshed and has expired, Tableau will stop working and you will have to activate a new product key in the Tableau Customer Portal.

On the other hand, if the product key is perpetual (legacy) and its maintenance has expired, Tableau will continue to operate but you will not have access to upgrades until the maintenance is renewed. After renewing the maintenance, refresh the existing product key to update its maintenance expiration date. The product key will never change. For more information about different product key types and associated licenses, see [tsm licenses list](#).

Note: This topic describes how to refresh the expiration date for Tableau Server, the Advanced Management, and the Data Management. For information about refreshing the maintenance date on Tableau Desktop see [Refresh the product key](#) in the Tableau Desktop and Tableau Prep Deployment Guide.

Before you begin

Verify the expiration date of your license(s). You can view the expiration date by following the TSM web interface procedure below, or by running `tsm licenses list` in the CLI.

- Compare the date with the date displayed in the [Tableau Customer Portal](#).
- If the portal does not display the date that you expect, contact [Customer Success](#).
- To renew your license, visit the [Tableau renewal](#) web page.

If the TSM date matches the portal date and the following refresh operation fails, contact [Tableau Support](#).

Note: To refresh your product key in an offline environment, visit the Tableau Customer Portal to obtain the latest term key, and then activate it. For more information about

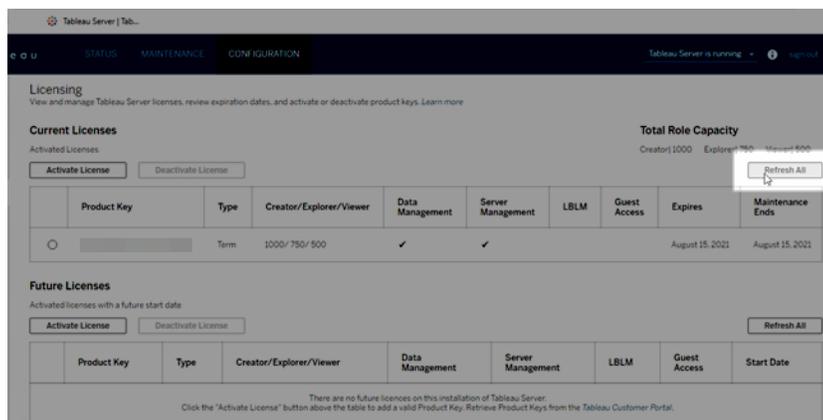
offline activation, see [Activate Tableau Server Offline - Adding a License](#). Activate all product keys before starting Tableau Server. Otherwise some users might become unlicensed.

Use the TSM web interface

1. Open TSM in a browser:

```
https://<tsm-computer-name>:8850
```

2. Click **Configuration** and **Licensing** and click **Refresh All**:



Use the TSM CLI

1. Open a command prompt as administrator on the initial node (the node where TSM is installed).
2. Run the following command:

```
tsm licenses refresh
```

Add Capacity to Tableau Server

You may need to add capacity to your Tableau Server installation to allow you to increase the number of users (if you have a user-based license) or the number of cores (if you have a

Tableau Server on Linux Administrator Guide

core-based license).

Tableau will provide you with a new product key that adds capacity to your existing Tableau Server installation. You need to activate this key and use it together with your existing product key(s) to get the combined capacity you are licensed for. Each feature (e.g., "Data management add-on") and license type (e.g., "Explorer") requires a key. For more information about the relationship between keys and licenses, see [tsm licenses list](#).

Follow the steps below to add a product key to Tableau Server.

If your Tableau Server is not connected to the internet, then you must perform an offline activation. See [Activate Tableau Server Offline - Adding a License](#).

Note: If you have upgraded to Tableau Server version 2021.1 or later, you no longer need to restart Tableau Server when you add capacity. For more information, see [Zero Downtime Licensing](#).

Use the TSM web interface

1. Open TSM in a browser:

```
https://<tsm-computer-name>:8850
```

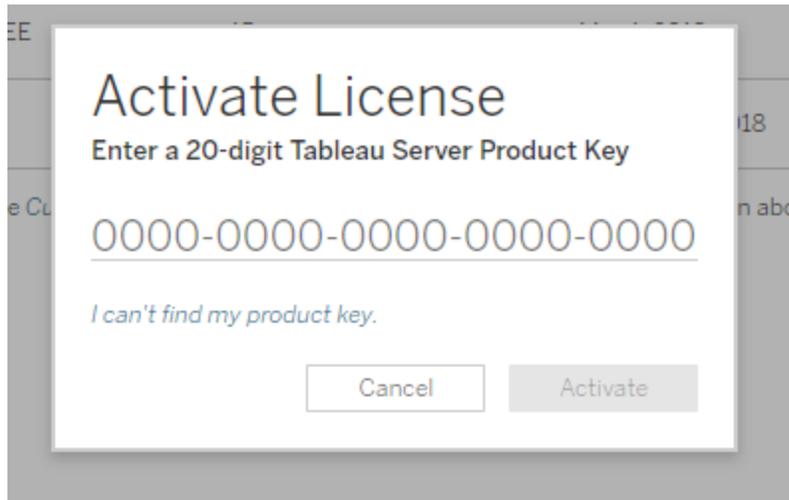
2. Click **Licensing** on the **Configuration** tab and click **Activate License**:

The screenshot shows the Tableau Server web interface. The top navigation bar includes 'STATUS', 'MAINTENANCE', and 'CONFIGURATION' tabs. The 'CONFIGURATION' tab is selected. On the left sidebar, 'Licensing' is highlighted. The main content area is titled 'Licensing' and contains a table with the following data:

Product Key	Seat Licenses	Expires
 XXXXXXXXXXXX	15	May 1, 2018
trial	10	January 22, 2018

Below the table, there is a note: "Access Tableau Server product keys in the [Customer Portal](#). View [Desktop License Usage](#) for more information about

3. Enter or paste your new product key and click **Activate**:



4. On the Register page, enter your information into the fields and click **Register**.

● ——— ○ ——— ○ ——— ○
Activate Register Setup Initialize

Register with Tableau. All fields are required.

Contact Information

First Name _____ Last Name _____

Phone Number _____ Email _____

Company Information

Organization _____ Industry _____ ▼

Department _____ ▼ Job Role _____ ▼

Region Information

City _____ Postal Code _____

Country/Region _____ ▼ State/Province _____ ▼

Register

Use the TSM CLI

1. Copy the product key to your computer.
2. Run the following command:

```
tsm licenses activate --license-key <license key>
```

3. After activation, you need to register Tableau Server. To do this, create a registration file and then pass it as an option with the `tsm register` command.

- a. Generate a template that you can edit by running the following command:

```
tsm register --template > /path/to/<registration_
file>.json
```

- b. Edit the template file to create your completed registration file.

Here is an example of a registration file in the required format:

```
{
  "zip" : "97403",
  "country" : "USA",
  "city" : "Springfield",
  "last_name" : "Simpson",
  "industry" : "Energy",
  "eula" : "yes",
  "title" : "Safety Inspection Engineer",
  "phone" : "5558675309",
  "company" : "Example",
  "state" : "OR",
  "department" : "Engineering",
  "first_name" : "Homer",
  "email" : "homer@example.com"
}
```

- c. After saving changes to the file, pass it with the `--file` option to register

Tableau Server:

```
tsm register --file /path/to/<registration_file>.json
```

For example:

```
tsm register --file /usr/share/tableau-reg-file.json
```

Activate Tableau Server Offline - Adding a License

When you install Tableau Server, you have to activate at least one product key. Doing this activates the server, and specifies the number of license levels you can assign to users. There are also times you may need to activate licenses after Tableau Server is installed, for example, if you add capacity to your server, or get a new product key. If you don't have your product key, you can get it from the [Tableau Customer Account Center](#).

In most cases, you can activate your key directly from Tableau, either during installation, or later, using the Tableau Services Manager (TSM) Licenses page, but there are some circumstances that don't allow you to do this. If your computer is not connected to the internet for example, or has a firewall that restricts access outside your intranet. In these cases you need to do an offline activation.

- Offline activation during install—To complete an offline activation when you are installing Tableau Server, see [Offline Activation When Installing Tableau Server](#).
- Offline activation of licenses after install—To complete an offline activation after your server is installed and running, follow the steps below.

Offline activation overview and prerequisites

Offline activation of Tableau Server involves several steps. If you have never activated a product key before on your computer, you need to run through the steps twice, once to initialize licensing on the computer, and a second time to activate the key. If you have had a license on the computer in the past, you run through the steps only once.

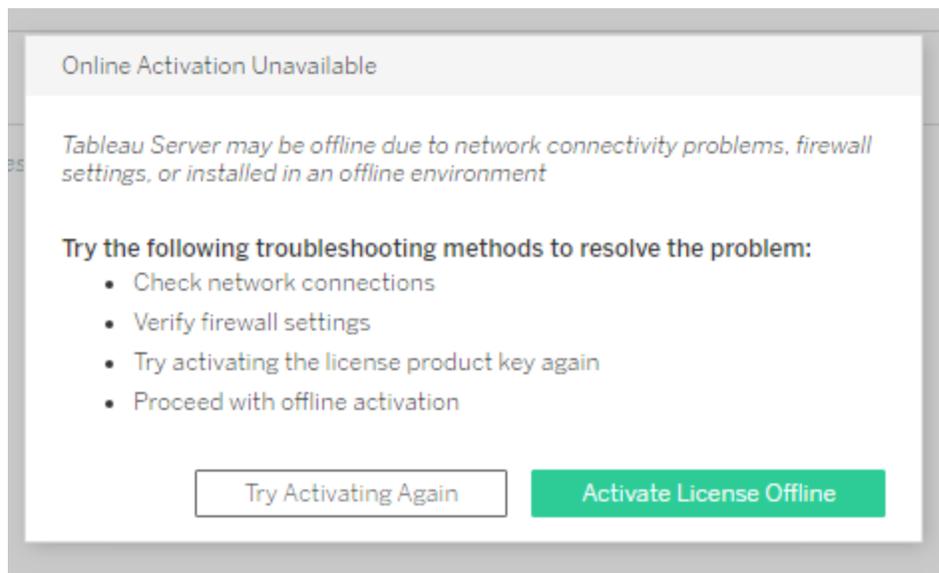
Offline activation steps:

1. Generate an offline activation request file (`offline.tlq`).
2. Upload the offline activation request to Tableau and download the resulting activation file (`activation.tlf`).
3. Upload the activation file to Tableau Server.

If this is the first time a Tableau product key has been activated on the computer, you repeat these three steps a second time.

Use the TSM web interface

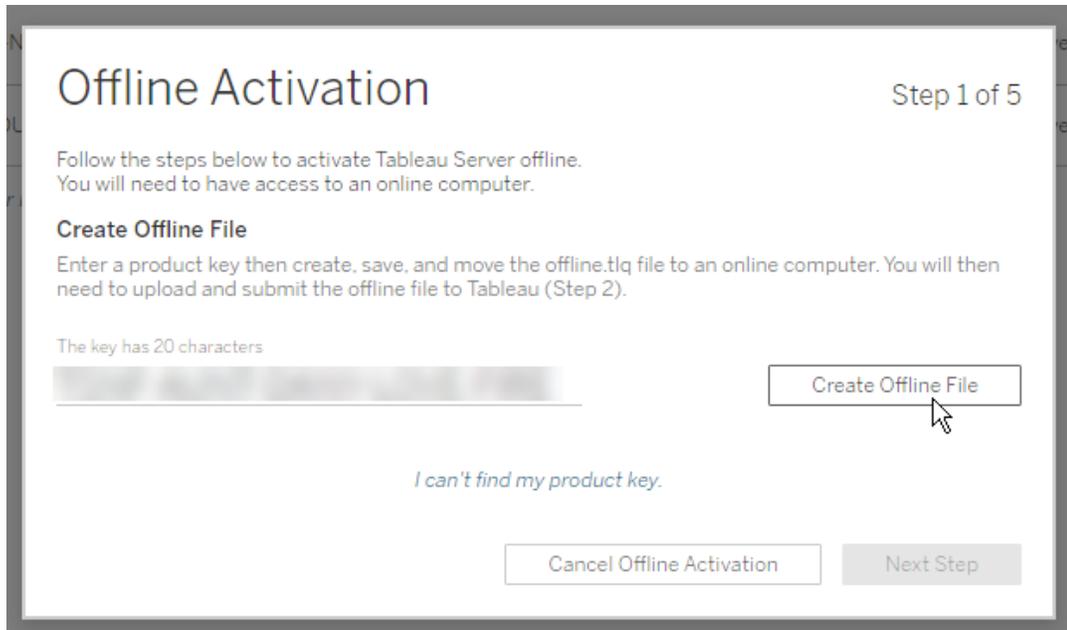
If you attempt to activate your product key from the TSM licenses page and see a dialog that says online activation is unavailable, you can activate the key offline. Click **Activate License Offline**.



1. Create Offline File

Your first step is to create an offline file (`offline.tlq`).

Create an offline file you will upload to Tableau. If your license key is not pre-filled in the form, enter your key and click **Create Offline File** to generate an offline file (`offline.tlq`) on the local computer.



Copy the `offline.tlq` file to a computer with internet access. You need to upload this file to Tableau to generate an activation file.

2. Upload and Submit Offline File

Next, on the computer where you copied the `offline.tlq` file, go to <http://www.tableau.com/support/activation> to upload your file and submit it. Doing this automatically generates an activation file (`activation.tlf`) that you can download and copy back to the Tableau Server computer.

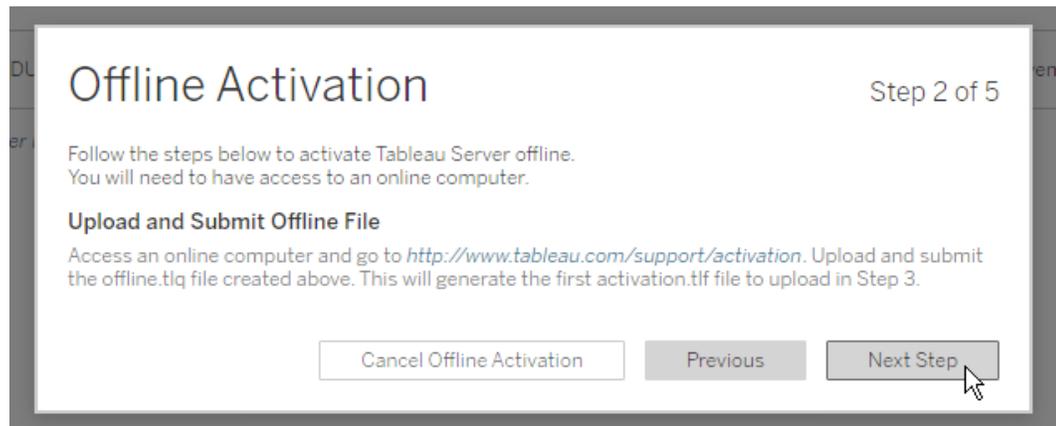
- a. On the Offline Activation page, click **Choose File** to select the `offline.tlq` file.
- b. Click **Upload Activation File** to submit the file to Tableau.
- c. Click the [here](#) link to download the resulting `activation.tlf` file to your computer.

Offline Activation

The activation was successful. Please click [here](#) to download your activation file.

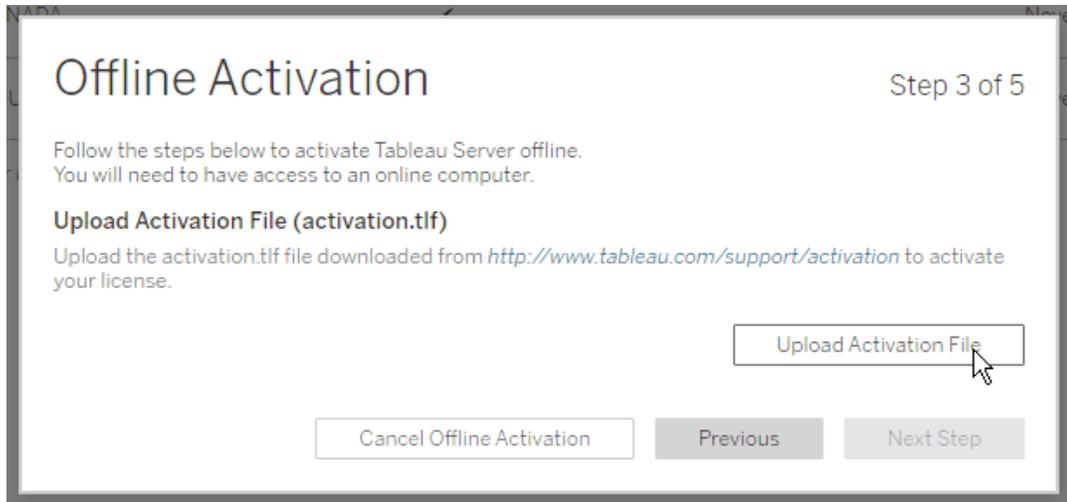
For help creating the offline activation file, see [Activate Tableau Desktop Offline](#) or [Activate Tableau Server Offline](#). ([Linux](#))

- d. Copy the downloaded `activation.tlf` file to the computer where Tableau Server is installed.
3. Click **Next Step** to move to the next dialog.



4. Upload Activation File

On the Tableau Server computer, click **Upload Activation File** to upload the `activation.tlf` file to Tableau Server.

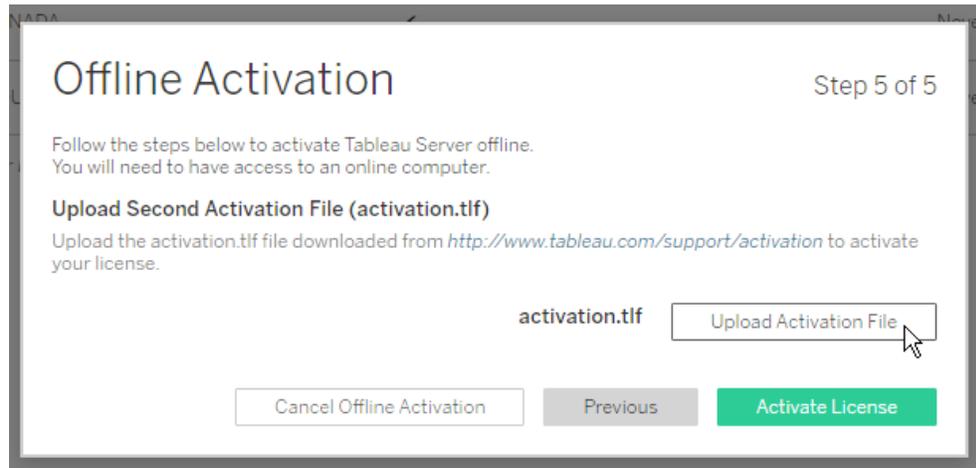


5. Create, Upload, and Submit a Second Offline File

Important: This step is only necessary if you are prompted to activate Tableau Server a second time. A second prompt only occurs if you are activating a computer that has never had an activated Tableau license on it.

- a. Click Create Offline File to generate a second `offline.tlq` file.
- b. Copy that file to the computer that has internet access.
- c. Open a browser session to <http://www.tableau.com/support/activation> and upload your new offline file.
- d. Download the resulting `activation.tlf` file.
- e. Copy the file to the Tableau Server computer.
- f. Click **Next Step** and then click **Upload Activation File** to upload the second

activation.tlf file to Tableau Server.



6. Click **Activate License** to complete the offline activation. You will need to restart Tableau Server for the changes to take effect.

Use the TSM CLI

Log in to Tableau Services Manager

Before you can proceed you must log in to Tableau Services Manager (TSM).

To log in to TSM, run the following command:

```
tsm login -u <username>
```

What if I can't log in?

The account that you use to configure the rest of the installation must be a member of the `tsmadmin` group that was created during initialization. To view the user accounts in the `tsmadmin` group, run the following command:

```
grep tsmadmin /etc/group
```

If the user account is not in the group, run the following command to add the user to the `tsmadmin` group:

Tableau Server on Linux Administrator Guide

```
sudo usermod -G tsmadmin -a <username>
```

After you have added the user to the `tsmadmin` group, run the `tsm login` command.

Step 1 - Generate an offline activation request file

1. On the initial node, open a terminal session.
2. Type this command to get your offline activation file:

```
tsm licenses get-offline-activation-file -k <product-key> -  
o <target-directory>
```

You can get your product key from the [Tableau Customer Portal](#). The target directory must already exist.

3. Copy the offline activation file (`offline.tlq`) from the target directory to a computer that has internet access.

Step 2 - Upload the offline activation request to Tableau

1. On the computer that has internet access, go to the Tableau [Product Activations](#) page.
2. Complete the instructions to upload your `offline.tlq` file.

This creates an activation file, `activation.tlf`.

3. Download the resulting activation file from Tableau.

Step 3 - Initialize or activate your license

1. Copy the activation file (`activation.tlf`) to a location accessible from your Tableau Server computer.
2. Run the following command:

```
tsm licenses activate -f <path-and-activation-file>
```

3. (Skip this step if you are installing Tableau Server for the first time.)

Restart the server for licensing changes to take effect:

```
tsm restart
```

First time license activation

If this is the first Tableau license activated on the computer, you will see this message:

```
Your license has been initialized. To complete the activation,
we need one more exchange. Generate and send to Tableau a
second activation request file.
```

Repeat steps 1 through 3 above to activate your license. You need to generate a second `offline.tlq` file, upload it to Tableau, and download the resulting `activation.tlf` file, then use that file to activate your license.

If you are not sure whether you need to repeat the steps, you can run this command:

```
tsm licenses list
```

If you see a message like this one you need to repeat the steps above:

```
No licenses are currently activated.
```

Subsequent license activation

If this is not the first Tableau license activated on the computer, or if you have completed the steps above two times, you should see a success message:

```
Activation successful.
```

Tableau Server is activated. If you need additional assistance, contact [Tableau Technical Support](#).

Deactivate Product Key

There are some scenarios where you must deactivate a product key:

- Changing a hardware configuration
- Changing product keys

Tableau Server on Linux Administrator Guide

- Moving a product key to a new installation

Note: With the Server ATR licensing model, it is not possible to deactivate a product key using the TSM web interface or CLI. Instead, you must use [atrdiag](#) to manage product keys. For more information, see [Activate Tableau Server Using the Authorization-To-Run \(ATR\) Service](#).

Before you begin

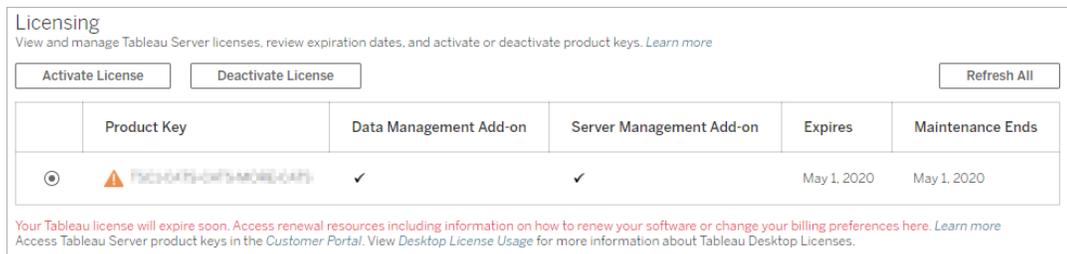
Verify that you are removing the correct product key(s). You can view license details by running `tsm licenses list` in the CLI.

Use the TSM web interface

1. Open TSM in a browser:

```
https://<tsm-computer-name>:8850
```

2. Click **Configuration** and **Licensing**.



	Product Key	Data Management Add-on	Server Management Add-on	Expires	Maintenance Ends
<input type="radio"/>	 TS2020-05-01-05-01-05-01	✓	✓	May 1, 2020	May 1, 2020

3. Select the product key that you want to deactivate, and then click **Deactivate License**.
4. After the key is deactivated, restart Tableau Server.

Use the TSM CLI

1. Open a command prompt as administrator on the initial node (the node where TSM is installed).
2. Run the following commands:

```
tsm licenses deactivate --license-key <product-key>
```

```
tsm pending-changes apply
```

```
tsm restart
```

Automate Licensing Tasks

You can use `tsm licenses` to perform licensing tasks such as activating or deactivating a Tableau Server product key on- or off-line, and getting associated files for offline activation or deactivation. However, Tableau Server must already be deployed and configured. You can automate these licensing tasks using the [Tableau Services Manager API](#).

```
C:\ProgramData\Tableau\Tableau Server\data\tabsvc\logs\
```

Troubleshoot Licensing

This topic includes instructions for troubleshooting issues related to Tableau Server licensing.

Handle an unlicensed server

Tableau offers two licensing models: role-based and core-based. To learn more about role-based and core-based licensing, see [Licensing Overview](#).

role-based licensing requires each active user account to be covered by a license. role-based licenses have a defined capacity, or number of users that they allow. Each user is assigned a unique user name on the server and is required to identify themselves when connecting to the server.

Tableau Server on Linux Administrator Guide

Core-based licensing has no constraints on the number of user accounts in the system, but it does restrict the maximum number of processor cores that Tableau Server can use. You can install Tableau Server on one or more machines to create a cluster, with the restriction that the total number of cores in all the machines does not exceed the number of cores you have licensed and that all of the cores on a particular machine are covered by the license.

Unlicensed role-based server

The most common reason for a server that has role-based licensing to be unlicensed is an expired product key or an expired maintenance contract.

Unlicensed core-based server

A core-based server can become unlicensed for a variety of reasons, such as an expired product key or when Tableau Server nodes running licensed processes cannot contact the Tableau Server node running the License Manager service. To learn more about licensed processes, see [Tableau Server Processes](#).

When the server is unlicensed you may not be able to start or administer the server. You can, however, manage your licenses using the `tsm licenses` command.

Unlicensed server administrator

All Tableau Server administrators require a user license. Tableau Server administrators will always consume the highest role available. If a Creator product key is activated, the Tableau Server Administrator(s) will take this role. If the highest role available on Tableau Server is an Explorer, the Server Administrator will take the Explorer role. If Creator licenses are added to the server, any existing Server Administrator accounts using Explorer licenses will automatically convert to use Creator licenses.

TSM administrator accounts do not require licenses.

If the license that the server administrator is using expires, then the account will become unlicensed and will be unable to sign in.

Verify the expiration date of your license(s) for the administrators on the server:

- Run `tsm licenses list`.
- Compare the date with the date displayed in the [Tableau Customer Portal](#).
- If the portal does not display the date that you expect, contact [Customer Success](#).
- To renew your license, visit the [Tableau renewal](#) web page.
- Run the `tsm licenses activate` command to activate a new license for the administrator account(s).

If the TSM date matches the portal date and the following refresh operation fails, contact [Tableau Support](#).

If the license for your administrator account has expired or will expire soon, you will need to activate a new license for the account. Alternatively, you can unlicense a non-administrator user to free a license for the server administrator account.

If a Tableau Server administrator is using a Creator, Explorer or Viewer license and their license expires, they will use another license of the same type, if available. If no license seats are available the user will become “unlicensed”.

Important: Do not restart Tableau Server until you have activated a new license or transferred a site role for the server administrator account.

Troubleshoot role-based licensing

This section provides information about resolving issues that can occur when adding the role-based Viewer, Explorer and Creator licenses to Tableau Server or Tableau Online, or when these licenses expire. The highest available license type is Creator, followed by Explorer, and finally Viewer. To learn more about role-based licensing, see [Licensing Overview](#).

A user or administrator is unlicensed due to license expiration

To avoid having users unexpectedly become unlicensed or move to another site role, you should always do one of the following before the license that they are currently using expires:

Tableau Server on Linux Administrator Guide

- Renew and activate a replacement license. If a user occupies a Creator, Explorer or viewer license and their license expires, they will use another license of the same type, if available.
- Change the site role of those users to allow the use of a license that is not due to expire.

To learn how site roles can be changed to require a different license, see [Set Users' Site Roles](#).

The reassignment of users to new licenses is governed by the following logic:

- When a Server Administrator user occupies a Creator license and their license expires (with no replacement licenses available), they are reassigned to an Explorer license if any Explorer licenses are available. This license reassignment occurs in order of most recent login. Server Administrators displace other users who might be currently using an Explorer license. If no Creator or Explorer licenses are available a Server Administrator becomes unlicensed.
- When a non-Server Administrator user occupies a Creator license and their license expires (with no replacement licenses available), they become unlicensed. To avoid having these users become unlicensed, change their site role prior to license expiration. This is especially important for users in the Site Administrator Creator site role, who must move to the Site Administrator Explorer site role before their Creator license expires to avoid losing Site Administrator capabilities.
- When a non-Server Administrator user occupies an Explorer or Viewer license and their license expires (with no replacement licenses available), they are upgraded to a higher license type, if licenses of that type are available. Specifically, the following occurs when a license expires:
 - Users who occupy an Explorer license will move to a Creator license, if available (with no change to site role).
 - Users who occupy a Viewer license will move to an Explorer license, if available. If no Explorer licenses are available, these users will move to a Creator license, if available (with no change to site role).
 - If no licenses are available at the higher license types, those users are moved to Unlicensed.

Users are reassigned to a new license as described above in order of most recent login, with lower license types reassigned first (first Viewer, then Explorer, and then Creator).

For example: Two users with a Viewer license, a user with the Creator license, and two Server Administrators with a Creator license all have their licenses expire. Four unexpired Explorer licenses are available for these users. In this situation, the following occurs in the order shown below:

1. The user with a Viewer license who logged in most recently is reassigned to an Explorer license.
2. The second user with a Viewer license is reassigned to an Explorer license.
3. The Server Administrator user with a Creator license who logged in most recently is reassigned to an Explorer license, and then the second Server Administrator with a Creator license is reassigned to the remaining Explorer license.
4. The user with the Creator license becomes unlicensed.

Server Administrator site role is unchanged when using a Creator license

Server Administrators gain Creator capabilities if Creator licenses are available in Tableau Server, with no change to their site role name. All other Tableau Server and Tableau Online users gain Creator licenses only if assigned to a site role that includes Creator in its name.

Licenses are not immediately available

When you add a role-based license to Tableau Server, those licenses become available to all users when you restart Tableau Server.

A user with a Viewer license cannot open Tableau Server or Tableau Online workbooks from Tableau Desktop

A user with a Viewer license who also has a separate Tableau Desktop license will be unable to open workbooks on Tableau Server or Tableau Online using Tableau Desktop. To open workbooks such using Tableau Desktop, that user will need an Explorer or Creator license on Tableau Server or Tableau Online.

Migrate from Core-Based to Role-Based Licensing

You can migrate Tableau Server from a core-based license metric (which counts the processor cores on which you have Tableau Server installed) to a role-based license metric

(which counts named users). To learn more about licensing metrics, see [Licensing Overview](#).

Prepare for migration to role-based licensing

Core-based licenses allow an unlimited number of users, including view-only guest accounts. Every user has a site role when they are added to Tableau Server, and these users and site roles persist when licensing is changed. Because role-based licenses limit the number of users, you should ensure that your new role-based licenses accommodate the number of users who are currently connecting to Tableau Server, including the users currently using guest accounts. If your new role-based licenses don't accommodate the full number of users, some users will move to the **Unlicensed** site role, which can be very CPU intensive. If you have a large number of users, your application server processes may become unavailable until all the users are processed. This operation could take hours to complete, so plan accordingly.

Before you migrate, verify that you have user licenses and corresponding product keys to allow all users to access Tableau Server after the migration is complete:

- Sign in to the [Tableau Customer Portal](#) to verify licenses and to copy the corresponding product key(s).
- To learn more about site roles, see [Set Users' Site Roles](#).
- To count the number of users in your Tableau Server installation, export a list of users to count them with a tool such as Microsoft Excel. To learn how to export a list of users, see [Export a User List](#).

Migrate to role-based licensing

To migrate to role-based licensing you must stop Tableau Server, deactivate the core-based product key, activate the role-based product key(s), and then start Tableau Server. Because this process will result in a restart of Tableau Server and cause downtime for Tableau Server users, you should migrate licensing during a period of low usage.

1. Stop Tableau Server:

Use the `tsm stop` command.

2. Deactivate the core-based product key(s):

Use the `tsm licenses deactivate` command with the core-based product key (s).

3. Activate the role-based product key(s):

Use the `tsm licenses activate` command with the role-based product key(s).

4. Start Tableau Server:

Use the `tsm start` command.

Use role-based licenses on a server with core-based licensing

The 2018.1 release of Tableau Server allows you to add Creator role-based licenses to Tableau Server installations with existing core-based licensing.

Note: If you upgrade a Tableau Server installation to 2018.1 without activating role-based licenses, Tableau Server will continue to operate as it did previously, with no changes to Tableau Server UI or permissions except that the legacy **Viewer** site role is renamed to **Read Only**. In version 2018.2, the **Read Only** site role was deprecated and once again became the **Viewer** site role.

When **Creator** licenses are introduced to Tableau Server, all Server Administrator users are required to have **Creator** license roles, which may require additional licenses. Administrators can activate additional **Creator** licenses using the `tsm licenses activate` command in Tableau Server to increase licensed user capacity. Activating these role-based licenses gives you a combination of the capabilities granted by role-based Creator licenses and the capabilities granted by your core-based license. To learn more about the different types of role-based licenses, see [role-based licenses](#).

Note: The unlimited number of users who have access to Tableau Server under core-based licensing have equivalent capabilities to users with an **Explorer** license under

role-based licensing. **Creator** functionality is limited to the defined user license model (s).

For example, if a Tableau Server installation has a 16 core license that includes guest access, and you added 10 **Creator** licenses, that server would have the following capabilities:

- A limit of 16 processor cores on hardware that runs Tableau Server
- Guest access
- Unlimited Explorer/Viewer licenses (from the unlimited user licenses that come with a core license)
- 10 Creator licenses

Example of completing a migration from core-based licensing

To extend the example above: If the core-based license was then deactivated, the following capabilities would be available:

- No limits on server hardware
- No guest access
- 10 Creator licenses

If you then added 50 Explorer licenses and 200 Viewer licenses, after deactivating your core-based license, the following capabilities would be available:

- No limits on server hardware
- No guest access
- 10 Creator licenses
- 50 Explorer licenses
- 200 Viewer licenses

Login-based License Management

Login-based license management helps you manage licensing for users with Creator roles on Tableau Server and Tableau Online. Users with Explorer or Viewer roles cannot use this feature. If you're using Role Based Subscriptions with Tableau Server or Tableau Online,

you can simplify your license management using login-based license management to eliminate separate Tableau Desktop and Tableau Prep Builder product keys. You only need to manage one or more product keys for on-premises Tableau Server, or in the case of Tableau Online, you don't need to manage any product keys at all.

You only need one Tableau Server or Tableau Online site to authorize an individual Tableau Desktop or Tableau Prep Builder. Login-based license management is enabled per production instance of your Tableau Server deployment. You can assign Creator roles to users who use Tableau Server, Tableau Desktop, and Tableau Prep Builder for license activation and centralized license management. In addition, login-based license management gives you more visibility into license usage.

You can use administrative views on Tableau Server and Tableau Online to see the assignment and use of Creator seats in one place. Administrative views provide information about who is using your Creator licenses, and shows the most recent license lease and version of both Tableau Desktop and Tableau Prep Builder, which helps you to monitor your Tableau deployments.

The following Tableau products support login-based license management:

- Tableau Online
- Tableau Server 2020.1+
- Tableau Desktop 2020.1+
- Tableau Prep Builder 2020.1.3+

If your Tableau Server is not enabled for login-based license management, contact your Tableau sales representative to obtain a special login-based license management-enabled product key.

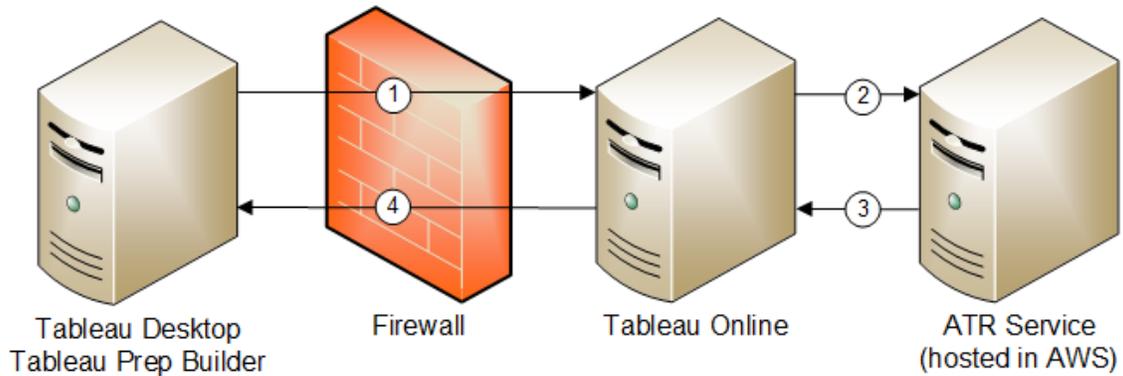
How login-based license management works

When login-based license management is in use, Tableau Desktop or Tableau Prep Builder periodically contacts Tableau Server or Tableau Online in order to verify that the user is a

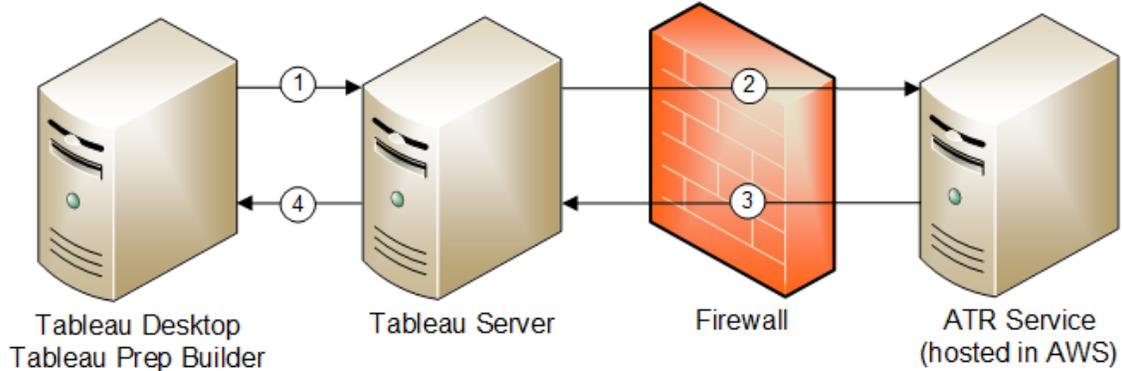
Tableau Server on Linux Administrator Guide

Creator and to obtain an authorization to run. Tableau Server or Tableau Online contacts the Tableau-hosted authorization-to-run (ATR) service to verify that the Tableau Desktop client is allowed activation. The ATR service, hosted on AWS, verifies and manages the license and the length of the authorization window. The diagrams below show the communication process between the client, Tableau Server or Tableau Online, and the licensing service.

Login-based license management with Tableau Online



Login-based license management with Tableau Server



1. You install Tableau Desktop or Tableau Prep Builder and select **Activate with your credentials**. You sign into your Tableau Online or Tableau Server site.
2. Tableau Online or Tableau Server verifies that you are a Creator. If not, you get an error. If yes, Tableau Online or Tableau Server communicates with the ATR service.
3. The ATR service returns an ATR lease to Tableau Online or Tableau Server.

4. Tableau Online or Tableau Server provides an ATR lease to Tableau Desktop or Tableau Prep Builder to finalize activation.

Use login-based license management

To use login-based license management, you either need to use Tableau Online or you need to install Tableau Server and activate it with a login-based license management enabled product key. You will also need to install Tableau Desktop.

Step 1: Install Tableau Server

If you already have Tableau Server installed, skip to (Optional) Step 2: Change login-based license management settings. Otherwise, you'll need to install Tableau Server:

1. Refer to the Deploy chapter to install Tableau Server.
2. During activation, you must enter a Tableau Server product key that supports role-based subscription and login-based license management.
3. Add users to your Tableau Server and set them to one of the three Creator roles: Server Administrator, Site Administrator Creator, or Creator. This assigns a Creator license to these users on Tableau Server. For more information, see [Set User's Site Roles](#).

(Optional) Step 2: Change login-based license management settings

Login-based license management is enabled by default for Tableau Online, Tableau Server, and Tableau Desktop starting with version 2020.1. You can, however, change some login-based license management settings.

Disable login-based license management on Tableau Server

Although login-based license management is the preferred method of activation for Tableau

Desktop and Tableau Prep Builder, you may have Tableau Server installations in your organization used for testing that don't need to use login-based license management. On Tableau Server, you use the Tableau Services Manager (TSM) command line utility to turn off login-based license management.

To turn off login-based license management, at a command prompt, type:

```
tsm configuration set -k licensing.login_based_license_management.enabled -v false
```

```
tsm pending-changes apply
```

Change login-based license management settings on Tableau Desktop or Tableau Prep Builder at install

To change login-based license management settings at the command line, you can run the installer .exe file from your computer's command line as an administrator. If you need to extract the .msi files, follow the instructions to [Extract and run the Windows \(MSI\) installer](#).

To use a duration length other than the default of 14 days/1209600 seconds, include the `ATRREQUESTEDDURATIONSECONDS` switch. For example:

```
tableauDesktop-64bit-2020-1-0.exe /quiet /norestart ACCEPTTEULA=1  
ATRREQUESTEDDURATIONSECONDS=43200
```

You must run the command from the directory where the .exe file is located or specify a full path to the location of the .exe file on the computer. Do not run the setup program from a shared directory on your network. Instead, download the .exe file to a directory on the computer where you're installing.

The following example shows the Windows installer command that disables login-based license management:

```
tableauDesktop-64bit-2020-1-0.exe /quiet /norestart  
ACCEPTTEULA=1 LBLM=disabled
```

or

```
tableauPrepBuilder-64bit-2020-1-0.exe /quiet /norestart  
ACCEPTTEULA=1 LBLM=disabled
```

To set the default URL for the Tableau Server you want users to use for activation when using login-based license management, add the `ACTIVATIONSERVER` option:

```
tableauDesktop-64bit-2020-1-0.exe /quiet /norestart  
ACCEPTTEULA=1 ACTIVATIONSERVER=http://<tableau_server_url>
```

Note: The `ACTIVATIONSERVER` option is only intended for first time activation. If you have previously signed-in to this version of Tableau, you cannot specify the activation server using this option. For example, if you are using Tableau Desktop version 2021.1 and have previously signed-in successfully, you cannot use the `ACTIVATIONSERVER` option to specify an activation server. However, if you install a different Tableau version or you are signing-in to Tableau for the first time on a new computer, you can use the `ACTIVATIONSERVER` option to specify the activation server.

Change login-based license management settings on Tableau Desktop by editing the registry

If Tableau Desktop is already installed, you can change login-based license management settings by editing the Windows registry.

To turn off login-based license management:

Tableau Server on Linux Administrator Guide

```
reg.exe add HKEY_LOCAL_MACHINE\SOFTWARE\Tableau\ATR /f /v LBLM  
/d disabled
```

To make login-based license management the only login option:

```
reg.exe add HKEY_LOCAL_MACHINE\SOFTWARE\Tableau\ATR /f /v LBLM  
/d required
```

Or, you can enable, disable, or require login-based license management by editing the registry directly:

1. As an administrator on the computer running Tableau Desktop, make a backup of the registry file before you make any changes to it.
2. Edit the registry, and in `HKEY_LOCAL_MACHINE\SOFTWARE\Tableau`, find the hive named `ATR` and modify the `LBLM` value to reflect the desired setting:
 - a. Name: `LBLM`.
 - b. Data: `enabled`, `disabled`, or `required`.
3. Restart Tableau so the changes take effect.

Change login-based license management settings on Tableau Desktop on macOS

To change login-based license management settings on macOS, run the following commands in a terminal window to update the preferences file, and then install or restart Tableau Desktop.

To turn off login-based license management:

```
sudo defaults write /Library/Preferences/com.tableau.ATR LBLM  
"disabled"
```

To make login-based license management the only login option:

```
sudo defaults write /Library/Preferences/com.tableau.ATR LBLM
"required"
```

Login-based license management settings

You use the following settings to change login-based license management, set the ATR duration, and set the activation server URL.

Setting	Value	Description
LBLM	enabled, disabled, or required	<p>Set to <code>enabled</code> (the default), the licensing screens will present the two options for activation (product key, or credentials).</p> <p>Set to <code>disabled</code>, login-based license management will not appear on the licensing screens.</p> <p>Set to <code>required</code>, login-based license management is the only way to activate the Tableau Desktop (when the licensing screen appears, it will offer only the credentials option for activation).</p> <div style="border: 1px solid #ccc; background-color: #f0f0f0; padding: 10px; margin-top: 10px;"> <p>Note: When <code>LBLM</code> is set to <code>enabled</code> or <code>required</code>, reporting is also enabled.</p> </div>

Tableau Server on Linux Administrator Guide

<p>ATRREQUESTEDDURATIONSECONDS</p>	<p>1209600</p>	<p>Sets the authorization to run (ATR) duration (in seconds), which is the length of time that an instance of Tableau Desktop and Tableau Prep Builder is authorized to run. The default is 1209600 seconds (14 days). Do not use commas as separators in the value.</p>
<p>ACTIVATIONSERVER</p>	<p>http://<tableau_server_url></p>	<p>Sets the default URL for the Tableau Server you want users to use for activation.</p>
<p>REPORTINGFREQUENCYSECONDS</p>	<p>3600</p>	<p>Sets the default (in seconds) for how often the login-based license management report is sent to Tableau Server or Tableau Online. The minimum setting is 60 seconds, and the default is 3600 seconds (one hour). Change this setting to reduce the load on Tableau Server or to reduce network traffic. No matter what the authorization-to-run (ATR) lease duration is set to, REPORTINGFREQUENCYSECONDS sets the time interval that the Tableau Desktop or Tableau Prep Builder client report login-based license management activations back to Tableau Server or Tableau Online.</p>

Additional configuration for virtual deployments

Tableau Desktop and Tableau Prep Builder periodically contact Tableau Online or Tableau Server to verify that Tableau is authorized to run, based on its license. Tableau Online or Tableau Server then contacts the ATR service to verify the license and the length of the authorization window.

When configuring a virtual (non-persistent) deployment of Tableau Desktop or Tableau Prep Builder, the duration time should be set to one of the lower values such as 4 or 8 hours in order to avoid an over-use activation error message. After the virtual machine (VM) is returned, the ATR service will handle the activation monitoring.

The following flags should be used on a source image prior to publishing Tableau Desktop to end users. Each end user will be activating the software by logging into Tableau Server or Tableau Online with each new VM delivered. No product keys need to be entered if the end user is a Tableau Creator on Tableau Server or Tableau Online.

If you are using login-based license management for Tableau Desktop or Tableau Prep Builder on a VM, you may get an error message that your license information has changed whenever you launch a new VM for Tableau Desktop or Tableau Prep Builder. This error forces a restart, which then asks you to register Tableau Desktop again. This error occurs because the ATR service sends a new token that doesn't match the license cache.

To prevent this from occurring, you can use the `SYNCHRONOUSLICENSECHECK` and `SILENTLYREGISTERUSER` options with the Windows installer. For example:

```
tableau-setup-std-tableau-2020 SYNCHRONOUSLICENSECHECK="true"
SILENTLYREGISTERUSER="true" ATRREQUESTEDDURATIONSECONDS=14400
```

or

```
tableauDesktop-64bit-2020-1-0.exe /quiet /norestart
ACCEPTEULA=1 ATRREQUESTEDDURATIONSECONDS=14400 ACTIVATIONSERVER-
```

Tableau Server on Linux Administrator Guide

```
=http://<tableau_server_url> SYNCHRONOUSLICENSECHECK="true"  
SILENTLYREGISTERUSER="true"
```

If Tableau Prep Builder and Tableau Desktop are being delivered on one Virtual Desktop, `ATRREQUESTEDDURATIONSECONDS` only needs to be set during Tableau Desktop installation. However, if you plan to install Tableau Prep Builder as a stand-alone, you'll need to set `ATRREQUESTEDDURATIONSECONDS` during Tableau Prep Builder installation.

Or, you can edit the following registry keys on the source image:

```
Reg key path: HKLM\SOFTWARE\Tableau\<Tableau version>\Settings\
```

```
Reg key (String value, need to set to true to make that feature  
enabled)
```

```
SynchronousLicenseCheck
```

```
SilentlyRegisterUser
```

(Optional) Step 3: Change the authorization to run (ATR) duration

The login-based license management default settings for the authorization to run (ATR) duration are appropriate for most environments, but you can change these default settings if needed. Login-based license management uses the default authorization to run (ATR) duration of 1209600 seconds (14 days), which is the length of time that an instance of Tableau Desktop and Tableau Prep Builder is authorized to run. This means that after the initial authorization, you could use Tableau without any network connection for 14 days before the activation expired.

The default duration value for login-based license management is not appropriate for delivering a non-persistent VM delivery solution to end users. The ATR duration should be lowered to 4 or 12 hours depending on VM use. When a new VM is delivered to an end user, a new authority to run token will be created. When the VM is returned, this token is also returned and is able to be used on the new VM authorization to run request.

Note: Login-based license management uses the following hierarchy when determining

ATR duration.

1. **ATR Service** – Establishes the minimum (4 hours/14400 seconds) and maximum (90 days/7776000 seconds) ATR durations applicable to all users/installations. It specifies the default ATR duration (14 days/1209600 seconds) if nothing is specified by Tableau Server or Tableau Desktop.
2. **Tableau Server** - Can optionally specify a maximum or default ATR duration (`licensing.login_based_license_management.max_requested_duration_seconds` or `licensing.login_based_license_management.default_requested_duration_seconds`) for all Tableau Desktop installations. Using these ATR duration settings, you can globally set the default ATR duration and maximum ATR duration for all Tableau Desktop clients, which eliminates the need to individually sign in to each Tableau Desktop client to set the ATR duration. The ATR Service maximum setting can be between the minimum of 4 hours (14400 seconds) and maximum of 90 days (7776000 seconds).
3. **Tableau Desktop** – Can optionally specify the ATR duration (`ATRREQUESTEDDURATIONSECONDS`) for the computer on which it is installed. If necessary, you can change the default ATR duration (14 days/1209600 seconds) to a setting within the ATR Service minimum (4 hours/14400 seconds) and maximum (90 days/7776000 seconds). This local ATR duration overrides any durations set by the ATR Service or Tableau Server. However, this default ATR duration cannot be more than the maximum ATR duration set on Tableau Server.

Change the ATR duration for Tableau Desktop or Tableau Prep Builder using Tableau Server

On Tableau Server, you use the Tableau Services Manager (TSM) command line utility to set the ATR duration.

To set the ATR duration, at a command prompt, type:

Tableau Server on Linux Administrator Guide

```
tsm configuration set -k licensing.login_based_license_management.default_requested_duration_seconds -v <value in seconds>
```

```
tsm pending-changes apply
```

To set the ATR maximum duration, at a command prompt, type:

```
tsm configuration set -k licensing.login_based_license_management.max_requested_duration_seconds -v <value in seconds>
```

```
tsm pending-changes apply
```

Change the ATR duration on Tableau Desktop by editing the registry

To use a duration length other than the default of 14 days/1209600 seconds, update the `ATRRequestedDurationSeconds` registry setting. For example:

1. As an administrator on the computer running Tableau Desktop, make a backup of the registry file before you make any changes to it.
2. Edit the registry, and in `HKEY_LOCAL_MACHINE\SOFTWARE\Tableau\ATR`, update the `ATRRequestedDurationSeconds` as follows (0 uses the default setting):
 - a. Name: Find the string value named `ATRRequestedDurationSeconds`.
 - b. Data: Update the number of seconds the duration should last. For example, add 43200 to set a duration of 12 hours.
3. Restart Tableau so that the changes take effect.

Change the ATR duration on Tableau Desktop on macOS

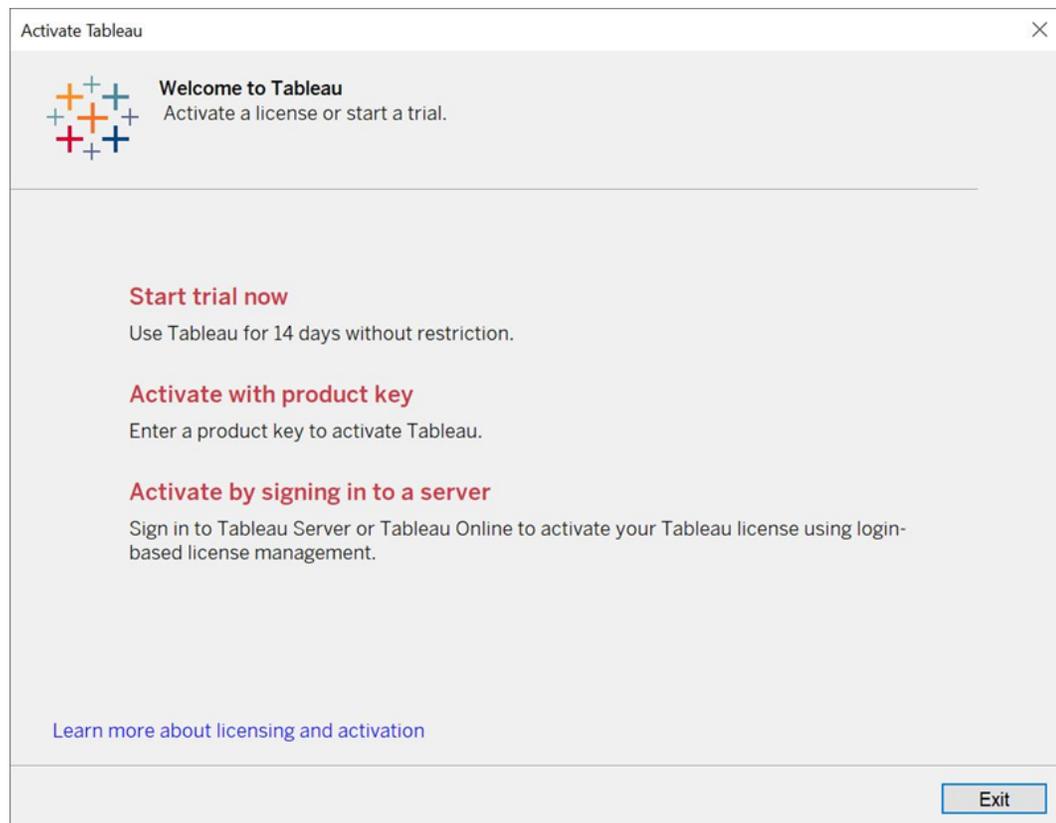
To use a duration length other than the default of 14 days/1209600 seconds, include the `ATRRequestedDurationSeconds` preferences setting. For example:

```
sudo defaults write /Library/Preferences/com.tableau.ATR  
ATRRequestedDurationSeconds -string "43200"
```

Step 4: Activate Tableau Desktop

The 2020.1 and later versions of both Tableau Desktop for Windows or macOS support login-based license management.

1. Run Tableau Desktop setup.
2. The Activate Tableau screen will include the **Activate by signing in to a server** option.



3. Click **Activate by signing in to a server** and then do one of the following:
 - If you're using Tableau Online, click the Tableau Online link.
 - If you're using Tableau Server, enter the Tableau Server URL, and then click **Connect**.

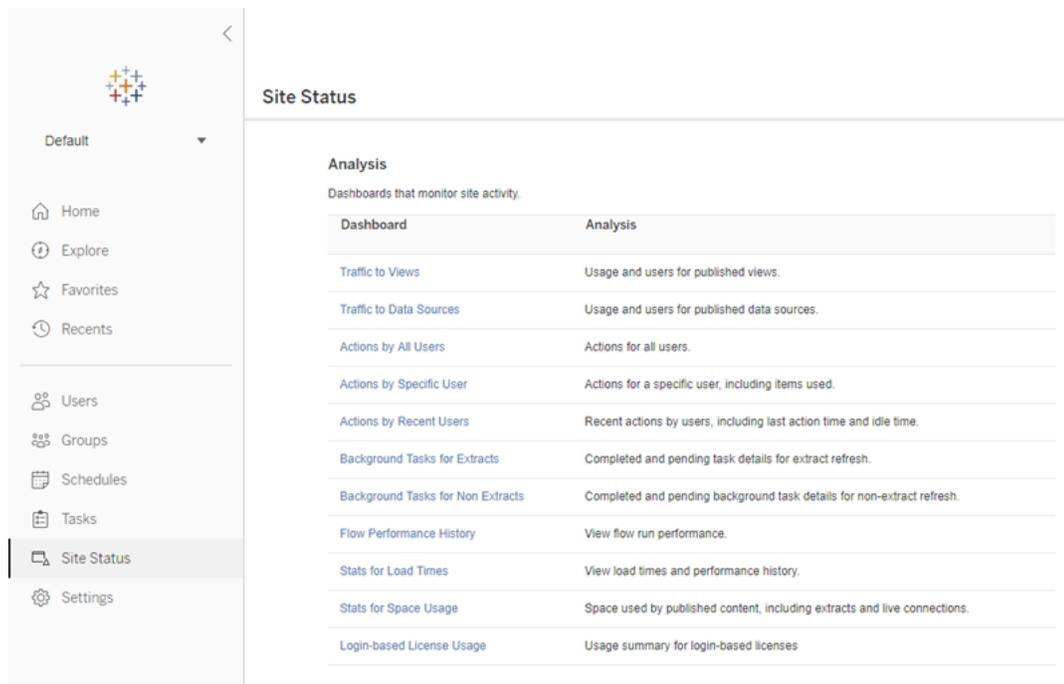
When prompted, enter valid credentials for a user with a Creator role subscription, and then click **OK**.

View login-based license usage

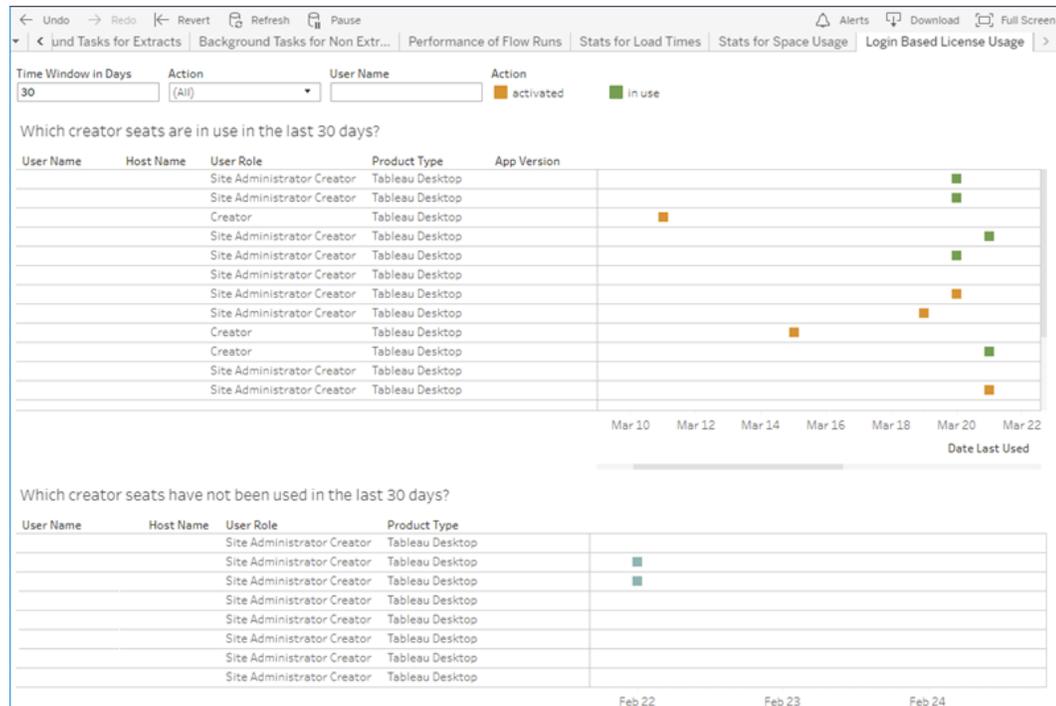
You can view login-based license usage for Tableau Online or Tableau Server. The report shows users, hosts, user role, product, version, activations, Creator seats in use, Creator seats not in use, and when a Creator seat was last used. You can view data for the past 30 days up to a maximum of 183 days.

To view the Login-based license management License Usage administrative view:

1. In Tableau Online or Tableau Server, in the navigation pane, click **Site Status**.



2. On the Site Status page, click **Login-based License Usage**.
3. Optional. On the report screen, you can change the time window to show when seats were last used, filter on actions, filter on user name, and sort by columns.



Troubleshooting

You may encounter one of the following error messages while using login-based license management. Use the information below to resolve the issue.

Login-based license management is not enabled on Tableau Server

If login-based license management is available as an activation option on Tableau Desktop, but not enabled on Tableau Server, or if Tableau Server does not have a login-based license management capable license, the following error message may be displayed:

Login-based license management is not enabled on the server you connected to. Select a different server to connect to, use a product key to activate your license, or start a trial to begin working immediately.

Tableau Server on Linux Administrator Guide

Contact your administrator to ensure that you're using a Tableau Server product key that's enabled for login-based license management. To verify that you have activated the appropriate product key, in the TSM web UI click **Configuration**, and then click **Licensing**. For guidance on activating licenses, see [tsm licenses activate](#). Login-based license management should be enabled by default unless it was previously turned off by the server administrator.

Login-based license management is not enabled on Tableau Desktop

If login-based license management is not enabled on Tableau Desktop, you will not have the option to activate Tableau using your credentials. Prior to version 2020.1, the ability to use login-based license management on Tableau Desktop is not turned on as a default. Check the Tableau Desktop version to ensure the correct version is being used.

If the login-based license management option has been turned off during installation or with an update, see [Step 2: Enable login-based license management](#).

Product key expiration date doesn't change after purchasing a year subscription

After purchasing a renewal of Tableau Server, and the new subscription term is reflected in Tableau Server, it may take up to 24 hours before the new expiration date appears in **Manage Product Keys** on Tableau Desktop.

You do not have a Creator license

When attempting to activate from Tableau Desktop, the following error message may be displayed:

You do not have a Creator license. Contact your administrator to obtain one.

This error is displayed if you have not been assigned a Creator role. If you belong to multiple sites on Tableau Online or Tableau Server, you need to sign in to the site where you have the Creator role when using login-based license management. Otherwise you'll get this error.

To verify that the product key you have activated on the server includes Creator licenses, open the TSM web UI and click **Configuration**, and then click **Licensing**.

You have activated the maximum number of computers

When attempting to activate from Tableau Desktop, the following error message may be displayed:

*You have activated Tableau the maximum number of times allowed under your account.
You must wait for the license activation on another computer to expire before you can activate Tableau again.*

This error is displayed when you activated Tableau from multiple computers with the same Creator user credentials and exceeded the maximum number of activations. You must wait until the authorization-to-run (ATR) token expires on one of the existing computers before attempting to activate a new computer. If you are using non-persistent virtual machines (VMs), you can shorten the ATR duration to prevent this error from occurring again.

To shorten the ATR token duration for maximum activation

If you encounter this maximum use error when using a non-persistent virtual deployment, it is possible to shorten the ATR duration to 4 hours (14400) seconds to avoid the error in the future. Alternatively, instead of changing the duration on an individual desktop, you can set the default duration on Tableau Server to affect all users.

The following steps shorten the lease on a computer previously activated with login-based license management that will no longer be used, in order to free up a seat to be activated on a new computer:

1. Open a Command Prompt as an administrator on a Tableau Desktop computer that will no longer be used.
2. Navigate to the Tableau binaries (\bin) directory, using the following command.

Windows: `cd Program Files\Tableau\Tableau <version>\bin`

Mac: `cd /Applications/Tableau\ Desktop\ <version>.app/Contents/MacOS`

Tableau Server on Linux Administrator Guide

3. Run the following command to set the duration to 4 hours, in seconds (e.g., 14400).

Windows: `atrdiag.exe -setDuration 14400`

Mac: `sudo ./atrdiag -setDuration 14400`

4. Delete the previous ATR token using the following command:

Windows: `atrdiag.exe -deleteAllATRs`

Mac: `./atrdiag -deleteAllATRs`

5. Next, overwrite the existing ATR token. Open Tableau Desktop. Tableau displays the “License has Changed” message. Click **Exit** to automatically close and reopen Tableau Desktop.
6. In the registration dialog box, click **Activate**, and then reactivate Tableau Desktop through Tableau Server using login-based license management, which will overwrite the existing token.
7. Close Tableau Desktop and wait for the ATR duration to elapse (e.g., 4 hours) so that the ATR token expires and frees-up a user seat. Do not open Tableau Desktop before the ATR duration has elapsed. Check to make sure the ATR duration has elapsed. The ATR token TTL End should show a date and time in the future (e.g., 4 hours from now).

Windows: `atrdiag.exe`

Mac: `./atrdiag`

8. After the ATR token expires and you can successfully sign in to Tableau Server on a new computer.

To return your computer to an unlicensed state

1. Open a Command Prompt as an administrator.
2. Navigate to the Tableau binaries (`\bin`) directory, using the following command:

```
cd Program Files\Tableau\Tableau <version>\bin
```

3. Run the following command:

Windows: `atrdiag.exe -deleteAllATRs`

Mac: `./atrdiag.exe -deleteAllATRs`

Note: This removes only the ATR token from the computer. It does not free-up any of the user seats. The user seat is only freed-up after the deleted ATR token expires.

Your Tableau credentials are invalid

When attempting to activate from Tableau Desktop, the following error message may be displayed:

Your Tableau credentials are invalid. Contact your administrator to reset your account.

This error is displayed when your Tableau license is not recognized. Contact your administrator..

Your computer's clock is not synchronized to the current time

When attempting to activate from Tableau Desktop, the following error message may be displayed:

Your computer's clock is not synchronized to the current time. Synchronize your computer's clock to the current time and then try to activate Tableau.

This error is displayed when your computer's clock is not synchronized with the current time. Synchronize your computer's clock with a time server on the internet or enable automatic time synchronization.

Unable to activate with your credentials

When attempting to activate Tableau, the following error message may be displayed:

Tableau Server cannot verify your licensing information over the internet. Contact your administrator to check your internet connection.

This error is displayed when the port `atr.licensing.tableau.com:443` is not open on all Application Server (VizPortal) nodes, or you have a proxy that is not configured properly to forward traffic to Tableau's licensing server.

To diagnose connectivity to Tableau's licensing server, paste the following URL (`https://atr.licensing.tableau.com/_status/healthz`) into a browser or at a curl command prompt.

Zero Downtime Licensing

With zero downtime licensing, which was introduced in Tableau Server version 2021.1, you can apply most licensing changes to Tableau Server without needing to restart when license end dates, capacity, or installed features are changed.

When should you restart Tableau Server?

When you apply a license update that does not require a restart, Tableau Server displays the following message: **Updated Licensing details are being applied across Server**. However, there are some situations when applying a license update that require you to restart Tableau Server. For example, if you are activating or deactivating a product key that reduces features or changes the allowed data source connections, you'll need to restart Tableau Server after making your changes. When a restart is required, Tableau Server will display the following message: **Restart Server to apply updated Licensing details**. The following table lists the times when license changes require you to restart Tableau Server.

License change	Restart required?
Extending a license term	No
Adding or reducing user license capacity	No
Adding core license capacity	No

Reducing core license capacity	Yes
Adding a Server Add-on feature	No
Removing a Server Add-on feature	Yes
Changing allowed data sources	Yes
Expired product key	Yes

Add Users to Tableau Server

You can add users to Tableau Server one at a time or in batches. You can add them to the server as unlicensed users, and then add them to sites and assign site roles as you onboard them to Tableau Server. Or you can add users to sites and specify their site roles at the same time, at which point they are ready to sign in.

Before you begin

- If your Tableau Server is configured with an Active Directory external identity store, review [User Management in Deployments with External Identity Stores](#) to understand how multiple domains, domain naming, NetBIOS, and Active Directory user name format influence Tableau user management.

Note: In the context of user and group synchronization, Tableau Server configured with LDAP identity store is equivalent to Active Directory. Active Directory synchronization features in Tableau Server function seamlessly with properly configured LDAP directory solutions.

- Verify that you have enough user licenses and role licenses for your user base. If you attempt to add a user for a site role that you do not have a license for, then you will receive an error.
- To add user licenses to your Tableau Server deployment, visit the [Tableau Customer Portal](#) to purchase licenses and the corresponding product key(s). After you have

purchased licenses, see [Add Capacity to Tableau Server](#) to update the server with the new key(s).

- The steps in this topic describe how to add an individual user and assign their site role. To add users in batches, see [Import Users](#).

Adding users at the server level vs. the site level

In a single-site environment, server administrators can add users on the **Users** page.

The screenshot shows the 'Users' page with 57 users. It includes a search bar, a filter for 'Site role' set to 'Any site role', and a table of users. The table has columns for 'Display name', 'Username', 'Site role', 'Groups', and 'Last signed in'.

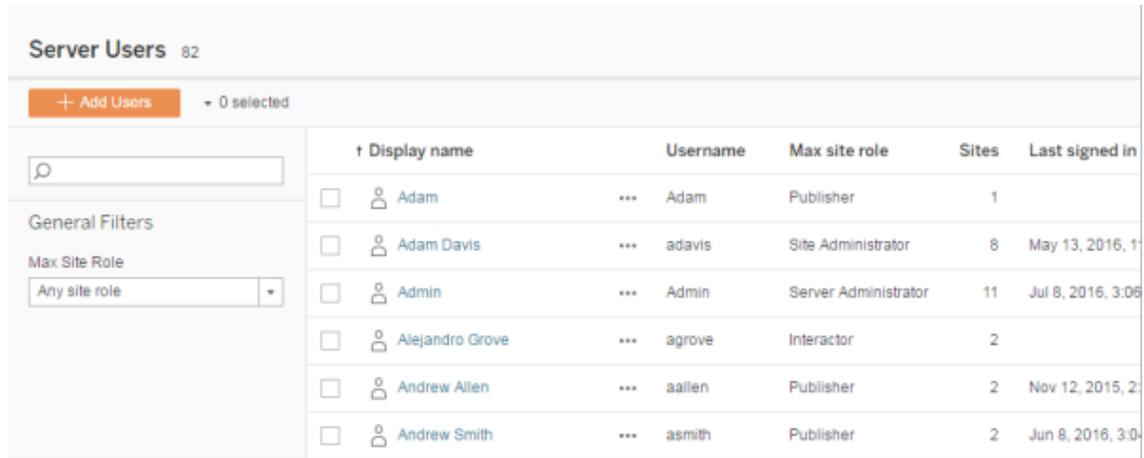
Display name	Username	Site role	Groups	Last signed in
Adam Davis	adavis	Site Administrator	4	May 13, 2016
Admin	Admin	Server Administrator	2	Jul 8, 2016
Alan Wang	awang	Publisher	4	
Alejandro Grove	agrove	Interactor	2	
Andrew Allen	aallen	Interactor	3	Feb 19, 2016

After you add a site to Tableau Server, it becomes a multi-site server with a **Server Users** page (all server users from every site appear here) and a **Site Users** page. Only server administrators can access the **Server Users** page, and both site administrators and server administrators can access the **Site Users** page.

The screenshot shows the 'Site Users' page with 64 users. It includes a search bar, a filter for 'Site role' set to 'Any site role', and a table of users. The table has columns for 'Display name', 'Username', 'Site role', 'Groups', and 'Last signed in'.

Display name	Username	Site role	Groups	Last signed in
Adam Davis	adavis	Site Administrator	2	May 13, 2016, 11:54 AM
Admin	Admin	Server Administrator	3	Jul 8, 2016, 3:06 PM
Andrew Allen	aallen	Publisher	2	Nov 12, 2015, 2:07 AM
Andrew Smith	asmith	Publisher	4	Jun 8, 2016, 3:04 PM
Ashley Garcia	agarcia	Site Administrator	4	Jun 2, 2016, 4:24 PM
Brendan Sweed	bsweed	Publisher	3	Sep 9, 2015, 1:33 AM

The **Server Users** page is the only place where you can assign users to multiple sites, delete users from the server, and if the server is using local authentication, reset user passwords.



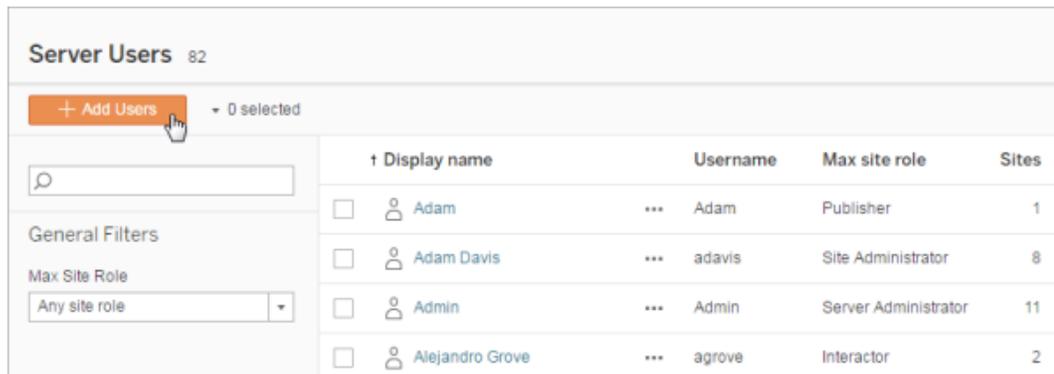
The screenshot shows the 'Server Users' page with 82 users. A search bar and a filter for 'Max Site Role' (set to 'Any site role') are visible. The table below lists several users with their display names, usernames, max site roles, and the number of sites they are associated with.

	Display name	Username	Max site role	Sites	Last signed in
<input type="checkbox"/>	Adam	Adam	Publisher	1	
<input type="checkbox"/>	Adam Davis	adavis	Site Administrator	8	May 13, 2016, 1
<input type="checkbox"/>	Admin	Admin	Server Administrator	11	Jul 8, 2016, 3:06
<input type="checkbox"/>	Alejandro Grove	agrove	Interactor	2	
<input type="checkbox"/>	Andrew Allen	aallen	Publisher	2	Nov 12, 2015, 2
<input type="checkbox"/>	Andrew Smith	asmith	Publisher	2	Jun 8, 2016, 3:0

Add a user to the server

1. In the site menu, select **Manage All Sites > Users**, and then click **Add Users**.

To add a user to a site, you select the site and go to the **Users** page.



This screenshot is identical to the previous one, but with a mouse cursor pointing to the '+ Add Users' button in the top left corner of the interface.

2. Do one of the following:
 - If the server is configured for **local authentication**, click **New User**, and enter a user name. With local authentication, the best way to avoid user name

collisions is to provide an email address for the user name. For example, *jsmith@example.com* instead of *jsmith*.

User names are not case sensitive. Characters not allowed in user names include the semi-colon (;) and colon (,).

- If the server is configured for **Active Directory authentication**, click **Active Directory User**. If you are adding a user from the same Active Directory domain that Tableau Server runs on, the server domain will be assumed, and you can type the AD user name without the domain.

Note: Do not enter the user's full name; this can cause errors during the importing process.

3. If the server is using local authentication, provide the following:

- **Display Name**—Type a display name for the user (e.g., *John Smith*).
- **Password**—Type a password for the user.
- **Confirm password**—Retype the password.
- **Email**—This is optional and can be added at a later time in the user profile settings.
- **Selected users are Server Administrators:** Specify whether the user should be a server administrator.
- **Name (Site Membership) / Site Role:** If the user is not a server administrator, you can assign a user to zero or more sites, along with a site role for each site. You do not have to choose site membership and site role at this time. If you don't specify site membership and site role for a new server user, the user will be added as a server user only, with a site role of Unlicensed. For details, see [Set Users' Site Roles](#).

4. Click **Create**.

New User

Username:
 Username available

Display name:

Password:

Confirm password:

Email (optional):

All sites

Site	Site role ⓘ
<input type="checkbox"/> Documentation - 20 User Limi...	
<input checked="" type="checkbox"/> Finance	<input type="text" value="Publisher"/>
<input type="checkbox"/> Human Resources	

Selected users are Server Administrators

Sign in to the Tableau Server Admin Area

As a server administrator on Tableau Server, you can access admin settings to configure sites, users, projects, and to do other content-related tasks.

If you want to change server settings such as processor, caching, authentication, distributed deployment, and other related configurations, see [Sign in to Tableau Services Manager Web UI](#).

If you are running Tableau Desktop and want to sign in to Tableau Server to publish or access content and data sources, see [Sign in to Tableau Server in Tableau Desktop](#).

Here's how to sign in to the Tableau Server admin pages:

Tableau Server on Linux Administrator Guide

1. Open your browser and enter the server URL. Here are some examples of what the URL might look like:

`http://localhost/` (if you're working directly on the server computer)

`http://MarketingServer/` (if you know the server's name)

`http://10.0.0.2/` (if you know the server's IP address)

If the server is *not* using port 80, you need to include the port number in the URL, as in these examples:

`http://localhost:8000/`

`http://MarketingServer:8080/`

`http://10.0.0.2:8888/`

... where 8000 or 8080 or 8888 is the port that you configured.

Tableau Server displays a page where you can enter a user name and password:



2. Enter the credentials for the server administrator that you created when you finished installing Tableau Server.

You're then taken to the main page of the **Default** site, and you're ready to create users, sites, and manage content.

Navigate the Admin Areas of the Tableau Web Environment

As an administrator on Tableau Server or Tableau Online, you can access admin settings that aren't available to other users to configure sites, users, projects, and to do other content-related tasks.

The settings in this article refer to the Tableau web environment. Tableau Server administrators with appropriate credentials can also change server settings such as processor, caching, authentication, distributed deployment, and related configurations using the TSM web environment. For information, see [Sign in to Tableau Services Manager Web UI](#).

Access based on site role and number of sites

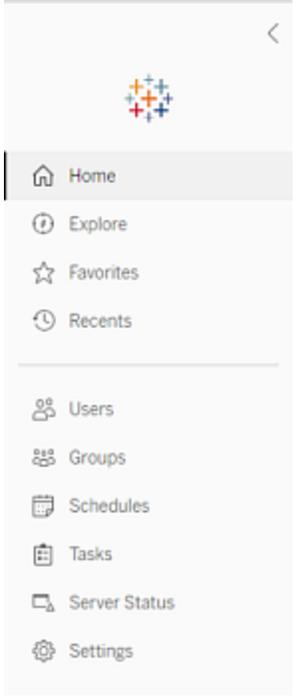
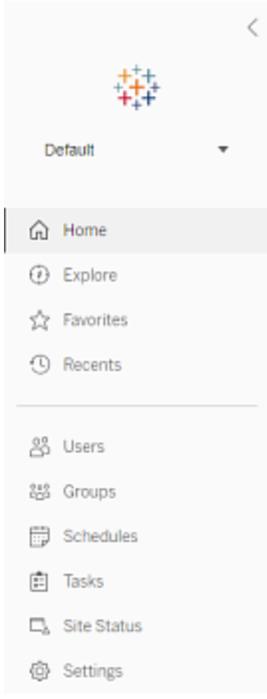
The menus you get when you sign in to Tableau Server or Tableau Online depend on the following conditions:

- Whether you're a site or server administrator.

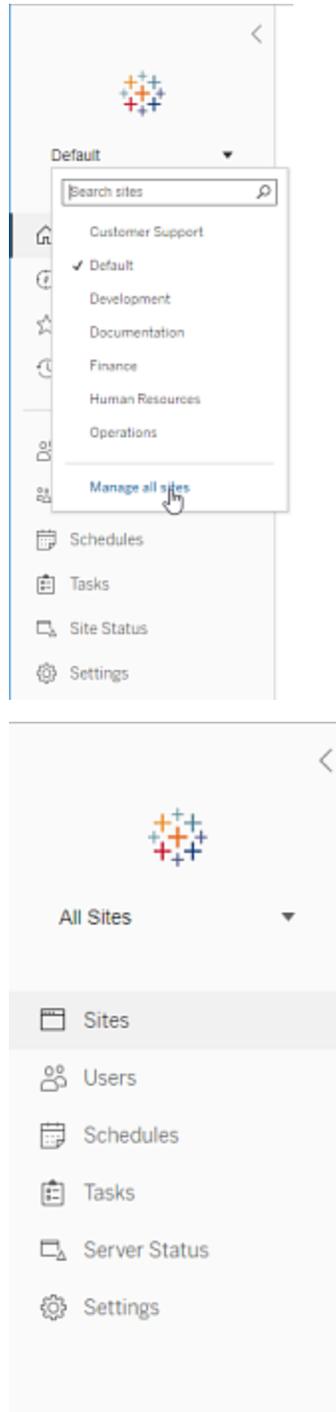
Site administrator access is available on Tableau Online and Tableau Server. Server administrator access is only on Tableau Server.

- Whether you have access to only one site or to multiple sites.

Server administrator

<p>On a single-site server, the site selector does not appear, and all other menus are the same.</p>	
<p>In a multi-site environment, menus along the left enable you to modify a specific site or all sites, and to configure users, groups, schedules, tasks, and server settings.</p> <p>To access server administrator settings that affect all sites, open the site menu by clicking the arrow next to the current site name, and then select Manage all sites.</p> <p>The Content and Group tabs go away, and the site menu text changes to All Sites to let you know you are managing server-wide settings, and options like Server Status reflect the server-wide view.</p>	

To return to the site administration menus, select **All Sites**, and then select the site you want to manage.

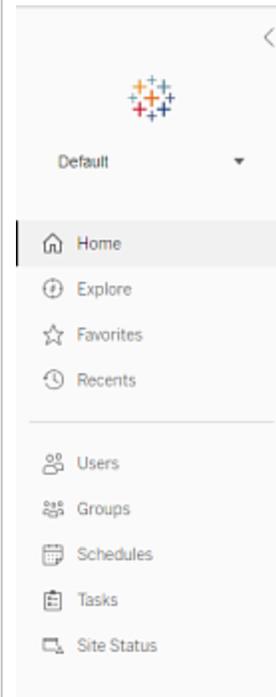


Site administrator

If you are a site administrator for Tableau Online or Tableau Server, and you have access to multiple sites, you'll get menus for selecting which site to manage, and for managing that site's content, users, groups, schedules, and tasks, and for monitoring its status.

The site selector displays the name of the current site. To go to another site, select the site menu, and then select the site name.

If you have access to only one site, the site selector does not appear, but all other menus are the same.



Server administrator tasks

Server administrators (available with Tableau Server Enterprise only) can do the following:

- Monitor server status and activity.
- Generate log files.
- Add sites and edit site settings. Only server administrators can add sites to the server.
- Add users to the server, and assign users to sites.
- Add and manage site groups.

To manage settings only for a specific site, you must first navigate to the site. Within each site, you can do the following:

- Administer content: Create projects, move content from one project to another, assign permissions, change ownership of a content resource, and so on.
- Manage schedules for extract refreshes and subscriptions.
- Monitor site activity and record workbook performance metrics.
- Manage storage space limits for content published by users.
- Allow web authoring.
- Enable revision history.
- Allow site administrators to add and remove users.
- Set the maximum number of licenses that site can consume for each license type (Creator, Explorer, Viewer).
- Allow users to subscribe to workbooks and views, and allow content owners to subscribe others to workbooks and views.
- Enable offline snapshots for favorites (iOS only).

Site administrator tasks

A site administrator on Tableau Online or Tableau Server can do the following tasks:

- Administer content: Create projects, move content from one project to another, assign permissions, change ownership of a content resource, and so on.
- View, manage, and manually run schedules for extract refreshes and subscriptions.
- Add and manage site users (if allowed by the server administrator; see Site Settings Reference).
- Add and manage site groups.
- Monitor site activity.

Sign in to Tableau Services Manager Web UI

This topic explains how to sign in to the Tableau Services Manager (TSM) web UI. The TSM web pages are used to configure Tableau Server settings such as user authentication, server processes, caching, and other server-related settings. You can also configure TSM from a command line shell. See [tsm Command Line Reference](#).

Important: Signing into TSM is *not* the same as signing into Tableau Server. TSM is used to configure the server, and requires an account with administrative privileges on the computer running TSM. Depending on how your enterprise is organized, the TSM administrator could be a user who does not have a Tableau Server account. A Tableau Server administrator has access to administrative pages for creating and editing sites, user, product, and other content-related tasks. For information about signing into Tableau Server as a Tableau Server administrator, see [Sign in to the Tableau Server Admin Area](#).

If you are running Tableau Desktop and want to sign in to Tableau Server to publish or access content and data sources, see [Sign in to Tableau Server in Desktop](#).

Requirements

- The account that you use to sign in to TSM must have authorization to make changes on the local computer where Tableau Server is installed.

Specifically, the account that you use to sign in to TSM must be a member of the TSM authorization group that was created during initialization. The default name of the TSM authorization group is `tsmadmin`.

To view the user accounts in the TSM authorization group, run the following command in the Bash shell. This example uses the default group name, `tsmadmin`:

```
grep tsmadmin /etc/group
```

If the user account is not in the group, run the following command to add the user to the `tsmadmin` group:

```
sudo usermod -G tsmadmin -a <username>
```

Tableau Server on Linux relies on PAM for core authentication scenarios. For more information about PAM integration for TSM administration, see [TSM Authentication](#).

- If you are running a distributed deployment of Tableau Server, then enter the host name, or IP address of the computer running the initial node. The credentials you enter must have administrative access to the computer running the initial node, as specified above.
- Specify an HTTPS protocol in the server URL. Tableau Server installs a self-signed certificate as part of the installation process. Therefore, the protocol must be specified as `https`. For more information about the self-signed certificate and certificate trust for TSM connections, see [Connecting TSM clients](#).
- Specify the port for TSM web UI (`8850`) in the URL.
- If you are running a local firewall, open port 8850. See [Configure Local Firewall](#).
- You must specify the hostname or IP address of the computer running TSM. If you have set up a load balancing or proxy solution in front of Tableau Server, do not specify the load balancer or proxy address.

Sign in to the TSM web UI

1. Open a browser and enter the Tableau Server URL, and append the dedicated TSM web UI port.

Here are some examples of what the URL might look like:

```
https://localhost:8850/ (if you're working directly on the server computer)
```

```
https://MarketingServer:8850/ (if you know the server's name)
```

```
https://10.0.0.2:8850/ (if you know the server's IP address)
```

2. In the sign-in page that appears, enter your administrator user name and password.

Note: Tableau Server creates and configures a self-signed certificate during the installation process. This certificate is used to encrypt traffic to the TSM Web UI. Because it's a self-signed certificate, your browser will not trust it by default. Therefore, your browser will display a warning about the trustworthiness of the certificate before allowing you to connect.



The screenshot shows the Tableau Services Manager login interface. At the top is the Tableau logo, which consists of a cluster of plus signs followed by the word 'tableau' in a lowercase, sans-serif font. Below the logo is the heading 'Sign In to Tableau Services Manager'. Underneath the heading is a note: 'Requires administrator access to the computer where Tableau Server is installed.' There are two input fields: one for 'Username' and one for 'Password'. Below these fields is a green button with the text 'Sign In' in white.

Customize Your Server

You can customize the Tableau Server web pages to personalize it for your company or group. You can perform these customizations:

- Change the server name that appears in the browser tab, tooltips, and messages.
- Change the logos that appear in the web environment.

For more information, see [tsm customize](#).

- Set the language used for the web environment and the locale used for views. See [Language and Locale for Tableau Server](#).
- Install custom fonts on Tableau Server and client computers that connect to Tableau Server. See [Use Custom Fonts in Tableau Server](#).
- Add a custom note to the server sign in page. The Sign In setting lets you add text. You can optionally add a URL to make the text a link. This note will also appear if a user receives a sign in error. Custom notes do not display on Tableau Mobile.

To set a custom note, sign in to a site on Tableau Server. On the left-side navigation pane, select **Manage all sites** from the drop-down site list. Select **Settings** and add a message to **Sign In Customization**.

For more information, see [Server Settings \(General and Customization\)](#).

- Add a custom message to the welcome banner on the home page for all server users to see. The custom message can contain up to 240 characters of text and hyperlinks as well as one paragraph break. Administrators can also disable the default Tableau welcome banner for the server.

To set a custom welcome banner, sign in to a site on Tableau Server. On the left-side navigation pane, select **Manage all sites** from the drop-down site list. Select **Settings**, then navigate to the **Customization** page.

Administrators and project leaders can also add images for projects in thumbnail view.

Language and Locale for Tableau Server

Tableau Server is localized into several languages. Server language and locale settings impact how this affects users. The **Language** setting controls user interface (UI) items such

as menus and messages. The **Locale** setting controls items in views such as number formatting and currency.

Administrators can configure language and locale on a server-wide basis and individual users can configure their own settings (search for "Your Account Settings" in the Tableau Server Help). If a user configures their own language and locale, their settings override the server settings.

Supported Languages

Tableau Server is localized into several languages. See the "Internationalization" section of the [Tableau Server Technical Specification](#) page for more information.

Default Settings

The default language for Tableau Server is determined during Setup. If the host computer is configured for a language Tableau Server supports, Tableau Server installs with that language as its default. If computer is configured for a language that is not supported, Tableau Server installs with English as its default language.

How Language and Locale are Determined

Another influence on which language and locale display when a user clicks a view is the user's web browser. If a server user has not specified a **Language** setting on their User Account page, and their web browser is set to a language that Tableau Server supports, the browser's language will be used—even if Tableau Server itself is set to a different language.

Here's an example: Assume that Tableau Server has a system-wide setting of English as the **Language** for all users. Server user Claude does not have a language specified on his Tableau Server User Account page. Claude's browser uses German (Germany) for its language/locale.

When Claude signs in to Tableau Server, the server UI displays in German and when he clicks a view, the view uses the Germany locale for numbers and currency. If Claude had set his user account **Language** and **Locale** to French (France), the UI and view would have

been displayed in French. His user account setting supersedes those of his web browser, and both of those have precedence over the Tableau Server system-wide setting.

Another setting to be aware of is the **Locale** setting in Tableau Desktop (**File > Workbook Locale**). This setting determines the locale of the data in the view, such as which currency is listed or how numbers are formatted. By default, **Locale** in Tableau Desktop is set to **Automatic**. However, an author can override that by selecting a specific locale. Using the above example, if the author of View A set **Locale** to **Greek (Greece)**, certain aspects of the data in View A would display using the Greek (Greece) locale.

Tableau Server uses these settings, in this order of precedence, to determine language and locale:

1. Workbook locale (set in Tableau Desktop)
2. Tableau Server User Account language/locale settings
3. Web browser language/locale
4. Tableau Server Maintenance page language/locale settings
5. Host computer's language/locale settings

Use Custom Fonts in Tableau Server

You can use custom fonts with Tableau Server. When you do this the safest way to guarantee that users have the experience you intend is to keep the following in mind:

-
- The fonts need to be installed on the computer where Tableau Server is running. After installing the fonts, restart Tableau Server to use the new fonts.
- The fonts need to be installed on any client computers that will connect to Tableau Server. You need to have the fonts installed locally in order for your browser to properly display them.

- As a best practice, use "web safe" fonts that are installed by default on all major browsers. This increases the likelihood that the fonts will display properly on client machines.
- Different browsers render the same fonts differently, so even when a client browser has the custom font installed, it may look different when viewed in different browsers. This can be especially noticeable with comments or titles where specific spacing is used for an intentional effect.

Note: For more information about installing fonts on Linux, refer to your Linux distribution's documentation and support.

Manage Sites Across a Server

You can plan and manage your sites in Tableau Server. You can manage users and groups for your sites, manage projects and content access, manage data, and create and interact with views on the web.

Sites Overview

The topics in this section describe the Tableau Server concept of a site and aspects of working with multiple sites. Topics include authentication type each site uses, and what to know about user licenses and administrator-level access to sites.

What is a site

You might be used to using the term *site* to mean "a collection of connected computers," or perhaps as the short form of "website." In Tableau-speak, we use site to mean a collection of users, groups, and content (workbooks, data sources) that's walled off from any other groups and content on the same instance of Tableau Server. Another way to say this is that Tableau Server supports multi-tenancy by allowing server administrators to create sites on the server for multiple sets of users and content.

All server content is published, accessed, and managed on a per-site basis. Each site has its own URL and its own set of users (although each server user can be added to multiple sites). Each site's content (projects, workbooks, and data sources) is completely segregated from content on other sites.

For site administrator recommendations for how to set up users on a site, how to structure a site for publishers and other content users, how to give users permissions to share and manage their content, and so on, see the [Manage Individual Sites](#) section.

For information about how users can get their content to Tableau Server, see [Publish Data Sources and Workbooks](#) in the Tableau user help.

Authentication and sign-in credentials

By default, all sites on a server use the same identity store type. You configure these settings when you install Tableau Server. For information, see [Configure Initial Node Settings](#).

Users who have access to more than one site on the same Tableau Server instance use the same credentials for each site. For example, if Jane Smith has a user name of *jsmith* and a password of *MyPassword* on Site A, she uses those same credentials on Site B. When she signs in to Tableau Server, she'll be able to choose which site she wants to access.

The Default site

Tableau Server installs with a site named Default. If you maintain a single-site environment on Tableau Server, this becomes the site you work with, and on which your users share their Tableau analysis. If you add sites, Default becomes one of the sites you can select when you sign in to Tableau Server. Default differs from sites that you add to the system in the following ways:

- It can never be deleted but, just like sites that you add, it can be renamed.
- It stores the samples and data connections that ship with Tableau Server.
- The URL used for Default does not specify a site. For example, the URL for a view

named Profits on a site named Sales is `http://localhost/#/site/sales/views/profits`. The URL for this same view on the Default site would be `http://localhost/#/views/profits`.

Why or why not add sites

On Tableau Server, users, projects, groups, data sources, and workbooks are managed per site. You can add users to multiple sites.

Each environment and its needs is unique. However, as a baseline, Tableau Zen Masters and Product Managers tend to recommend using sites for true multi-tenancy needs. In other words, create a new site only when you need to manage a unique set of users and their content completely separately from all other Tableau users and content.

For site administrator recommendations for how to set up users on a site, how to structure a site for publishers and other content users, how to give users permissions to share and manage their content, and so on, see the Manage Individual Sites section.

Examples for which it makes sense to use sites

- You are a consultant who manages Tableau analysis for multiple clients, and you want to create a site for each client, to ensure that data from one client is not exposed to another.
- You want to allow Guest user access to a small and contained area of the server.

Examples for which projects can work better than sites

- A content-development process in which data sources and reports evolve from sandbox to production phases.

Migrating users and content from one site to another is a laborious process. Although you might have good reasons to use sites for this and similar processes, by creating sites, you as the site administrator compound your ongoing maintenance burden. For each configuration update you make to one site (for example creating new projects

and setting permissions), you usually would need to duplicate the same work on each additional site.

- You want to separate areas of the server by functional area.

Among a group of Tableau users, it's common that some users need to access content in multiple areas. Using sites would encourage publishing the same data sources and reports to multiple sites. This leads to data source proliferation and can negatively impact server performance. Using projects is a simpler way to work with this scenario.

For additional ideas, see the following resources:

- Why use projects in the topic [User Projects to Manage Content Access](#).
- Discussions about sites on the Tableau Community forums. [Here's a link](#) to get you started.

Administrator-level access to sites

Tableau Server includes three administrator-level site roles: Server Administrator, Site Administrator Creator, and Site Administrator Explorer.

The **Server Administrator** site role always takes the highest license available, and it allows full access to Tableau Server, including all content access. You can find more information about this role in [Server Administrator Overview](#). Server administrators also create sites as needed. (Site administrators don't have permissions to do this.)

A server administrator can assign one of the **Site Administrator** site roles to users to delegate creating and maintaining a specific site's user and content framework. The content framework enables Tableau users to share, manage, and connect to data sources and workbooks.

- Assign **Site Administrator Creator** to administrators who also connect to data, and create and publish data sources or workbooks. This site role takes a **Creator** license.
- Assign **Site Administrator Explorer** if the user manages the content framework

but doesn't need to edit the content itself. This site role takes an **Explorer** license, and it allows viewing and interacting access.

By default, the Site Administrator site roles allow creating and managing the site's users and groups, creating projects to organize content on the site, assigning permissions to allow users (groups) to access the content they need, scheduling extract refreshes, and a few other tasks.

A server administrator can deny site administrators' user management tasks. For example, you might do this if you use the Site Administrator Creator role for the data experts. In other words, you want to allow these users to manage connections to underlying data, create and publish "single source of truth" data sources, create top-level projects, and organize content across projects without restriction; but not necessarily add and remove site users.

For each site the server administrator can also limit site administrator access, so that site administrators can manage groups and content, but not add or remove users or set users' site roles.

In some organizations, the same person might be both a server administrator and site administrator for one or more sites. Even so, the tasks performed by a site administrator and a server administrator are distinct.

Licensing and user limits

You can add server users to multiple sites, and set their site roles and permissions on each site. A user who belongs to several sites does not need a license for each site. Each server user needs only one license. The license that user will use corresponds to the highest site role they have on the server. To learn more about how licenses and site roles intersect, see [Set Users' Site Roles](#).

Server administrators can use the **Limit number of users** setting (select **Site <name> > Settings**) to specify a user limit for the site, or set a site role limit, which limits the number of Creators, Explorers, and Viewers allocated at the site level. To learn more, see [Manage Site Role Limits](#).

Only licensed users are counted. For example, if a site has 90 licensed users (including administrators), 20 unlicensed users, the user count is 90. For information about how to view the number of licenses and site roles across the server, see [View Server Licenses](#).

Export or Import a Site

You can provision a new Tableau Server site by importing (migrating) information from another site. You do this by exporting the existing site's (the *source* site) information to a file. Then you complete steps to verify and import that information to the *target* site.

Site Migration Options

You can migrate a site in any of these ways:

- To another site on the same Tableau Server instance.
- To a site on a separate Tableau Server instance.
- From Tableau Server on Windows to Tableau Server on Linux or vice-versa.

Site Migration Limitations

What information is preserved in a site export

- The export file you create preserves workbooks, projects, data sources, and users. This includes permissions set on content, user favorites lists, and site quotas.
- Users' custom views are preserved; however, depending on the type of site migration, custom view URLs might change in a way that breaks users' bookmarks to their views.
- When you export a site on Tableau Server to import to another Tableau Server site, subscription and extract refreshes schedules are preserved.

What information isn't preserved in a site export

- Usage data, which appears in the site's administrative views, is not preserved. For example, view and data source counts, user actions, and performance data.
- Backgrounder jobs that are in-progress while a site is being exported, will not be exported and will not show up on the new site once the import is complete.
- OAuth access tokens embedded in data connections are reset. For those data sources, you will need to edit the connections and re-authenticate to the underlying data.

Prepare the Source and Target Sites

Before you export a site, complete the following checklist to prepare both environments. Some of these instructions depend on whether both sites are on the same server instance or on separate ones.

Identify the Tableau Server node

The `tsm sites` commands will use your local file store to hold the export and import data. If you are running a multi-node Tableau cluster, then you must run the `tsm sites` commands on a Tableau Server that is running the Data Engine process. For information about the Data Engine process and the processes that require it, see [Tableau Server Processes](#).

Delete stale content

Make sure the source site contains only what you want to import to the new site. Delete unused workbooks, data sources, or projects.

Remove obsolete users

Confirm that all server users are licensed, and remove accounts that are no longer in use. You can't remove users during the import process, so if the two sites are on the same server instance, all users you export from the source site are imported to the target site.

Create or identify the target site

You must import a site file to a site that already exists on the target Tableau Server instance. Because the import process removes anything from the target site that is not included in the import file, we recommend that you import to an empty site. For more information, see [Add or Edit Sites](#).

If your source site has workbooks that use published data sources, the data connections in these workbooks continue to refer to the source site name. To maintain those connections without having to republish the workbooks, make sure the target and source site names match.

Locate site IDs

The `tsm` command you use to export or import a site requires a parameter that takes the site ID. You can get the site ID from the URL when you are signed in to the site from a web browser.

If only one site exists on the server, that site is named **Default**. When you are signed in to the Default site, the browser URL looks something like this:

```
https://server-name/#/projects
```

In the URL, the absence of the `/site` parameter indicates that it's the Default site. The site ID for the Default site is "Default" without the quotation marks.

On a multi-site Tableau Server deployment, the browser URL includes `#/site/` followed by the site ID. The following URL would appear if you navigate to the Views page on a site whose site ID is `finance`:

```
https://localhost/#/site/finance/views
```

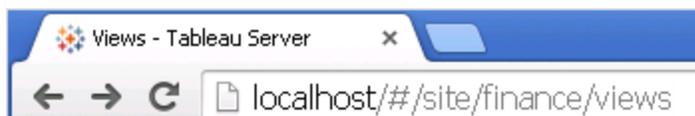


Tableau Server on Linux Administrator Guide

Check the identity store

You can export from and import to sites that do not use the same user identity store type, but you will need to modify the mapping files used for the import. This step is built into the import process and described in Step 3: Verify that site settings are mapped correctly.

Create users on the target server if necessary

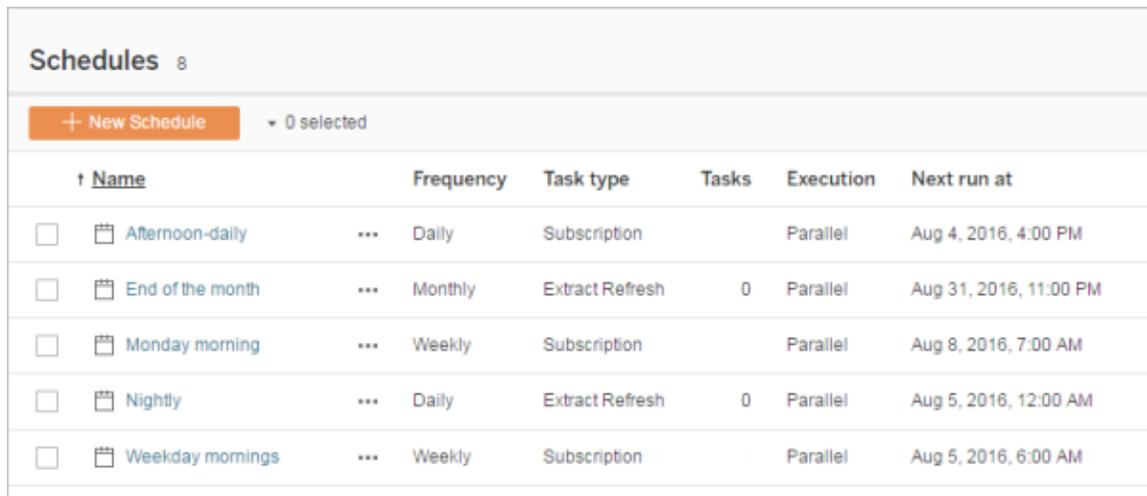
The site import process assigns users to the target site. If the source site is on a Tableau Server instance other than the target site, you must create users on the target server before you can perform the import. If the two sites are on the same Tableau Server instance, the target site has access to the existing users, and you can skip this step.

Configure the target server to deliver subscriptions

Subscriptions are imported, but you must configure the server to deliver them. For more information, see Set Up a Site for Subscriptions.

Check schedules

The **Schedules** page lists the existing schedules for extract refreshes and subscriptions.



The screenshot shows the 'Schedules' page in Tableau Server. At the top, there is a header 'Schedules 8' and a '+ New Schedule' button. Below the header, there is a table with columns: Name, Frequency, Task type, Tasks, Execution, and Next run at. The table contains five rows of schedules, each with a checkbox on the left and a three-dot menu icon to the right of the name.

	Name	Frequency	Task type	Tasks	Execution	Next run at
<input type="checkbox"/>	Afternoon-daily	Daily	Subscription		Parallel	Aug 4, 2016, 4:00 PM
<input type="checkbox"/>	End of the month	Monthly	Extract Refresh	0	Parallel	Aug 31, 2016, 11:00 PM
<input type="checkbox"/>	Monday morning	Weekly	Subscription		Parallel	Aug 8, 2016, 7:00 AM
<input type="checkbox"/>	Nightly	Daily	Extract Refresh	0	Parallel	Aug 5, 2016, 12:00 AM
<input type="checkbox"/>	Weekday mornings	Weekly	Subscription		Parallel	Aug 5, 2016, 6:00 AM

For migrations **from one Tableau Server site to another**, refreshes and subscriptions assigned to default schedules on the source site are mapped to the same schedules on the target site. If the source site has schedules that do not exist on the target site, and the target

site is on another Tableau Server instance, you must create the schedules on the target site that you want the source schedules to map to. You can edit the mapping files to make sure this is done as you expect.

Tips for importing to a target with fewer users or schedules

When a target site has fewer users or schedules than the source site, many-to-one importing is not supported. How you can address this depends on whether the source and target sites are on the same Tableau Server instance.

You can take any of the following approaches that apply to your site migration use case:

- Remove extra users or schedules from the source site before you export. This is the preferred option if the two sites are on the same server instance.
- Add missing users or schedules to the target site before beginning the import. This is required if the target site is on another server instance.
- Add the missing users or schedules to the target site in the middle of the import process and manually update the mapping files. This is an option only if the sites are on the same instance.
- Manually map the users or schedules to different users and schedules in the target site during the import process. This is required if a user name differs between servers—for example, the exported user named *adavis@company.com* is defined on the target site as *davisa*.

Migrating a Site

You must use the `tsm sites` commands to complete the site migration process. The steps below walk you through exporting information from the source site, mapping site settings, and importing the mapped files to the target site.

Step 1: Export a site

On the source Tableau Server machine, type the following command:

Tableau Server on Linux Administrator Guide

```
tsm sites export --site-id <source-siteID> --file <filename>
```

Tableau Server must be running when you use the `export` command. During the export process, Tableau Server locks the site you are exporting.

For example, to export a site with site ID **weather-data** to the file **export-file.zip**, type the following:

```
tsm sites export --site-id weather-data --file export-file
```

By default, Tableau Server saves `<export-file>.zip` to `/var/opt/tableau/tableau_server/data/tabsvc/files/siteexports`. For more information, see [tsm File Paths](#).

Step 2: Generate the import mapping files

To generate import files for the target site, you need the `.zip` file you created when you completed the steps in Step 1: Export a site.

Note: The exported file does not give "Others" read permission by default. Depending on who is importing the file, you may need to adjust the permissions on the file to allow a non-owner to read it.

1. On the target Tableau Server machine, copy the exported `.zip` file to the directory Tableau Server expects to find the files for importing. For example:

```
/var/opt/tableau/tableau_server-  
/data/tabsvc/files/siteimports
```

2. Verify that the target site already exists on Tableau Server, as the import process will not create a new site. For more information, see [Prepare the Source and Target Sites](#).
3. Run the following command on the target Tableau Server machine (Tableau Server must be running):

```
tsm sites import --site-id <target-siteID> --file <export-  
file.zip>
```

This command generates a set of .csv files that show how source site settings will map to the target site. In the steps described in the next section of this article, you confirm these mappings and adjust them where needed.

By default, these .csv files are generated to a `mappings` directory created under `siteimports`. For example:

```
/var/opt/tableau/tableau_server-  
/data/tabsvc/files/siteimports/working/import_<id>_<date-  
time>/mappings
```

For more information, see [tsm File Paths](#).

Step 3: Verify that site settings are mapped correctly

The .csv files you generated in the previous section describe how the source site's resources will be assigned to the target site when the import is complete. Items in the files that Tableau Server was unable to map, and that you need to edit, are indicated by a series of question marks (???). Before you can complete the import process, you must replace the question marks with valid assignments on the target site.

Important: Some requirements apply to mapping users, schedules, and published content resources, particularly when the source and target sites are on separate Tableau Server instances. For more information, see [Prepare the Source and Target Sites](#) earlier in this article.

To verify mapping files

1. Navigate to the directory that contains the .csv map files generated by the `tsm sites import` command. By default:

Tableau Server on Linux Administrator Guide

```
/var/opt/tableau/tableau_server-  
/data/tabsvc/files/siteimports/working/import_<id>_<date-  
time>/mappings
```

2. Use your preferred text editor to open one of the .csv files in the `mappings` directory, and do the following.
 - a. Confirm that the mappings are correct.
 - b. If an entry shows a series of question marks (???), replace them with a valid value.

For descriptions of the settings in each of these files, use the tables in Mapping File Content Reference later in this article.

- c. Save the changes and preserve the CSV file formatting.

Repeat this process for the remaining .csv files.

Step 4: Import the correctly mapped files to the target site

After you verify the site mappings in the .csv files, you can import the settings to the new site to complete the migration process.

1. Run the following command on the target Tableau Server machine:

```
tsm sites import-verified --import-job-dir <import-id-dir-  
ectory> --site-id <target-siteID>
```

For example:

```
tsm sites import-verified --import-job-dir /var/-  
opt/tableau/tableau_server-  
/data/tabsvc/files/siteimports/working/import_ff00_  
20180102022014457  
--site-id new-site
```

- When the success message appears, sign in to the new site and confirm that everything was imported as you expected.

Note: The `tsm sites import` and `tsm sites export` commands can leave a site in a locked state if an error occurs. To unlock a site, use the `tsm sites unlock` command.

Mapping File Content Reference

The following tables list the columns in each of the mapping files created when you run the `tsm site import` command.

CSV file name: `mappingsDomainMapperForGroups`

Column title	Can it be edited?	Description
<code>source_name</code>	No	A user group name on the source site.
<code>source_domain_name</code>	No	The identity store type on the source site: either local (for local identity store) or a domain name (for Active Directory or LDAP external identity store).
<code>target_domain_name</code>	Yes*	The identity store type on the target site: either local for local identity store, or a domain name (such as <code>example.com</code> or <code>example.lan</code>) for Active Directory or LDAP external identity store. *For the All Users group, keep the target_domain_name value set to local , even if your target server is configured for Active Directory identity store. The All

	<p>Users group is a special default user group that must exist on every Tableau Server.</p>
--	--

CSV file name: mappingsScheduleMapper

Column title	Can it be edited?	Description
source_name	No	The names of custom and default extract or subscription schedules on the source site.
source_scheduled_action_type	No	The type of schedule, either Refresh Extract , for extract refreshes, or Subscriptions , for subscription deliveries on the source site.
target_name	Yes	The names of custom schedules on the target site. You can edit this value. For example, if the schedule is named Friday Update on the source site you can rename it Friday Refresh on the target site.
target_scheduled_action_type	No*	<p>The type of schedule, either Refresh Extract, for extract refreshes, or Subscriptions, for subscription deliveries on the target site.</p> <p>*In the rare case that you see question marks (???) in this column, replace them with either Refresh Extract or Subscriptions, to match the entry you see under source_scheduled_action_type.</p>

CSV file name: mappingsSiteMapper

Column title	Can it be edited?	Description
source_url_namespace	No	The site ID of the source site.
target_url_namespace	No	The site ID of the target site.

CSV file name: mappingsSystemUserNameMapper

Column title	Can it be edited?	Description
source_name	No	The user name attribute of a user on the source site.
source_domain_name	No	The identity store type on the source site: either local (for local identity store) or a domain name (for Active Directory or LDAP identity store), or external .
target_name	Yes	<p>The user name attribute for users who will be assigned to the target site upon import.</p> <p>Confirm that all the user names in the list exist on the target server, and replace question marks (???) with user names that exist on the target server.</p> <p>You cannot create user names by adding rows to the CSV file. Similarly, you cannot remove user names by deleting rows.</p> <p>You can edit a user name in the target_name column to be different from its</p>

		<p>source user name, as long as the user already exists on the target server with that name.</p> <p>For example, a user can have a source_name value of agarcia@company.com and a target_name value of ash-leygarcia@company.com.</p> <p>You can map a user on the source site to only one user name on the target site.</p>
target_domain_name	Yes	The identity store type on the target site: either local (for local identity store) or a domain name (for Active Directory or LDAP external identity store).

CSV file name: MappingsScheduleRecurrenceMapperWithAutoCreation

This file does not require updates.

Add or Delete Sites

Tableau Server comes with one site named Default. Server administrators can add or delete sites as the needs of an organization change.

Add a site

1. Do one of the following:
 - If you're adding a site to the server for the first time, select **Settings > Add a Site**, and then click **Add a Site**.

Host Multiple Sites on Tableau Server

Create independent sites for different organizations on a single installation of Tableau Server.

Each site's workbooks, data, and user lists are isolated from those of other sites. As the server administrator, only you can see every site and perform actions such as creating sites and making server-wide changes.

[+ Add a Site](#)

- If you've added sites before, in the site menu, click **Manage All Sites**, and then click **New Site**.

Sites 9				
+ New Site 1 selected Actions				
	Name	Users	Site administrators	Max users
<input type="checkbox"/>	Customer Support	4	2	Server limit
<input type="checkbox"/>	Default	63	8	Server limit
<input type="checkbox"/>	Development	4	2	Server limit
<input type="checkbox"/>	Documentation - 20 User Limit	5	1	20
<input type="checkbox"/>	Finance	13	2	Server limit

2. [Edit the site's settings](#) to customize it for your organization.

Delete sites

Server administrators can delete sites that have been added to Tableau Server. Deleting a site also removes workbooks and data sources that were published to the site, as well as users. If a user belongs to additional sites, they will not be removed. To permanently delete a user, go to the Server Users page.

Note: The Default site cannot be deleted.

1. On the site menu, click **Manage all sites**, and then click **Sites**.
2. Select the site you want to remove, and then on the **Actions** menu, click **Delete**.

3. Click **Delete** in the confirmation dialog box that appears.

Site Availability

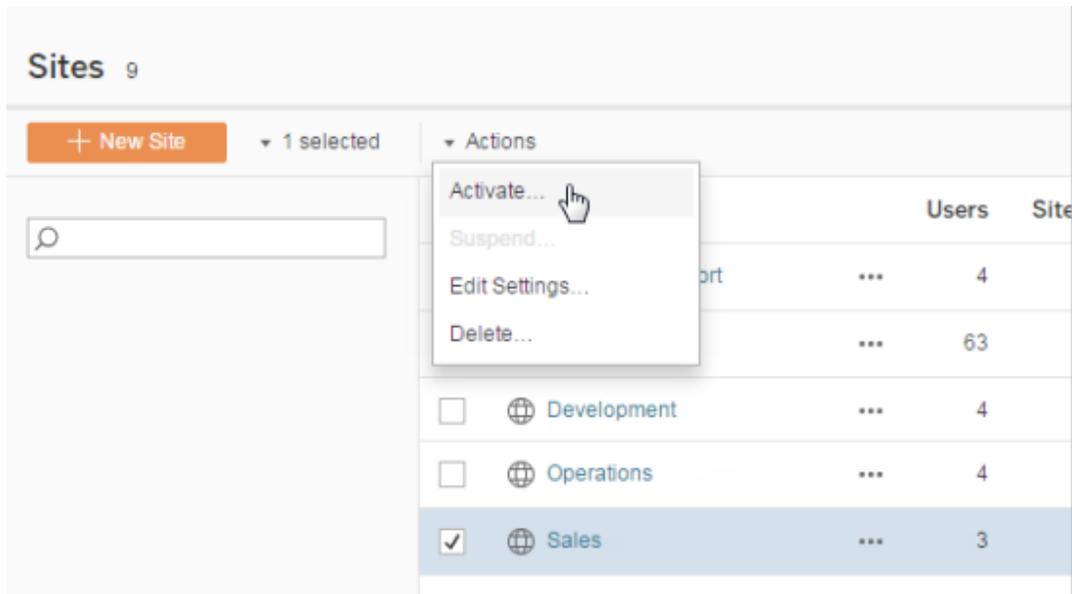
A site can become suspended or locked due to a site import failure, or because a server administrator chooses to suspend the site for a period of time.

When a site is suspended, only the server administrator can activate the site to make it available again.

Note: If a site becomes locked and you cannot access the Sites page through the Server interface, use the `tsm sites unlock` command to change the state to active.

To activate or suspend a site

1. In the site menu, click **Manage All Sites**, and then click **Sites**.
2. Select the site, and then select **Actions > Activate** or **Suspend**.



Manage Site Role Limits

Server administrators can create site role limits to set a maximum number of licenses of each type (Creator, Explorer, or Viewer) that can be consumed on a given site. After a server administrator sets a site role limit, site administrators can add users (thereby automatically consuming licenses) up to that site role limit. If a user is a member of multiple sites across the server, then that user will count against each site's role limit, but will only consume one license (which corresponds to the highest site role they have on the server). Server administrators do not count against site role limits.

To learn more about the capabilities of each site role, see [User-based licenses](#) in the Tableau Server help [Licensing Overview](#). For information for server administrators on how to set up site role limits, see [Site Settings Reference](#).

Create role limits on a site

Before you can enable site administrators to manage their users, you can [View Server Licenses](#) to determine how to allocate licenses across the Server, or [Add Capacity](#) for new users to the Server.

Server administrators can configure site role limits through the [Settings](#) page in the web UI, or through the [REST API](#).

To set a site role limit on a site:

1. Go to the General tab of the Settings page for your site.
 - If you have a single site, on the side navigation, click Settings and General.
 - If you have multiple sites, select the site you want to configure and click Settings and General.
2. Under Managing Users, specify that Server and site administrators can add and remove users.
3. Under Limit the number of users to: select Site Role Limit
4. Set a limit for Creators, Explorers, and Viewers.
5. Click Save

Server administrators can set site role limits within the following restrictions:

Tableau Server on Linux Administrator Guide

- A site role limit cannot exceed the number of licenses of that type that have been activated on the server
- A site role limit cannot be less than the number of users of that site role already present for that site
- If a limit is set for any license type, a limit must be set for every license type.
- If the limit for a role is left blank, the server license limit is used.

When site role limits are met

If a user is added to a site as an Explorer, they will consume an Explorer license, unless the site has met its role limit set for Explorer. When this happens, a few different things can occur:

- If there are available licenses at a higher tier, the user will be added to the site as an Explorer, but will consume a Creator license.
- If no higher licenses are available, the user is added to the site as an Unlicensed user.

Site role limits can be affected by users consuming licenses that are different than their given site roles (Viewers consuming Explorer licenses, for example). In this case, administrators can unlicense those users, then update the site role quotas. For more information on how to troubleshoot licensing issues, see [Troubleshoot Licensing](#).

Allow Users to Save Revision History

Revision history enables your users to see how workbooks and data sources (content resources) have changed over time. Each time someone saves (publishes) a content resource, Tableau Server creates a new version, which becomes the current version. It makes the previous version the most recent revision in the revision history list. Revision history gives users confidence to experiment with their content, knowing that their older versions are available.

Notes

- This information applies to Tableau Server, and is for server administrators who want to allow publishers to work with revisions.
- On Tableau Online, workbook and data source revision history are enabled on all

sites. Users can save up to 10 revisions.

- For information about working with the content revisions themselves, including potential issues, see [Work with Content Revisions](#) in the User/Analyst section of the Tableau help.

Permissions users need to work with revision history

To access revision history, a user must have a site role of **Creator** or **Explorer (Can Publish)**, plus the following permissions, depending on the content type:

- Project: **View** and **Save**
- Workbooks in the project: **View**, **Save**, and **Download Workbook/Save As**
- Flows in the project: **View**, **Save**, and **Download Flow/Publish As**
- Data sources in the project: **View**, **Save**, and **Download Data Source**

Enable revision history and set the number of revisions allowed

Revision history is set at the site level, and is enabled by default, with a limit of 25 revisions for each content resource.

1. Sign in to a site as a Server Administrator, and click **Settings**.
2. Under **Revision History**, select **Save a history of revisions**, and enter the maximum number of revisions you want to allow for each content resource.
3. Click **Save**.

When you lower the number of revisions, the most recent revisions are saved. For example, if you set the limit to 15, the 15 most recent versions of the workbook or data source are saved.

Note: A content resource's revision history list might not show the changes to the limit until a background cleanup process runs on the server.

Clear all revisions

Server administrators can delete all previous revisions of published workbooks and data sources from a site. The most recent version of each published workbook and data source is always retained.

1. Sign in to a site as a Server Administrator, and click **Settings**.
2. Under **Revision History**, click **Clear Revision History**.
3. Click **Save**.

Security for previewing and restoring workbooks

When users select **Restore** or **Preview** for workbook revisions, user passwords are exchanged between the user's browser and the server. Tableau Server encrypts these passwords using public/private key encryption. To ensure these public keys are provided by Tableau Server, you must configure the server to use SSL (HTTPS). For more information, see [SSL](#).

See also

[Potential revision history issues](#) in the User/Analyst section of the Tableau help.

Extract Refresh Schedules

Tableau Desktop authors and data stewards can create and publish *extracts*. Extracts are copies or subsets of the original data. Because extracts are imported into the data engine, workbooks that connect to extracts generally perform faster than those that connect to live data. Extracts can also increase functionality.

Before refreshing extracts

When an extract refresh is performed on extracts created in Tableau 10.4 and earlier (that is, a .tde extract), the extract is upgraded to .hyper extract automatically. While there are many benefits of upgrading to a .hyper extract, you will be unable to open the extract with previous versions of Tableau Desktop. For more information, see [Extract Upgrade to .hyper Format](#).

Setting up refresh schedules

As a server administrator, you can enable scheduling for extract refresh tasks, and then create, change, and reassign schedules. General scheduling options you change on the server are available as part of the publishing process when a Tableau Desktop user publishes an extract.

Schedules that you create have the following options:

Priority

The priority determines the order in which refresh tasks are run, where 0 is the highest priority and 100 is the lowest priority. The priority is set to 50 by default.

Execution mode

The execution mode indicates to the Tableau Server background processes whether to run refreshes in parallel or serially. Schedules that run in parallel use all available background processes and serial schedules run on only one background process. However, a schedule can contain one or more refresh tasks, and each task will only use one background process, whether in parallel or serial mode. This means that a schedule in parallel execution mode will use all **available** background processes to run the tasks under it in parallel, but each task will only use one background process. A serial schedule uses only one background process to run one task at a time.

By default, the execution mode is set to parallel, so that refresh tasks finish as quickly as possible. You might want to set the execution mode to serial (and set a lower priority) if you have a very large schedule that prevents other schedules from running.

Frequency

You can set the frequency to hourly, daily, weekly, or monthly.

For information, see [Create or Modify a Schedule](#).

Refreshing extracts manually

In the Tableau Server web environment, both server and site administrators can run extract refreshes on-demand on the **Schedules** page:

- Select the schedule and click **Actions > Run Now**.

You can also refresh extracts from the command line using the `tabcmd refreshextracts` command. For more information, see [tabcmd Commands](#).

Refreshing extracts from Tableau Desktop

Tableau Desktop users can refresh extracts they publish and own. They can do this the following ways:

- **At publish time:** When an author publishes a workbook or data source that uses an extract, that author can add it to server refresh schedule. The refresh can be a full or an incremental refresh.

Incremental refreshes reference a column in the extract that has a data type of date, date/time, or integer; such as a timestamp. Tableau uses this column to identify new rows that need to be added to the extract. For more information, see [Refreshing Extracts](#) and [Schedule Extract Refreshes as You Publish a Workbook](#) in the Tableau Help.

- **User interface:** In Tableau Desktop, you can use the **Refresh from Source, Add**

Data From File, and **Add Data From Data Source** commands to upload an addition to or refresh an extract on Tableau Server. A user might want to do this if Tableau Server doesn't have sufficient credentials to access the underlying data. For more information, see [Updating Extracts on Tableau Server](#) in the Tableau Help.

- **Data Extract command line utility:** The Data Extract command line utility installs with Tableau Desktop. You can use it to append to or refresh a published extract. For more information, see [Tableau Data Extract Command Line Utility](#) in the Tableau Help.

Enable Extract Refresh Scheduling and Failure Notification

Your publishers can schedule extract refreshes when two conditions are met:

- Tableau Server is configured to send email messages when extract refreshes fail. This is configured by a Tableau Server Manager (TSM) administrator and is on by default. For details, see [Configure Server Event Notification](#).
- The site or sites in which you want to allow publishers to schedule extract refreshes is configured to send email when the refresh fails. This is configured by a server administrator in Tableau Server and is on by default. The instructions below explain how to do this if it is not enabled.

While you're enabling scheduling, you can decide whether also to enable sending email to owners of data sources or workbooks that are refreshed when those extract refreshes do not complete successfully. You can read more about these emails below. When you enable refresh failure notification, the owners of the content that has scheduled refreshes can opt out individually by changing their account settings.

1. Sign in to Tableau Server as a server administrator.
2. Go to the General tab of the Settings page for the site you want to configure for subscriptions:
 - If you have a single site, at the top the browser window, click **Settings** and **General**.

- If you have multiple sites, select the site you want to configure and click **Settings** and **General**.

3. On the **General** page, do the following:

- Under **Notifications for Extract Jobs and Flow Runs**, select or clear the **Send notifications to data source, flow, and workbook owners** check box.

To clarify, if a scheduled refresh for a particular data source fails, the email goes only to the owner of that data source, not to owners of workbooks that connect to that data source.

- Under **Embedded Credentials**, select both options to let publishers embed credentials and schedule extract refreshes. (Automatic refresh schedules require embedded credentials so Tableau Server can directly access data.)

Note: On a multi-site server, failure notifications are a site setting, and embedded credentials are a server setting. To configure embedded credentials on a multi-site server, select **Manage All Sites** and then **Settings**.

Managing schedules from the server

In your organization it might be more appropriate to manage embedded credentials and refresh schedules centrally from the server. If you do that, you might clear the check boxes in the **Embedded Credentials** section described in the steps above, so that Tableau Desktop publishers do not see schedule options during publishing.

Managing schedules centrally enables you to distribute extract refresh and subscription tasks, so you can run them when most people are offline. It also enables you to oversee which credentials are embedded in connections.

For more guidelines for managing schedules and refreshes on the server, see [Provide access to data sources](#) and [Keep data fresh](#) in *Everybody's Install Guide*.

How refresh failure emails work

The email notification for a failed extract refresh lists the extract name and location on the server, gives the time of last successful refresh, the number of consecutive times the refresh has failed, and suggests the reason for the failure and possible solution.

After five consecutive failures, the refresh schedule is suspended until you or the data owner takes an action to address the cause of the failure, such as updating database credentials or a path to the original data file.

How the last successful refresh date is determined

The last successful refresh date and time are shown when that last refresh occurred within a number of days. By default it is 14 days, and this value is set in `wgserver.alerts.observed_days`. If the number of days since the last successful refresh exceeds the number specified in this setting, the message in the email shows “not in the last *N* days.”

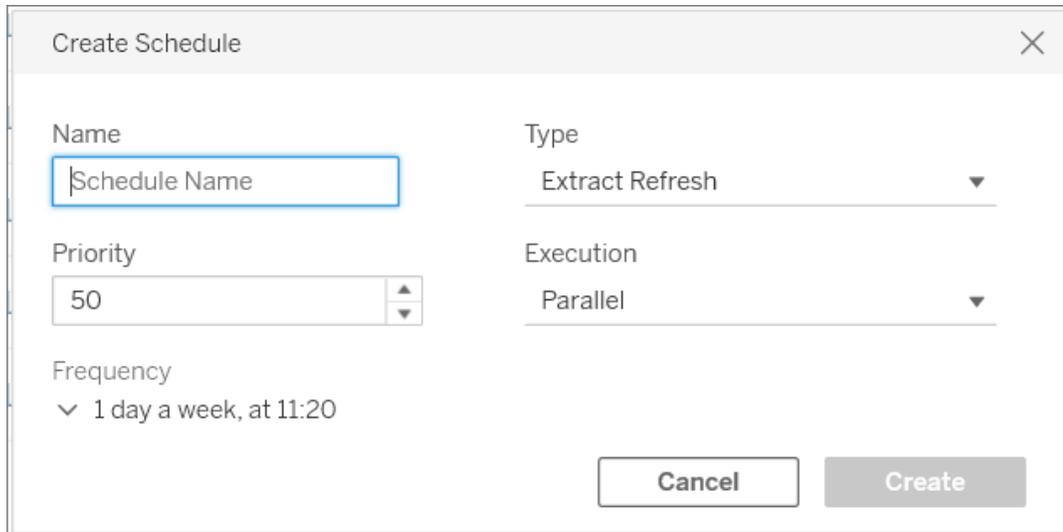
Create or Modify a Schedule

The Schedules page is accessible only by Tableau Server Administrators. It shows a list of schedules, including their name, type, what they're for (scope), the number of tasks, behavior (concurrent or serial processing), and when they are scheduled to run.

Note: If you enable custom schedules for subscriptions on one or more sites, users will control schedules for their subscriptions on those sites. To learn more, see [Enable Custom Schedules for Subscriptions](#).

To create a new schedule

1. In a site, click **Schedules**.
2. Click **New Schedule**.



3. Specify a descriptive **Name** for the schedule.
4. Select a **Task type** the schedule will handle—refreshing extracts, running flows, or delivering subscriptions.
5. You must define a priority from 1 to 100, where 1 is the highest priority. This is the priority that will be assigned to the tasks by default. If two tasks are pending in the queue, the backgrounder will evaluate the task priority for extract refreshes and flows, and the schedule priority for subscriptions to determine which one runs first. For more information, see [How Scheduled Server Jobs are Prioritized](#).
6. **Execution**: choose whether a schedule will run in parallel or serially. Schedules that run in parallel run on all available backgrounder processes so that they can complete faster.

Note: Schedules for the same workbook will always run serially, even if you set this option to parallel.

7. Finish defining the schedule. You can define an hourly, daily, weekly, or monthly

schedule. The **Frequency** is populated automatically based on the selections you make.

The screenshot shows the 'Create Schedule' dialog box with the following settings:

- Priority: 50
- Execution: Parallel
- Frequency: ^ 3 times a month at 11:20
- Repeats: Monthly
- On: Day

The calendar grid shows the following dates selected:

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	Last			

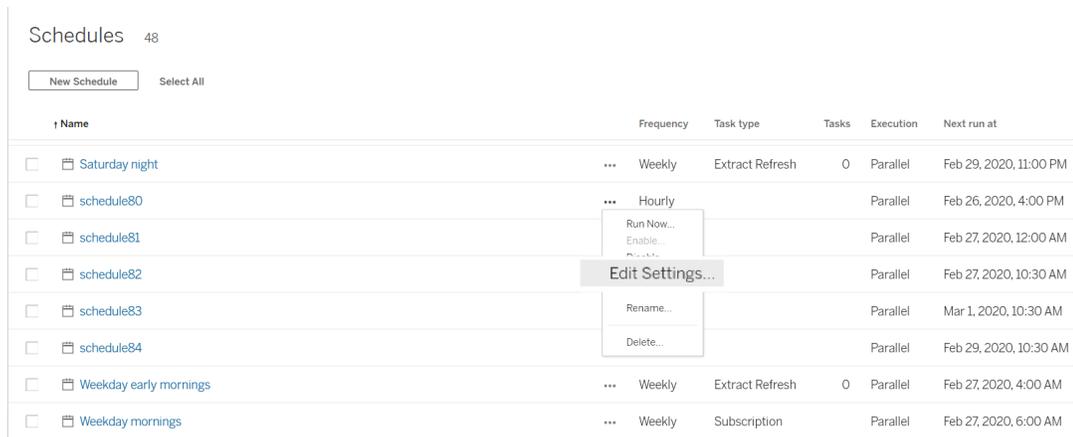
The 'Day' dropdown menu shows the selected days: 16 | 20 | 24

8. Click **Create**.

To modify an existing schedule

1. Navigate to the Schedules page.
2. Select an existing schedule, click the Actions drop-down arrow, and then select **Edit Settings**.

Tableau Server on Linux Administrator Guide



Schedules 48

New Schedule Select All

Name	Frequency	Task type	Tasks	Execution	Next run at
<input type="checkbox"/> Saturday night	...	Weekly	Extract Refresh	0 Parallel	Feb 29, 2020, 11:00 PM
<input type="checkbox"/> schedule80	...	Hourly		Parallel	Feb 26, 2020, 4:00 PM
<input type="checkbox"/> schedule81				Parallel	Feb 27, 2020, 12:00 AM
<input type="checkbox"/> schedule82				Parallel	Feb 27, 2020, 10:30 AM
<input type="checkbox"/> schedule83				Parallel	Mar 1, 2020, 10:30 AM
<input type="checkbox"/> schedule84				Parallel	Feb 29, 2020, 10:30 AM
<input type="checkbox"/> Weekday early mornings	...	Weekly	Extract Refresh	0 Parallel	Feb 27, 2020, 4:00 AM
<input type="checkbox"/> Weekday mornings	...	Weekly	Subscription	Parallel	Feb 27, 2020, 6:00 AM

Context menu for 'schedule82':

- Run Now...
- Enable...
- Edit Settings...
- Rename...
- Delete...

3. Finish editing the schedule, and click **Save**.

Rules for Creating or Modifying Schedules

Following are rules you must follow when creating new or modifying existing schedules.

Note: If you have existing schedules that do not follow these rules, you must modify them accordingly. Not doing so, may result in unexpected behavior and the tasks may not run at the scheduled time.

- Schedules that run every 15 or 30 minutes must have start and end times that are on the hour. Examples of on the hour: 5:00 AM to 6:00 AM.
- Daily schedules on any recurrence must have the same start and end minute. For example, 10:35 am to 4:35 pm. The hour can be different. However, if the daily schedule is set to only happen once a day, it needs only a start time and not an end time.

See also

Manage Refresh Tasks

Extract Refresh Schedules

[Schedule a flow task](#)

Enable Custom Schedules for Subscriptions

Custom schedules for subscriptions allow users to receive email messages on a schedule that they define, rather than using a fixed schedule defined by a server administrator. Custom schedules for subscriptions have been available to users of Tableau Online since March 2017, and these schedules can now be enabled on a per-site basis in Tableau Server version 2018.2. Enabling custom schedules for subscriptions is a permanent change on any sites where you make this change. Any sites where you don't enable custom schedules remain on fixed schedules defined by a server administrator.

Enable custom schedules

Before you can enable custom schedules on one or more sites, you must first enable custom schedules on Tableau Server, and then enable custom schedules on one or more sites on that server. To learn more about enabling subscriptions on Tableau Server, see [Set Up a Site for Subscriptions](#).

Step 1: Enable custom schedules on Tableau Server

From a command prompt with Tableau administrator permissions, run the following commands:

```
tsm configuration set -k features.SelfServiceSchedules -v true
tsm pending-changes apply
```

This operation will restart Tableau Server.

Step 2: Enable custom schedules on a site

1. Log in to Tableau Server as a server administrator using a web browser:

```
https://<hostname>/#/login
```

2. Browse to the **Site Settings** page for a site, and then enable custom schedules:
 1. Click **All Sites**, and then choose one of the sites from the drop-down list.
 2. Click **Settings**.

3. On the **General** tab, under **Subscriptions**, select the following checkbox: **Permanently convert from fixed schedules created by administrators to custom schedules created by users. (You can't undo this).**
4. Click **Save**.

When custom schedules are enabled on a site, all fixed schedules with a subscription are converted to an equivalent custom schedule.

How Scheduled Server Jobs are Prioritized

Jobs and Tasks

In Tableau Server, users can schedule extract refreshes, subscriptions, or flows to run periodically. These **scheduled items** are referred to as **tasks**. The Backgrounder process initiates unique instances of these tasks to run them at the scheduled time. The **unique instances of the tasks** that are initiated as a result are referred to as **jobs**. **Jobs** are also created for runs that are initiated **manually**, by clicking the **Run now** option.

For example, an extract refresh task is created to run daily at 9 AM. This is an extract refresh task, and every day at 9 AM, a job will be created for the Backgrounder to run.

You can assign a priority number to Tasks and Schedules using values from 1 to 100. Lower the number, higher the priority, 1 is the highest priority, and 100 is the lowest.

Priority Rules for Jobs

When processing scheduled extract refreshes, subscriptions and flow runs, Tableau Server prioritizes background jobs in this order:

1. Any job already in process is completed first.
2. Any task or schedule that you initiate manually using **Run now** starts when the next backgrounder process becomes available. Exception to this are the flow tasks and schedules. Flow runs use the assigned task priority to determine the order in when they should run. If there is no task priority assigned it defaults to 0 which is the highest

priority.

Note: Learn how to restrict Run now settings- Server Settings (General and Customization).

- Jobs with the highest priority (the lowest number) start next, independent of how long they have been in the queue.

For extract refreshes and flows, this is the task priority. The task priority is inherited from the schedule priority when the task is first created. The task priority can be subsequently changed but the task priority returns to the default value when the data source is republished.

For subscriptions, this is the schedule priority. If you have enabled custom schedules for subscriptions, then the priority of those jobs is set to 50.

For example, a job with a priority of 20 will run before a job with a priority of 50, even if the second job has been waiting longer. To change task priority, see [Create or Modify a Schedule](#).

- Jobs with the same priority are executed in the order they were added to the queue. The first job added to the queue starts first; then the second job starts.
- When multiple jobs with the same priority are scheduled to run at the same time, they start in the order they were created or enabled. Jobs scheduled for the same time are executed by task type with the fastest category of jobs starting first: flow runs, followed by data driven alerts, followed by system jobs, followed by subscriptions, followed by extract creation, followed by incremental extracts, and then full extracts.
- As the last tie breaking measure, the Backgrounder uses the historical run time. Jobs that have run faster in the previous runs, will be prioritized over jobs that have taken longer historically.

The following limitations also impact when the jobs are run:

- The number of concurrent jobs is limited to the number of backgrounder processes you have configured for Tableau Server.
- Separate refreshes for the same extract or data source cannot run at the same time.
- Jobs associated with a schedule that is set to run serially run one at a time.

Configure Workbook Performance after a Scheduled Refresh

To improve the load times for workbooks, Tableau Server caches the results of queries included in workbooks. For most workbooks, query results are computed and cached when they are first viewed by a user on Tableau Server. However, for workbooks that connect to data extracts, Tableau Server can recompute query results when the corresponding extract refresh tasks run. This reduces the load time for these workbooks when they are first viewed, so this option is turned on by default for workbooks that have been viewed recently.

Determine the performance impact

Although this option reduces the initial load time for workbooks, recomputing query results also increases the load on Tableau Server. If your Tableau Server installation is already performance-constrained, you might want to turn this option off or lower the threshold for workbook caching.

Here are some possible reasons why you might want to turn this option off or lower the threshold:

- The Background Tasks for Non Extracts administrative view displays many long-running jobs in the **Warming up external query cache on data change** category.
- The Background Task Delay administrative view displays long delays.
- CPU and memory consumption for the backgrounder processes is consistently high.

However, note that this is only one of the options that impacts the performance of background tasks. For more information about performance, see [Performance](#).

Turn off workbook caching for the server

To decrease the load on Tableau Server, you can turn off workbook caching after a scheduled refresh at the server-level. If you turn this option off, Tableau Server caches query results for workbooks the first time the workbooks are viewed.

Use the following tsm configuration set option to turn off workbook caching after a scheduled refresh:

```
backgrounder.externalquerycachewarmup.enabled
```

For more information on how to use and apply tsm set options, see [tsm configuration set Options](#).

Turn off workbook caching for a site

You can also turn off workbook caching after a scheduled refresh for an individual site. For example, you might do this if there is one site in particular that contains many slow workbooks which increase load on the server.

1. Select the site for which you want to turn off workbook caching in the sites dropdown.
2. Click **Settings**.
3. In the **Workbook Performance after a Scheduled Refresh** section, clear the check box.

Note: Although this option is available in the settings for an individual site, you must have server administrator permissions to view it.

Configure the workbook caching threshold

Tableau Server only recomputes query results for workbooks that both have scheduled refresh tasks and have been viewed recently.

You can increase or decrease the number of workbooks that are cached after a scheduled refresh with the following tsm configuration set option:

```
backgrounder.externalquerycachewarmup.view_threshold
```

By default, the threshold is set to 2.0. The threshold is equal to the number of views that a workbook has received in the past seven days divided by the number of refreshes scheduled in the next seven days. (If a workbook has not been viewed in the past seven days, it is unlikely that it will be viewed soon, so Tableau Server does not spend resources recomputing queries for the workbook.)

Ensure Access to Subscriptions and Data-Driven Alerts

To ensure that users see the Subscribe and Alert buttons in the Tableau Server toolbar and can receive related emails, do the following:

- **Configure SMTP and event notifications on Tableau Server:** See [Set Up a Site for Subscriptions](#).
- **Ensure that users have an email address in Tableau Server:** Users can update their email address on [their account settings page](#).
- **Embed database credentials or don't require them:** To email data in a view, Tableau Server needs to access the data without user involvement. This can be accomplished by using a workbook with embedded database credentials, a Tableau Server data source, or data that doesn't require credentials (such as a file that's included with the workbook at publish time).
- **Ensure that users can access needed workbooks and views:** Access to workbooks and views on the server is controlled by the **View** permission. To receive

images of content in email messages, users also need the **Download Image/PDF** permission. For more information, see [Permissions](#).

- **Avoid trusted authentication for embedded views:** If you use restricted tickets (the default) to render an embedded view, the Subscribe and Alert buttons don't appear.

(Alerts only) Ensure that users can access published data sources with View and Connect capabilities.

Set Up a Site for Subscriptions

When users subscribe to a workbook or view, a snapshot of the view is email to them on scheduled basis, so they can see the latest updates without having to sign into Tableau Server. Administrators, project leaders with appropriate site roles, and content owners have the option to subscribe other users to workbooks and views. For more information, see [Subscribe to Views](#).

Note: To create and receive subscriptions, users need access to related databases and views. [See this list of requirements](#) for details.

Looking for Tableau Server on Windows? See [Setup a Server for Subscriptions](#).

Prerequisite: Configure the server to send subscription emails

Before you can enable subscriptions for a site, you need to complete the steps to enable subscriptions on the server. Follow the steps in these topics to configure subscriptions on the server.

1. [Configure SMTP Setup](#)
2. [Configure Server Event Notification](#)

Enable subscriptions

After you have configured SMTP and server event notifications, you can enable subscriptions.

To enable subscriptions:

1. Sign into Tableau Server as a server administrator.
2. Go to the General tab of the Settings page for the site you want to configure for subscriptions:
 - If you have a single site, on the side navigation, click **Settings** and **General**.
 - If you have multiple sites, select the site you want to configure and click **Settings** and **General**.
3. On the General tab, scroll down to **Subscriptions** and select the subscription options for your users.

Note: Subscription options are only visible after the TSM administrator has enabled the server-wide configuration option, **Allow users to receive email for views that they have subscribed to**. For details, see Configure Server Event Notification.

- a. Select **Let users subscribe to workbooks and views**
 - b. (Optional) To allow content owners to subscribe other users to their content, select **Let content owners to subscribe other users**.
 - c. (Optional) To allow users to include attachments with their subscriptions, select **Let users add attachments to subscribed workbooks and views**. This option will not be available if the TSM administrator has not enabled attachments in TSM. For details, see Configure Server Event Notification.
4. (Optional) Scroll to **Email Settings**.

- a. Enter an **Email From Address** that will show as the "From" address in email messages.
- b. Enter an **Email Footer** for email messages.

A site's "From" address and message footer are also used in emails for [data-driven alerts](#).

5. Click **Save**.

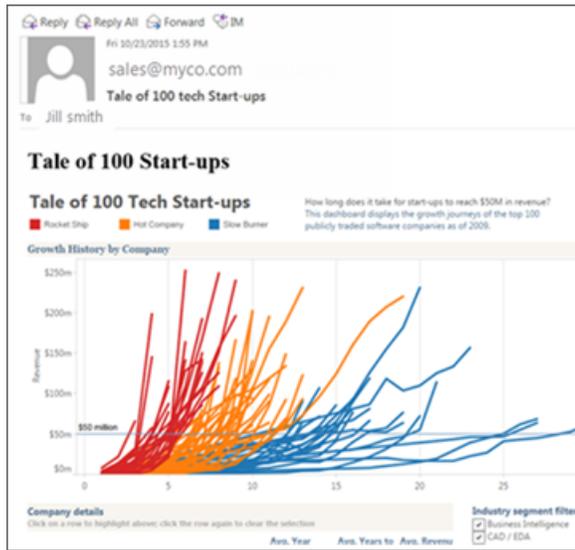
To specify the subscription schedules available to users, see [Create or Modify a Schedule](#).

Test subscriptions in a site

1. [Subscribe to a view](#).
2. In the site with the subscription you want to test, on the side navigation, click **Schedules**.
3. Select the schedule you chose for the subscription, and then click **Actions > Run Now**.

A snapshot of the view should be emailed to you within 10 minutes. If you experience

an issue, see [Troubleshoot Subscriptions](#).



Manage all user subscriptions

1. In the side navigation, click **Tasks**, and then click **Subscriptions**.

All user subscriptions for the current site appear, including information like subscriber name, view name, and delivery schedule.

2. Select any subscription you want to update. From the **Actions** menu, select **Change Schedule**, **Change Subject**, **Change Empty View Mode**, or **Unsubscribe**.

(The empty-view option sends subscription emails only when data exists in a view. It's a good choice for high-priority alerts.)

Suspended Subscriptions

By default, a subscription is suspended after 5 consecutive subscription failures and result in the subscription emails not sent. To change the threshold number of subscription failures that can occur before they are suspended, use the tsm configuration set option, `backgrounder.subscription_failure_threshold_for_run_prevention`. This sets the threshold for the number of consecutive failed subscriptions necessary before suspending the subscription. This is a server-wide setting.

Only Server administrators can configure the threshold number of subscription failures before a subscription is suspended.

Server administrators can opt in to receive email notifications when a subscription is suspended. You can do this by navigating to **My account settings -> Subscription Notifications**. This setting is at a site-level so has to be configured for site separately.

Resume suspended subscriptions

If a subscription fails more than five times, you'll receive a notification email that your subscription has been suspended. There are a few ways to resume a suspended subscription if you're a subscription owner or administrator:

- From the **My Content** area of Tableau web pages, an icon appears in the Last update column to indicate that the subscription is suspended. **Select ... > Resume Subscription** to resume.
- From the **Subscriptions** tab of the affected workbook, an icon appears in the last update column to indicate that the subscription is suspended. **Select ... > Resume Subscription** to resume.
- From the **Subscriptions** tab under **Tasks**, an icon appears in the last update column to indicate that the subscription is suspended. **Select ... > Resume Subscription** to resume (Server administrators only).

When a subscription is resumed, the alert failing count goes back to zero. The next evaluation of the subscription will occur at the next scheduled evaluation time.

See also

[Subscribe to Views](#) in the Tableau Desktop and web authoring Help.

Project-level administration to learn which site roles allow full Project Leader capabilities.

Set Up for Data-Driven Alerts

When data reaches important thresholds for your business, data-driven alerts automatically send email notifications to key people users specify. As a Tableau Server administrator, you set up data-driven alerts much like you do subscriptions. For information about how users create and manage these alerts, see [Send Data-Driven Alerts](#) in Tableau User Help.

Note: To create and receive data-driven alerts, users need access to related databases and views. [See this list of requirements](#) for details. If alerts are enabled for a site, any user on that site can create them.

Configure email for data-driven alerts

1. Complete the steps in [Configure SMTP Setup](#) so the server can send email.
2. While viewing a site, click **Settings** at the top of the browser window.
3. Under Email Settings, enter a site-specific "From" address or message footer.

A site's "From" address and message footer are also used in emails for subscriptions.

4. Click **Save**.

Manage all data-driven alerts in a site

1. At the top the browser window, click **Tasks**, and then click **Alerts**.
2. Select any alerts you want to update.
3. From the **Actions** menu, do any of the following:
 - Add or remove yourself as a recipient.
 - Edit alerts to change data thresholds, delivery schedules, and the full list of

recipients.

- Change alert ownership to different users, or delete alerts.

Disable data-driven alerts for a site

Data-driven alerts are supported for all sites by default, but administrators can disable them for specific ones.

1. While viewing a site, click **Settings** on the left side navigation pane.
2. Under Data-Driven Alerts, uncheck **Let users create alerts and receive alert emails**.
3. Click **Save**.

Suspend data-driven alerts

By default, an alert is suspended after 350 consecutive alert failures. Server administrators can configure the threshold number of alert failures before an alert is suspended. To change the threshold number of data-driven alert failures that can occur before alerts are suspended, use the [tsm configuration set](#) option, `dataAlerts.SuspendFailureThreshold`.

This sets the threshold for the number of consecutive failed alerts necessary before suspending the alert. This is a server-wide setting. The threshold value applies to every configured data-driven alert on the server.

Resume suspended alerts

If an alert fails enough times, you'll receive a notification email that your alert has been suspended. There are a few ways that administrators or alert owners can resume a suspended alert:

- From the Tasks > Alerts area of Tableau web pages, an  icon appears in the Last checked column to indicate that the alert is suspended. Select ... > **Resume Alert** to resume the alert.
- Click **Resume Alert** in the notification email to resume the alert. A notification will either allow you to resume the alert, or indicate that the view has changed and the alert should be deleted.
- From the Alerts panel of the affected view or workbook. To resume the alert from a view or workbook, select **Alert** to open the Alerts panel. An icon appears next to the suspended alert. Select **Actions** > **Resume Alert** on the affected alert to resume.

Alert owners will receive an email notification when the alert is working again.

Control how often the server checks data-driven alerts

By default, Tableau Server checks every 60 minutes to confirm whether data conditions for alerts are true. If you notice performance impacts, you can customize this time interval with the [tsm configuration set](#) option, `dataAlerts.checkIntervalInMinutes`.

Regardless of the `dataAlerts.checkIntervalInMinute` setting, the server also checks alerts whenever extracts in the related workbook are refreshed. To check an alert more frequently than the setting specifies, change the extract-refresh schedule.

Track the server's alert-checking process

In the Background Tasks for Non Extracts view, you can track the server's alert-checking process by looking for these tasks:

- Find Data Alerts to Check
- Check If Data Alert Condition Is True

The "Find" task limits "Check" tasks to alerts that can currently send related emails. For example, if a user has chosen an email frequency of "Daily at most", after the alert condition becomes true, the server waits 24 hours before checking the alert again.

Each "Check" task uses one server background process, loading the related view to evaluate the alert condition. If all users see the same version of a view, it loads only once. But if users have applied filters to a view, or the data they see is limited by user-level security, the view loads once for each recipient.

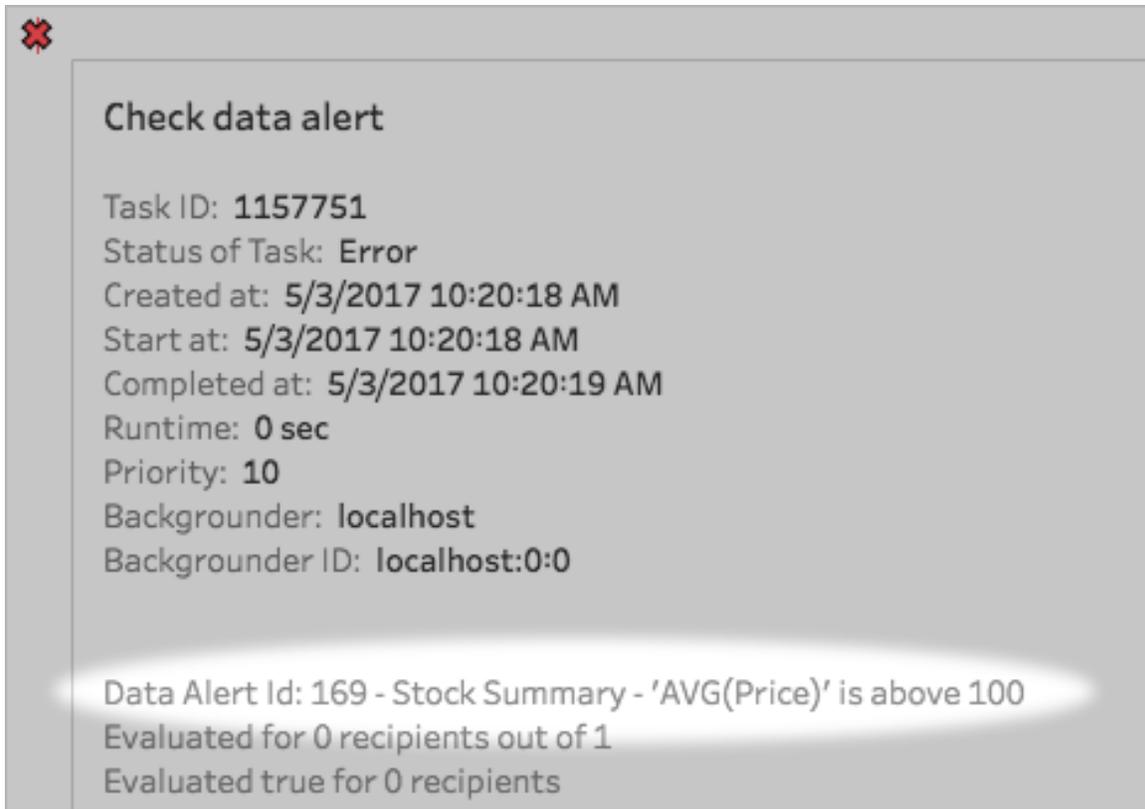
Identify and fix failing alerts

As an administrator, you can proactively identify failing alerts that users may be unaware of. To check:

1. Select Status in your site menu
2. Select Background tasks for non-extracts
3. From the Task drop-down menu, select Check if Data Alert is True
4. In the far right, click on Error to see a list of failing alerts
5. Hover over the red failure icon to display a tooltip with alert details

To determine the alert owner, look for the alert ID number in the `data_alerts` table of the Tableau Server Repository. (In the [alert management](#) area of a site, you can also look for the alert name following the number, but be aware that multiple different alerts may use the same name.)

Note: Alert owners will be automatically notified when an alert fails ten times. Administrators can customize when alert owners receive notifications. Users will not be notified for alerts that failed prior to upgrading to Tableau Server 2018.1.



Failing alerts are often caused by content changes on Tableau Server. Encourage users to recreate alerts if changes like the following occur:

- A workbook, view, or data field is removed or renamed.
- Database credentials embedded in workbooks expire. (Alerts require workbooks to use either embedded credentials, or none at all.)
- A data source becomes inaccessible.

Tip: To automatically get emailed when alerts fail, follow the steps in Collect Data with the Tableau Server Repository, and connect to the "background_jobs" table. From that table, create a custom view that includes the "Check If Data Alert Condition Is True" job

name and its finish code. Then [set up a data-driven alert](#) to email you whenever a finish code equals 1 (failure).

Set Up for Metrics

Metrics are a type of Tableau content that tracks the value of an aggregate measure, such as sum of sales. Because metrics refresh frequently and display their current value in an easy-to-glance format, they are useful for monitoring data. To learn more about how users work with metrics, see [Create and Troubleshoot Metrics](#).

As a Tableau Server administrator, you have the ability to control how often metrics refresh and how failing refreshes are handled. You also can make sure that users are able to create metrics or disable metrics for particular sites.

Ensure that users can create metrics

When metrics are enabled for a site, all users with a Creator or Explorer (can publish) site role can create metrics, if they have the correct permissions.

Metrics are created from existing views on a Tableau site. To ensure that users can create metrics on a view, verify that:

- Users have Download Full Data permission for the view. For more information, see [Permissions](#).
- The password for the data source is embedded, if it is required. For more information, see [Edit Connections on Tableau Server](#).

Disable metrics for a site

Metrics are enabled on all sites by default. You can disable metrics on a per-site basis.

1. On the site where you want to disable metrics, from the navigation panel, click **Settings**.
2. Under **Metrics Content Type**, uncheck **Enable metrics**.
3. Click **Save**.

When you disable the metrics content type, metrics no longer appear on the site. The data for any existing metrics is retained, but these metrics will no longer refresh. If you re-enable metrics, these metrics will reappear and resume refreshing.

Configure how often metrics refresh

When a metric refreshes, it checks for new data via the view it was created from, known as the connected view. You might want to increase the time between refreshes if you notice a performance impact on your server—or decrease it if your users require more up-to-date data.

Metrics that rely on live data refresh every 60 minutes, by default. To adjust the refresh interval for live data, use the **tsm configuration set** option `metricservices.checkIntervalInMinutes`. This is a server-wide setting.

Metrics that rely on extract-based data refresh when the extracts refresh. To control how often these metrics refresh, change the extract refresh frequency. For more information, see [Extract Refresh Schedules](#).

Configure failure notifications for metric refreshes

If a metric is not able to connect to the data it needs to refresh, the refresh will fail. When a metric refresh fails 10 times in a row, the metric owner receives an email notification.

To adjust the number of consecutive failures before a warning email is sent, use the **tsm configuration set** option `metricservices.failureCountToWarnUser`. This is a server-wide setting.

Configure when metric refreshes are suspended

If a metric refresh fails 175 times in a row, the refresh is suspended. Once a metric refresh is suspended, the server will no longer attempt to check for new data, until the refresh is manually resumed.

To adjust the number of consecutive failures before a refresh is suspended, use the **tsm configuration set** option `metricservices.maxFailedRefreshAttempts`. This is a server-wide setting.

Manage metrics

Though metrics are created from a view, they are not tied to the view like alerts or subscriptions. This means you can manage metrics similar to how you manage workbooks, by renaming, moving, tagging, deleting, or setting permissions on a metric.

Find metrics to manage either by navigating the project hierarchy or via the following paths.

- To see all metrics on a site: navigate to the Explore section, then select **All Metrics**.
- To see metrics created from the views in a workbook: open the workbook, then select the **Connected Metrics** tab.
- To see metrics related to a search term: perform a search, then select the **Metrics** tab.

Address failing and suspended metric refreshes

Metric refreshes may fail for one of the following reasons.

- The connected view was deleted or modified.
- Permissions changed for the connected view.
- The password for the data source is no longer embedded or is no longer valid.
- There was a temporary connectivity issue, which will resolve itself.

For more information on why metric refreshes fail and what users can do to fix them, see [Fix failing refreshes](#).

Encourage users to overwrite a metric if the connected view was modified in a way that caused the refresh to fail, but the view is still available. Users can overwrite a metric by creating a metric with the same name in the same project as the existing metric.

Resume suspended refreshes

If the cause of the failure is fixed, for example by embedding the correct password for the data source, you can resume the metric refresh.

1. Locate the affected metric. Metrics with suspended refreshes display the text **Refresh Suspended**, instead of the time of last refresh, in grid and list view.
2. On the warning message, click **Resume refresh**.

Tableau attempts to perform the refresh. If this attempt succeeds, you'll receive a confirmation, and the refresh will resume on schedule. If the attempt doesn't succeed, the refresh remains suspended. You or the metric owner can delete or overwrite the metric, or keep it to reference historical data.

Monitor metric activity with administrative views

Use the administrative views for Tableau Server to monitor metric refreshes and see which users are creating and viewing metrics.

1. Navigate to the site you want to monitor, or monitor server-wide activity by selecting **All Sites** from the site picker.
2. From the navigation panel, click **Site Status** or **Server Status**.
3. Select the dashboard you want to inspect.
 - To monitor metric refresh activity, open the Background Tasks for Non Extracts dashboard.

Filter for the tasks **Find Metrics to Update** or **Update All Metrics on a View**.

- To see which users are creating metrics, open the Actions by All Users or Actions by Specific User dashboard.

Filter for the action **Create Metric**.

- To see recent activity involving metrics, open the Actions by Recent Users dashboard.

Look at the list of actions under **What Actions Were Recently Performed?**

Managing Background Jobs in Tableau Server

In Tableau Server, users can schedule extract refreshes, subscriptions, or flows to run periodically. These scheduled items are referred to as **Tasks**. The Backgrounder process initiates unique instances of these tasks to run them at the scheduled time. The unique instances of the tasks that are initiated as a result are referred to as **Jobs**. Jobs are also created for runs that are initiated manually, by clicking the **Run Now** option in the web interface, programmatically through REST API, or `tabcmd` commands.

For example, an extract refresh task is created to run daily at 9 AM. This is an extract refresh task, and every day at 9 AM, a job will be created for the Backgrounder to run. In addition to user-generated jobs, the Backgrounder also does a number of System jobs on behalf of the user to support general Tableau work flows, such as thumbnail generation.

Running all these jobs can mean that Backgrounder uses a lot of resources at various times during the day. Using the Job Management feature, Server and Site administrators can get more details on these jobs that happen in their Server or Site, and take action on those jobs to better manage server resource usage. System jobs are only viewable by Server administrators, and by default are filtered out..

The **Run Now** settings on the **General** settings page also allows you to manage your resources by either allowing or blocking users from running jobs manually. By default, this option is selected to allow users to run jobs manually. Clear the check box to prevent users from running jobs manually. To learn more about managing Backgrounder resources, see [Tableau Server Backgrounder Process](#).

The Jobs page which contains the information about jobs can be accessed by navigating to the **Existing Tasks** menu of the left navigation menu.

Information about jobs can only be viewed by Server and Site administrators.

Overview

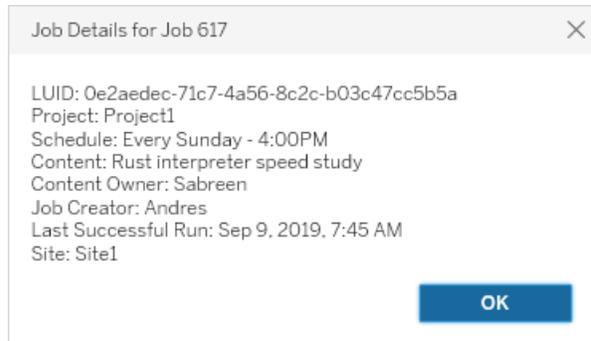
This topic describes how to view and understand the information displayed in the Jobs page.

At the top of the page there are high level statistics for the number of **Failed**, **Completed**, and **Cancelled** jobs within the past 24 hours. For Server administrators, this also includes System jobs. Applying filters do not change these values.

ID	Status	Priority	Task Type	Job Requested Time	Run Time (min)	Queue Time (min)	Average Run...	Average Queue...
975	Pending	78	Extract Refresh/Creation	Sep 9, 2019, 9:15 AM	0.0	15	5.4	21.6
678	Pending	24	Subscription	Sep 9, 2019, 9:04 AM	0.0	13.0	28.6	20.5
356	Cancelled	63	Extract Refresh/Creation	Sep 9, 2019, 9:03 AM	13.7	0.4	21.9	28.8
168	In Progress	73	Subscription	Sep 9, 2019, 9:01 AM	3.3	12.4	32.6	18.1
404	Completed	73	Extract Refresh/Creation	Sep 9, 2019, 8:58 AM	10.2	8.1	17.0	4.6
563	Pending	72	Extract Refresh/Creation	Sep 9, 2019, 8:54 AM	0.0	22.2	30.8	8.2
817	Pending	42	Extract Refresh/Creation	Sep 9, 2019, 8:54 AM	0.0	22.8	16.5	3.5
824	In Progress	91	Extract Refresh/Creation	Sep 9, 2019, 8:52 AM	4.3	20.6	6.6	17.4
357	Completed	90	Extract Refresh/Creation	Sep 9, 2019, 8:50 AM	14.1	12.6	24.8	12.3
726	Completed	92	Extract Refresh/Creation	Sep 9, 2019, 8:49 AM	1.1	26.4	33.7	36.6
239	Cancelled	40	Flow	Sep 9, 2019, 8:47 AM	22.2	7.3	38.7	22.8
49	In Progress	96	Subscription	Sep 9, 2019, 8:47 AM	17.2	12.8	27.4	2.2
652	Pending	3	Extract Refresh/Creation	Sep 9, 2019, 8:46 AM	0.0	31.0	3.1	20.3

For each job generated, there is a Job ID, the status of that job, the priority, the type of task that the job was generated from, the current run time - if the job is in-progress, current queue time - if queued, as well as the average run time, and average queue time. Tableau records historical run and queue times to compute the average run time and average queue time.

The Job ID can be useful when viewing jobs on **Admin views** and can also be used to query the **Workgroups Database**. When you click on the Job ID, you will see more detailed information about the job, such as the Job LUID, the project name, the schedule, the content name, content owner, job creator, the last time the job ran successfully, and the site name. The site name is displayed if you navigate to the Jobs page using the **Manage All Sites** menu.



Note: Doing a **Refresh Now** from the **Data Sources** page will only show the LUID information in the **Job Details** dialog box.

Important! Jobs that existed 24 hours or newer before an upgrade to Tableau Server 2019.4 will not have data for **Average Queue Time**, **Average Run Time**, **Last Successful Run Time**, and **Job Creator** on the **Jobs** page.

Task Types

There are several types of tasks:

- **Extracts:** This includes extract creation, incremental extract refreshes, and full extract refreshes. For more information on extract refreshes, see [Quick Start: Refresh Extracts on a Schedule](#).
- **Subscriptions:** Includes subscriptions for workbooks and views. For more information, see [Set Up a Site for Subscriptions](#).
- **Flow:** This includes scheduled flows and manual flow runs.
- **Encryption:** Includes the following:

Tableau Server on Linux Administrator Guide

- Extract encryption and decryption
 - Flow encryption and decryption
 - Re-key extracts and flows
- System: This is all system Jobs that the Backgrounder handles behind the scenes to support Tableau Server.

Filters

You can filter to see only certain jobs. The available filters are by Job Status type, Task Type, and Time Range. For the Time Range filter, you can choose from past one to 24 hours, in four hour increments. The option to filter in System Jobs is available if you are a Server Administrator.

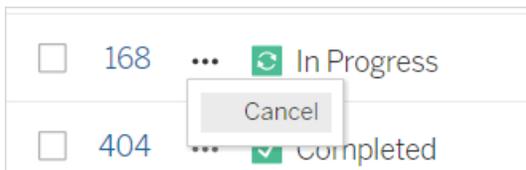
Canceling Jobs

Extract refreshes, subscriptions and flow run jobs can be canceled. You can only cancel one job at a time, and selecting multiple jobs at one time for cancellation is not supported.

When you cancel a job, an email with the time the job was canceled, the affected content, and the time the job ran before being canceled is sent to the recipients that you select in the **Cancel Job** dialog box . In addition you can add your customized notes to be included in the email.

If you do not select any recipients, the job will be canceled, but no email will be sent.

To cancel a job, click on the ellipses next to the Job ID and use the dialog to cancel the job:



Cancel Job

Tableau Server will send an e-mail with the information about the cancelled job. You can provide additional information below, such as a reason for cancelling the job:

0/500

Send an email to:

- Job Creator
- Project Owner
- Site Administrator(s)
- Server Administrator(s)

Cancel OK

Status

There are seven types of status that jobs can be in, and hovering over each status will display more relevant information.

- **Completed:** This job shows as **Completed successfully** and you can see the time when the job completed in the tooltip that is displayed when you hover over the status.

Completed

This job successfully completed on Jun 13, 2019, 10:23 AM. [Learn more](#)

- **In Progress:** This job shows as **In Progress**. A time for how long the job has been

running for is displayed in the tooltip when you hover over the status.



- **In Progress:** This job is **In Progress**, but is **running late**. Tableau keeps track of the average run times for the same job, and if the current run time is longer than the average run time, then it is considered running late. Times for how much longer than average the job has been running and its average run time is provided in the tooltip that is displayed when you hover over the status.



- **Pending:** This job is currently **Pending**, waiting to be run when there is available Backgrounder capacity. A time for how long the job has been in the queue for is provided in the tooltip that is displayed when you hover over the status.



- **Pending:** This job is currently **Pending**, but is **running late**. Tableau keeps track of the average queue times for the same job, and if the current queue time is longer than

the average queue time then it is considered running late. Times for how much longer than average the job has been queued is provided in the tooltip that is displayed when you hover over the status.



This job has been on the queue 4.8 min longer than previous runs (0.1 min). [Learn more](#)

- **Cancelled:** This job was **Cancelled** by a Server or Site administrator. The time the job was cancelled and how long it ran for before cancellation is provided in the tooltip that is displayed when you hover over the status.



This job was cancelled on Jun 11, 2019, 9:55 PM after running for 29.5 min. [Learn more](#)

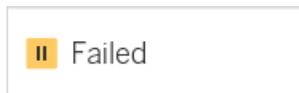
- **Failed:** This job is showing as **Failed**. The time when the job failed, how long it ran for before it failed, and why the job has failed is provided in the tooltip that is displayed when you hover over the status.



This job failed on Jun 11, 2019, 4:23 AM after running for 51.6 min because of: Invalid credentials [Learn more](#)

- **Suspended:** This job is showing as **Failed** with a pause icon. If the job fails 5 times

consecutively, then the job is suspended. Suspended tasks are still available but Backgrounder will not create jobs for these tasks until they are resumed by the user.



This job did not run since it was initiated from a task that is suspended. To run jobs initiated from this task, you must first resume the task. [Learn more](#)

Tableau Service Manager Jobs

TSM jobs are administrative tasks that help configure and maintain Tableau Server. These jobs run by Tableau Services Manager.

Here are some key TSM jobs:

- **Cleanup:** This job is created when a cleanup command is issued to Tableau Server. The cleanup command deletes old log files and temporary files. For more information on the TSM CLI cleanup command, see [tsm maintenance](#).
- **Deployments:** This job is created to apply any configuration updates that you make to Tableau Server. This can be initiated either through the TSM web interface or TSM CLI. For more information on the TSM CLI command, see [tsm pending-changes](#). Here are a few examples of configuration updates: enabling SSL, enabling Run as User, and server topology changes.

Depending on how many topology changes are being made, and the complexity, this job can take longer than the previous time it ran successfully. For example, if the previous change was a hot topology change, and the current one is not, the current job can take longer to complete than the previous one.

- **Generate backup:** This job is created when a backup command is issued to Tableau

Server. The backup command creates a backup file of Tableau data (data in the File Store and repository). For more information on the TSM CLI backup command, see `tsm maintenance backup`.

The time taken for the backup job to complete depends on the amount of data that needs to be backed up. If the amount of data to be backed up has increased from the last time this job was run, it will take longer for the job to complete compared to the previous time.

- **Initialize Tableau Server:** This job is initiated to initialize Tableau Server during the installation process. For more information on the TSM CLI command, see `tsm initialize`
- **Restore:** This job is created when a restore command is issued to Tableau Server. The restore command restores a Tableau Server data backup file. For more information on the TSM CLI restore command, see `tsm maintenance`. The time taken for the restore job to complete depends on the size of the backup file. If the backup file is larger than the previous times, the restore job will take more time to complete.
- **Start Server:** The job is created to start all the stopped Tableau Server processes. This can be initiated either through the TSM web interface or TSM CLI. For more information on the TSM CLI command see, `tsm start`.
- **Stop Server:** This job is created to stop all the running Tableau Server processes. This can be initiated either through the TSM web interface or TSM CLI. For more information on the TSM CLI command, see `tsm stop`.

Canceling tsm Jobs

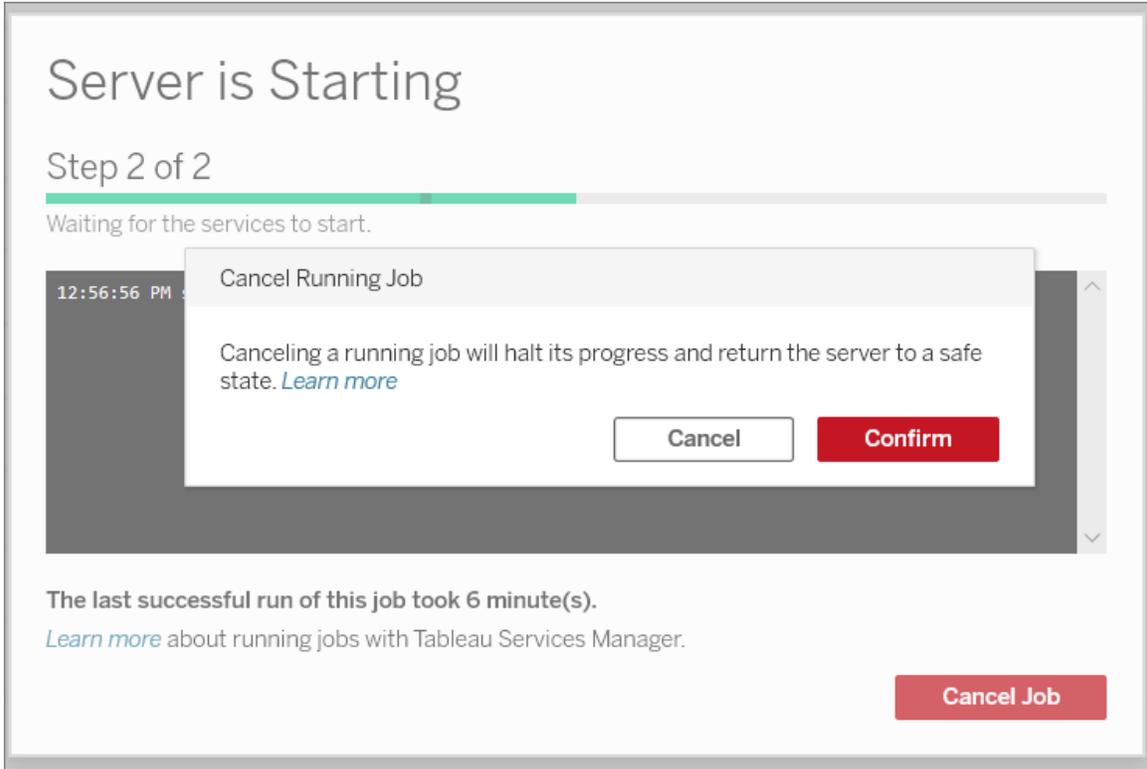
You can cancel jobs either through the TSM web interface or using the TSM CLI. For more information, see [Cancel TSM Jobs](#).

Cancel TSM Jobs

TSM Jobs can be canceled through TSM web interface or TSM CLI. There are many TSM jobs, but only certain jobs can be canceled once they are in progress. Any job that hasn't yet

started can be canceled using TSM CLI.

TSM web interface: For jobs that can be canceled while they are running, the Cancel option is available in the Job dialog box as shown below:



TSM CLI: To cancel jobs using TSM CLI, see `tsm jobs`.

To see more information about TSM jobs in general, see [Tableau Service Manager Jobs](#).

Canceling Jobs that are in progress

Only certain jobs can be cancelled while they are already running, and the cancel behavior can be different depending on the job and the state of the job at the time it was canceled. This is explained in detail below:

- **Cleanup:** If you cancel a cleanup job, it will stop any services that were started in order to do the cleanup. Depending on when it was canceled, some files may be deleted and some may not have been deleted yet resulting in partial cleanup.
- **Decommission File Store:** If you cancel this job, it returns the Tableau Server File Store topology to the state that it was prior to starting the decommissioning process.
- **Generate Backup:** If you cancel this job, any services used for backup are stopped and Tableau Server will try to delete any files that it created as part of the backup process.
- **Restart Server:**
 - Job is cancelled when Tableau Server processes are stopping: The job is canceled, but the services will try to get to a stopped state.
 - Job is cancelled while Tableau Server processes are restarting: The job is canceled, but the services will try to restart.
- **Start Server:** The job will be canceled, but the processes will still try to start.
- **Stop Server:** The job will be canceled, but the services will try to stop.

Here are some of the main reasons why you may want to cancel a job:

1. Since tsm jobs can only be run one at a time, you might need to cancel a current job if you need to run another job instead.
2. If the running job includes changes to the Tableau Server that you did not intend to make.

Administrative Views

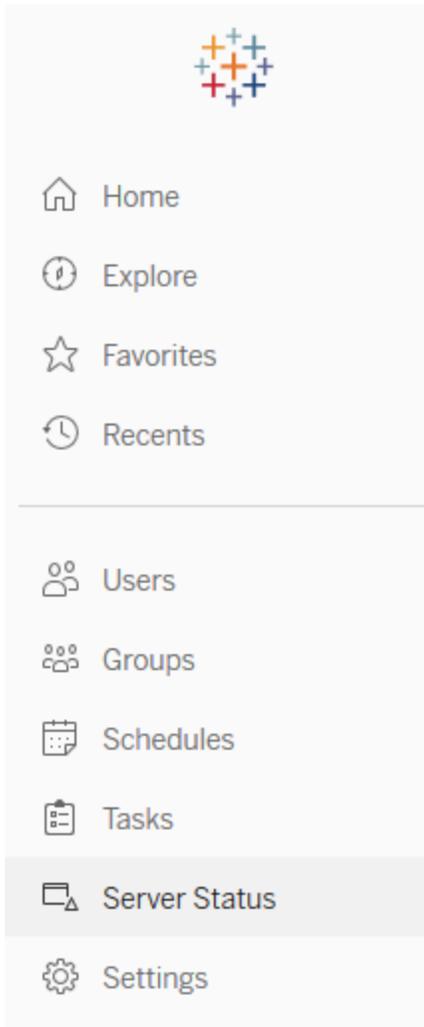
The Status page contains an embedded Tableau workbook with various administrative views. These views help you to monitor different types of server or site activity.

Note: You must install PostgreSQL drivers before you can see Administrative views. For more information, see Database Drivers.

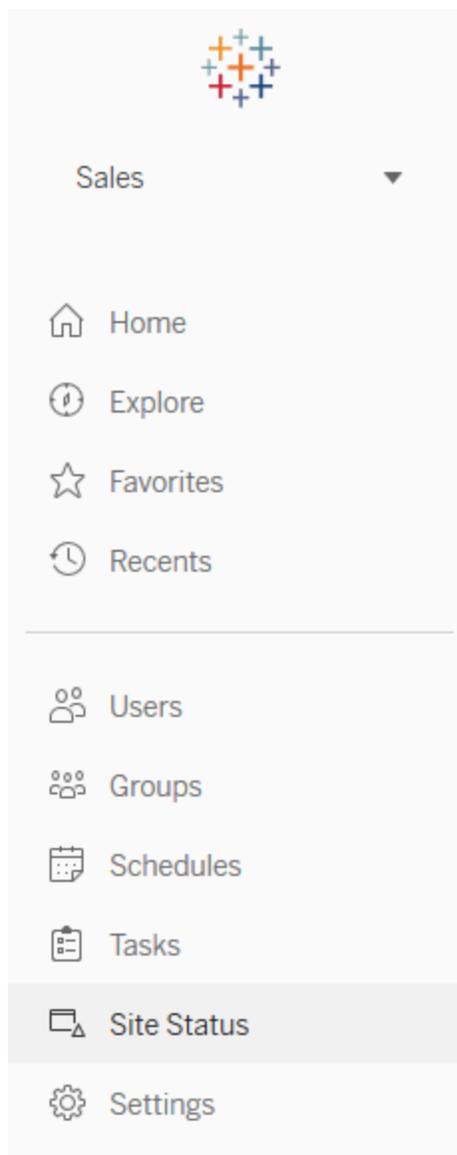
Navigating to administrative views

To see administrative views, click **Status**. Site administrators can see administrative views for their site. Administrators of multiple sites can see views for the current site.

On a multi-site server, server administrators can see views for the entire server. Click the site menu, and then click **Manage All Sites** to access the server menus.



To see views for individual sites on a multi-site server, click the site menu, select the site name, and then click **Site Status**.



Pre-built Administrative Views

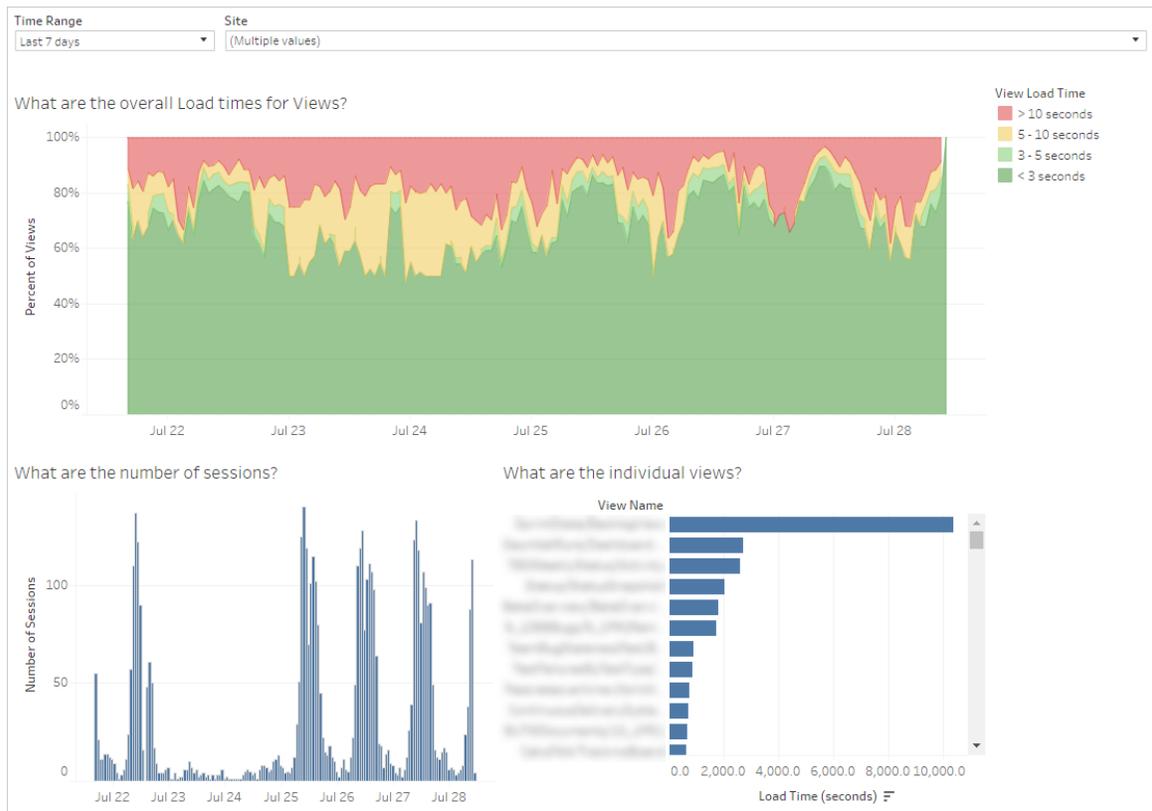
Administrative views are powerful monitoring tools that can help you optimize Tableau Server and better understand how your users are interacting with Tableau content. The administrative views listed to the right are included with Tableau Server. Click on the link for a view to learn more about how to interpret and act on the information the view provides.

To create your own administrative view, see [Create Custom Administrative Views](#).

Performance of Views

Note: This view is only available to server administrators. For information about how to navigate to administrative views, see Administrative Views .

The Performance of Views administrative view displays how long it takes for views to load and how many sessions are running at a time on the server.



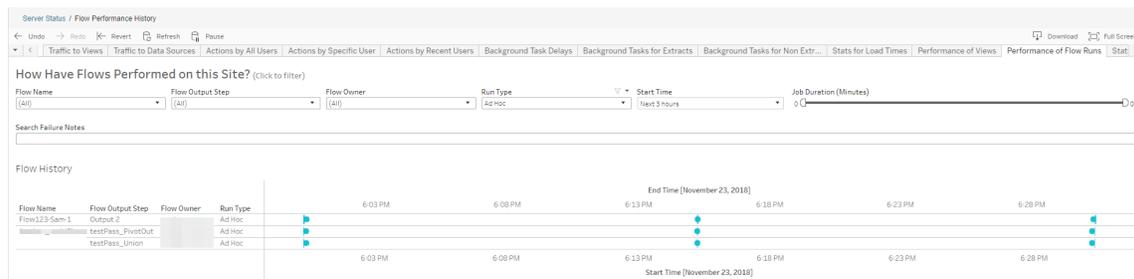
You can compare spikes in the number of sessions with spikes in slow load times to identify the times of day when high user traffic is slowing down the server. You can also look at the individual views by load time to understand which views take the longest to load.

Some views might take a long time to load regardless of when they are viewed. You can identify which workbooks need to be optimized with the **Stats for Load Times** administrative view. Some simple ways to optimize workbooks includes the following:

- Display less information in each view.
- Break up views.
- Reduce the number of filters.
- Use data extracts.

Performance of Flow Runs

Use this view to see the performance history for all the flows on a site. You can filter by Flow Name, the Output Step Name, Flow Owner, Run Type (Scheduled or Ad Hoc), and the time the flow runs were started. For information about other administrative views available for flows, see [Monitor Flow Health and Performance](#) in the Tableau Prep help.



Here are some questions you can answer using this view:

- **What flow tasks are currently scheduled?** – To do this, use the Start Time filter and select the time frame you want to look at. For example, to see flow tasks that are scheduled in the next 3 hours, select **Hours -> Next ->** and enter **3**.
- **What is the duration of flow tasks?** - To answer this, click on a mark in the view and you should see details including the task duration.

How many flows were run ad hoc, and how many were scheduled runs? - To answer this, use the **Run Type** filter and select **Ad hoc** or **Scheduled**.

Note: This is not functional in this release and will not actually filter the data.

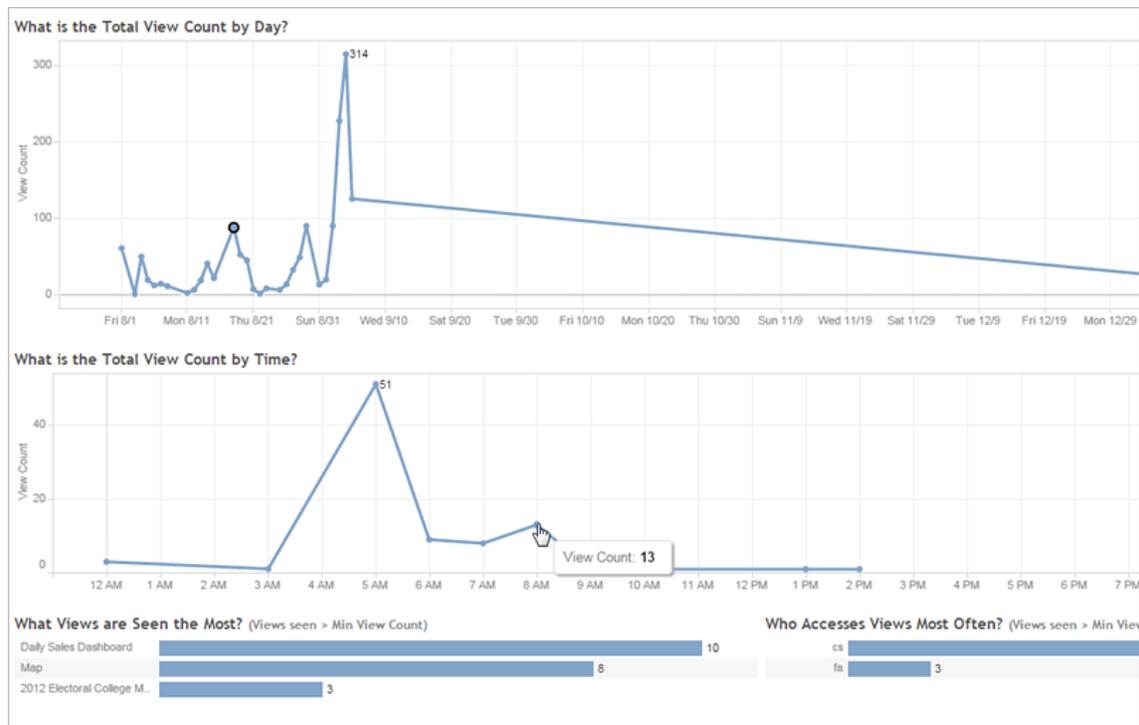
In addition to the questions described above, here are some examples of insights you might be able to gather:

- The flows that are running most frequently will have the most marks.
- To see how many flows are running at the same time currently, hover over a mark that shows “**In Progress**” or “**Pending and select “Keep Only**” to filter all flow runs that are currently running.
- To see how many flows are running at the same time during a specific time range, select a range for the **Start Time** filter. For example, you can choose “**Next three hours**” to see which flows will be running in the next three hours.

Traffic to Views

The Traffic to Views view gives you the ability to see how much of your user traffic goes to views.

You can filter what information is displayed and the time frame it comes from by selecting the view, the workbook, and the time range. Server administrators can specify the site.



Two time lines at the top of the view show you how views are being used over a time range you specify (the default is the last 7 days):

- **What is the Total View Count by Day**—This shows total view count by day, based on the filters you set. Hover your mouse pointer over a point on the line to see the count of views. Select the point to update the other sections of the view based on your selection.
- **What is the Total View Count by Time**—This shows the view count by time of day. The filters and any selection impact this graph.

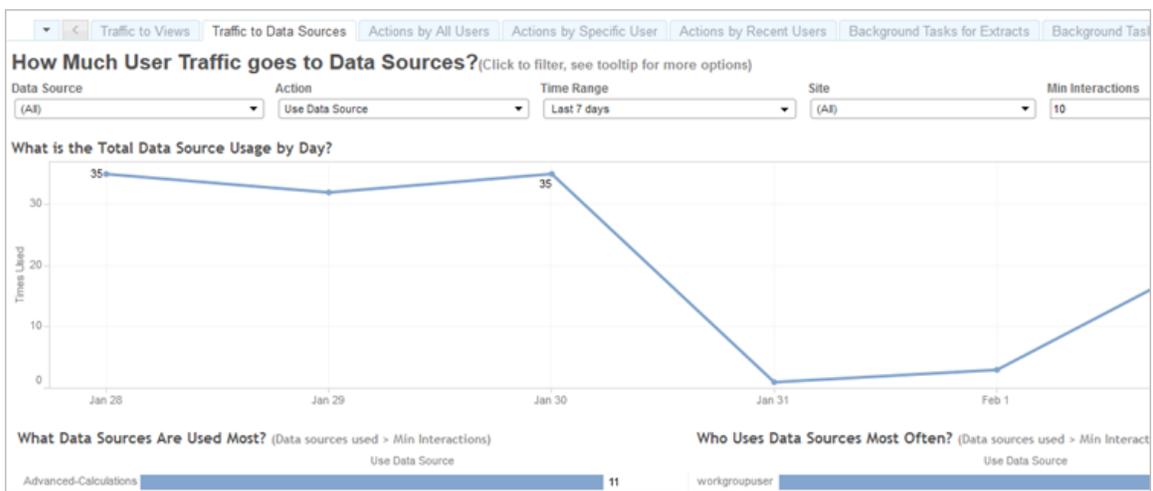
Two bar graphs at the bottom of the view show results that are filtered by the **Min View Count** filter at the top of the view. These show you the views that are most often accessed, and the users who most frequently access views. Only those views and users with counts greater than or equal to the minimum view count value are displayed:

- **What Views are Seen the Most**—This is a list of the most visited views. Like the other sections of the view, the information is limited by filters and any selection you make.

- **Who Accesses Views Most Often**—This shows the users who most often access the views and is limited by filters and any selection you make.

Traffic to Data Sources

The Traffic to Data Sources view gives you the ability to see usage of data sources on your Tableau Server installation. This can help you determine which data sources are most heavily used and those that are less often used. You can filter the information you see by selecting the data source, the action taken on that data source, and the time range. Server administrators can specify the site.



A time line at the top of the view shows you how data sources are being used over a time range you specify (the default is the last 7 days):

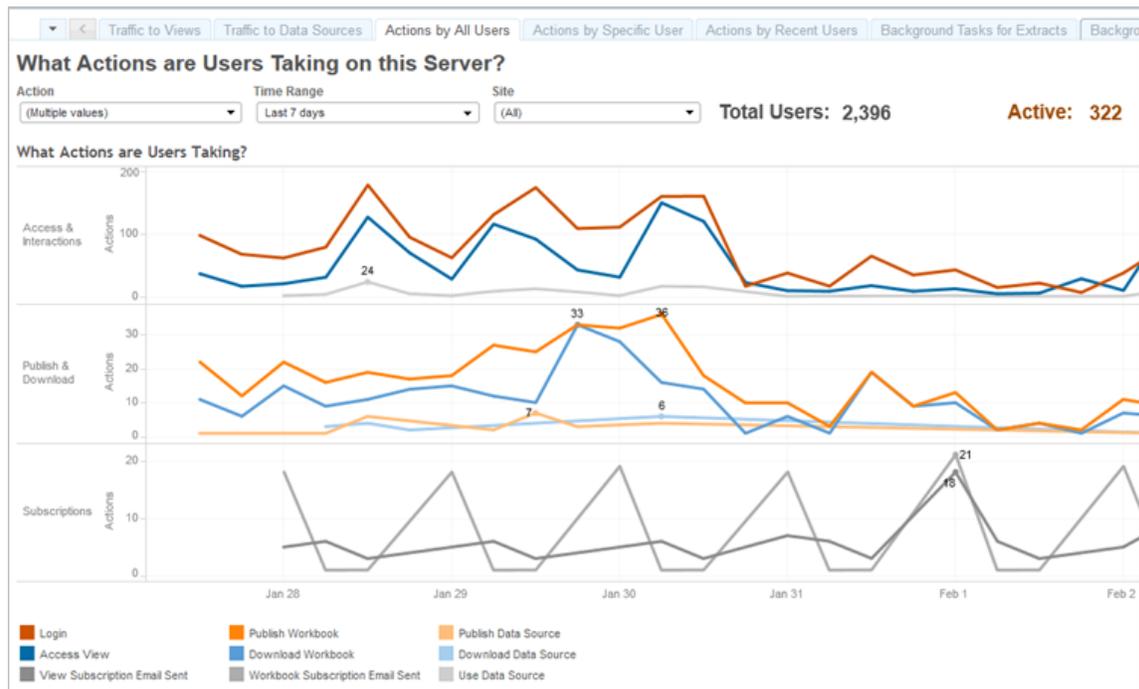
- **What is the Total Data Source Usage by Day**—This shows total data source usage by day, based on the filters you set. Hover your mouse pointer over a point on the line to see the count. Select the point to update the other sections of the view based on your selection.

Two bar graphs at the bottom of the view show results that are filtered by the **Min Interactions** filter at the top of the view. These show you which data sources are most used, and who uses data sources most often. Only those data sources and users with interaction counts greater than or equal to the minimum interactions value are displayed:

- **What Data Sources are Used Most**—This is a list of the most used data sources. Like the other sections of the view, the information is limited by filters and any selection you make.
- **Who Uses Data Sources Most Often**—This shows the users who most often use the data sources. This is impacted by filters and any selection you make.

Actions by All Users

The Actions by All Users view gives you insight into how your Tableau Server installation is being used. You can filter the view by actions and by time range. Server administrators can filter by site. The Total Users count shows the number of users who have performed an action. This value is not affected by any filtering. The Active user count shows the number of active users who have performed one of the selected actions.



Up to three separate groups of time lines show you how users are using Tableau Server over a time range you specify (the default is the last 7 days). If no actions are selected for a particular group, that group does not display. Possible groups are:

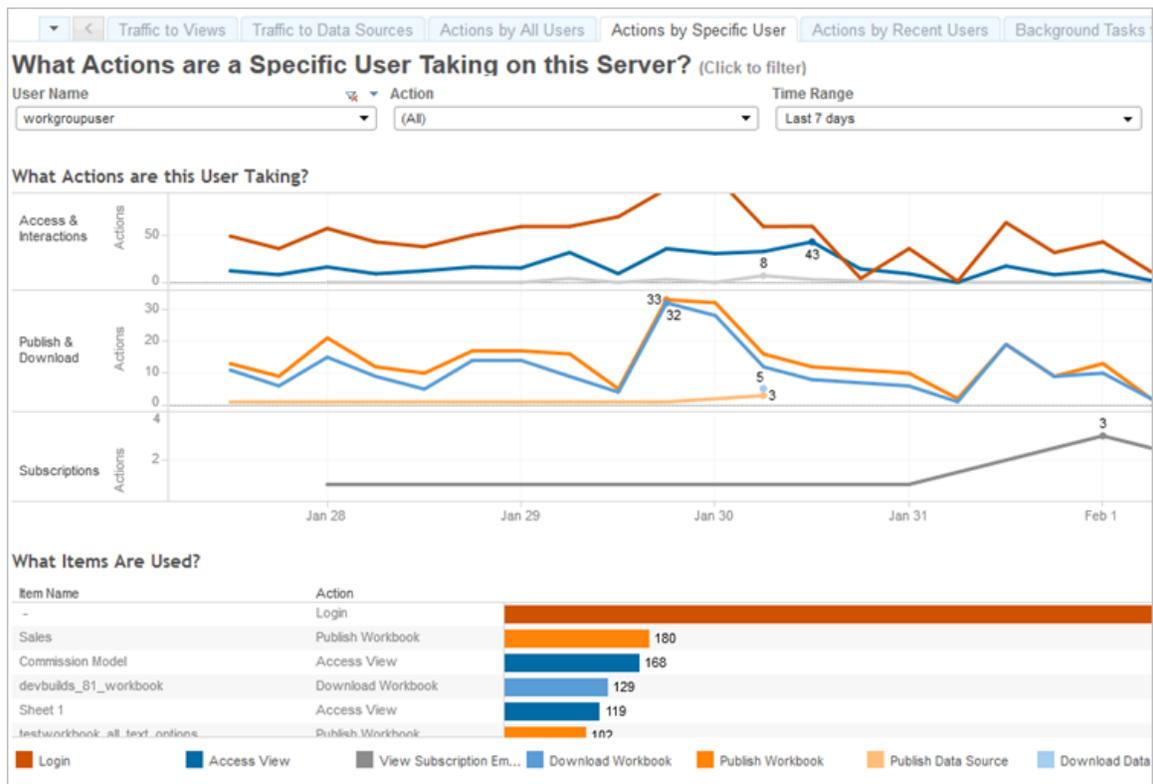
Tableau Server on Linux Administrator Guide

- **Access & Interactions**—This shows you sign in (log on) activity, view access and data source use.
- **Publish & Download**—This shows publishing and downloading of flows, workbooks and data sources.
- **Subscriptions**—This shows counts of subscription email sent for workbooks and views. It also shows the counts of flow runs.

Use the legend at the bottom to view a subset of the displayed actions. Click a single action to highlight the line for the action, or **Ctrl + Click** on multiple actions to highlight more than one. To clear the selection and display all the selected actions, click on any action in the legend.

Actions by Specific User

The Actions by Specific User view gives you insight into how individual users are working in your Tableau Server installation. You can filter the view by user name, actions, and time range. Server administrators on multi-site installations can filter by site.



Up to three separate groups of time lines show you how a selected user is using Tableau Server over a time range you specify (the default is the last 7 days). If no actions are selected for a particular group, or if no actions were taken, that group does not display. Possible groups are:

- **Access & Interactions**—This shows you sign in (log on) activity, view access and data source use. This means any interaction with a Data Source by the selected user, including extract refreshes scheduled by that user, or the user accessing a workbook that is associated with that data source.
- **Publish & Download**—This shows publishing and downloading of flows, workbooks and data sources.
- **Subscriptions**—This shows counts of subscription email sent for workbooks and views. It also shows the counts of flow runs.

A bar graph at the bottom of the view shows which items the selected user is using.

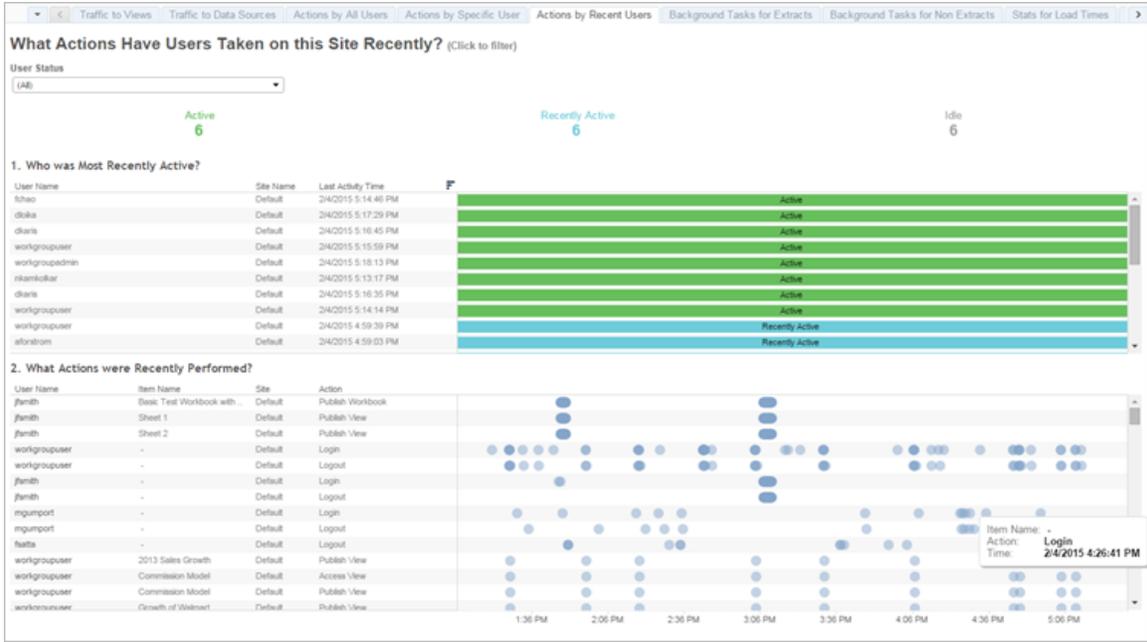
Use the legend at the bottom to view a subset of the displayed actions. Click a single action to highlight the line for the action, or **Ctrl + Click** on multiple actions to highlight more than one. To clear the selection and display all the selected actions, click on any action in the legend.

Actions by Recent Users

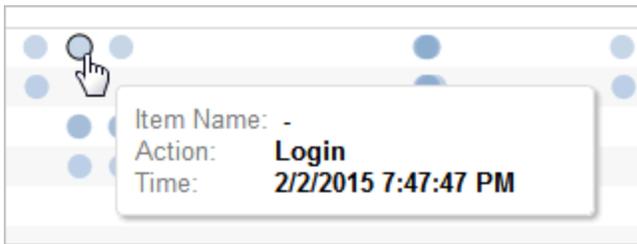
The Actions by Recent Users view shows you which signed-in users have been active on Tableau Server over the past 24 hours. This can be useful if you need to perform some maintenance activity and want to know how many and which users this will affect, and what they are doing on Tableau Server.

The view **Active**, **Recently Active**, and **Idle** users that are currently signed in to Tableau Server. For this view, an active user is one who took an action in the last 5 minutes, a recently active user is one who last took an action within 30 minutes, and an idle user is one who last took an action more than 30 minutes ago. The actions are displayed in the lower section of the view.

Tableau Server on Linux Administrator Guide

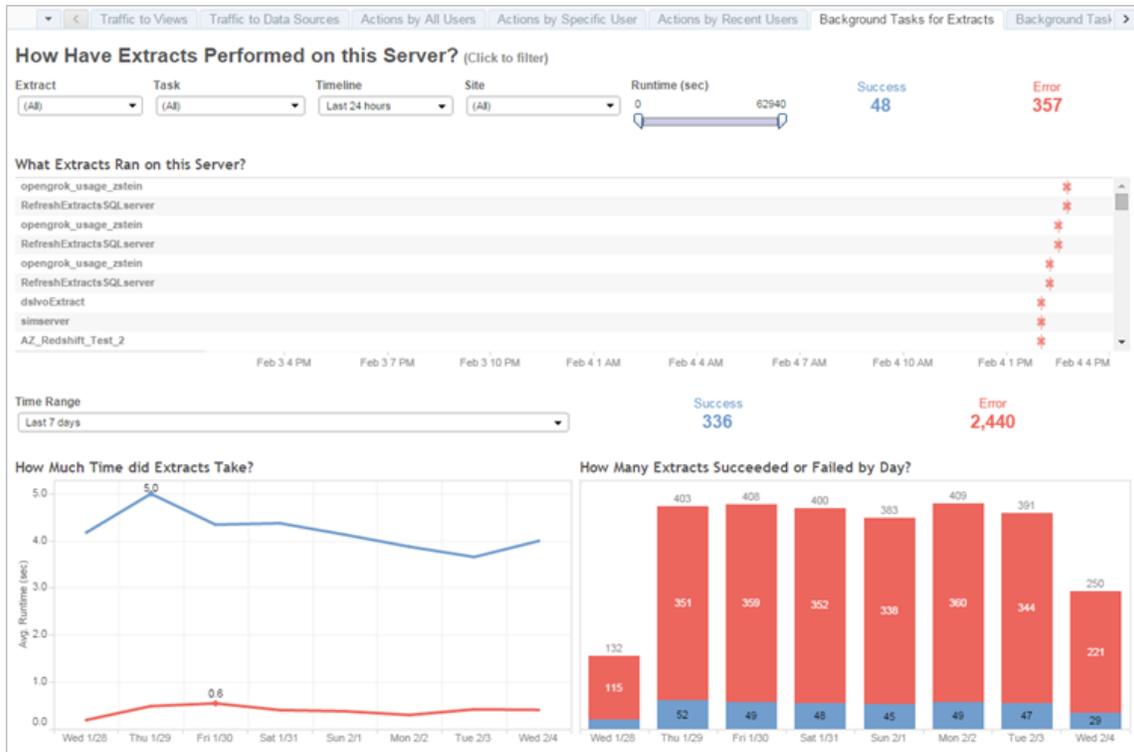


Select a user to see only the actions that user performed recently. Hover over an action to see details of the action.



Background Tasks for Extracts

The Background Tasks for Extracts view displays extract-specific tasks that run on the server.



Understand this view

To better understand this pre-built administrative view, make note of the following:

- The table, "What Extracts Ran on this Server," lists the extracts that ran in the time period specified in **Timeline**.
- You can click **Success** or **Error** to filter the table based on status.
- You can also click a specific task to update the "How Much Time did Extracts Take" graph for the selected task.
- The table, "How Many Extracts Succeeded or Failed," updates for the status (success or failure) of the task, but the count of extracts that succeeded or failed does not change.

Status

Tasks can have a status of success or error.

Icon

Description

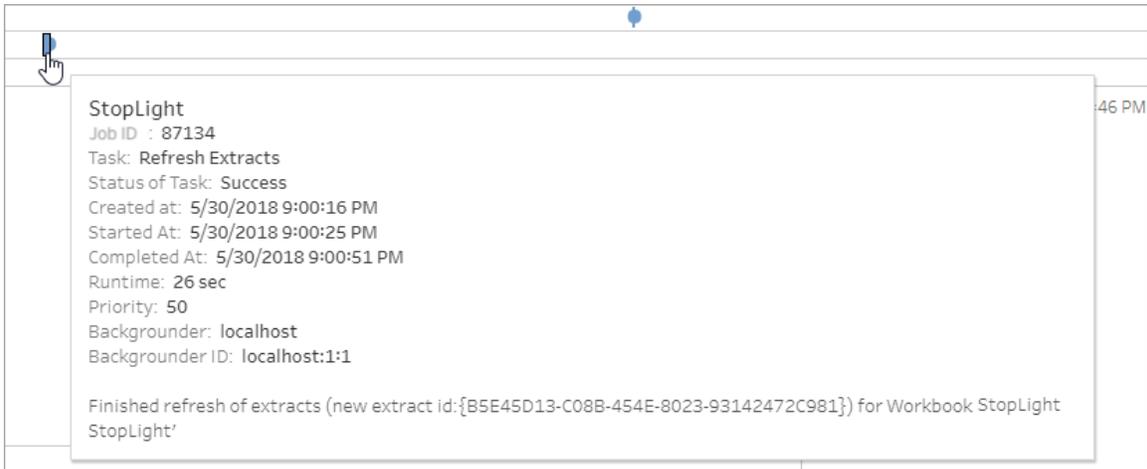


Error—Server was unable to complete the task.

Icon	Description
	Success —Server completed the task.

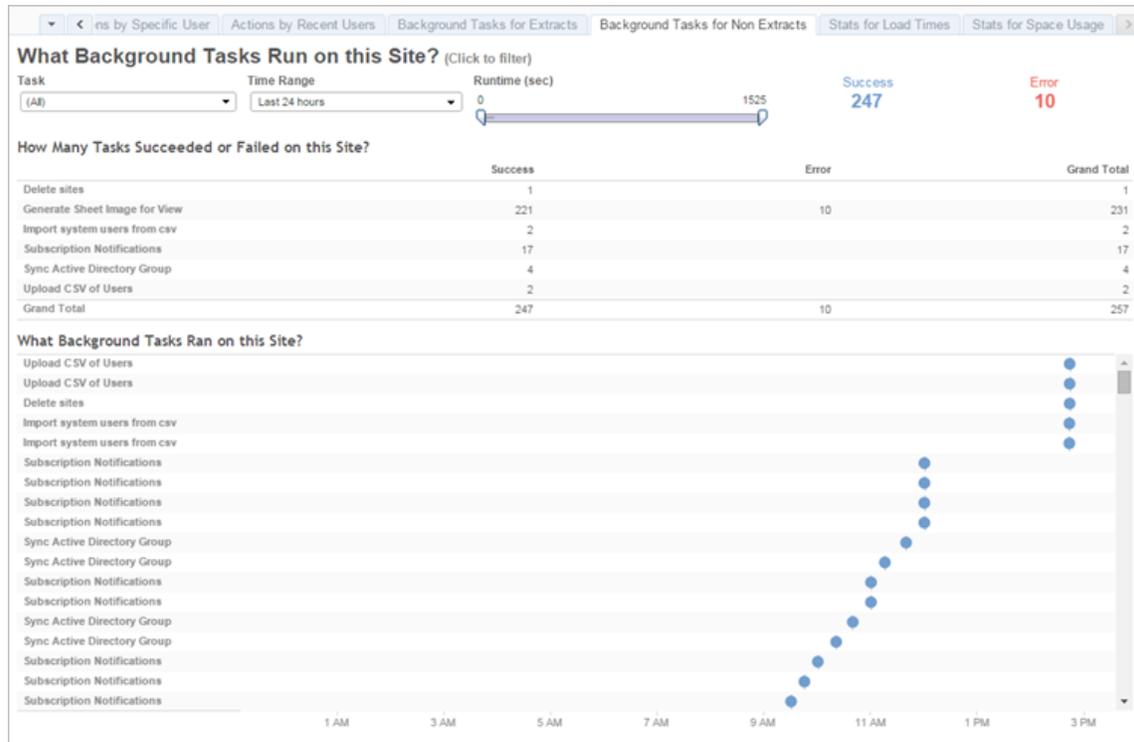
See details about a task

For details on about the task, use your mouse to hover over the success or error icon.



Background Tasks for Non Extracts

The Background Tasks for Non Extracts view displays tasks that the server runs that are not related to extract refreshes. For example, edited OAuth connections, subscription notifications, and so on.



A table lists the tasks that ran in the time range specified. Click **Success** or **Error** to filter the table based on status. Select a specific task in the **How Many Tasks Succeeded or Failed on this Site** table to update the **What Background Tasks Ran on this Site** graph for the selected task.

Tasks can have a status of success or error. For details about the task, use your mouse to hover over the success or error icon.

Icon	Description
	Error —Server was unable to complete the task.
	Success —Server completed the task.

Details that you can see about the task are its ID, status, priority, when it was created, started and completed. You can also see its runtime: the total run time of the background job, which includes the run time of the job plus background job overhead such as initialization and cleanup. You can also see which background runner the job is running on.

thumbnail images used in your workbooks. We recommend that you wait for the Upgrade Thumbnails job to complete before you back up Tableau Server.

After the first run of the Upgrade Thumbnails job, it runs on a predefined weekly schedule. The Upgrade Thumbnails job runs at lowest priority and creates one task per workbook in the `background_jobs` table to upgrade any low resolution thumbnails. Low resolution thumbnails published to Tableau Server by Tableau Desktop version 2018.3 and earlier are automatically cleaned up each week when the Upgrade Thumbnails job runs.

Troubleshooting

You can check the status of the Upgrade Thumbnails job using the [Background Tasks for Non Extracts administrative view](#). The Upgrade Thumbnails job can display either the success or error status.

Upgrade Thumbnails job failed, or it completed but some thumbnails are still low-resolution.

The Upgrade Thumbnails job might show error status if your credentials are wrong. In that case, the workbook thumbnails will still appear in fuzzy, low-resolution (192 x192 pixels). Update your credentials, and the Upgrade Thumbnails job will update the workbook thumbnails the next time it runs.

Background Task Delay

Note: This view is only available to server administrators. For information about how to navigate to administrative views, see [Administrative Views](#) .

The Background Task Delay view displays the delay for flow tasks, extract refresh tasks and for subscription tasks—that is, the amount of time between when they are scheduled to run and when they actually run. You can use the view to help you identify places you can improve server performance by distributing your task schedules and by optimizing tasks.

Tableau Server on Linux Administrator Guide



Here are possible reasons for the delays, and ways that you might reduce the delays:

- Many tasks are scheduled for the same time. In the example view, tasks that show long delays are clustered at the same time every day, which creates spikes in the wait time. Note that you can set the **Timeline** filter to a single day to view task delays by hour and identify the hours of the day which have many tasks scheduled at the same time. A solution to this issue can be to distribute the tasks to off-peak hours to reduce load on the server.
- Specific tasks take a long time to run and are preventing other tasks from running. For example, there might be an extract refresh job that is connecting to a slow data source or that is processing a large amount of data. Use the **Background Tasks for Extracts** administrative view to identify which extract refresh tasks are running slowly. You can then optimize the extract refresh task by filtering the data, aggregating the data, or creating multiple data sources for individual tables in a data source.

- Other server processes are running at the same time and are consuming server resources and slowing down performance. Monitor the CPU and memory usage of server processes to see which processes are consuming the most resources and then adjust the configuration of processes on your server.

Stats for Load Times

The Stats for Load Times view shows you which views are the most expensive in terms of server performance. You can filter by view and time range. Server administrators can filter by site. You can also limit the view based on load time in seconds, using the sliding Load Time filter. Load times are for the server. Depending on your client browser and networking, actual load time may vary slightly.

The **Average Load Times** graph shows average load times for views based on the filters you set. Hover over a point to see details. Select a point on the line to update the rest of the view for the selection:

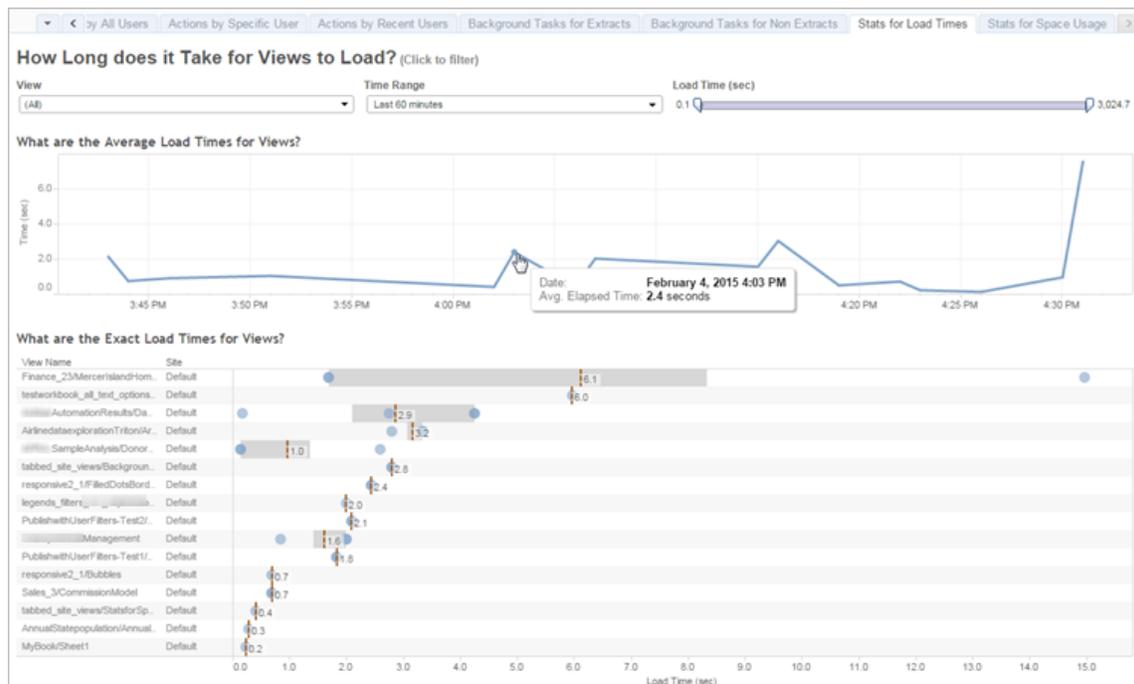
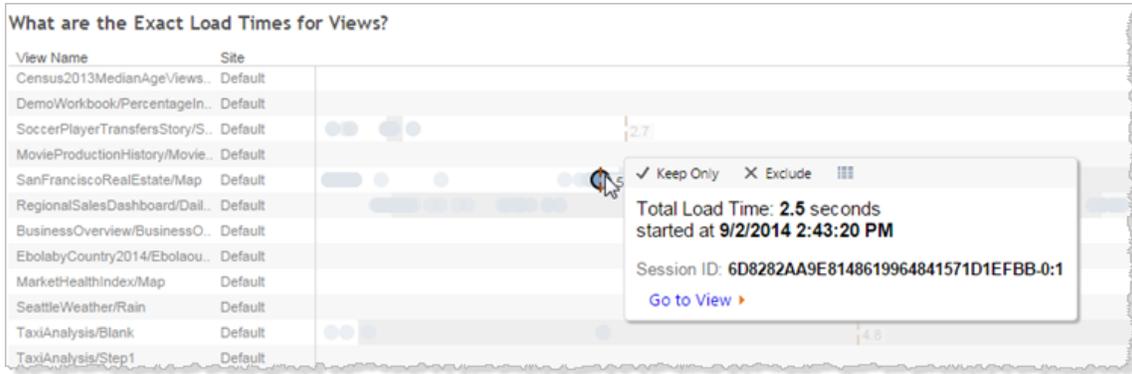


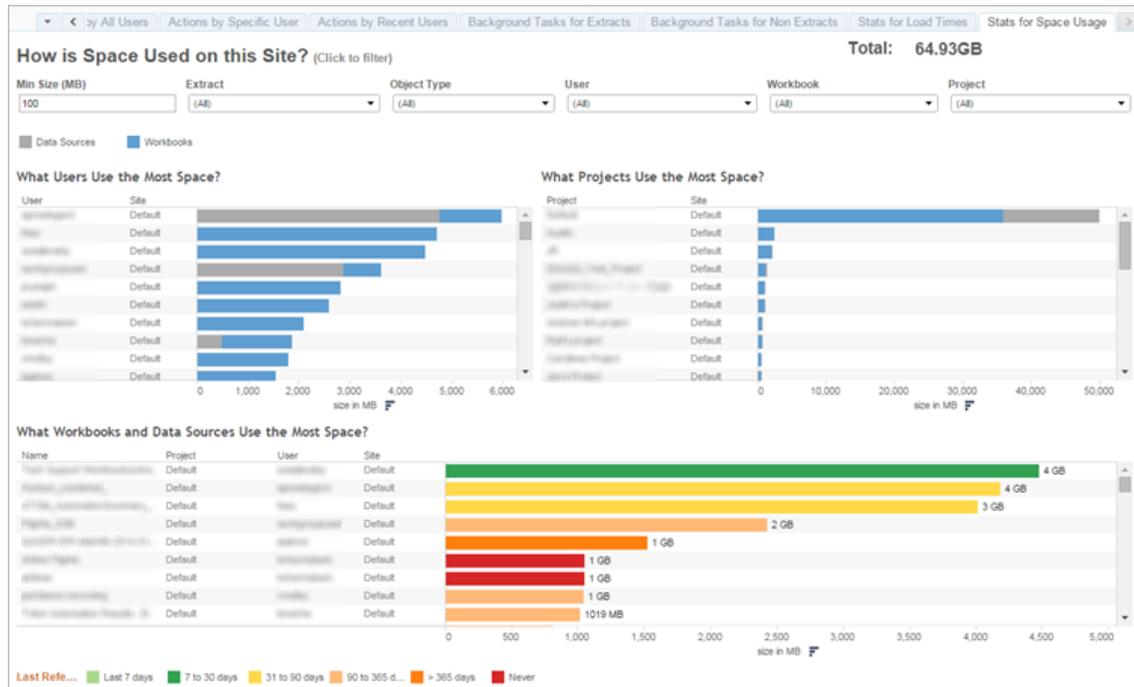
Tableau Server on Linux Administrator Guide

The **Exact Load Times** view shows exact time to load the listed views. A vertical line shows the average load time for each view. Select a mark to see details of a specific instance of the view loading:



Stats for Space Usage

The Stats for Space Usage view can help you identify which workbooks and data sources are taking up the most disk space on the server. Disk space usage is displayed by user, project, and by the size of the flow output, workbook or data source and is rounded down to the nearest number:



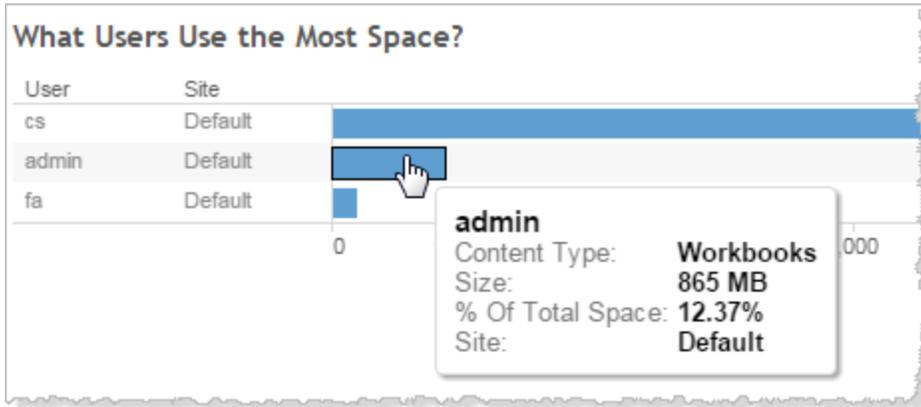
Use the **Min Size** filter to control which data sources and workbooks are displayed, based on the amount of space they take up.

Three bar graphs give you information about space usage on your Tableau Server:

- **What Users Use the Most Space**—This shows the users who own data sources and workbooks that are taking up the most space. Click a user name to filter the next two graphs for that user. Click the data source bar or the workbook bar for a user to filter the next two graphs for that type of object for that user. Click the selected user or bar to clear the selection.
- **What Projects Use the Most Space**—This shows the projects with the data sources and workbooks that are using the most space. If a user or object type is selected in the What Users Use the Most Space graph, this displays information specific to the selection.
- **What Workbooks, Flows, and Data Sources Use the Most Space**—This shows

the workbooks and data sources that are taking the most space. The bars are color-coded based on the length of time since the last refresh.

Move your cursor over any bar to display usage details:



Click on a bar to select it and update the other areas of the view based on that selection.

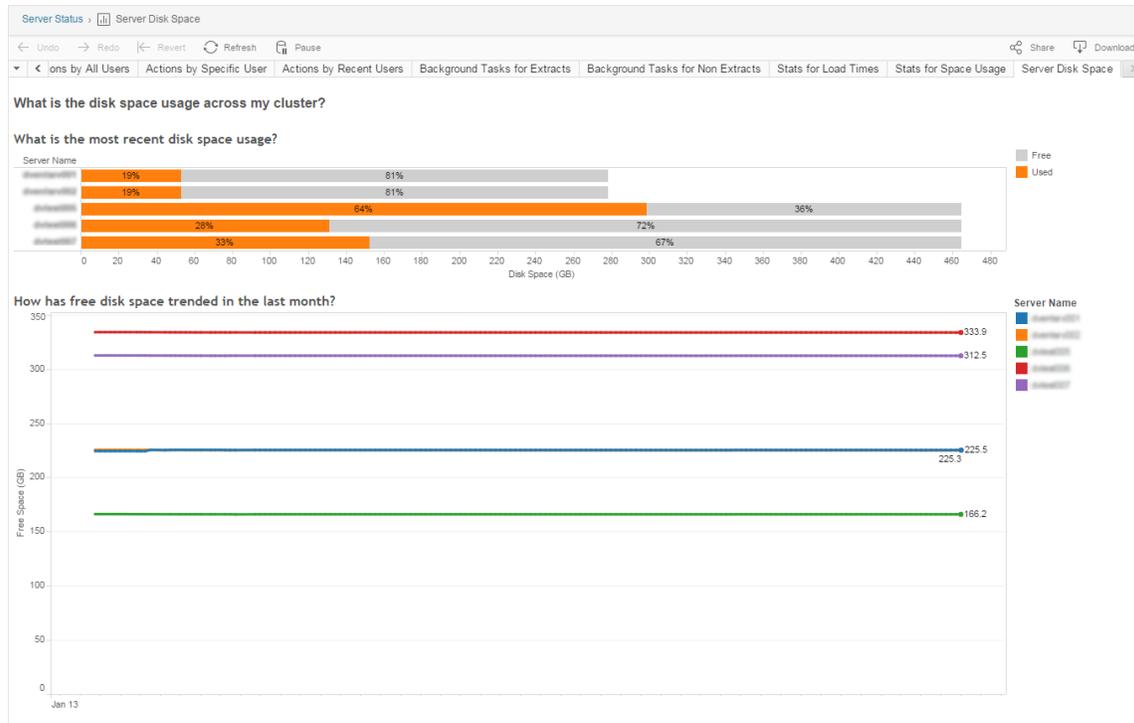
Server Disk Space

Note: This view is only available to server administrators. For information about how to navigate to administrative views, see Administrative Views .

Use the Server Disk Space view to see how much disk space is in use on the computer or computers that run Tableau Server, where disk space refers only to the partition where Tableau Server is installed. You can also use this view to identify sudden changes in disk space usage.

This view reports disk space usage as a decimal GB value. If the operating system of your Tableau Server computers report the value using binary GB, the amounts can differ.

For a distributed installation, the view displays information about each computer in the cluster.



The Server Disk Space view includes two graphs:

- **What is the most recent disk space usage?**—This graph shows disk space usage for the last 30 days both in gigabytes and as a percentage. Disk space refers only to the partition where Tableau Server is installed.
- **How has free disk space trended in the last month?**—This graph shows changes to disk space usage over the last month. Rest your pointer on a line to view the exact amount of free disk space for a point in time.

When Tableau Server is low on disk space, you can remove files to free space.

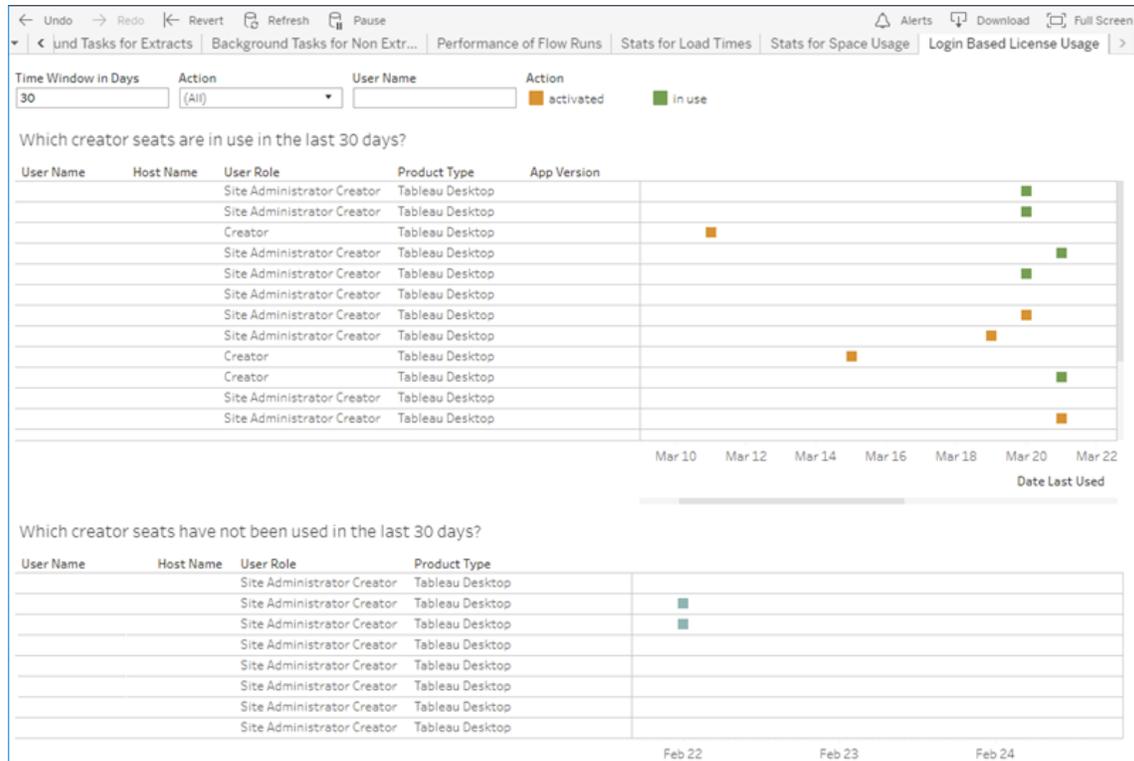
Tip: You can have Tableau Server notification you when free disk space falls below a threshold that you specify. For more information, see [Configure Server Event Notification](#).

Login-based License Usage

Note: This view is only available to site administrators and server administrators when login-based license management is enabled on Tableau Server. For information about how to navigate to administrative views, see [Administrative Views](#) .

The Login-based License Usage view lets server administrators view login-based license activation usage for Tableau Online or Tableau Server. The Login-based License Usage view can help you manage licenses efficiently and determine if you need more or fewer licenses. This view can help you answer the following questions:

- Who is using a Tableau Desktop or Tableau Prep Builder license in my enterprise?
- Has a Creator role been shared or transferred?
- Has any activation activity occurred on a computer where it should not be?
- On which host is the activation being used?
- Which role is assigned to the user?
- On which Tableau product is the license in use?
- On which Tableau version is the license in use?
- Did the Creator role activate through Tableau Desktop or Tableau Prep Builder?
- Has the Creator seat been activated?
- How many Creator seats are in use?
- How many Creator seats are not in use?
- When was a Creator seat was last used?



In addition to using the login-based license usage administrative view, you can also access login-based license usage data (`identity_based_activation_reporting`, `identity_based_activation_user_role_change`, and `identity_based_activation_admin_view`) in the "workgroup" PostgreSQL database of the Tableau Server repository. Before you can access this data, you must [enable access to the Tableau Server repository](#).

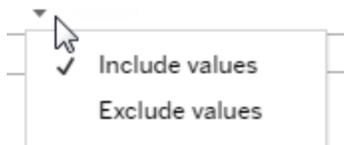
Filters

On the report screen, you can change the time window to show when seats were last used, filter on actions, filter on user name, and sort by columns.

- **Time Window in Days.** Enter the number of days for which to view login-based license management activated client usage data. You can view data for the past 30 days up to a maximum of 183 days.
- **(All).** Apply all filters to the view.

- **Activated.** Show Creator users that have activated using login-based license management.
- **in use.** Show Creator users who activated using login-based license management whose seats are in use.
- **last used.** Show when the login-based license management client was last used.
- **unassigned.** Show which login-based license management activated Creator seats are currently unassigned.
- **user name.** Show login-based license management activations in use by the specified user.

When you hover over the filter card, a drop-down icon appears. Click the icon to specify whether the view should include data that matches the filter (the default) or exclude data that matches the filter:



Which creator seats are in use in the last <nn> days?

This area of the dashboard shows a list of three types licenses (activated, in use, and unassigned). Hovering over an activated, in use, or unassigned mark gives you information including the registered user of the copy of Tableau. Click a column head to sort the list.

Which creator seats have not been used in the last <nn> days

This area of the dashboard shows a list of licenses that have not been used during the specified time period. A timeline shows the last use date. Hovering over a last use mark gives you information including the registered user of the copy of Tableau.

Desktop License Usage

Note: This view is only available to server administrators. For information about how to navigate to administrative views, see [Administrative Views](#) .

The Desktop License Usage view lets server administrators see usage data for Tableau Desktop licenses in your organization. This can help you manage licenses efficiently and determine if you need more or fewer licenses. This view can help you answer the following questions:

- Who is using a Tableau Desktop license in my enterprise?
- Have any licenses been shared or transferred?
- Is any license being used on a computer where it should not be?
- Does a specific user use their license?
- What types of licenses are being used in my enterprise?
- Do I need to convert any trial licenses?

Note: To get data about licenses, each copy of Tableau Desktop version 10.0 or later needs to be configured to send data to Tableau Server. This configuration can be done at installation time, using scripting or third-party software to install and configure Tableau, or after installation, by modifying the registry or property list file. For more information, see [Configure Desktop License Reporting](#).

In order to view license data, Desktop License Reporting must be enabled on Tableau Server. See [Enable and configure Desktop license reporting](#).

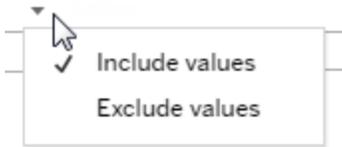
Filters

- **Product Keys.** Type a string to filter the dashboard to only those licenses that

include the string anywhere in the product key. For example, to see only licenses that begin with TDTD, type TDTD and press Return to filter the view. Click the **X** after the string to reset the filter.

- **Action.** Use this filter to control what the dashboard displays, based on the action taken. Actions are **Activate**, **Use**, and **Return** (deactivate). If the **Use** action is not selected, nothing is displayed in the top bar graph.
- **Department.** Use this filter to control what departments the dashboard displays licenses for. The filter is populated based on the **Department** values specified when Tableau Desktop is registered.
- **Select time duration in days.** Use this slider to specify the time length in days that the dashboard displays information for. The default value is 183 days.

When you hover over the filter card in the first three filters, a drop-down icon appears. Click the icon to specify whether the view should include data that matches the filter (the default) or exclude data that matches the filter:



Who has used Tableau in the last <nn> days?

This area of the dashboard shows a bar graph of three types of Tableau Desktop licenses (Perpetual, Trial, and Term) and the number of users who have used each license type during the specified time period. Hover over a license type segment to see an explanation of the license type. Click a segment to filter the rest of the dashboard for only that license type. This action filters both the tables that show licenses that have been used and those that have not been. For example, to see a list of term licenses that have been used during the time period, click the Term bar. The "used" and "not been used" lists are filtered to just show term licenses.

A table of detailed information shows under the bar graph. For each row in the table, action icons display on the right, above a timeline that shows you when the action last took place.

To see a list of the underlying data in a format that allows you to select and copy values like email or product key, click a row in the list of licenses and click the View Data icon:



The data displays in summary form. Click **Full data** to see all the data. From this view you can select and copy individual values, or download the data as a text file.

What licenses have not been used in the last <nn> days

This area of the dashboard shows a list of licenses that have not been used during the specified time period. A timeline shows the last use date. Hovering over a last use mark gives you information including the registered user of the copy of Tableau.

Desktop License Expiration

Note: This view is only available to server administrators. For information about how to navigate to administrative views, see [Administrative Views](#) .

The Desktop License Expiration view gives server administrators information about which Tableau Desktop licenses in your organization have expired or need maintenance renewal. This can help you manage licenses efficiently. This view can help you answer the following questions:

- What trial or term licenses have expired?
- What perpetual licenses have expired maintenance?
- What perpetual licenses have maintenance renewals coming up?

To renew a license or get additional renewal information, see [How to renew your Tableau licenses](#).

Note: In order to get data about licenses, each copy of Tableau Desktop version 10.0 or later needs to be configured to send data to Tableau Server. This configuration can be done at installation time, using scripting or third-party software to install and configure Tableau. For more information, see [Configure Desktop License Reporting](#).

In order to view license data, Desktop License Reporting must be enabled on Tableau Server. See [Enable and configure Desktop license reporting](#).

Filters:

- **Product Keys**—Type a string to filter the dashboard to only those licenses that include the string. For example, to only see licenses that begin with TDTD, type TDTD and press return to filter the view. Click the "x" after the string to reset the filter.
- **Department**—Use this filter to control what department(s) the dashboard displays licenses for. The filter is populated based on the Department values used when registering copies of Tableau Desktop.
- **Time Duration**—Use this filter to control the length of time for which the dashboard displays information.

The view includes the following tables, which are affected by the filters you set at the top of the view:

- **What keys have expired maintenance**—This table shows the product keys for which maintenance has expired, with a vertical line indicating the point at which the six month window for renewing maintenance closes. If maintenance for a key is expired for more than six months you need to purchase a new key in order to qualify for

support or upgrades.

- **What trial and term licenses have expired**—This shows the trial or term product keys that have expired.
- **What is the maintenance schedule for my keys**—This shows the keys and their maintenance status.

Backgrounder Dashboard

The **Backgrounder Dashboard** view is an overview of the background jobs. Using this view, you can find more information about:

- The time it takes for jobs to run.
- When backgrounder is busy or overloaded.
- Jobs that completed successfully, failed, or canceled.

Note: This view does not include flow run jobs.

In previous versions of Tableau Server, admin views were all displayed in the same workbook, in separate tabs. However, the two new admin views are displayed as separate workbooks and not part of the existing admin view workbook. You can navigate to the new admin views from the Server or Site Status page:

Dashboard	Analysis
Traffic to Views	Usage and users for published views.
Traffic to Data Sources	Usage and users for published data sources.
Actions by All Users	Actions for all users.
Actions by Specific User	Actions for a specific user, including items used.
Actions by Recent Users	Recent actions by users, including last action time and idle time.
Background Tasks for Extracts	Completed and pending task details for extract refresh.
Background Tasks for Non-Extracts	Completed and pending background task details for non-extract refresh.
Flow Performance History	View flow run performance.
Stats for Load Times	View load times and performance history.
Stats for Space Usage	Space used by published content, including extracts and live connections.
Login-based License Usage	Usage summary for login-based licenses.
Backgrounder Dashboard	Backgrounder job runtimes and busy periods.
Stats for Unused Content	Age and usage details for content that hasn't been recently used.

Summary and Filters



A - At the top, is a summary chart that tells you the number of jobs that have succeeded, failed or canceled. You can click on the sections of the bar chart to filter the information by job status. This filter is applied to the entire view and the information displayed includes only jobs with the job status that you selected.

B - You can find additional filter options by clicking on the filter icon. A Filter pane is displayed that allows you to filter by **Task type**, **Job executed at**, **Site**, **Project**, **Content owner**, **Schedule**, and **Backgrounder ID** . The Backgrounder ID is unique to a Backgrounder process. You can use this to see the information about the work done by each Backgrounder process. When you select one or more of these filters, they are applied to the entire view.

×

FILTERS

Task type

Job executed at ⓘ

Site 18 19 20 21 22 23

Project

Owner

Schedule name

Backgrounder

Job Status

Priority

0 100

Job Status

Failed

Success

Details

The Backgrounder Dashboard has four sections each showing different information about jobs. Each of these sections have more filters in the drop down menu that you can apply to that specific section.

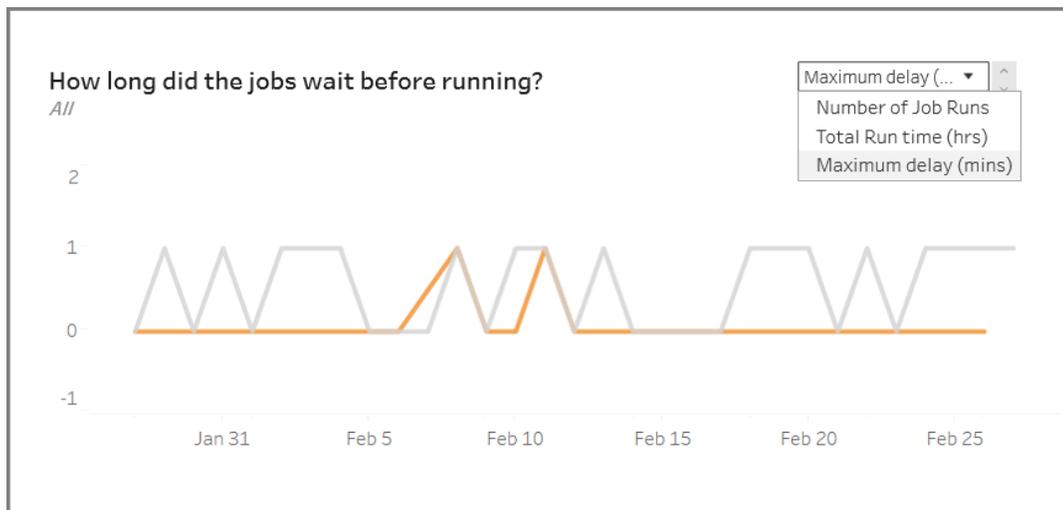
1. **The section at the top left** gives you information about the jobs with the selected job status or any filters you selected using the Filter pane. If no selection is made, all jobs are included.

The information displayed also varies based on the selection you make using the drop down:

1. **Total run time** : This shows you the total run time for all jobs with the selected job status, and any other filters you selected.

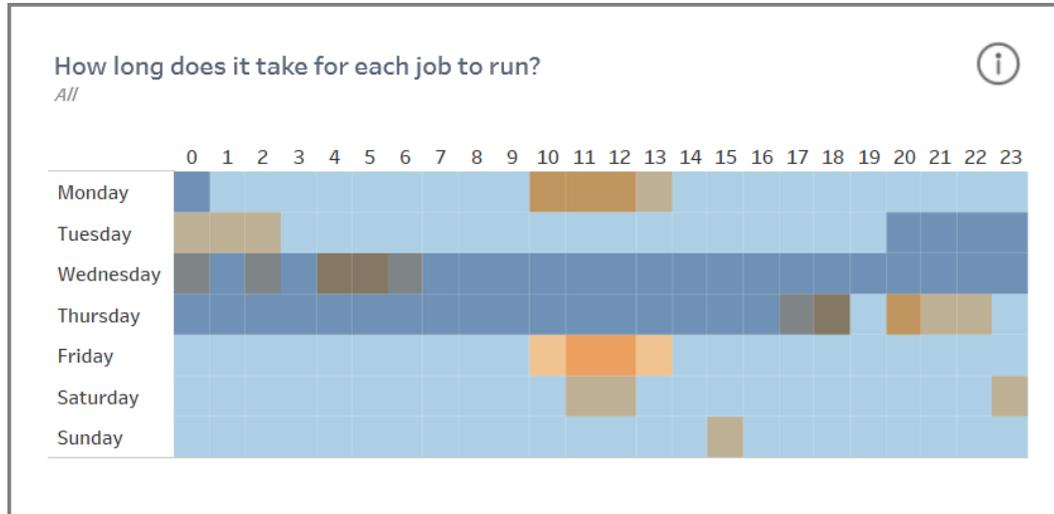
For example, if you selected, Job Status: Failed, and Task type : Subscriptions, the graph shows all subscription jobs that failed.

2. **Number of Jobs**: This shows you the number of jobs that ran for the selected job status and any other selected filters.
3. **Maximum delay**: This shows you the how long the jobs were queued before running.



2. **The section at the top right** gives you information on how busy or overloaded the

Backgrounder is on a given day and time. The information displayed however depends on the filter selections you made for the job status and other options in the Filter pane.



If you click on the information icon at the top right part of this section it, a color legend is displayed.

- The variations in the blue color correlates to the number of jobs running in that time period. The darker the blue, the more jobs that are running.
- The variations in the orange color correlates to the queue times . The darker the orange, the longer the queue time.
- Using both of these metrics, you can identify when the backgrounder is busy (dark blue) or is overloaded (dark orange).



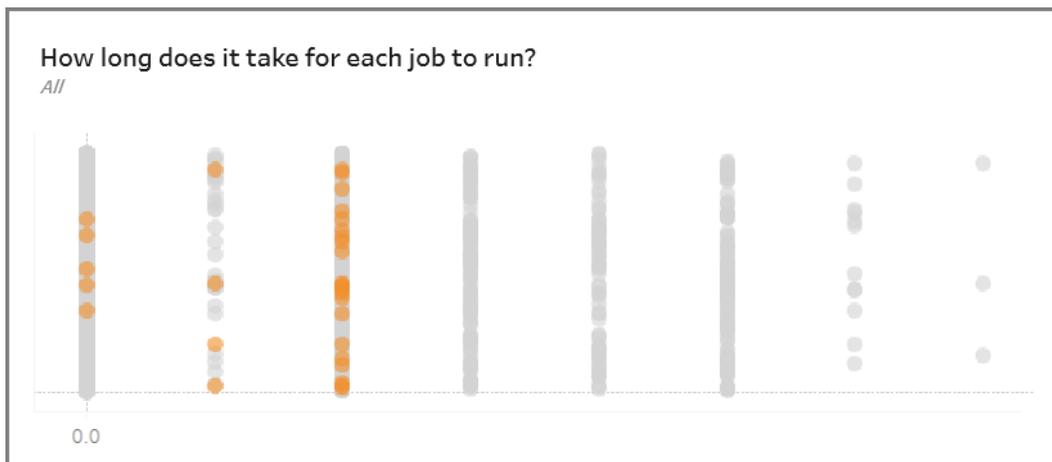
- The section at the bottom left** gives you run time information about the jobs with the selected status or any other filters you selected using the Filter pane. If no selection is made, all jobs are included. Select marks in the top sections to populate this section with details for the selected content.



The drop down selections gives you more options and the information displayed changes accordingly:

- When **Title** is selected, the run time information for each individual job is displayed.
- When **Project** is selected, the total run time for each project is displayed.
- When **Job Creator** is selected, the total run time for jobs created by a specific user is displayed.
- When **Site** is selected, the total run time for jobs on that specific site is displayed.
- When **Schedule** is selected, the total run time for jobs using that specific schedule is displayed.

4. **The section at the bottom right** tells you how long it took for each job to run.



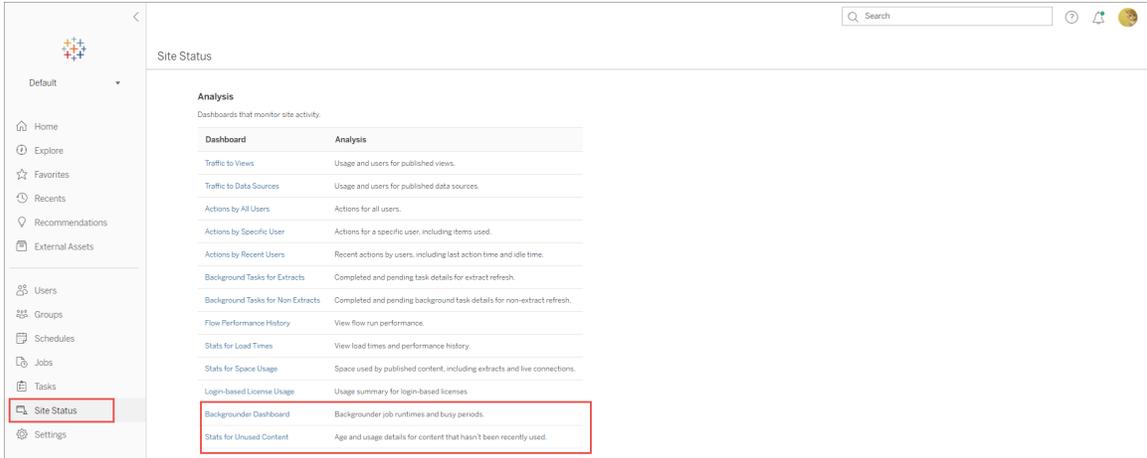
Stale Content

The Stale Content view can be used to identify content that hasn't been used or accessed in the specified time period (displayed as Stale Access Threshold). You can set that time period in days. The minimum value for the time period is 1 day and the maximum is 120 days.

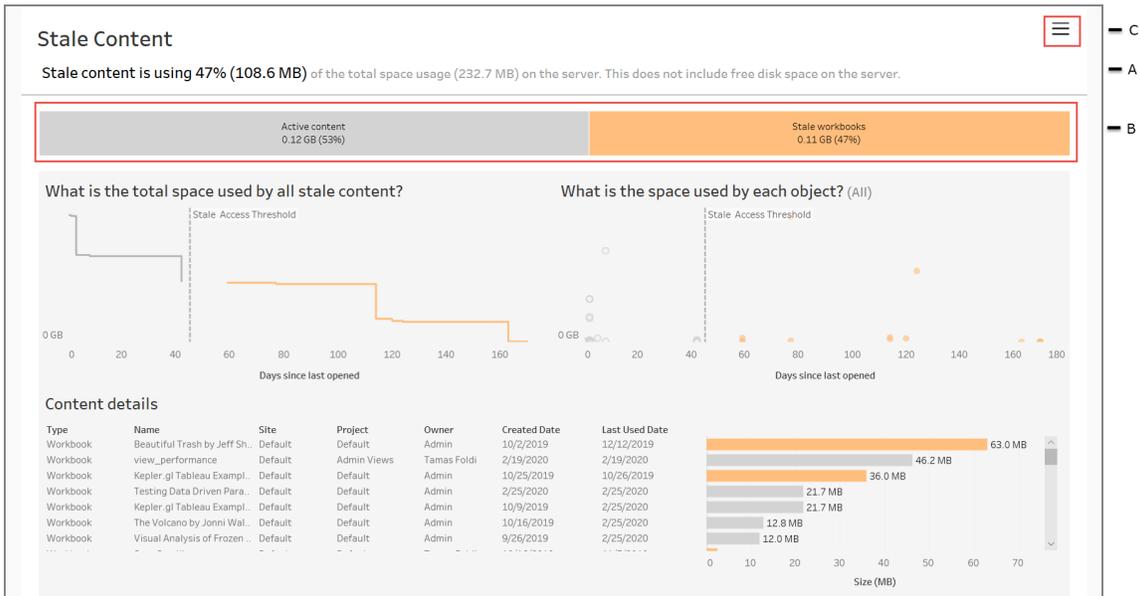
This view also provides the information about the disk space used by stale and active content.

Tableau Server on Linux Administrator Guide

In previous versions of Tableau Server, admin views were all displayed in the same workbook, in separate tabs. However, the two new admin views are displayed as separate workbooks and not part of the existing admin view workbook. You can navigate to the new admin views from the Server or Site Status page:



Summary and Filters

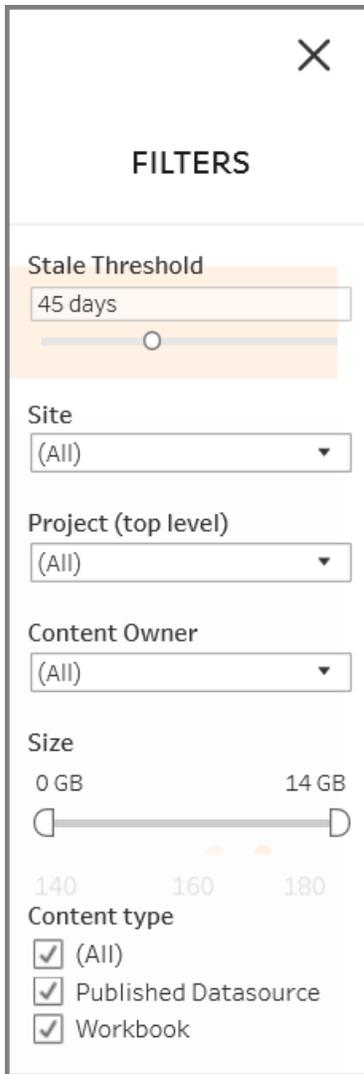


A - At the top of the view, you will see a statement that summarizes the amount of space that is used by stale content compared to the total space used. The total space used is defined as the sum total of disk space used by active and stale content.

B - This summary is followed by a chart that gives you a further breakdown of the types of stale content and content that is considered active - meaning content that has been accessed in the time period below the stale threshold. You can click on the bar chart and apply it to filter the data displayed in the view.

C - You can see and apply additional filter options by clicking the filter icon. This filter pane includes:

- Stale Threshold
- Site
- Project
- Content Owner
- Size
- Content type



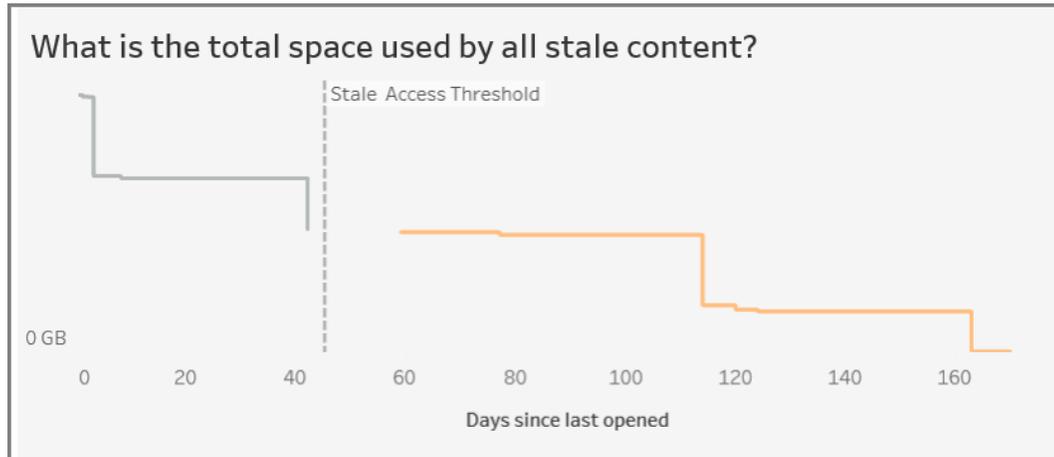
These filters are applied to the entire view.

Details

The Stale Content view has three sections as described below that provides details:

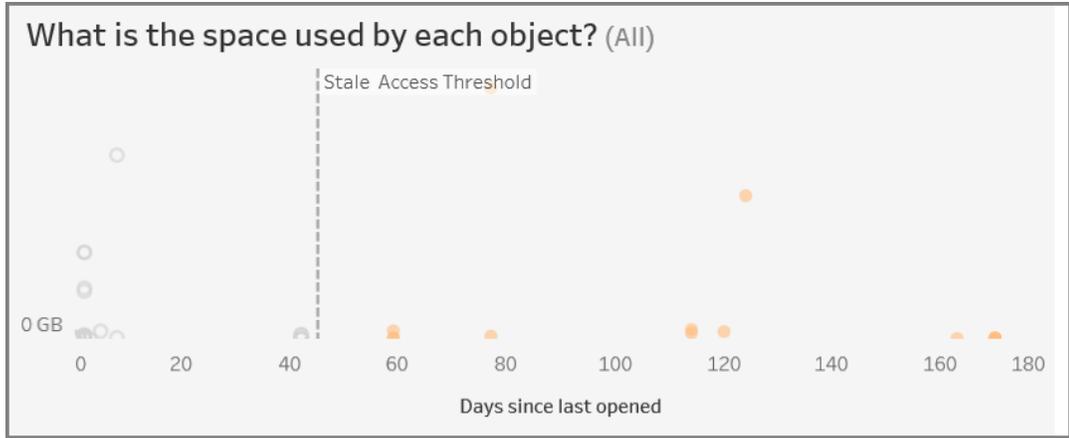
1. **The top left section** shows you the total space used for the selected content. The x-axis shows the number of days that have passed since the content was last opened, and the y-axis shows you the size. The graph also shows the stale threshold.

Set your desired staleness threshold, then use this view to identify content that is the most stale. Click the **Stale Workbooks** or **Stale Datasources** in the bar at the top to filter to the content of interest. Select the marks to the right of the **Stale Access Threshold** to see more details about content. The details are displayed in the bottom left section.



2. **The top right section** shows you the amount of space that is used by each selected content. For example, if you select Stale workbooks, the space used by each stale workbook is displayed. You can use this section to find out which content is the most stale, or is taking the most space.

This section helps you identify content that hasn't been used in a long time. Click the **Stale Workbooks** or **Stale Data Sources** on the bar at the top. Select the oldest set of unused content (marks further to the right) to see more details. The details are displayed in the bottom left section. This can be your next set of content to consider for archiving or deleting as these are contents that nobody has been using, regardless of size.



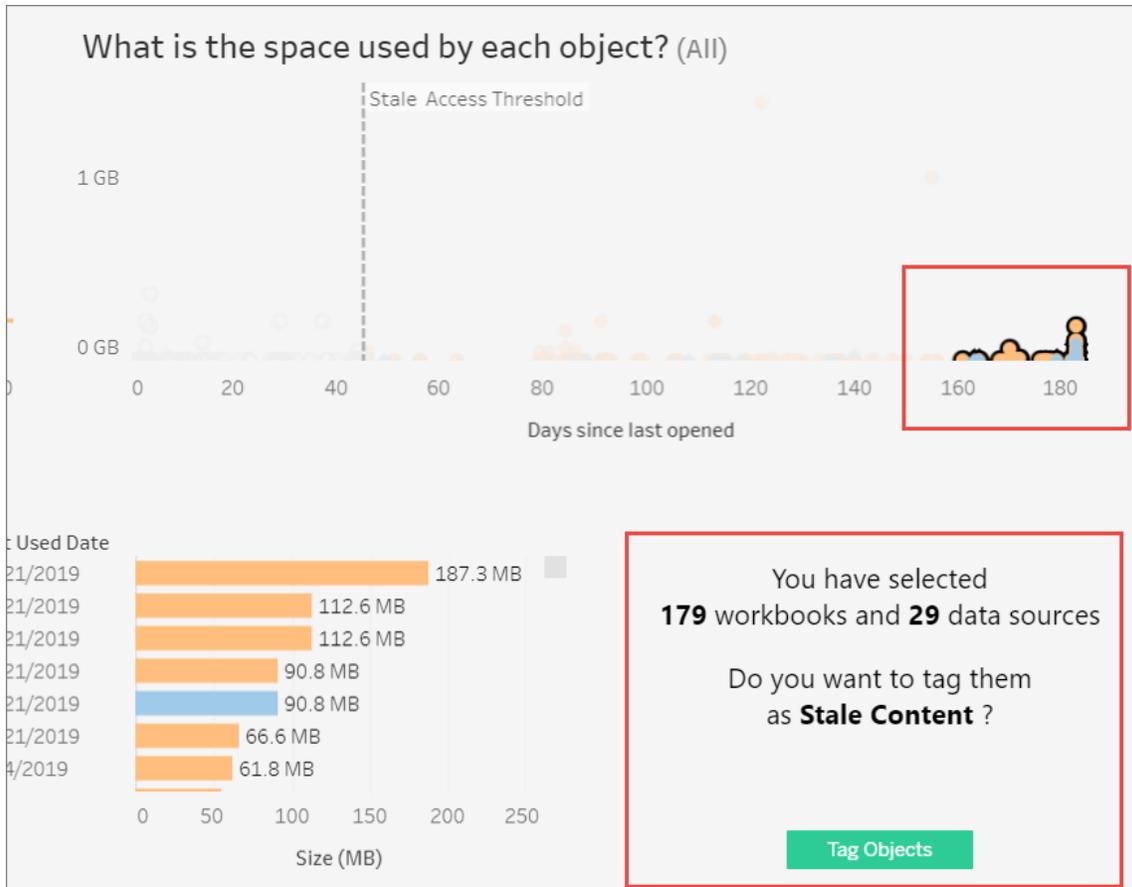
3. The section at the bottom shows detailed information about the selected content as shown below:

Type	Name	Site	Project	Owner	Created Date	Last Used Date	Size
Workbook	Beautiful Trash by Jeff Sh.	Default	Default	Admin	10/2/2019	12/12/2019	63.0 MB
Workbook	view_performance	Default	Admin Views	Tamas Foldi	2/19/2020	2/19/2020	46.2 MB
Workbook	Kepler.gl Tableau Examp...	Default	Default	Admin	10/25/2019	10/26/2019	36.0 MB
Workbook	Testing Data Driven Para...	Default	Default	Admin	2/25/2020	2/25/2020	21.7 MB
Workbook	Kepler.gl Tableau Examp...	Default	Default	Admin	10/9/2019	2/25/2020	21.7 MB
Workbook	The Volcano by Jonni Wal...	Default	Default	Admin	10/16/2019	2/25/2020	12.8 MB
Workbook	Visual Analysis of Frozen ...	Default	Default	Admin	9/26/2019	2/25/2020	12.0 MB

Archive or Delete Stale Content

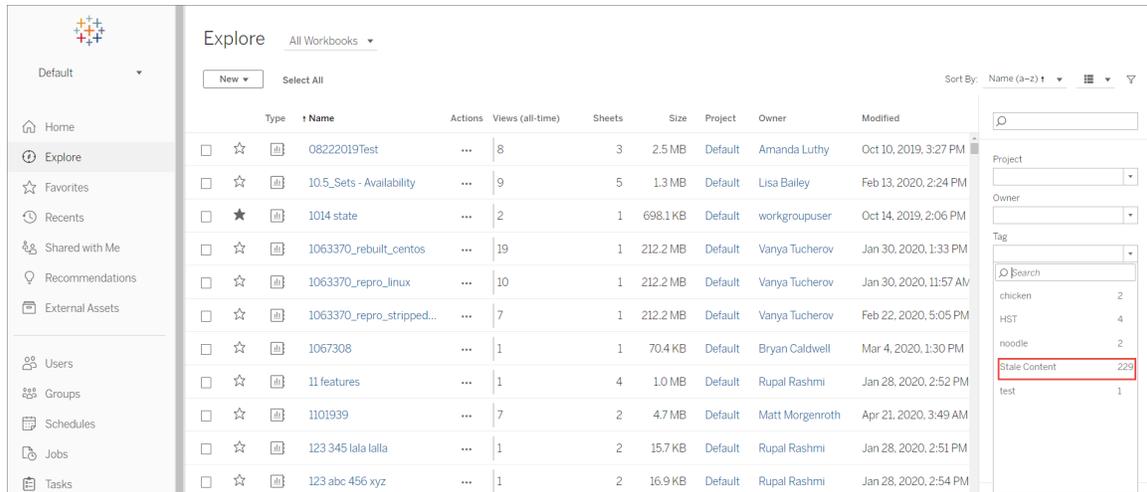
Starting in 2020.3, the Stale Content admin view includes a feature that allows you to select and tag content as stale. You can select content from either the section at the bottom or the top right section. When you make a selection, you will see the number of objects and the type of content that are selected, as seen in the screen shot below. Click the **Tag Objects** button to tag the selected content.

In the screen shot shown below, content that has not been opened in the last 160 days or more are selected to be tagged as stale content.



To find all the tagged content, on Tableau Server web interface, navigate to **Explore**. Select the **Stale Content** filter to see all the content that have the stale content tag. You can now select the content and either move it to a project that you use for archiving or delete the content.

Tableau Server on Linux Administrator Guide



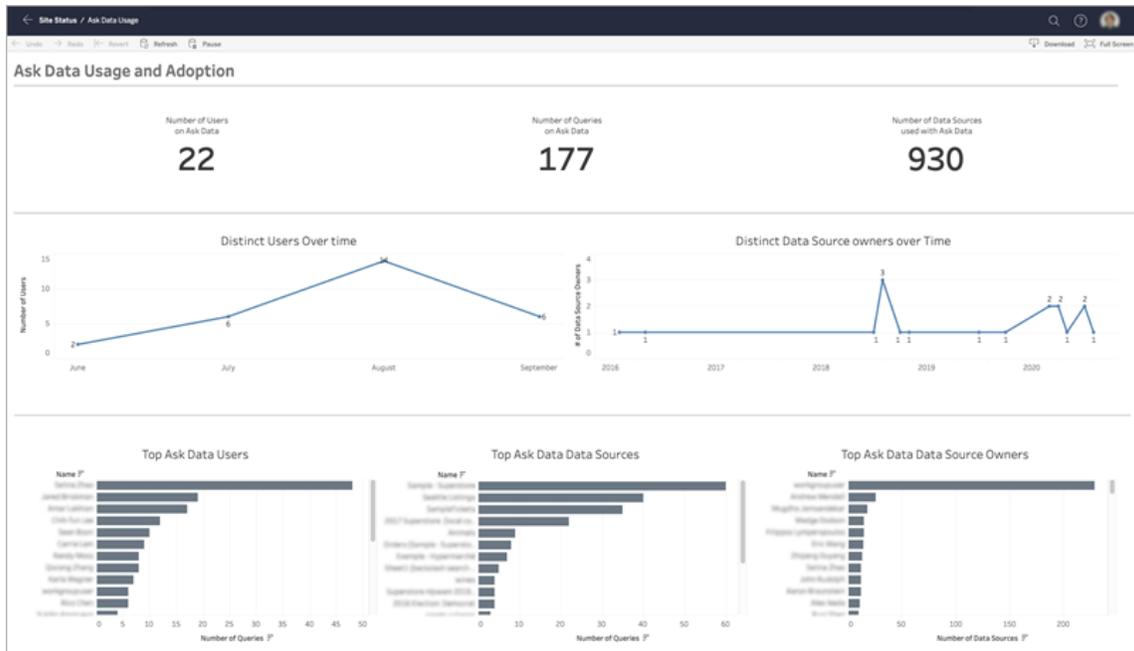
The screenshot shows the Tableau Server 'Explore' interface. On the left is a navigation sidebar with options like Home, Explore, Favorites, Recents, Shared with Me, Recommendations, External Assets, Users, Groups, Schedules, Jobs, and Tasks. The main area displays a table of workbooks with columns for Name, Actions, Views (all-time), Sheets, Size, Project, Owner, and Modified. A search filter on the right is set to 'Stale Content', showing a count of 229 for that tag. Other tags visible include 'chicken' (2), 'HST' (4), 'noodle' (2), and 'test' (1).

Type	Name	Actions	Views (all-time)	Sheets	Size	Project	Owner	Modified
□ ☆	08222019Test	...	8	3	2.5 MB	Default	Amanda Luthy	Oct 10, 2019, 3:27 PM
□ ☆	10.5_Sets - Availability	...	9	5	1.3 MB	Default	Lisa Bailey	Feb 13, 2020, 2:24 PM
□ ★	1014 state	...	2	1	698.1 KB	Default	workgroupuser	Oct 14, 2019, 2:06 PM
□ ☆	1063370_rebuilt_centos	...	19	1	212.2 MB	Default	Vanya Tucherov	Jan 30, 2020, 1:33 PM
□ ☆	1063370_repro_linux	...	10	1	212.2 MB	Default	Vanya Tucherov	Jan 30, 2020, 11:57 AM
□ ☆	1063370_repro_stripped...	...	7	1	212.2 MB	Default	Vanya Tucherov	Feb 22, 2020, 5:05 PM
□ ☆	1067308	...	1	1	70.4 KB	Default	Bryan Caldwell	Mar 4, 2020, 1:30 PM
□ ☆	11 features	...	1	4	1.0 MB	Default	Rupal Rashmi	Jan 28, 2020, 2:52 PM
□ ☆	1101939	...	7	2	4.7 MB	Default	Matt Morgenroth	Apr 21, 2020, 3:49 AM
□ ☆	123 345 lala lalla	...	1	2	15.7 KB	Default	Rupal Rashmi	Jan 28, 2020, 2:51 PM
□ ☆	123 abc 456 xyz	...	1	2	16.9 KB	Default	Rupal Rashmi	Jan 28, 2020, 2:54 PM

If you have Advanced Management on your Tableau Server, you can use the Tableau Content Migration Tool to manage archiving stale content on a regular schedule. For example, you can build a plan that runs on a regular schedule that can automatically pick up content tagged as Stale Content and move it to an Archive project. After a certain amount of time, the content in this project can be purged from the system. For more information see, [Migration Plans: Workbooks](#).

Ask Data Usage

The Ask Data Usage view is a pre-built dashboard that allows site or server admins to see and understand the usage patterns and value of Ask Data for a site. Admins can see the growth of engagement with Ask Data and monitor the results of internal training or roll-outs. The dashboard highlights the top Ask Data users, data sources, and data source owners, along with some headline value metrics.



To enable Ask Data, see [Disable or Enable Ask Data for a Site](#).

Explore the dashboard

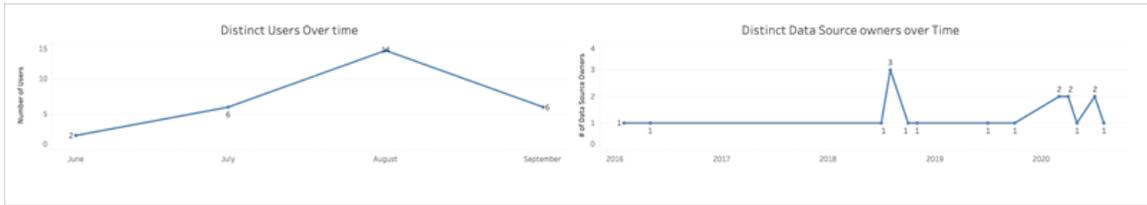
The Ask Data Usage view provides information about Ask Data across the entire site. You can use the following metrics to understand user engagement and help drive self-service analytics adoption in your organization.



At the top of the dashboard, three headline metrics provide an overview of Ask Data usage on the site.

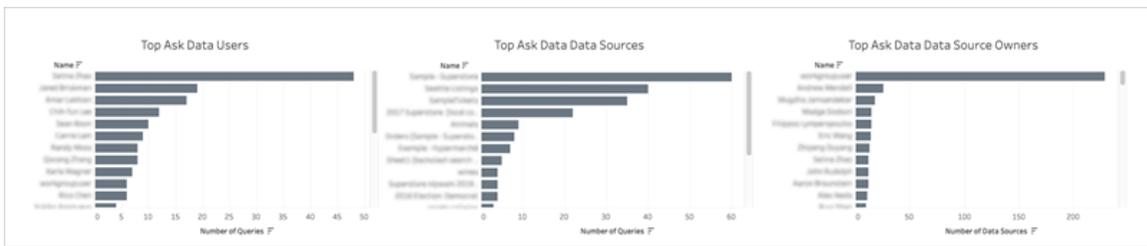
- **Number of Users on Ask Data** - This shows the total number of Ask Data users on the site.
- **Number of Queries on Ask Data** - This shows the total number of Ask Data queries issued on the site.

- **Number of Data Sources Used with Ask Data** - This shows the total number of data sources used with Ask Data.



In the middle of the dashboard, two line charts show you how Ask Data is used over time.

- **Distinct Users Over Time** - This shows the distinct number of Ask Data users over time.
- **Distinct Data Source Owners Over Time** - This shows the distinct number of data source owners over time.



At the bottom of the dashboard, three bar charts list the top Ask Data users, data sources, and data source owners.

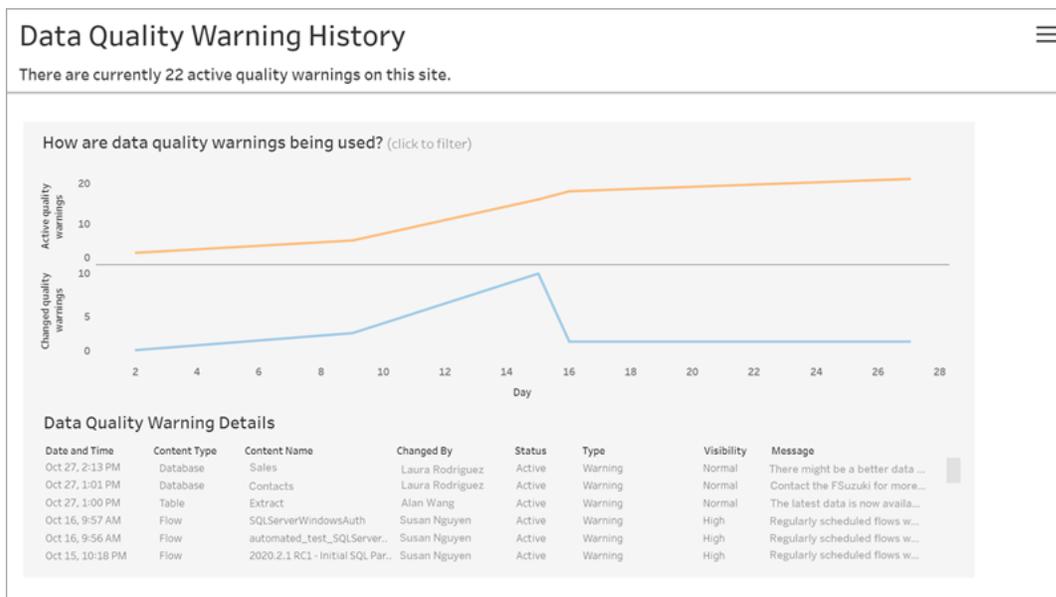
- **Top Ask Data Users** - This lists the top Ask Data users and the total number of queries issued by each user.
- **Top Ask Data Data Sources** - This lists the top Ask Data data sources and the total number of queries issued for each data source.
- **Top Ask Data Data Source Owners** - This lists the top Ask Data data source owners and the total number of data sources owned by each user.

Data Quality Warning History

When Tableau Catalog is enabled in your environment, site administrators can see how data quality warnings are being used on the site using the pre-built admin view, Data Quality Warning History.

For more information about Tableau Catalog, part of the Data Management Add-on, see "About Tableau Catalog" in the [Tableau Server](#) or [Tableau Online](#) Help.

From the Site Status page, select the Data Quality Warning History dashboard:



The dashboard shows how many data quality warnings are active over a period of time. It also shows how many warnings have been changed (created, updated, and deleted) over that same time period.

See warning details

Under the line charts are the details about the data quality warnings, including:

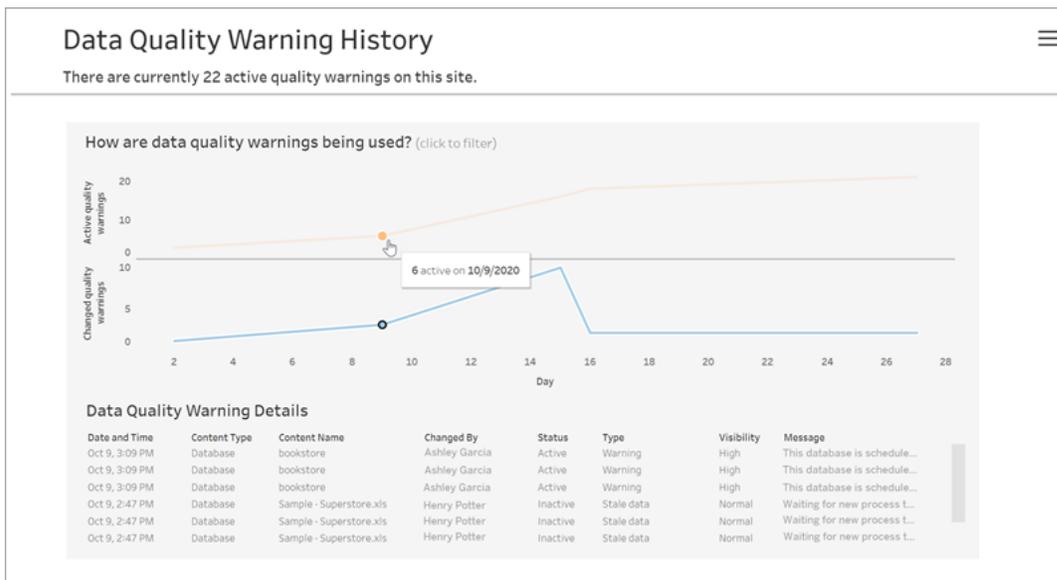
- **Date and Time** - When the warning was created or last changed.
- **Content Type** - The type of asset on which the warning is set. For example, database, table, data source, or flow.

Tableau Server on Linux Administrator Guide

- **Content Name** - Name of the asset on which the warning is set.
- **Changed By** - Name of the person who created or last changed the warning.
- **Status** - If the warning is active or inactive.
- **Type** - Warning type can be Stale data, Warning, Deprecated, Sensitive data, or Under maintenance.
- **Visibility** - The warning can be configured to have normal (the default) or high visibility.
- **Message** - The message the warning creator wrote to display to users when they see the details of the warning.

Filter warning history

When you review data quality warning history, you can click a mark on the view to filter the details shown below the view.

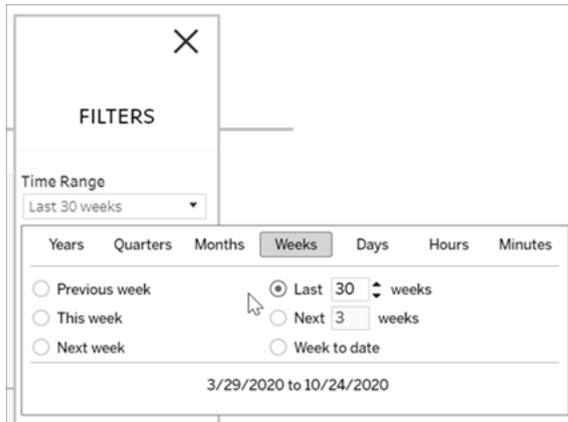


The numbers on the Day axis represent the date within the time range. For example, if today is November 18, and you filter for the last 7 days, the Day axis shows 12-18.

More filters are available when you click the filter icon in the upper right corner: filter by time range and by content type.

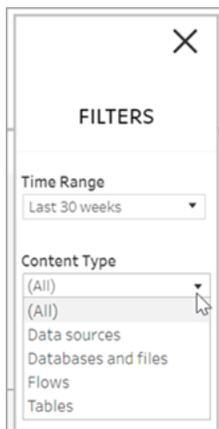
Filter by time range

You can configure the time range from years to minutes.



Filter by content type

You can see all the data quality warnings on your site, or you can filter to see warnings for specific types of assets, like data source or table:



Access data quality warning history data

In addition to using the Data Quality Warning History admin view, you can also access data quality warning history data in the "workgroup" PostgreSQL database of the Tableau Server repository. Before you can access this data, you must [enable access to the Tableau Server repository](#).

See [About the Tableau Data Dictionary](#) for a link to open the latest data dictionary where you can search for these tables with quality warning history data:

- `historical_events`
- `historical_event_types`
- `hist_data_quality_indicators`
- `data_quality_indicators`

Who can do this

To set a data quality warning, you must be a server or site administrator.

Create Custom Administrative Views

In addition to the pre-built administrative views available on the Maintenance page on the Server, you can use Tableau Desktop to query and build your own analyses of server activity. To do this, you can connect to and query views in the Tableau Server repository using one of two built-in users: the "tableau" or "readonly" user.

To connect to the Tableau Server repository, see [Collect Data with the Tableau Server Repository](#).

- The **tableau** user—The tableau user has access to special views and a subset of tables in repository database. These views and tables are provided so that administrators can create custom administrative views. Tableau makes an effort to limit changes to these tables and views so that custom views built with them do not break.
- The **readonly** user—The readonly user has access to a large number of the repository tables, providing more data about server usage. Administrators can use these to create custom administrative views too, but many of the tables are intended primarily to support the functioning of Tableau Server and may be changed or removed without warning. This means that views created from these tables can break when the database structure is changed.

For examples of custom administrative views, see the [Tableau Community](#). You can also use the temporary workbook that is generated when you view the build-in Administrative views.

Before you can connect using one of the built-in users, you must enable access to the Tableau Server database. After doing this you can use Tableau Desktop to connect to and query the database as the tableau user or the readonly user.

The `tsm configuration set` option `auditing.enabled` controls whether Tableau Server collects historical user activity and other information in the repository. It is enabled by default. Be aware that collecting historical events impacts the size of Tableau Server's backup file (.tsbak).

- All `hist_` tables are controlled by the `tsm configuration set` option `wgserv-er.audit_history_expiration_days`, which controls how many days of event history are kept in the repository and has default value of 183 days.
- The `_http_requests` table is cleaned of all data older than 7 days when you run `tsm maintenance cleanup` with the `--http-requests-table` option. For more information, see [Remove Unneeded Files](#).
- The `_background_tasks` table is cleaned automatically and keeps data for the last 30 days.
- All other tables with names that begin with a "_" prefix contain current data.

For more information about the tables in the Tableau Server repository, see [Workgroup Database Data Dictionary](#).

Performance

You can monitor and tune the performance of Tableau Server.

Tableau Server Performance Overview

When you take the time to understand the performance of Tableau Server, you make it easier to serve your users by improving the efficiency of Tableau Server. Although every server environment is unique, and there are many variables that can impact performance, the general steps that you take to understand and act on performance data in Tableau Server are the same.

- **Notifications.** Configure email notifications for important server events. For example, you can receive notifications when server processes become unavailable and when the server is running out of disk space.
- **Monitoring.** Collect and analyze data about Tableau Server to understand how well the server is performing.
- **Tuning.** Make adjustments to tasks, process configurations, and more to improve the performance of Tableau Server.
- **Troubleshooting.** Identify bottlenecks in resources, workbooks, and more to improve the performance of Tableau Server.

General Performance Guidelines

Hardware and Software

Add more cores and memory: Regardless of whether you're running Tableau Server on one computer or several, the general rule is that more CPU cores and more RAM will give you better performance. Make sure you meet the Tableau Server recommended hardware and software requirements.

If you are running Tableau Server in a virtual environment, use your VM host's best practices for vCPU allocation in relation to the number of physical CPU cores on the VM host.

Configuration

Schedule refreshes for off-peak hours: Backup tasks tend to stall other background tasks until the backup is completed. Use the Background Tasks for Extracts administrative view to see your refresh and backup task schedules. Your refresh tasks should be scheduled for off-peak hours that don't overlap with your backup window.

Look at caching: Caching helps Tableau Server respond to client requests quickly, especially for views that connect to live databases. Use the `tsm data-access caching list` command to confirm the caching frequency is set to `low` (this is the default).

Tableau Server uses a query cache to store query results. The size of the query cache is automatically set based on the amount of available system memory, as long as you have not set it manually. The query cache consists of the logical query cache, the metadata cache, and the native query cache. The default settings are suitable for most situations but it is possible to manually configure them using the TSM command line interface. The TSM settings are: `native_api.InitializeQueryCacheSizeBasedOnWeights`, `native_api.QueryCacheMaxAllowedMB`, `native_api.LogicalQueryCacheMaxAllowedWeight`, `native_api.MetadataQueryCacheMaxAllowedWeight`, `native_api.NativeQueryCacheMaxAllowedWeight`, and `native_api.QueryCacheEntryMaxAllowedInPercent`. For more information, see `native_api.InitializeQueryCacheSizeBasedOnWeights`.

Consider changing two session memory settings:

- **VizQL session timeout limit:** The default VizQL session timeout limit is 30 minutes. Even if a VizQL session is idle, it is still consuming memory and CPU cycles. If you can make do with a lower limit, use `tsm configuration set Options` to change the `vizqlserver.session.expiry.timeout` setting.
- **VizQL clear session:** By default, VizQL sessions are kept in memory even when a user navigates away from a view. This reduces the need to rebuild views but consumes more session memory. To free up memory, you can end sessions when users

leave views by changing the value of the `vizqlserver.clear_session_on_unload` setting to `true`. (Regardless of this setting, sessions for the Tableau Mobile app are always kept in memory, improving mobile performance.)

Assess your process configuration: Tableau Server is divided into six different components called server processes. While their default configuration is designed to work for a broad range of scenarios, you can also reconfigure them to achieve different performance goals. Specifically, you can control on which computers the processes run and how many are run. See Performance Tuning Examples for general guidelines for one-, two-, and three-node deployments.

Server Resource Manager (SRM)

The Server Resource Manager (SRM) monitors the system resources each Tableau process is using as well as tracking the total usage of Tableau Server on the system. If either a specific process or the product as a whole takes up too much system resources, SRM can notify the processes to free the resources or restart those processes.

The thresholds that determine when SRM will notify or restart a process are set in the SRM configuration options. The Tableau development team has set the default settings based on internal testing and don't recommend you change these settings directly.

If you are seeing excessive system resource usage, we recommend that you contact Tableau Support to help determine if these configuration options need to be modified to solve the problem or issue that you are seeing.

Performance Monitoring Overview

When you monitor a server, you collect and analyze data that signals whether the server is performing badly or running into problems. For example, if you notice that your server is using 100% of its processing capacity for long periods of time, you know that there's a problem.

The data that you need to collect and analyze can be broken down into the following broad categories:

- Resource usage data—how Tableau Server uses hardware resources like disk space, memory, and processors.
- Session and load time data—how users interact with Tableau Server, including how long it takes for views to load and how many concurrent users there are.
- Background task data—how Tableau Server runs tasks that are not directly tied to a user action. For example, background tasks include extract refresh tasks, subscription tasks, and more.

Some of this data, including load time data and extract refresh data, is already accessible from the administrative views that are built into Tableau Server. However, to collect resource usage data you need to use an external performance monitoring tool. To collect additional load time data and background task data, you can connect to the Tableau Server repository.

For more information on the built-in administrative views, see [Administrative Views](#).

Note: To use the sample workbook and to publish views to Tableau Server, you must have Tableau Desktop.

Collect Data with the Tableau Server Repository

The Tableau Server repository is a PostgreSQL database that stores data about all user interactions, extract refreshes, and more. You can enable access to the repository and use the data in it to help analyze and understand Tableau Server performance.

Looking for Tableau Server on Windows? See [Collect Data with the Tableau Server Repository](#).

After you enable access to the Tableau Server repository, you can create views with data from the repository. The views that you create with this data are sometimes called custom

administrative views. In addition to being used for performance monitoring, custom admin views can be used for tracking user activity, workbook activity, and more. For more information on the type of data that you can use for these views, see [Create Custom Administrative Views](#) and [About the Tableau Server Data Dictionary](#). Alternatively, if you are only interested in performance data, you can use the preselected database tables in the sample performance workbook.

Enable access to the Tableau Server repository

You can use Tableau Desktop to connect to and query the Tableau Server repository using two built-in users. The user named `tableau` has access to several database views you can use as part of building your own analyses of Tableau Server activity. The user named `readonly` has access to additional database tables that you can use to create views for even more in-depth analysis and this is the user we recommend you use.

Before you can connect to the repository, you need to enable access for the `readonly` user to the database. Use the `tsm data-access repository-access enable` command to enable repository access. When you enable repository access, you also create a password for the `readonly` user. You will use this password to access to the repository. You may also need to have port 8060 opened on the repository node so you can connect to the database.

1. Verify that port 8060 is opened on the computer where the repository is installed. This is a requirement if you are connecting remotely.
2. Enable repository access and create a readonly user password:

```
tsm data-access repository-access enable --repository-user-name readonly --repository-password <PASSWORD>
```

If your password includes special characters, you may need to escape the characters or enclose the password in quotes. Refer to the documentation for the Linux distro that you are running for information about passing special characters in bash shell.

This command will restart Tableau Server.

Note: If you later decide that you want to disable remote access to the Tableau Server repository, use the `tsm data-access repository-access disable` command. The command disables external access to the repository. This will not disable access from localhost. For more information, see `tsm data-access repository-access disable`.

Connect to the Tableau Server repository

This section describes how to connect to a custom set of tables from Tableau Server repository. For more information on the tables that you can connect to, see [About the Tableau Server Data Dictionary](#).

1. In Tableau Desktop select **Data > Connect to Data**, and then select **PostgreSQL** as the database to connect to.

Note: You might need to install the PostgreSQL database drivers. You can download drivers from www.tableau.com/support/drivers.

2. In the PostgreSQL connection dialog box, enter the name or URL for Tableau Server in the **Server** box. If you have a distributed server installation, enter the name or IP address of the node where the repository is hosted.

Connect using the port you have set up for the `pgsql.port`, which is 8060 by default.

3. Specify `workgroup` as the database to connect to.
4. Connect using the user and the password you specified.
5. Click the **Require SSL** option if you have configured Tableau Server to use SSL for connecting to the repository. For more information, see [Configure Postgres SSL to Allow Direct Connections from Clients](#).
6. Click **Connect**.

PostgreSQL

Server: Port:

Database:

Enter information to sign in to the database:

Authentication:

Username:

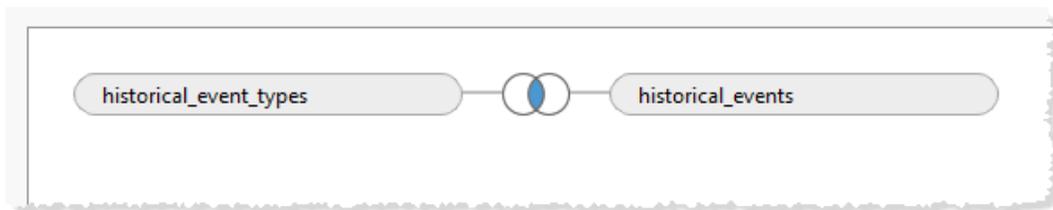
Password:

Require SSL

[Initial SQL...](#)

7. Select one or more tables to connect to.

The `tableau` user has access to all of the tables that start with an underscore or with `hist_`. For example, you can connect to `_background_tasks` and `_data-sources`. The `hist_` tables include information about server users that isn't currently presented in the Actions by Specific User view. The `readonly` user has access to additional tables that can be used to query other information about server usage.



8. Click **Go to Worksheet**.

PostgreSQL Version

Use the following steps to find the version of PostgreSQL used by Tableau Server:

1. Log into Tableau Server as a user with sudo access.
2. Use the following command to view the version of PostgreSQL installed on the machine:

```
$ postgres --version psql --version
```

If the above command results in an error, you may need to locate the directory. Use the following steps to locate the directory:

1. `$ locate find /opt/tableau -name psql`
2. Navigate to the path and issue the version command to find the PostgreSQL version:

```
$ postgres psql --version
```

You can also connect to the workgroup database and issue the following query to get the version: `select version()`

About the Tableau Server Data Dictionary

The Tableau Server Data Dictionary includes information about the tables and views in the "workgroup" PostgreSQL database of the Tableau Server repository. This database provides persistent storage for Tableau Server and is primarily intended to support that application. The Data Dictionary is not a complete description of all tables and fields in the database, and is provided for customers who want to query the database for information about usage on Tableau Server. Because the database and its contents are intended to support Tableau Server, the structure and contents may change without warning. This means any custom views you build from directly querying the database could break.

[Open the Data Dictionary](#) (new window).

Performance Tuning

This section describes how to use the performance data that you collect to identify ways to improve the performance of Tableau Server. Because no two server environments are identical, we can't provide hard and fast rules for tuning server performance. However, you can draw conclusions about performance from patterns in the data that you collected.

For example, are there recurring spikes? Do any of the patterns that you notice in the administrative views correspond to similar patterns in a monitoring tool? Observing patterns like this can guide you in testing and incremental tuning.

Most performance tuning for Tableau Server boils down to these general approaches:

- **Optimize for User Traffic:** This tunes the server to respond to user requests and to display views quickly.
- **Optimize for Extracts:** This tunes the server to refresh extracts for published data sources. You might want to optimize for extract refreshes if your organization has a lot of data and the data needs to be as up to date as possible.
- **Optimize for Extract Query Heavy Environments:** This is a specialized server configuration to optimize for query performance of workbooks that use extracts as their data source.

Rendering views and refreshing extracts generate the most load on the server, so you should optimize for the task that your organization is most interested in.

As a best practice, optimize your workbooks for performance. For more information and resources on how to optimize your workbooks, see [Optimize Workbook Performance](#).

Topics in this section:

Optimize for User Traffic

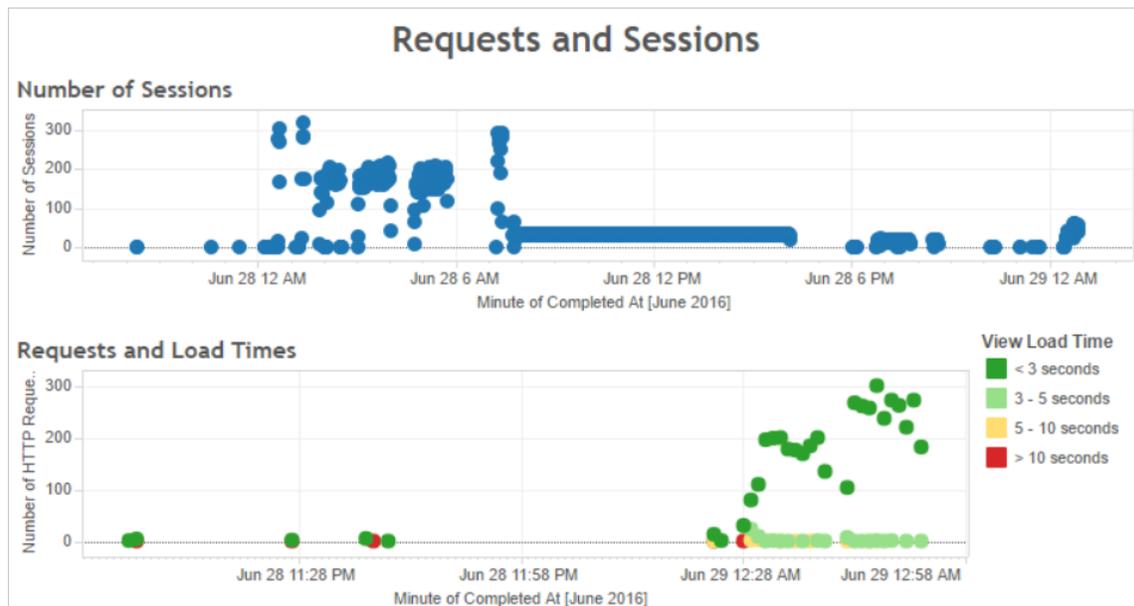
You can optimize for traffic if you have many active Tableau Server users and few published data sources that need extract refreshes.

- When to optimize for user traffic
- [Ways to optimize for user traffic](#)

When to optimize for user traffic

Slow load times for views

Use the **Requests and Sessions** dashboard of the sample performance workbook to analyze how long views take to load.



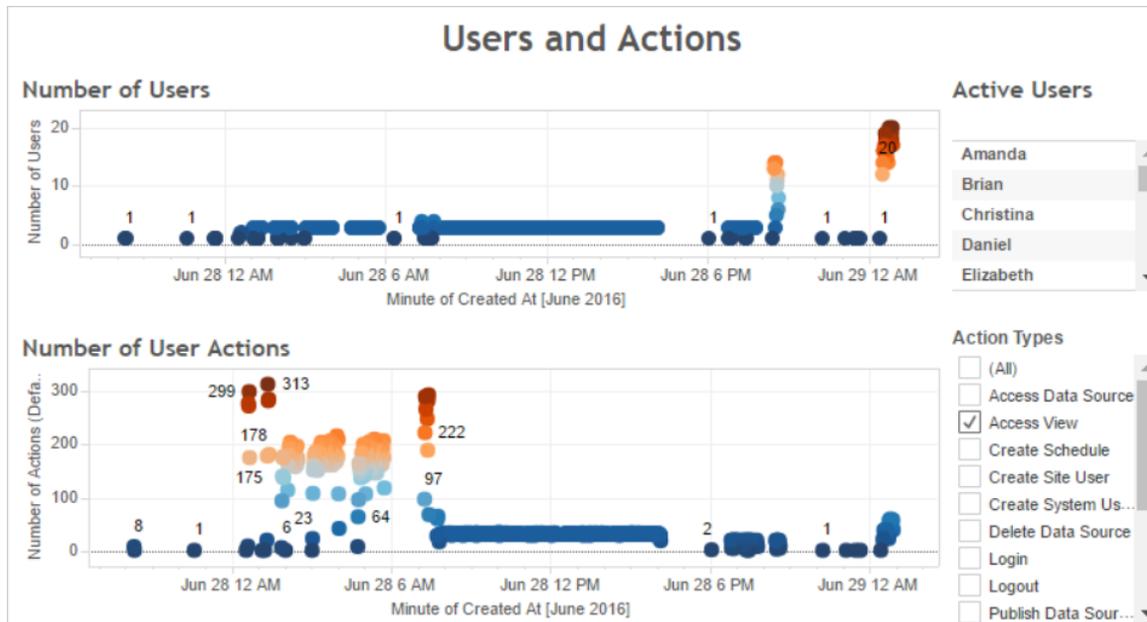
If multiple views take longer than 10 seconds to load, and if the slow load times correspond to a large number of sessions, that can indicate that user traffic is slowing down the server.

However, if a particular view takes a long time to load regardless of when it is viewed, it's a sign that the workbook for the view needs to be optimized. You can identify which workbooks need to be optimized with the Stats for Load Times administrative view. Some simple ways of optimizing workbooks includes displaying less information in each view or breaking up views, reducing the number of filters, and using data extracts.

Tableau Server on Linux Administrator Guide

High resource usage corresponding to user traffic

If your server displays high CPU and memory usage during peak traffic hours, you should optimize for user traffic. To determine peak traffic hours and analyze how many concurrent users are on your server, use the **Users and Actions** dashboard. In addition, you can use the Traffic to Views administrative view to see how much user traffic involves accessing views (as opposed to performing administrative functions, publishing, or other tasks).

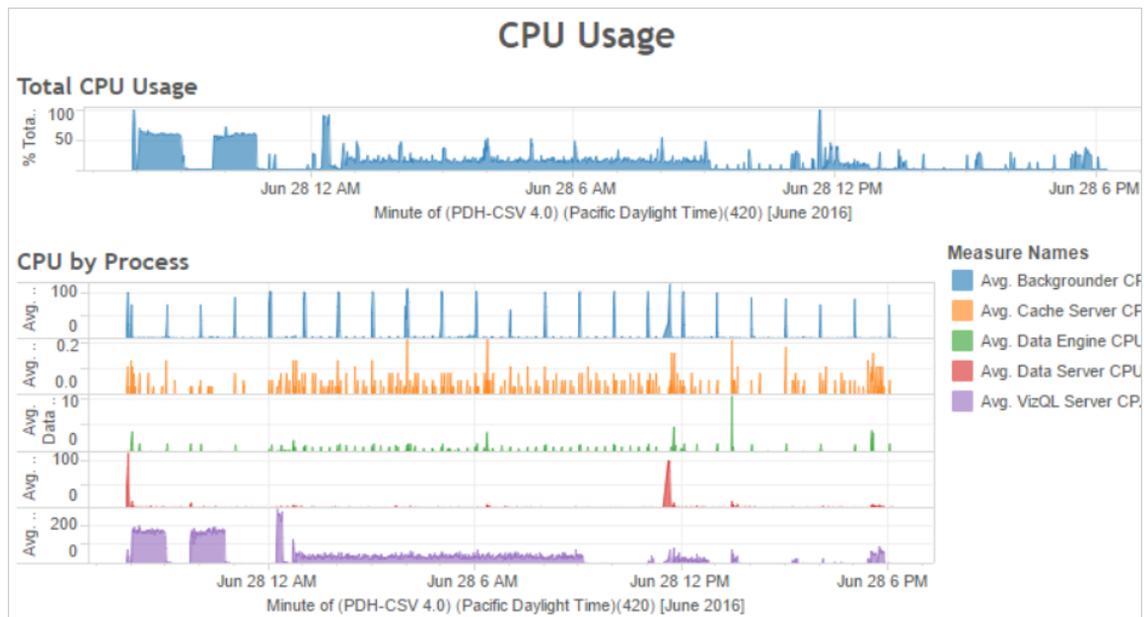


If you click a point in the **Number of Users** view, the dashboard displays the users that were active at the time and the number of user actions that those users performed. By default, the only user actions displayed are user views, but you can use the **Action Types** filter to display additional user actions.

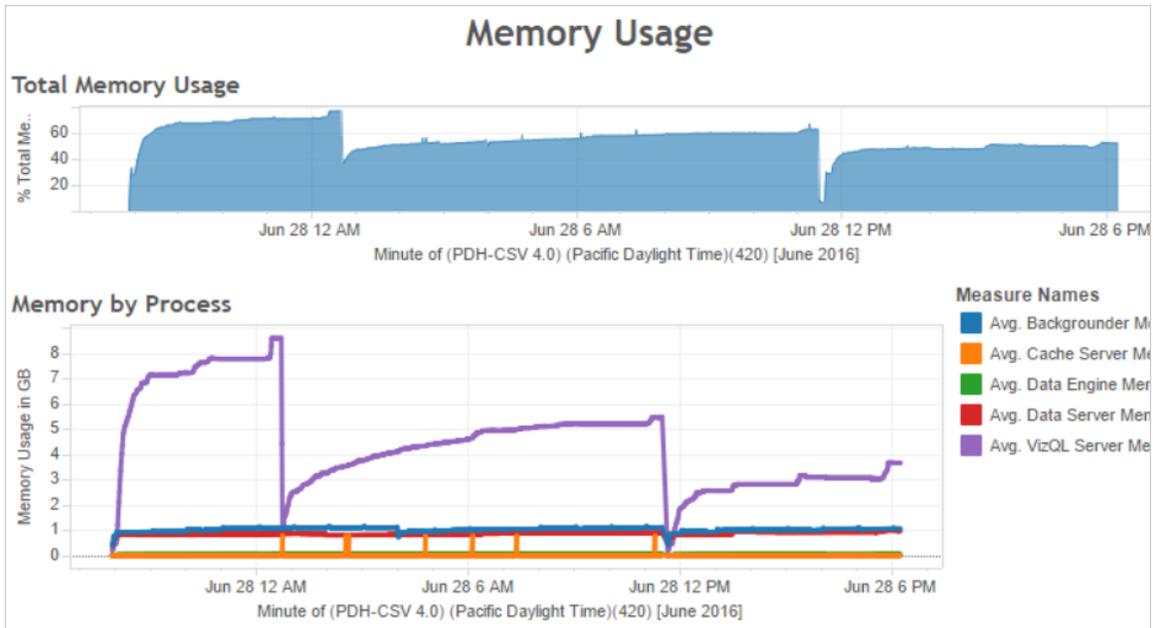
Make a note of the times of day when there are many concurrent users and views so that you can compare this to resource usage. As a rule of thumb, the number of users should correspond to a high number of user actions. However, the view in this example displays an artificially high number of actions for a single user as part of a load generation test. As an example, you can compare the high number of views at 12 AM on June 28th with the resource usage in the dashboard illustrated later.

Use the **CPU Usage** dashboard to display the percent of total CPU usage and the percent of CPU usage for each process. In the following example, note the large spike in total CPU usage and in the VizQL server process at 12 AM on June 28th. Because the VizQL server process loads and render views, the VizQL server process is often the first process to show strain under high user traffic.

Note: The percent of CPU usage for individual processes may add up to more than 100 percent. This is because processor utilization for individual processes is measured for a given processor core. By contrast, the total CPU usage is measured for all processor cores.



Use the **Memory Usage** dashboard to display the percent of total memory usage and the average memory usage in gigabytes. As a general rule, memory usage increases steadily with user traffic. Here again the VizQL server process is the first to show strain under high traffic.



Ways to optimize for user traffic

When high user traffic corresponds to high resource usage as it does in the example shown previously, you should optimize for user traffic.

Adjust the number of VizQL server processes

The most effective way of optimizing for user traffic is to adjust the number of VizQL server processes. Add one VizQL server process at a time and measure the effect with more performance monitoring. Because VizQL server processes can consume a lot of CPU and memory, adding too many processes can slow down the server instead. If you see consistently high memory usage, try to reduce the number of VizQL server processes to reduce the amount of memory reserved.

For more information about configuring processes, see [Configure Nodes](#).

Adjust the number of other processes

Although the most effective way of improving performance for user traffic is to adjust the number of VizQL server processes, you can also tune other processes that support the VizQL server process or that prevent the VizQL server process from accessing resources.

For example, the VizQL server process makes frequent requests to the cache server process, so you might also want to increase the number of cache server processes. On the other hand, the Backgrounder processes might contend for CPU resources with the VizQL server process. As a result, if you do not need to run frequent extract refreshes, you might reduce the number of processes for the backgrounder. If you do need additional instances of the backgrounder, and if you're running Tableau Server on a cluster, you can move the Backgrounder process to a dedicated node.

Adjust the VizQL session timeout limit

In the example shown previously, the amount of memory used by the VizQL server process increases with user traffic, and it remains reserved by Tableau Server for some time after the traffic finished. This is because the VizQL server process reserves memory for each session for a specified amount of time. If the VizQL server process uses a high percentage of the available memory, try reducing the timeout for each session to make memory available more quickly.

To do this, use the `tsm configuration set` command to reduce the `vizqlserver.session.expiry.timeout` setting. The default is 30 minutes.

Refresh the cache less often

If your users do not always need the most up-to-date data, you can optimize for user traffic by configuring Tableau Server to cache and reuse data as much as possible.

To do this, use the `tsm data-access caching list` command to confirm the refresh frequency. The default is `Low`. Use the `tsm data-access caching set` command to change the refresh frequency.

Assess view responsiveness

When a user opens a view, the components of the view are first retrieved and interpreted, then displayed in the user's web browser. For most views, the display rendering phase occurs in the user's web browser and in most cases, this yields the fastest results and highest level of interactive responsiveness. Handling most interactions in the client web browser reduces bandwidth and eliminates round-trip request latencies. If a view is very

complex, Tableau Server handles the rendering phase on the server instead of in the client web browser, because that generally results in the best performance. If you find that views aren't as responsive as you'd like, you can test and change the threshold that causes views to be rendered by the server instead of in the client web browser. For more information, see [Configure Client-Side Rendering](#).

Configure Client-Side Rendering

When you navigate to a view in Tableau Server, the processing required to display the view, called *rendering*, can be performed by either your client device or Tableau Server. The choice depends on the complexity of the view, which is determined by the number of marks, rows, columns, and more. If a view is less complex, it's faster for a client device to render the view. If a view is more complex, it's faster to send a request to Tableau Server and take advantage of the server's greater computing power.

Note: If a view uses the polygon mark type or the page history feature, server-side rendering is always performed, even if client-side rendering is enabled.

Supported browsers

Client-side rendering is supported in Internet Explorer version 9.0 or higher, Firefox, Chrome, and Safari. All of these web browsers include the HTML 5 `<canvas>` element, which client-side rendering requires.

Client-side rendering is also supported by the Tableau Mobile app.

Configure the complexity threshold for computers and mobile devices

Because computers have more processing power than mobile devices, Tableau Server performs more client-side rendering on computers than on mobile devices.

As a server administrator, you can configure when client-side rendering happens on computers and mobile devices by adjusting the complexity threshold for each. For example, you

might lower the threshold for mobile devices if you notice that views display slowly on them. Or, you might increase the threshold to reduce the number of requests to Tableau Server.

By default, the complexity threshold for computer web browsers is 100. To adjust the complexity threshold for computers, use the following command:

```
tsm configuration set -k vizqlserver.browser.render_threshold -v [new value]
```

By default, the complexity threshold for mobile devices is 60. To adjust the complexity threshold for mobile devices, use the following command:

```
tsm configuration set -k vizqlserver.browser.render_threshold_mobile -v [new value]
```

For example, to change the mobile threshold to 40, you might enter the following command:

```
tsm configuration set -k vizqlserver.browser.render_threshold_mobile -v 40
```

For more information on how to use tsm option set, see [tsm configuration set Options](#).

Disable client-side rendering

Client-side rendering is enabled by default and is recommended to improve the performance of views. However, you might want to disable client-side rendering temporarily for testing or if your server is being accessed primarily by computers or mobile devices with very little processing power.

Use the following command to disable client-side rendering:

```
tsm configuration set -k vizqlserver.browser.render -v false
```

For more information on how to use tsm option set, see [tsm configuration set Options](#).

Tableau Server on Linux Administrator Guide

Testing with the URL Parameter

To test server-side rendering on a session basis, type `? : render=false` at the end of the view's URL. For example:

```
http://localhost/views/Supplies/MyView?:render=false
```

If client-side rendering is disabled on Tableau Server, enter `? : render=true` to enable it for the session:

```
http://localhost/views/Supplies/MyView?:render=true
```

You can also test particular complexity thresholds on individual views to see if it's appropriate to adjust the server-wide threshold for your server and network conditions. For example, you may find that lower complexity (such as 80) or higher complexity (such as 120) tipping points result in more responsiveness to user interactions. To test a threshold, you can keep the server's default configuration (client-side-rendering enabled) and enter the test threshold number at the end of the view's URL. For example:

```
http://localhost/views/Supplies/MyView?:render=80
```

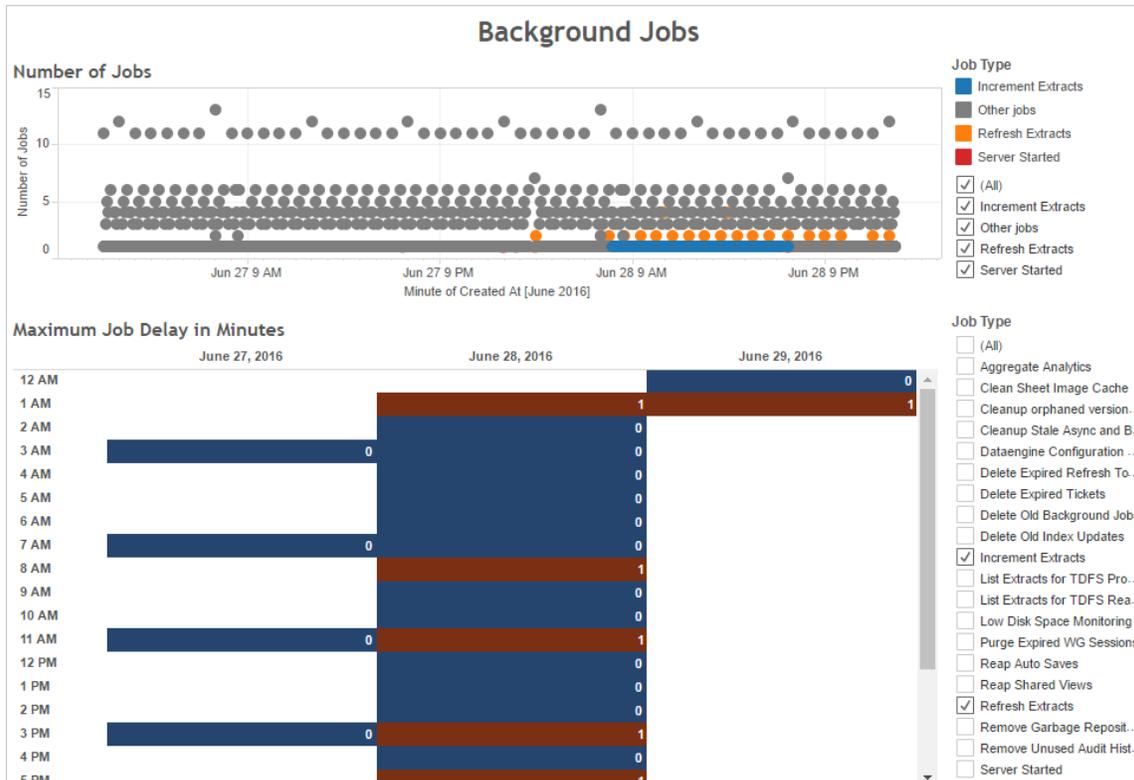
Optimize for Extracts

Try to optimize for extracts if the extract schedules correspond to high resource usage or if extracts take a long time to finish.

When to optimize for extracts

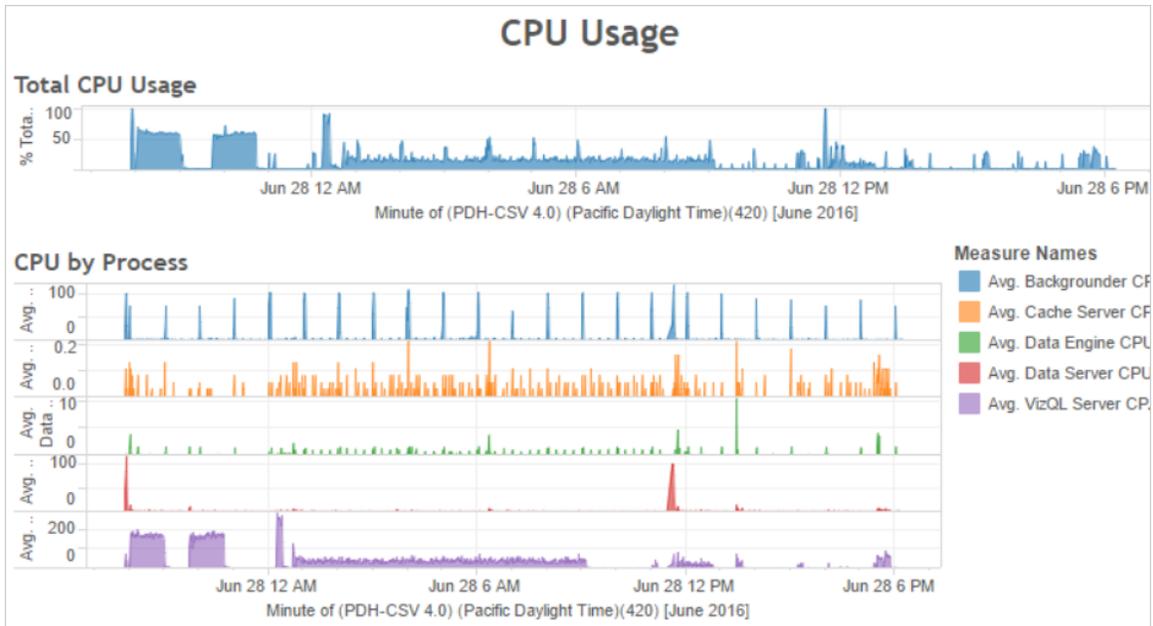
High CPU usage corresponds to extract schedules

Use the **Background Jobs** dashboard of the sample performance workbook to view the number of background jobs run by Tableau Server, including extract refresh jobs. The dashboard also displays how long background jobs are delayed—that is, the amount of time between when a background job is scheduled and when it actually runs. If you see long delays at particular times of the day or if many jobs are running at the same time, try distributing the job schedules across different times of the day to reduce the load on the server.



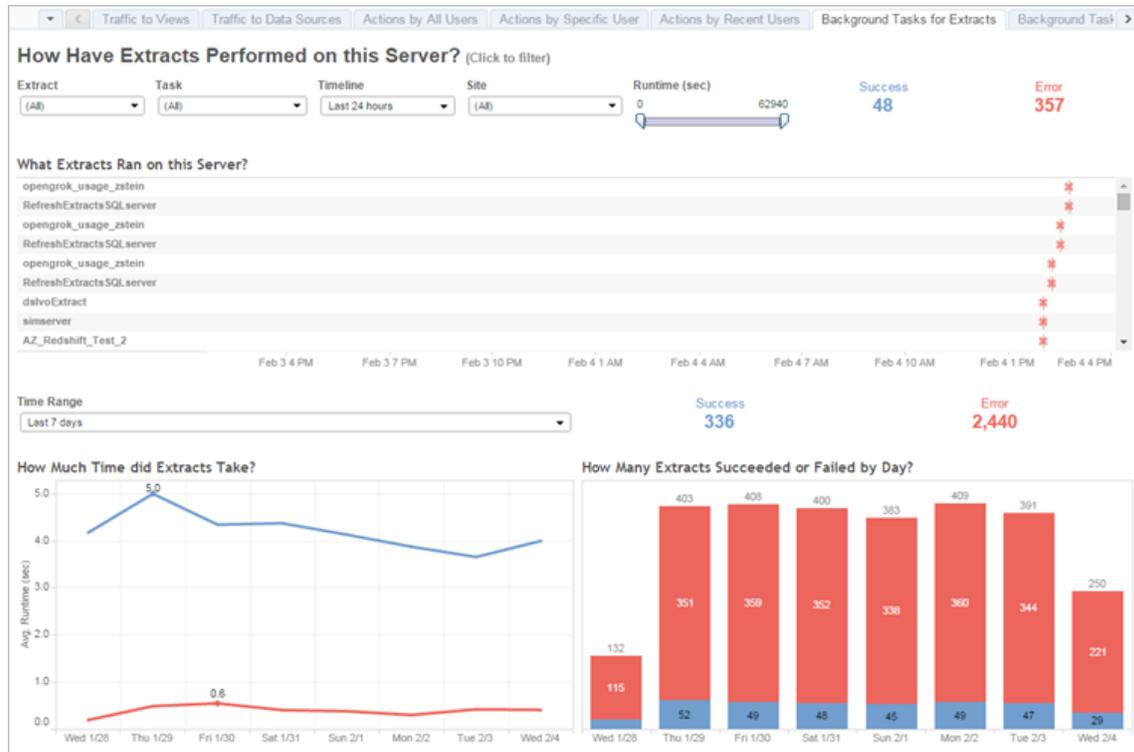
Also compare the times when there are many background jobs or long delays with the CPU usage of the server. Use the **CPU Usage** dashboard to display the percent of total CPU usage and the percent of CPU usage for each process. Because the background process runs background jobs, it is the first process to show strain when there are many extract refresh jobs or when there are slow extract refresh jobs. Note that the CPU usage of the background process periodically but briefly reaches 100 percent. This indicates that there are intensive refresh jobs on a recurring schedule.

Note: The percent of CPU usage for individual processes may add up to more than 100 percent because processor utilization for individual processes is measured for a given processor core. By contrast, the total CPU usage is measured for all processor cores.



Extracts fail or run slowly

Use the Background Tasks for Extracts administrative view to determine how many extracts fail and how long extracts take to complete. Frequent failures can indicate a problem with a particular data source.



Ways to optimize for extracts

When high CPU usage corresponds to extract refresh schedules like it does in the example shown previously, you should optimize for extracts.

Adjust the extract refresh schedule

Use the **Background Jobs** dashboard of the sample performance workbook to identify optimal times for running extracts. In addition to running extracts in off-peak hours, you can distribute extract refreshes to minimize concurrent server load. If extract refreshes continue to cause problems, reduce the frequency of extract refreshes as much as possible in these ways:

- Schedule extracts for times when the server isn't busy.
- Reduce the frequency of refreshes.

Speed up specific extracts

Use the Background Tasks for Extracts administrative view to identify failing extracts and long-running extracts.

- Reduce the size of extracts. You can help improve server performance by keeping the extract's data set short, through filtering or aggregating, and narrow, by hiding unused fields. To make these changes, use the Tableau Desktop options **Hide All Unused Fields** and **Aggregate data for visible dimensions**. For more information, see [Creating an Extract](#) in the Tableau Help.

For general tips on building well-performing workbooks, search for “performance” in the Tableau Help. To see how workbooks perform after they've been published to Tableau Server, you can create a performance recording. For more information, see [Create a Performance Recording](#).

- Use incremental refresh jobs. Incremental refresh jobs append new rows to an existing extract instead of creating the extract from scratch. This type of extract refresh runs quickly because it processes only the data that has been added since the last time the extract refresh job ran. However, it does not account for data that has been updated rather than appended to a data source. As a result, if you run incremental refresh jobs, you should still occasionally run full refresh jobs. For example, you might run a full refresh job once or twice a week for a data source instead of every day.

Configure the execution mode for extract refreshes

When you create extract refresh schedules, ensure that they run in parallel execution mode. When you run a schedule in parallel, it runs on all available background processes, even if the schedule contains only one refresh task. When you run a schedule serially, it only runs on one background process. By default, the execution mode is set to parallel so that refresh tasks finish as quickly as possible.

However, in some circumstances, it can make sense to set the execution mode to serial. For example, you might set the execution mode to serial if a very large job is preventing other schedules from running because it uses all available background processes.

Increase the number of backgrounder processes

A single background process can consume 100 percent of a single CPU core for certain tasks. As a result, the total number of instances you should run depends on the computer's available cores. If you have Tableau Server installed in a cluster and you run backgrounder processes on a separate node, a good rule of thumb is to set the number of backgrounder process to between half the number of cores and the full number of cores of the computer running the backgrounder processes.

For more information about configuring processes, see [Configure Nodes](#).

Isolate processes

If you have Tableau Server installed in a cluster, you see the largest benefit from moving the backgrounder processes to a separate node to avoid resource contention. This is because the backgrounder process is very CPU-intensive and running it on the same node where other CPU-intensive processes are running can slow down the server. For example, both the VizQL server process and the data engine process can be CPU-intensive. Read the two-node configuration in the [Recommended Baseline Configurations](#) topic for more details.

Optimize for Extract Query Heavy Environments

The topic provides guidance on setting up a specific Tableau Server topology and configurations to help optimize and improve performance in an extract query environment.

What is an extract query-heavy environment? Extracts and federated data sources are queried while loading workbooks, views, and dashboards creating a lot of query workload. Therefore, if you have a lot of extracts and federated data sources, you can be said to have a 'extract-query-heavy environment'.

If your environment is extract-query-heavy as defined above, the next couple of sections can help you decide if this configuration is right for you.

When to use this configuration

Key reasoning behind this configuration: Hyper is Tableau's memory-optimized Data Engine technology suited for fast data ingests and analytical processing, making it key to optimizing query-heavy workloads. As your extract use grows, we recommend **configuring Data Engine on dedicated nodes of the Tableau Server cluster**. This configuration allows Tableau Server to scale-out the infrastructure to optimize performance when querying extracts.

There are several factors that affect Tableau Server performance when viewing content using extracts and federated data sources. The goal here is to **achieve consistent and reliable query performance** when viewing content on the Server. Use this configuration if one of the following conditions apply to your environment:

- You are seeing wide variability in workbook load times and the workbook uses extracts or federated data sources.
- Your Tableau Server deployment is growing in the number of Creators, Explorers, Viewers, and extract-based content, so you want to scale out efficiently.
- You are seeing resource contention between Data Engine and VizQL Server when File Store is present on the machine.
- You analyze large amounts of data. This configuration helps in optimizing performance in big data scenarios, in both data ingestion and analysis. To learn more about Tableau and big data, see [Hyper-charge big data analytics using Tableau](#).

Note: Use Server-side performance recording to determine query execution times. To determine resource usage of Tableau use **Performance monitor** for Windows installations, and *sysstat* or *vmstat* tools for Linux installations.

Benefits of using this configuration

These are the key benefits to configuring dedicated nodes for Data Engine:

- Dedicated Data Engine nodes will reduce resource contention between extract queries and other resource-intensive workloads such as those processed by VizQL Server.
- Extract queries are load balanced dynamically on the dedicated nodes, taking into account the current state of the system to ensure that no one node is over or under-utilized.
- More consistent performance in user experience when loading extract-dependent workbooks. The focus here is to establish a consistent and reliable performance rather than making individual queries better.
- You have more control over scaling out Tableau Server processes that need more resources. If VizQL Server, Data Engine, and Backgrounder are all running on the same node and slow extract queries are the problem, it will be difficult to see performance improvements by adding a second node with all three processes. With this configuration, you can add more nodes that will specifically improve extract query workloads.
- Helps improve availability and up-time. In the event of a failure and one of the dedicated Data Engine nodes is unavailable, VizQL Server will attempt to route the pending requests on the problem node to other dedicated Data Engine nodes.
- The Data Engine leverages as many cores as available on the machine. Given this, you have the flexibility to add more resources to the dedicated Data Engine nodes to reduce query response time and variability on expensive extract queries or add more dedicated Data Engine nodes to get more extract query throughput in your Server.

When not to use this configuration

- If you are not experiencing issues with extract-based query load, hardware resources may be better allocated to other portions of Tableau Server.
- On nodes where File Store, Data Engine, and VizQL Server co-exist, you are not seeing resource contention between Data Engine and VizQL Server.
- Before implementing this configuration, it is highly recommended that you evaluate

your CPU usage for VizQL Server and for the node where Data Engine that installed with the File Store.

Configuration

The main goal of this configuration is to have Data Engine on one or more dedicated nodes.

- In deployments where File Store is installed locally, this means configuring File Store on one or more dedicated nodes. Data Engine is automatically installed on the same node as the File Store.
- In deployments where you are configuring External File store, you can still configure Data Engine on dedicated nodes on Tableau Server.

By separating VizQL Server and File Store processes, the load between querying extracts and viewing or interacting with views can be balanced and better managed. This configuration is targeted at consistent performance when querying extracts.

Below is a visual representation of the configuration where the Data Engine/File Store processes have two dedicated nodes, nodes 5 and 6. This is an example where File Store is configured locally which is why the Data Engine and File Store processes are co-located.

The same configuration works for deployments with External File Store, but Node 5 and 6 will have only Data Engine configured in that case.

Additionally, since Node 1 also has the Repository and File Store processes, all of the data needed to perform a backup exist on Node 1 which can improve backup performance.

External Load Balancer						
Process	Node 1 (Initial Node)	Node 2	Node 3	Node 4	Node 5 (DE)	Node 6 (DE)
Cluster Controller	✓	✓	✓	✓	✓	✓
Gateway	✓	✓	✓	✓		
Application Server	✓	✓				
VizQL Server	✓ ✓	✓ ✓				
Cache Server	✓ ✓	✓ ✓				
Search & Browse	✓	✓				
Backgrounder			✓ ✓	✓ ✓		
Data Server	✓ ✓	✓ ✓	✓ ✓	✓ ✓		
Data Engine	✓	✓	✓	✓	✓	✓
File Store	✓				✓	✓
Repository	✓	✗				

Hardware Guidance

To get the most out of this configuration, you will need to experiment with various hardware sizes and configurations to see what best fits your peak load performance objectives. Hyper is a high-performance database technology and the key resources that impact performance are memory, cores, and storage I/O. Understanding how Hyper uses resources to process queries will help you make your hardware selection and understand the reason between different configurations.

- Memory:** When an extract-based query is processed for a user or background process, Tableau Server selects a dedicated Data Engine node to process the query. That dedicated Data Engine node will then copy the extract from local storage, most often the server hard disk, into memory. Having more available system memory allows the operating system better manage memory usage for Tableau. Dedicated Data Engine nodes uses system memory to store the result set of executed queries. If the result set is still valid and the operating system has not cleared it from memory, the result set in memory can be reused.

Tableau Server's minimum hardware recommendation is 32 GB of memory but if you are expecting a high volume of extract-based workbook loads, you should consider

64 GB or 128 GB. If you are hitting other resource limits in addition to memory (like cores), instead of scaling up to 128 GB of memory, it might be better to scale out to an additional 64 GB dedicated Data Engine node.

The process of copying the extract from local storage into memory can take time and optimizing disk performance may be necessary. Optimizing disk performance is covered in the **Storage I/O** section.

- **Cores:** When processing an extract-based query, the number of cores is an important hardware resource that can impact performance and scalability. CPU cores are responsible for executing a query and having more available cores will result in faster execution time. Generally speaking, doubling the number of cores will reduce the query execution time in half. For example, a 10 second query currently utilizing 4 physical cores or 8 vCPUs, will take 5 seconds if you upgrade to 8 physical cores or 16 vCPUs.

The current Tableau Server minimum hardware recommendation is 8 cores, but if your deployment utilizes extracts, consider 16 or 32 core machines. An important thing to note is that if memory and I/O are your bottlenecks, then increasing available cores will not improve your query performance.

- **Storage I/O:** Hyper is designed to leverage the available performance of your extract storage device to speed up query processing. We recommend picking fast disk storage like Solid State Drives (SSD) with high read/write speeds. Currently, SSDs that utilize NVMe storage protocol offers the fastest available speeds.

Note: Sizing resources for dedicated Data Engine nodes only impacts the extract query performance. When loading a workbook, there are many other processes involved that make up total VizQL load request time. The VizQL Server process, for example, is responsible for taking the data from the Data Engine and rendering the visualization.

Other Performance Tuning and Optimizations:

There are additional features you can use to optimize performance beyond the basic configuration described above. The optimizations described below are applicable to both local File Store and External File Store deployments.

- **Extract Query Load Balancing:** To determine where to route the extract query, Data Engine uses a server health metric- the amount of resources Data Engine is consuming and the load from other Tableau processes that may be running on the same node. In addition to evaluating system resources, whether an extract already exists in memory on the node is also taken into account to make sure an extract query is sent to the node that has the most available resources to process the query. This results in more efficient memory and disk utilization and extracts are not duplicated in memory across nodes. See the Extract Query Load Balancing help article for more details.

The extract query load balancing feature is enabled by default in Tableau Server version 2020.2 and later..

- **Workload optimizations using node roles:** With Backgrounder and File Store node roles, server administrators have more flexibility and control over which nodes should be dedicated for running extract queries and extract refreshes. As mentioned in the topology diagram above, certain Data Engine nodes are dedicated to processing extract queries and run only the File Store and Data Engine processes. Node Roles is available with the Server Management Add-on. For more information on node roles, see [Workload Management through Node Roles](#).

The diagram below uses the same topology as the basic configuration described above but with the node roles.

Tableau Server on Linux Administrator Guide

External Load Balancer						
Process	Node 1 (Initial Node)	Node 2	Node 3	Node 4	Node 5 (DE)	Node 6 (DE)
Cluster Controller	✓	✓	✓	✓	✓	✓
Gateway	✓	✓	✓			
Application Server	✓	✓				
VizQL Server	✓ ✓	✓ ✓				
Cache Server	✓ ✓	✓ ✓				
Search & Browse	✓	✓				
Backgrounder			✓ ✓ Extract refresh	✓ ✓ No Extract refresh		
Data Server	✓ ✓	✓ ✓	✓ ✓			
Data Engine	✓	✓	✓	✓	✓	✓
File Store	✓		✓		✓ Extract query	✓ Extract query
Repository	✓	✓				

- Extract Refreshes Backgrounder node role:** By setting Node 3 to extract-refreshes Backgrounder node role, only incremental refreshes, full refreshes, and encryption/decryption jobs will run on this node. By setting Node 4 to no-extract-refreshes Backgrounder node role, all background jobs other than extract refreshes will run on this node. Data Server and Gateway help the extract refresh jobs when using federated and shadow extracts. For more information on Backgrounder node roles, see File Store Node Roles.

Additionally, since Node 1 also has the Repository and File Store processes, all of the data needed to perform a backup exist on Node 1 which can improve backup performance.

The Backgrounder node roles are available with the Server Management Add-on in Tableau Server version 2019.3 and later.

- Extract Queries File Store node role:** Node 5 and 6 which are the **dedicated** Data Engine nodes have the extract-queries File Store node role to ensure they only process queries for viz loads, subscriptions, and data-driven alerts.
- Extract Queries Interactive File Store node role:** For dedicated Data Engine nodes which have extract-queries File Store node role, server administrators can

further isolate the interactive and scheduled workloads to run on specific **dedicated** Data Engine nodes. This is useful for times when there are a lot of users interacting and loading workbooks during high volume subscription times. For example, let's say there are 1000 subscriptions scheduled for the 8 AM Monday mornings. At the same time, many users are also loading dashboards at the beginning of their day. The combined volume of subscription and user queries can result in users experiencing slower, more variable workbook load times. With the extract-queries-interactive File Store node role, you can designate dedicated Data Engine nodes to only accept queries for interactive users (the ones who are looking at their screens waiting). These dedicated Data Engine nodes that are prioritized for interactive workloads would be protected from the high volume of competing subscription jobs and provide more consistent query times. Additionally, Server Admins can use this node role to better plan for growth since they can add dedicated Data Engine nodes for interactive and scheduled workloads independently. For more information, see File Store Node Roles.

The File Store node roles are available with the Server Management Add-on in Tableau Server version 2020.4 and later.

- **Optimizations using External File Store:** This feature allows you to use a network share as the storage for File Store instead of using the local disk on a Tableau Server node. By having the storage on a centralized location, you can significantly reduce the amount of network traffic spent on replicating data between the File Store nodes. For example, in the case when File Store is using a local disk, when a 1 GB extract is refreshed using local File Store, the 1 GB of data is replicated across the network to all nodes that are running the File Store process. In the case where Tableau Server is configured with External File Store, the 1 GB extract only needs to be copied to the network share once and all File Store nodes can access that single copy. The centralization of storage also reduces the total amount of local storage needed on File Store nodes.

Additionally, Tableau Server backups leverage snapshot technology to significantly reduce the time to complete a backup.

While you don't need a dedicated Data Engine node configuration to gain the benefits of External File Store, the additional workload management features with File Store node role and the Extract Query Interactive node role can be used together. See the Tableau Server External File Store topic for more details.

External File Store is available with the Server Management Add-on in Tableau Server version 2020.1 and later.

When to Add Nodes and Reconfigure

Tableau Server can scale up and out as your needs and requirements evolve. Here are some guidelines to help you figure out whether it's time to add more nodes to your system, reconfigure the server, or both:

- **More than 100 concurrent users:** If your deployment is user-intensive (>100 simultaneous viewers), it's important to have enough VizQL processes—but not so many that they exceed your hardware's capacity to handle them. Also, enabling the Tableau Server **Guest User account** can increase the number of potential simultaneous viewers beyond the user list you may think you have. The administrative view can help you gauge this. For more information, see [Actions by Specific User](#).
- **Heavy use of extracts and frequent extract refreshes:** Extracts can consume a lot of memory and CPU resources. There's no one measurement that qualifies a site as extract-intensive. Having just a few, extremely large extracts could put your site in this category, as would having very many small extracts. Sites where extracts are frequently refreshed (for example, several times a day) are often helped by more emphasis on the background process, which handles refresh tasks. Use the [Background Tasks for Extracts](#) administrative view to see your current refresh rate. Extract heavy sites benefit from isolating the Backgrounder process on its own machine. For more information, see the [two-node configuration in the Recommended Baseline Configurations](#) topic.
- **Query heavy environments:** If you are experiencing slow query performance for

Workbooks that use extracts, isolating nodes that handle queries on extracts from VizQL processes can improve and stabilize performance. For more information, see [Optimize for Extract Query Heavy Environments](#).

- **Downtime potential:** If your server system is considered mission critical and requires a high level of availability, you can configure it so there's redundancy for the server processes that handle extracts, the repository, and the gateway.

Performance Tuning Examples

Performance tuning examples have been expanded and moved to separate topics. For more information, see [Performance Tuning](#)

For recommended baseline configurations that serve as starting point for installations with one, two, and three nodes, see [Recommended Baseline Configurations](#).

Performance Recording

This section describes how to create performance recordings and use the results to improve workbook performance. With performance recordings, you can view how long workbook events take. For example, you can see how long it takes to connect to a data source, run a query, render data, and more.

Create a Performance Recording

The Performance Recording feature in Tableau records performance information about key events as you interact with a workbook. You can then view performance metrics in a workbook that Tableau creates to analyze and troubleshoot different events that are known to affect performance:

- Query execution
- Compiling query
- Geocoding

Tableau Server on Linux Administrator Guide

- Connections to data sources
- Layout computations
- Extract generation
- Blending data
- Server rendering (Tableau Server only)

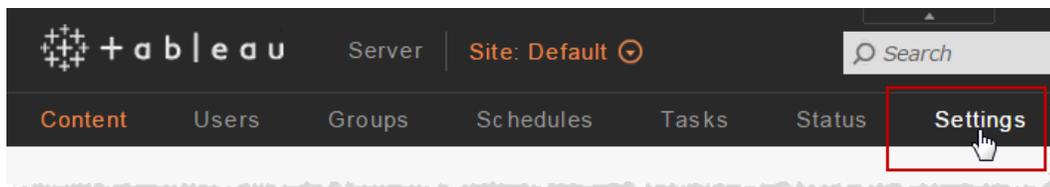
Tableau support may ask that you create a performance workbook as they work with you to diagnose performance issues.

Looking for Tableau Server on Windows? See [Create a Performance Recording](#).

Enable Performance Recording for a Site

By default, performance recording is not enabled for a site. A server administrator can enable performance recording site by site.

1. Navigate to the site for which you want to enable performance recording.
2. Click **Settings**:



3. Under Workbook Performance Metrics, select **Record workbook performance metrics**.
4. Click **Save**.

Start a Performance Recording for a View

1. Open the view for which you want to record performance.

When you open a view, Tableau Server appends ":iid=<n>" after the URL. This is a session ID. For example:

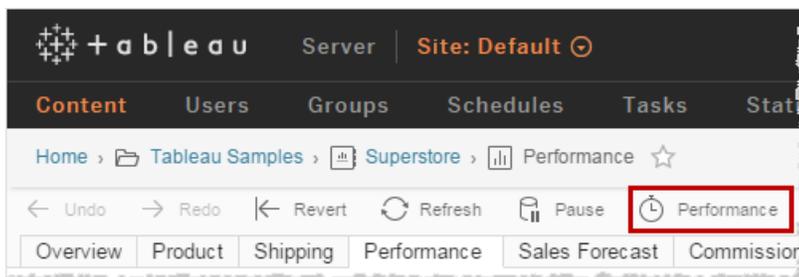
```
http://10.32.139.22/#/views/Coffee_Sales2013/USSalesMar-
ginsByAreaCode?:iid=1
```

2. Type `:record_performance=yes&` at the end of the view URL, immediately before the session ID. For example:

```
http://10.32.139.22/#/views/Coffee_Sales2013/USSalesMar-
ginsByAreaCode?:record_performance=yes&:iid=1
```

3. Click the **Refresh** button in the toolbar.
4. Load the view.

A visual confirmation that performance recording has started is the **Performance** option in the view toolbar:



View a Performance Recording

1. Click **Performance** to open a performance workbook. This is an up-to-the-minute snapshot of performance data. You can continue taking additional snapshots as you continue working with the view; the performance data is cumulative.

2. Move to a different page or remove `:record_performance=yes` from the URL to stop recording.

Interpret a Performance Recording

Create a recording to evaluate the performance of your workbook. Once you have completed the recording, you can download the resulting workbook and open it in Tableau Desktop for analysis.

A performance recording workbook contains two main dashboards: Performance Summary and Detailed Views. The Performance Summary dashboard provides a high-level overview of the most time-consuming events. The Detailed Views dashboard provides a lot more detail and is intended to be used by advanced users when building workbooks. The **Detailed Views** dashboard is only visible when the performance recording workbook is opened using Tableau Desktop.

For information on how to create a performance recording in Tableau Server, see [Create a Performance Recording](#).

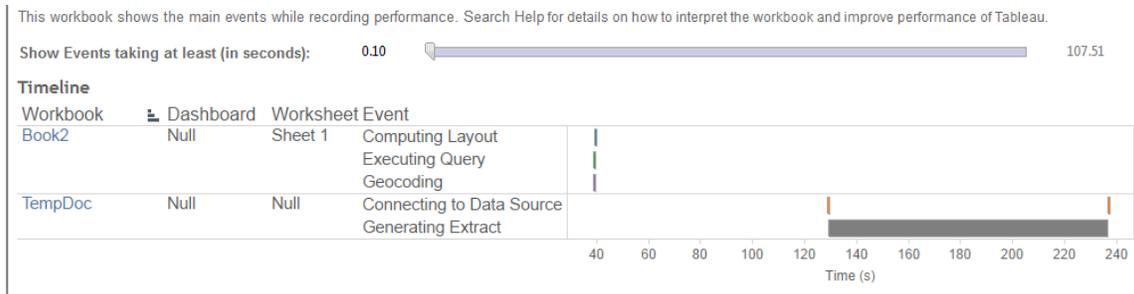
Performance Summary

The **Performance Summary** dashboard contains three views: **Timeline**, **Events**, and **Query**.

Timeline

The uppermost view in the performance summary recording dashboard shows the events that occurred during recording, arranged chronologically from left to right. The bottom axis shows elapsed time since Tableau started, in seconds.

In the Timeline view, the **Workbook**, **Dashboard**, and **Worksheet** columns identify the context for events. The **Event** column identifies the nature of the event, and the final column shows each event's duration and how it compares chronologically to other recorded events:



Events

The middle view in a performance summary dashboard shows the events, sorted by duration (greatest to least). Events with longer durations can help you identify where to look first if you want to speed up your workbook.



Different colors indicate different types of events. The range of events that can be recorded is:

- Computing layouts

If layouts are taking too long, consider simplifying your workbook.

- Connecting to data source

Slow connections could be due to network issues or issues with the database server.

- Compiling query

This event captures the amount of time spent by Tableau in generating the queries.

Long compile query times indicate that the queries generated are complex. The

complexity may be due to too many filters, complex calculations, or generally due to a complex workbook. Examples of complex calculations include, lengthy calculations, LOD calculations, or nested calculations. Try simplifying the workbook, using action filters or moving calculations to the underlying database.

- Executing query
 - For live connections, if queries are taking too long, it could be because the underlying data structure isn't optimized for Tableau. Consult your database server's documentation. As an alternative, consider using an extract to speed performance.
 - For extracts, if queries are taking too long, review your use of filters. If you have a lot of filters, would a context filter make more sense? If you have a dashboard that uses filters, consider using action filters, which can help with performance.

- Generating extract

To speed up extract generation, consider only importing some data from the original data source. For example, you can filter on specific data fields, or create a sample based on a specified number of rows or percentage of the data.

- Geocoding

To speed up geocoding performance, try using less data or filtering out data.

- Blending data

To speed up data blending, try using less data or filtering out data.

- Server rendering

You can speed up server rendering by running additional VizQL Server processes on additional machines.

Query

If you click on an **Executing Query** event in either the **Timeline** or **Events** section of a performance summary dashboard, the text for that query is displayed in the Query section.

If you are connected to a published data source, the query text is displayed in XML. If you are connected to the data source directly, the query is displayed in SQL like shown below:

Query

```
SELECT "State"."ID" AS "ID",
       "StateSynonyms"."Name" AS "State_Name",
       "State"."ParentID" AS "State_ParentID"
FROM "StateSynonyms"
     INNER JOIN "State" ON (("State"."ID" = "StateSynonyms"."ParentID") AND ("State"."MapCode" = "StateSynonyms"."MapCode"
```

If it makes sense, you can use the query text to work with your database team on optimizing at the database level. Sometimes the query is truncated and you'll need to look in the Tableau log to find the full query. Most database servers can give you advice about how to optimize a query by adding indexes or other techniques. See your database server documentation for details.

Sometimes for efficiency, Tableau combines multiple queries into a single query against the data. In this case, you may see an **Executing Query** event for the Null worksheet and zero queries being executed for your named worksheets.

Detailed Timeline

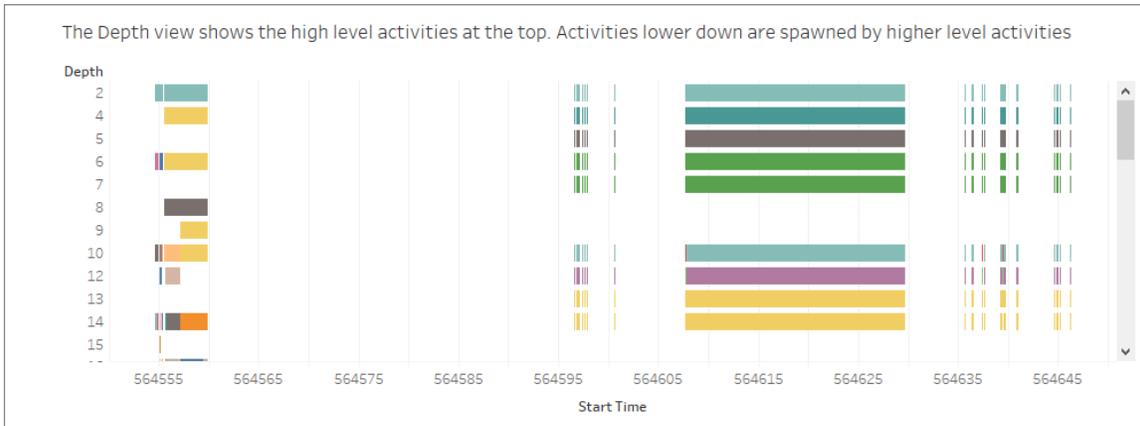
This view is the detailed version of the **Timeline** view that shows all events and separating individual items that were grouped in the **Timeline** view. It is intended to be used by advanced users during workbook designs.

Detailed Views

The **Detailed Views** dashboard contains **Depth**, **Exclusive CPU**, **Inclusive CPU**, and **Elapsed Time** views.

Depth

The **Depth** view is the uppermost view in the **Detailed Views** dashboard and provides insight into what happens when a request is made. This view is the most useful when filtered to a single user request. Examples of user requests are: loading a view, selecting a mark, or changing a filter.



Each bar on the depth view represents a single activity. An activity is a unit of work that is done as part of processing a user request. A single user request results in multiple activities. The length of each bar on the depth view is proportional to the elapsed time for the activity that the bar represents.

High level activities appear at the top of the view. Activities lower down are child activities generated by high level activities.

Hovering over each bar provides additional details about the activity and also highlights the corresponding row in the **CPU** and **Elapsed Time** view described in the next section.

In order to narrow down on what parts of the request took the most time, investigate long running activities at the highest levels.

CPU and Elapsed Time

The **CPU** and **Elapsed Time** views appear lowermost in the **Detailed Views** dashboard. You can toggle between **Exclusive CPU**, **Inclusive CPU** and **Elapsed Time** views by

clicking on the radio buttons.

<p>Select a View</p> <p><input checked="" type="radio"/> Exclusive CPU</p> <p><input type="radio"/> Inclusive CPU</p> <p><input type="radio"/> Elapsed Time</p>	<p>- Exclusive CPU time is useful for identifying activities that consume majority of the CPU</p> <p>- Inclusive CPU time is useful for identifying high level activities that consume majority of the CPU (either themselves or due to activities that they sponsor)</p> <p>- Elapsed time is useful for identifying activities that took the most wall clock time</p>
---	--

While the **Depth** view can help in quick visual identification of long running activities, it may not necessarily highlight activities that happen multiple times with each instance taking a small amount of time. The **Exclusive CPU**, **Inclusive CPU**, and the **Elapsed Time** views provide aggregate statistics for each activity. The number of times an activity took place is shown in the **Count** column and the total amount of time taken by a single activity is shown using the bar chart.

Sometimes for efficiency, Tableau combines multiple queries into a single query against the data. In this case, you may see an **Executing Query** event for the Null worksheet and zero queries being executed for your named worksheets.

Performance Monitoring Tools

This topic describes external resources that you can use to monitor and tune performance.

Tableau Server includes several tools that you can use to monitor server performance and health. For more information about these tools, see [Performance Monitoring Overview](#).

Disclaimer: This topic includes information about third-party and community supported products. Please note that while we make every effort to keep references to third-party and community content accurate, the information we provide here might change without notice. For the most up-to-date information, please consult the documentation for products referenced here. To learn more about community supported tools, see [Support levels for IT and developer tools](#).

- **TabJolt.** A load and performance testing tool that you can use to understand how Tableau Server scales with your workloads, in your environment, and to inform your scalability and capacity needs. Here are some key use cases for when you would use TabJolt:
 - To establish a baseline for server performance and test deployments before pushing them to production environments.
 - In a new Tableau Server, to help understand how the new server scales in your environment, specifically to your hardware and workload?
 - Before upgrading to understand the new version will scale in your environment.
 - To find the best server deployment configuration, given your hardware, workbooks and environments.
- **Replayer.** A tool that can replay log-based real user traffic from a Tableau Server against any other server or configuration. It replays Tableau Server single- or multi-user sessions. Here are some ways that Replay can be used:
 - Playback specific Tableau Server sessions, and filter the session based upon start time or RequestID.
 - Use it to simulate load conditions so that you can test how to scale and balance your Tableau Server installations.
 - Perform regression testing by running and comparing end-to-end user scenarios for Tableau Server upgrades.
 - Capture and report HTTP exceptions that occur in a single-user session.
 - Replay a defect, so that you can troubleshoot and verify that it is fixed.
- **Scout.** An exploratory tool that captures performance metrics across any workbooks on both Tableau Desktop and Tableau Server Here are some ways that Scout can be

used:

- Find slow workbooks on Server.
- Validate performance improvements or regressions after making server configuration or topology changes.
- Validate that workbooks and dashboards are loading properly after upgrading to new Tableau Server.
- Validate that workbooks are still working properly after data source changes.
- [Sitescope](#). An agentless application monitoring tool.
- [Zabbix](#). An open-source, real-time monitoring tool.
- [Splunk](#). A tool for monitoring and analyzing machine data, including logs.
- [Graylog](#). An open-source log management tool.

Configure Client-Side Rendering

When you navigate to a view in Tableau Server, the processing required to display the view, called *rendering*, can be performed by either your client device or Tableau Server. The choice depends on the complexity of the view, which is determined by the number of marks, rows, columns, and more. If a view is less complex, it's faster for a client device to render the view. If a view is more complex, it's faster to send a request to Tableau Server and take advantage of the server's greater computing power.

Note: If a view uses the polygon mark type or the page history feature, server-side rendering is always performed, even if client-side rendering is enabled.

Supported browsers

Client-side rendering is supported in Internet Explorer version 9.0 or higher, Firefox, Chrome, and Safari. All of these web browsers include the HTML 5 `<canvas>` element, which client-side rendering requires.

Client-side rendering is also supported by the Tableau Mobile app.

Configure the complexity threshold for computers and mobile devices

Because computers have more processing power than mobile devices, Tableau Server performs more client-side rendering on computers than on mobile devices.

As a server administrator, you can configure when client-side rendering happens on computers and mobile devices by adjusting the complexity threshold for each. For example, you might lower the threshold for mobile devices if you notice that views display slowly on them. Or, you might increase the threshold to reduce the number of requests to Tableau Server.

By default, the complexity threshold for computer web browsers is 100. To adjust the complexity threshold for computers, use the following command:

```
tsm configuration set -k vizqlserver.browser.render_threshold -v [new value]
```

By default, the complexity threshold for mobile devices is 60. To adjust the complexity threshold for mobile devices, use the following command:

```
tsm configuration set -k vizqlserver.browser.render_threshold_mobile -v [new value]
```

For example, to change the mobile threshold to 40, you might enter the following command:

```
tsm configuration set -k vizqlserver.browser.render_threshold_mobile -v 40
```

For more information on how to use `tsm option set`, see `tsm configuration set Options`.

Disable client-side rendering

Client-side rendering is enabled by default and is recommended to improve the performance of views. However, you might want to disable client-side rendering temporarily for testing or if your server is being accessed primarily by computers or mobile devices with very little processing power.

Use the following command to disable client-side rendering:

```
tsm configuration set -k vizqlserver.browser.render -v false
```

For more information on how to use `tsm option set`, see [tsm configuration set Options](#).

Testing with the URL Parameter

To test server-side rendering on a session basis, type `? : render=false` at the end of the view's URL. For example:

```
http://localhost/views/Supplies/MyView?:render=false
```

If client-side rendering is disabled on Tableau Server, enter `? : render=true` to enable it for the session:

```
http://localhost/views/Supplies/MyView?:render=true
```

You can also test particular complexity thresholds on individual views to see if it's appropriate to adjust the server-wide threshold for your server and network conditions. For example, you may find that lower complexity (such as 80) or higher complexity (such as 120) tipping points result in more responsiveness to user interactions. To test a threshold, you can keep the server's default configuration (client-side-rendering enabled) and enter the test threshold number at the end of the view's URL. For example:

```
http://localhost/views/Supplies/MyView?:render=80
```

Data Acceleration

Administrators can enable data acceleration for specific workbooks to improve their performance.

Loading a published workbook or dashboard is a two-step process:

1. **query**: fetching the data needed after connecting to the underlying data source.
2. **rendering**: performing visual rendering such as layout, drawing shapes, assigning colors, and so on.

The time it takes to load a workbook depends on the combined time it takes to do these two steps. Data acceleration makes the query step faster in many cases. Thus, an accelerated workbook loads faster because Tableau Server precomputes and fetches the workbook's data in a background process. However, not all workbooks will benefit from data acceleration. Data acceleration only improves the performance of the query step, so a workbook may not load observably faster if:

- its query step is already fast and is not a performance bottleneck on load.
- it uses transient functions, such as `now()` or `today()`.
- it relies on user-based functions to provide different results for different users. Even though Tableau Server could precompute results for all users, doing so could significantly increase CPU, memory, and storage usage.

Supported in This Release

To precompute the data, Tableau Server needs to connect to the data source in the background without requiring user interaction. Therefore, data acceleration is only supported for workbooks with embedded connection credentials.

Once enabled, data for workbooks with embedded extracts is automatically accelerated whereas workbooks with published and live data sources need to be added to an acceleration schedule.

Not Supported in This Release

The following are currently not supported:

- Workbooks that use encrypted extracts
- Workbooks that prompt user for credentials
- Workbooks that fetch data from federated data sources.
- Data blending is partially supported but data queried against secondary data sources is not accelerated.

Enabling Workbooks for Data Acceleration

Workbooks are not enabled for acceleration by default.

The easiest way to configure data acceleration is to use the `accelerate_workbooks.py` Python script that you can download from the [Tableau Data Acceleration Client](#) GitHub repository.

Alternatively, you can use the Tableau Server REST API. For more information, see [Data Acceleration](#) in the Tableau Server REST API documentation.

Scheduling Data Acceleration

To keep the data of accelerated workbooks that use published or live data sources fresh, Tableau Server needs to run the background precomputation periodically. This is achieved by defining a schedule for the precomputation (similar to scheduling extract refreshes). A schedule of type `DataAcceleration` should be used for this purpose.

User Context for Precomputation

Precomputation for accelerated workbooks is performed with the user context of only one user. This is either:

- the owner of the workbook (if there are no user filters in the workbook or data source or if there are user filters on the data source but the data source is a published data source).
- or-

- the user that was selected for thumbnail generation the last time workbook was published (if there are user filters on the workbook and the data source is not a published data source).

Cost of Data Acceleration

Enabling this feature will increase the computation load and number of jobs on Tableau Server background processes because data acceleration fetches the required data from data sources in a background process. A background job to precompute the data of an enabled workbook is run if any of the following happen:

- The workbook is republished (this includes the web-authoring save).
- An extract used by the workbook is refreshed.
- A `DataAcceleration` scheduled task associated with the workbook runs.

Administrators should consider those costs before enabling Data Acceleration for many workbooks, or scheduling acceleration jobs too frequently.

- Workbooks that are being heavily edited and republished may not be suitable because each republish triggers a precomputation. We recommend acceleration for workbooks that are published for consumption.
- If a workbook uses multiple extracts then their refresh will trigger precomputation of the data. Thus, frequent extract refreshes for enabled workbooks could cause a spike in background job load, especially given that by design data acceleration jobs are run after the successful extract refresh.
- The precomputed data for workbooks using embedded extracts is stored on disk, and the data for published and live data sources is kept in the cache server. Thus, acceleration may increase the disk storage or cache memory usage of the server.

Note: When using this feature, it is recommended to increase the size of the Tableau Server external cache to 2 GB or larger. To view your current Tableau Server external cache size setting, type: `tsm configuration get -k redis.max_memory_in_mb`.

To set the Tableau Server external cache size to 2 GB, type: `tsm configuration`

```
set -k redis.max_memory_in_mb -v 2048 and then type tsm pending-
changes apply. For more information, see tsm configuration set Options.
```

Extract Query Load Balancing

In Tableau Server version 2020.2 and later, load balancing for extract-based queries has improved and may result in faster load times for extract-based dashboards. Large deployments that are extract heavy and have a high volume of dashboard loads will probably see the most improvement, particularly if they have Hyper running on standalone nodes. Smaller deployments where Hyper is running on nodes that are shared with other server processes may also see performance improvements.

Hyper logs a server health metric about the amount of resources Hyper is consuming and also takes into account load from other Tableau processes that may be running on the same server node. Based on this information, extract queries will be sent to a node that has available resources to process the query. In addition to evaluating system resources, the load balancer improves the chance that the node you are routed to has the extract already cached. This reduces the number of duplicate extracts across nodes and improves memory and I/O usage.

To use this feature, the Cache Server process must be active. For more information, see [Tableau Server Cache Server](#). If Cache Server process is not enabled, load balancing will automatically revert to the previous functionality.

This feature is turned on by default. To disable it, use the following tsm commands:

```
tsm configuration set -k hyper_standalone_con-
sistent.hashing.enabled -v false
```

```
tsm configuration set -k hyper_standalone.health.enabled -v
false
```

Apply the changes using the following tsm command: `tsm pending-changes apply`

For more information, see `tsm configuration set Options`.

Monitoring Tableau Server

You can configure SMTP and alerts and subscriptions to aid in monitoring Tableau Server.

Configure SMTP Setup

Tableau Server can email server administrators about system failures, and email server users about subscribed views and data-driven alerts. First, however, you need to configure the SMTP server that Tableau Server uses to send email. After configuring SMTP, complete the steps to configure notifications (Configure Server Event Notification), then when you start or restart the server, it will trigger an email notification, which confirms that you have set up notifications correctly.

Configuring SMTP requires that you restart Tableau Server services.

Secure SMTP

To enable and configure TLS for SMTP, you must use the TSM CLI as described in this topic. Tableau Server only supports STARTTLS (Opportunistic or Explicit TLS).

If your organization does not use public certificates for verifying TLS connections, then you can upload a private certificate to Tableau Server to verify trusted connections. For more information, see the `tsm security custom-cert add` command.

You may also configure SMTP TLS for encryption-only by disabling the certificate validation process. For more information, see the section, *Configuration file reference*, in the *Use the TSM CLI* tab below.

Use the TSM web interface

1. Open TSM in a browser:

`https://<tsm-computer-name>:8850`. For more information, see [Sign in to Tableau Services Manager Web UI](#).

2. Click **Notifications** on the **Configuration** tab and click **Email Server**.
3. Enter the SMTP configuration information for your organization:

The screenshot shows the Tableau Server Configuration page for Email Server settings. The page has a dark blue header with tabs for STATUS, MAINTENANCE, and CONFIGURATION. Below the header, there's a section for Notifications, followed by an Email Server tab. The main content area is titled 'Configure email server' and contains several input fields with labels and placeholder text:

- SMTP server address:** smtp.example.lan
- Username:** tableau-notify@example.lan
- Password:** [Redacted]
- Port Number:** 25 (Default)
- Send all emails from:** no-reply@example.lan (with a note: 'Type an email address that all emails will be sent from (example: no-reply@example.com)')
- Send server health email to:** tableau-health@example.lan (with a note: 'Type email addresses, separated by a comma, that will receive Tableau Server health emails. Tableau Server health emails are typically sent to server administrators or other IT admins.'))
- Tableau Server URL:** https://tableau.example.lan (with a note: 'Choose a footer link to embed in all email alerts and subscriptions. This link is typically the sign-in page of Tableau Server.'))

At the bottom of the form, there are two buttons: 'Cancel' and 'Save Pending Changes'.

4. Click **Save Pending Changes** after you've entered your configuration information.
5. Click **Pending Changes** at the top of the page:



6. Click **Apply Changes and Restart**.
7. Run the `tsm email test-smtp-connection` to view and verify the connection configuration. See [tsm email test-smtp-connection](#).

Use the TSM CLI

For the initial configuration of SMTP, we recommend that you use the configuration file template below to create a json file. You can also set any single configuration key listed below with the syntax described in `tsm configuration set`.

1. Copy the following json template to a file.

Important: The template below includes common options for most deployments. After you copy the template to a text file, you must edit the option values for your SMTP server requirements. You may need to remove or add options. See the reference section that follows for more information about all supported SMTP key options.

```
{
  "configKeys": {
    "svcmonitor.notification.smtp.server": "SMTP server host
name",
    "svcmonitor.notification.smtp.send_account": "SMTP user
name",
    "svcmonitor.notification.smtp.port": 443,
    "svcmonitor.notification.smtp.password": "SMTP user
account password",
    "svcmonitor.notification.smtp.ssl_enabled": true,
    "svcmonitor.notification.smtp.from_address": "From email
address",
    "svcmonitor.notification.smtp.target_addresses": "To email
address1,address2",
    "svcmonitor.notification.smtp.canonical_url": "Tableau
Server URL"
  }
}
```

2. Run the `tsm settings import -f file.json` to pass the json file with the

appropriate values to Tableau Services Manager to configure Tableau Server for SMTP. Tableau Services Manager will validate the entity values.

3. Run the `tsm pending-changes apply` command to apply the changes. See `tsm pending-changes apply`.
4. Run the `tsm email test-smtp-connection` to view and verify the connection configuration. See `tsm email test-smtp-connection`.

SMTP CLI configuration reference

This table lists all of the options that can be used to configure SMTP with TSM CLI.

Option	Description
<code>svc-mon-itor.notification.smtp.server</code>	Address of SMTP server. Example: "svc-mon-itor.notification.smtp.server": "mail.example.com"
<code>svc-mon-itor.notification.smtp.send_account</code>	User name for SMTP account.
<code>svc-mon-itor.notification.smtp.port</code>	Port number for SMTP server. The default is 25.
<code>svc-</code>	Password for SMTP server account.

Option	Description
<code>mon-itor.notification.smtp.password</code>	<p>Example:</p> <pre>"svc-monitor.notification.smtp.password": "password"</pre>
<code>svc-monitor.notification.smtp.ssl_enabled</code>	<p>Specifies whether the connection to the SMTP server is encrypted. The default is false.</p>
<code>svc-monitor.notification.smtp.ssl_required</code>	<p>If enabled, Tableau Server will refuse to connect to SMTP servers without using TLS. The <code>svc-monitor.notification.smtp.ssl_enabled</code> option must also be set to true.</p> <p>The default is false.</p>
<code>svc-monitor.notification.smtp.ssl_check_server_identity</code>	<p>If set to true, Tableau Server will check the SMTP server identity as specified by RFC 2595. These additional checks based on the content of the server's certificate are intended to prevent man-in-the-middle attacks.</p> <p>The default is false.</p>
<code>svc-monitor.notification.smtp.ssl_trust_</code>	<p>When using TLS, trust certificates from all mail servers, ignoring the validity of the certificate's chain of trust. By setting this key to true, TLS will be used only to encrypt the traffic to the</p>

Option	Description
<code>all_hosts</code>	<p>SMTP host.</p> <p>The default is false.</p>
<code>svc- mon- itor.no- tification.smtp.ssl_ ciphers</code>	<p>The default and supported sets of cipher suites is defined by the version of JDK that is installed with Tableau Server. See the section below, TLS ciphers, for a list of supported and default ciphers.</p> <p>To update the cipher suites used by Tableau Server for SMTP TLS connections, enter a white space-separated list of cipher suites for this value. For example, "TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256 TLS_DHE_DSS_WITH_AES_128_GCM_SHA256 TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384".</p>
<code>svc- mon- itor.no- tification.smtp.ssl_ versions</code>	<p>The default TLS versions enabled on this version of Tableau Server are TLSv1, TLSv1.1, TLSv1.2 and TLSv1.3.</p> <p>TLS version support is defined by the version of JDK that is installed with Tableau Server.</p> <p>Supported versions of TLS are SSLv2Hello, SSLv3, TLSv1, TLSv1.1, TLSv1.2, TLSv1.3.</p> <p>To update the versions used by Tableau Server for SMTP TLS connections, enter a white</p>

Option	Description
	<p>space-separated list of versions for this value. For example, "TLSv1.2 TLSv1.3".</p>
<pre>svc- mon- itor.no- tification.smtp.from_ address</pre>	<p>Email address that will send an notification if there's a system failure. The email address must have valid syntax (for example, ITalerts@bigco.-com or noreply@mycompany), but it does not have to be an actual email account on Tableau Server. (Some SMTP servers may require an actual email account, however.)</p> <div data-bbox="751 737 1349 957" style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;"> <p>Note: You can override the system-wide email address on a per-site basis. For more information, see What is a site.</p> </div> <p>Example:</p> <pre>"svc- monitor.notification.smtp.from_ address": "donot- reply@example.com"</pre>
<pre>svc- mon- itor.no- tification.smtp.target_ addresses</pre>	<p>Email address to receive notifications. If email notifications are enabled, you need to include at least one address. Separate multiple addresses with commas.</p> <p>Example:</p> <pre>"svc- monitor.notification.smtp.target_ addresses":</pre>

Option	Description
	"iluvdata@example.com"
svc- mon- itor.no- tification.smtp.canonical_ url	<p>URL of the Tableau Server. Enter <code>http://</code> or <code>https://</code>, followed by the name or IP address of the Tableau server. Used in the footer of subscription email.</p> <p>Example:</p> <pre>"svc- mon- itor.notification.smtp.canonical_ url": "http://myserver.example.com"</pre>

TLS ciphers

The following is a list of TLS ciphers that are supported by the JDK that is included with Tableau Server. In this version of Tableau Server, all of these ciphers are enabled by default. You can specify a custom list of ciphers for your SMTP configuration by entering a white-space separated list with the option, `svcmonitor.notification.smtp.ssl_ciphers`, as described in the table above.

TLS_RSA_WITH_AES_128_CBC_SHA256	TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384
TLS_DHE_DSS_WITH_AES_256_GCM_SHA384	TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256
TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA	TLS_RSA_WITH_AES_256_CBC_SHA256

TLS_RSA_WITH_AES_128_GCM_SHA256	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
TLS_DHE_DSS_WITH_AES_128_CBC_SHA256	TLS_DHE_DSS_WITH_AES_256_CBC_SHA
TLS_DHE_RSA_WITH_AES_128_CBC_SHA256	TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
TLS_RSA_WITH_AES_256_GCM_SHA384	TLS_EMPTY_RENEGOTIATION_INFO_SCSV
TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	TLS_DHE_DSS_WITH_AES_256_CBC_SHA256
TLS_RSA_WITH_AES_256_CBC_SHA	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256	TLS_DHE_RSA_WITH_AES_256_CBC_SHA
TLS_DHE_DSS_WITH_AES_128_GCM_SHA256	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	TLS_ECDH_RSA_WITH_AES_128_CBC_SHA
TLS_DHE_RSA_WITH_AES_256_GCM_SHA384	TLS_DHE_RSA_WITH_AES_128_CBC_SHA

TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256	TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256
TLS_ECDH_RSA_WITH_AES_256_CBC_SHA	TLS_DHE_DSS_WITH_AES_128_CBC_SHA
TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
TLS_RSA_WITH_AES_128_CBC_SHA	TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	TLS_AES_256_GCM_SHA384
TLS_AES_128_GCM_SHA256	

Configure Server Event Notification

A Tableau Services Manager (TSM) administrator can configure Tableau Server to allow notifications for the following events:

- Content updates
 - Extract failures (enabled by default)
 - Subscription views for users (disabled by default)
- Server health monitoring
 - Server status changes (disabled by default)
 - Desktop License reporting (disabled by default)
- Drive space
 - Email alerts when disk space crosses or remains below pre-configured thresholds (disabled by default)
 - Recording usage history (enabled by default)

Note: You need to configure SMTP before you can configure subscriptions or notifications. For more information, see [Configure SMTP Setup](#).

Use the TSM web interface

1. Open TSM in a browser:

```
https://<tsm-computer-name>:8850.
```

For more information, see [Sign in to Tableau Services Manager Web UI](#).

2. Click **Notifications** on the **Configuration** tab and click **Events**.
3. Configure notification settings for your organization:

- Content updates

- **Send emails for extract refresh failures**

When this option is enabled (the default), a server administrator can configure email notifications to be sent when extract refreshes fail. These messages are configured at the site level, so even if this option is enabled, messages are not sent unless the **Send email to data source and workbook owners when scheduled refreshes fail** option is enabled for a site (this is enabled by default). For details, see [Enable Extract Refresh Scheduling and Failure Notification](#).

- **Allow users to receive email for views that they have subscribed to**

When this option is enabled (by default is it disabled), a server administrator can configure a site to send subscription email. These email messages are configured at the site level and can only be configured when this option is enabled. For details, see [Set Up a Site for Subscriptions](#).

When users subscribe to a workbook or view, a snapshot of the view is emailed to them on a scheduled basis, so they can see the latest updates without having to sign into Tableau Server.

To allow users to attach PDF renderings on subscription emails, select **Let users add attachments to subscribed views.**

- Server health monitoring
 - **Send emails for Tableau Server process events (up, down, and failover)**

Tableau Server sends an email message when the data engine, repository, or gateway server processes stop or restart, or when the initial Tableau Server node stops or restarts.

If you are running a single-server installation (all processes on the same computer), health alerts are only sent when Tableau Server is up. No "down" alerts are sent. If you are running a distributed installation that's configured for failover, a DOWN alert means that the active repository or a data engine instance has failed and the subsequent UP alert means that the passive instance (repository) or second instance (data engine) of that process has taken over.

Note: Tableau Server is designed to be self-correcting. If a service or process stops responding or goes down, Tableau Server attempts to restart it. This can take 15 to 30 minutes to complete. Because of this, reacting immediately to service or process alerts can be counter-productive, especially in an installation with redundant services that can handle requests while one restarts.

- **Enable Tableau Desktop License reporting**

License reporting data originates in Tableau Desktop and is sent to Tableau Server. When this option is enabled, Tableau Server will generate and display the administrative report for Desktop License reporting. For information on the report, see Desktop License Usage.

- Drive space

Enable notifications (alerts) for remaining disk space on your Tableau Server.

- **Send emails when unused drive space drops below thresholds**

You can configure Tableau Server to send email notifications when disk space usage on any node crosses a threshold, or remains below the threshold. And you can configure how often threshold notifications are sent.

There are two thresholds you must set, **Warning threshold** and **Critical threshold**. Thresholds are expressed in percentage of disk space remaining. The critical threshold must be less than the warning threshold.

You also specify the **Send threshold alert every** option. This determines how often, in minutes, warning and critical notifications should be sent. The default value is 60 minutes.

- **Record disk space usage information and threshold violations for use in custom administrative views**

When you configure Tableau Server to record disk space usage, information about free disk space is saved in the repository and you can view the usage history using the Administrative Views.

4. Click **Save Pending Changes** after you've entered your configuration information.

5. Click **Pending Changes** at the top of the page:



6. Click **Apply Changes and Restart**.

Use the TSM CLI

The various notification values described above can be set individually with the `tsm configuration set` command. Alternatively, you can construct a json file and pass all configuration values in one operation. Both methods are described in this section.

Set notification values individually

The following table shows the key/value pairs that map to the notification events described earlier in this topic. Use the `tsm configuration set` command with the following syntax to set a single key/value pair:

```
tsm configuration set -k <config.key> -v <config_value>
```

For example, to enable job failure notifications, run the following command:

```
tsm configuration set -k backgrounder.notifications_enabled -v true
```

Notification option	Key	Value
Extract failures or Flow run failures	<code>backgrounder.notifications_enabled</code>	<code>true false</code>
Enable subscription views for user	<code>subscriptions.enabled</code>	<code>true false</code>

Tableau Server on Linux Administrator Guide

Enable PDF attachments for sub-scriptions	<code>subscriptions.attachments_enabled</code>	<code>true false</code>
Maximum attachment size (MB) for subscription notifications	<code>subscriptions.max_attachment_size_megabytes</code>	integer value, default is 150
Server status changes	<code>svcmonitor.notification.smtp.enabled</code>	<code>true false</code>
License reporting	<code>features.DesktopReporting</code>	<code>true false</code>
Remaining space thresholds: enable email notifications	<code>storage.monitoring.email_enabled</code>	<code>true false</code>
Remaining space thresholds: warning percentage	<code>storage.monitoring.warning_percent</code>	integer value, for example, 20
Remaining space thresholds: critical percentage	<code>storage.monitoring.critical_percent</code>	integer value, for example, 15
Set email interval	<code>storage.monitoring.email_interval_min</code>	integer value, in minutes, for example, 25

Record usage history	storage.monitoring.record_history_enabled	true false
----------------------	---	--------------

After you are done setting values, you must run the following command:

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

Set all notification values with a single json file

To make all notifications settings with a single configuration, you can pass a json file.

Copy and edit the following template to create a file for your configuration.

```
{
  "configKeys": {
    "backgrounder.notifications_enabled": true,
    "subscriptions.enabled": true,
    "subscriptions.attachments_enabled": true,
    "subscriptions.max_attachment_size_megabytes": 150,
    "svcmonitor.notification.smtp.enabled": true,
    "features.DesktopReporting": true,
    "storage.monitoring.email_enabled": true,
    "storage.monitoring.warning_percent": 20,
    "storage.monitoring.critical_percent": 15,
    "storage.monitoring.email_interval_min": 25,
    "storage.monitoring.record_history_enabled": true
  }
}
```

After you have saved the file, pass it with the following command:

```
tsm settings import -f <path-to-file.json>
```

To apply changes, run the following command:

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

Maintenance

You should perform regular maintenance on Tableau Server, such as creating backups, synchronizing Active Directory groups, and removing unneeded files.

Backup and Restore

A Tableau Server administrator should perform regular database maintenance, monitor disk usage on the server, clean up unnecessary files to free up space on the server, and back up Tableau Server and its data. Taking these steps can help ensure that Tableau Server runs with maximum efficiency.

You can use the Tableau Services Manager (TSM) command line tool to back up and restore your Tableau data. Tableau data includes data extract files, as well as Tableau Server's own PostgreSQL database, which stores workbook and user metadata, and server configuration data. Tableau Server log files capture activity and can help you diagnose problems. Logs are written to folders on the server and you can archive and remove them to save

disk space.

Note: You can use the `tsm maintenance restore` command to restore Tableau Server backups created using `tabadmin backup` and `tsm maintenance backup`. Database backups made in other ways, and virtual machine snapshots are not valid sources for restoring Tableau Server.

You can only restore from a backup that has the same type of identity store as the running server. For example, a backup from a server using local authentication can be restored to a Tableau Server initialized with local authentication, but a backup from a server using Active Directory authentication cannot be restored to a server initialized with local authentication.

Platform compatibility

You can use a backup created with Tableau Server on Linux to restore Tableau Server on Windows version 2018.2 and later.

You cannot use a backup created with Tableau Server on Linux to restore earlier versions of Tableau Server on Windows (version 2018.1 and earlier).

You can use a backup created with Tableau Server on Windows (version 2018.2 and earlier) to restore Tableau Server on Linux. For more information about restoring a Windows backup on Tableau Server on Linux, see [Migrate Tableau Server from Windows to Linux](#).

Disk Space Usage for Backup and Restore

The free disk space required to create a backup varies depending on the amount of data in the Tableau Server repository and file store services, and their collocation with the `tabadmincontroller` service. During backups, the background tasks for cleaning up old extracts are temporarily paused. This means that, for the duration of the backup, extract refreshes will leave extra files in place, adding to disk space usage. If your backup takes a long time, or if your organization uses many extracts that are regularly updated, this can result in a

significant amount of temporary disk space usage. These temporary files will be removed after the backup is complete.

The following table lists the disk space requirements for backup based on whether the node hosts the repository, file store, controller, or some combination of them.

Repository	File Store	Controller	Disk Space Required
✔			<p>3x repository data + 250 MB</p> <p>To obtain an estimate of the repository data, check the size of <code><data directory>/pgsql/data/base</code> directory.</p> <p>To obtain the exact size of the repository data, open the backup file and use the size of the <code>workgroup.pg_dump</code> file.</p>
	✔		<p>1.5x file store data</p> <p>To obtain an estimate of file store data (extracts, flows, etc.), check the size of <code><data directory>/dataengine</code> directory.</p>
		✔	<p>3x repository data + 250 MB + 2.5x file store data</p>
✔	✔		<p>3x repository data + 250 MB + 1.5x file store data</p>
	✔	✔	<p>3x repository data + 250 MB + 1.5x file store data</p>
✔		✔	<p>3x repository data + 250 MB + 2.5x</p>

			file store data
✓	✓	✓	3x repository data + 250 MB + 1.5x file store data

Restore disk space requirements

You must have adequate disk space for the database restore process to run successfully.

To restore Tableau Server:

- On controller nodes, you need free space equal to at least the size of the backup archive.
- On repository nodes, you need free space equal to at least three times the size of the repository data in the backup archive, plus 250 MB, plus the size of the pgsql data directory.
- On file store nodes, you need free space equal to at least twice the size of the dataengine folder in the backup archive.

Best Practices for Backing Up Tableau Server

We recommend following these security and performance best practices.

Protect backup file

While configuration secrets are encrypted when stored on disk internally, when these configurations are exported to a backup file, some secrets are written into the file in plain text. It is up to the administrator to take measures to protect the backup file. There are a variety of options available:

- Write the file to an encrypted file system.
- Write the file to a disk that is physically protected and restricted to specific users.
- Encrypt the backup file.

Maximize backup efficiency

There are several ways you can maximize backup efficiency. Your environment can impact how effective each of these is, so test with your data to see what works best.

Optimizing with topology configurations:

- Co-locating File Store on the same node as the Administration Controller can reduce the length of time it takes to back up Tableau Server by reducing or eliminating the need to transfer data between nodes during the backup process. This is especially true if your organization uses many extracts.
- Co-locating the repository (pgsql) with the Administration Controller node can also help to reduce back up time, but the time savings is less significant than that of the File Store.

The Administration Controller is usually on the initial node, unless you have had an initial node failure and moved the controller to another node.

Optimizing with backup strategies:

Backup is a resource intensive process. If possible, doing your backups during off peak hours is a generally a good strategy. But this however, depends on your requirements and how often Tableau Server data is updated and what your restore requirements. For a detailed explanation of backup and disaster recovery, see [Tableau Server Disaster Recovery](#). Here are some backup strategies and adopt them to your requirements

- **Type of storage:** Solid State disks are recommended in general for backups. SSD helps make your backups faster and complete sooner compared to traditional spinning disks.
- **Backup compression:** You have the option of running your backups with or without compression. When you do your backup with compressions, your backup size will be comparatively smaller, but you may see a slower performance. So if your goal is more focused on speed, choose the `--skip-compression` option:

Use the `--skip-compression` option when backing up Tableau Server. This creates the backup without using compression, and results in a larger backup file but can reduce the amount of time it takes for the backup to complete. For more information, see `tsm maintenance backup`.

- **Snapshot backup:** This option is only available if you have configured your Tableau Server with External File Store. Although the performance of snapshot backups depend on the type of network attached storage, in general snapshot backups are faster than the traditional Tableau Server backups. For more information see, [Tableau Server External File Store](#).

Perform a Full Backup and Restore of Tableau Server

You can use the following steps to back up your Tableau Server deployment. Specifically, these steps describe how to recover a clone of a server from a collection of backup data and assets.

Note: The backup process can take a long time to run. Since no other jobs can be run while backup is running, we recommend that you run backup during non-business hours.

Backup data types

There are two types of backup data that Tableau Server can generate. We recommend performing regular backups of each type in case you must restore a server in a recovery scenario:

- **Data managed by Tableau Server:** consists of the Tableau PostgreSQL database or repository and File Store, which contains workbook and user metadata, data extract files, and site configuration data. When you use TSM to create a backup, all of this data is saved in a single file with a `.tsbak` extension. This data is backed up with the `tsm maintenance backup` command.

Note: When File Store is configured external to Tableau Server you cannot use the `tsm maintenance backup` command to backup Tableau Server Data. For more information on how to backup this data, see [Backup and Restore with External File Store](#).

You can only restore from a backup that has the same type of identity store as the running server. For example, a backup from a server using local authentication can be restored to a Tableau Server initialized with local authentication, but a backup from a server using Active Directory authentication cannot be restored to a server initialized with local authentication.

- You can only restore a backup file to a version of Tableau Server version that is the same or newer than the version the backup was created on. You cannot restore to an older version of Tableau.
- You can only restore a backup file to a version of Tableau Server version that is the same or newer than the version the backup was created on. You cannot restore to an older version of Tableau.
- Configuration and Topology data: includes most of the server configuration information required to fully recover a server. SMTP, alerting, some authentication assets, are all examples of configuration data that are exportable for backup. Topology data defines how your Tableau Server processes are configured in both single-server and multiple node deployments. Configuration and topology data is backed up with the `tsm settings export` command.

Note: You can change the file path used by the `tsm maintenance backup` command from the default value. For more information, see [tsm File Paths](#).

Backup assets that require a manual process

Some configuration data is not included in the `tsm settings export` command and must therefore be documented and restored manually. The following configuration data is excluded from the `tsm settings export` operation. Your backup maintenance process should include documenting the following Tableau Server configuration data:

- **System user accounts.** Tableau Server setup creates an unprivileged user account, `tableau`. This account is used to access Tableau Server resources. This account can be changed during setup. If you have not changed this account, then you do not need to document it.
- **TSM group membership.** There are two groups created by Tableau Server: `tableau` and `tmsadmin`. If you configured alternative groups when you installed Tableau Server, then you'll need to document the group names.

In all cases you should document the user accounts that are in these groups. To view membership in a group, run the following command `grep <group_name> /etc/group`.

- **Coordination Service deployment configuration.** If you are running a multinode cluster, document which nodes are running the Coordination Services process. To view process configuration on your nodes, run `tsm topology list-nodes -v`.
- **Customization settings.** If your organization uses custom header or sign-in logos for Tableau Server web pages, you should include a copy of those assets with your back up portfolio. See `tsm customize`.
- **Most authentication assets.** Most certificate files, key files, keytab files or other authentication-related assets are not backed up by TSM. There are three exceptions:
 - The public certificate and private key for the internal PostgreSQL database (if enabled) are backed up.
 - The certificate and key for external SSL are backed up and included in the configuration data.
 - The custom certificate installed by `tsm security custom-cert add` (if added) is backed up.

However, all other authentication-related assets are not backed up. For example, if you have enabled access to the PostgreSQL database with the `tsm data-access repository-access enable` command, be sure to document the name/password pairs

for each account you've configured. These credentials are not backed up. The certificate and key for mutual SSL are not included in the back up.

- LDAP assets. Keytab files, configuration files, and or other LDAP-related assets are not backed up by TSM.

Internal server secrets and repository passwords are crypto-related configurations that are not exported. However, you do not need to document configuration values. New secrets will be created as part of the restoration process when you initialize the new instance.

Backing up Tableau Server for recovery

Tableau Server includes commands that you run to generate backup data for Tableau Server.

Note: When backing up Tableau Server on Linux, the unprivileged user must have write access to the network share where the backup files are written. Otherwise, backup will fail.

To back up server topology and configuration data, use the `tsm settings` command.

1. Topology and configuration data are included when you run the `tsm settings export` command. The data is exported as a json file. Specify the name and location of the json file by running the following command:

```
tsm settings export -f <filename>.json
```

Note: Because the backup contains secrets, we recommend that you encrypt the backup and store it in a secure place. For more information about Tableau Server secrets, see [Manage Server Secrets](#).

2. Back up repository and File Store data. Repository data is backed up with the `tsm`

`tsm maintenance backup` command. Specify the name and location of the backup file by running the following command:

```
tsm maintenance backup -f <filename>.tsbak -d
```

The backup file is assembled in a temporary location in the data directory and then written to the directory defined in the TSM `basefilepath.backuprestore` variable:

```
/var/opt/tableau/tableau_server-  
/data/tabsvc/files/backups/<filename>.tsbak
```

For more information about where backup files are written, and how to change that location, see [tsm File Paths](#). **Note:** Even when you change the backup location, the backup process uses a temporary location in the data directory to assemble the backup file.

Note: When File Store is configured external to Tableau Server you cannot use the `tsm maintenance backup` command to backup Tableau Server Data. For more information on how to backup this data, see [Backup and Restore with External File Store](#).

Restoring core Tableau Server functionality

The procedure below uses the assets from the previous two sections to rebuild a Tableau Server in a recovery scenario.

Note: If you need to restore only the repository on an otherwise functional Tableau Server, see [Restore from a backup](#). If you are running a distributed deployment, and your initial node has failed, see [Recover from an Initial Node Failure](#).

Topology and configuration backup data must be from Tableau Server on Linux. You cannot restore configuration data from a backup file that was generated on Tableau Server on Windows. To restore a backup made from Tableau Server on Windows to Tableau Server on Linux, see [Migrate Tableau Server from Windows to Linux](#).

Tableau Server on Linux Administrator Guide

You must have the following assets ready:

- **Topology and configuration data:** This is the json file that is generated by the `tsm settings export` command.
- **Repository backup file:** This is the file with a `.tsbak` extension that is generated by the `tsm maintenance backup` command.

You can only restore from a backup that has the same type of identity store as the running server. For example, a backup from a server using local authentication can be restored to a Tableau Server initialized with local authentication, but a backup from a server using Active Directory authentication cannot be restored to a server initialized with local authentication.

When you use `tsm maintenance restore` to restore your Tableau data, data extract files and the contents of the PostgreSQL database are overwritten with the content in the backup file (`.tsbak`). If you are running a distributed installation of Tableau Server, perform the restore on the node running the TSM Controller (this is usually the initial node).

- **Backup assets:** These assets include the list of documented configurations as noted in the previous section.

To restore a standalone Tableau Server

1. On the computer where you want to restore Tableau Server, Install and Initialize TSM. If your organization used non-default system user accounts, as described in an early section of this topic, then you must specify the users during this step.
2. Activate and Register Tableau Server.
3. (Optional). Configure Local Firewall.
4. (Optional). Verify LDAP.

5. Initialize Tableau Server. See Configure Initial Node Settings.
6. Import topology and configuration data. Copy the topology and configuration json backup file to the computer. Import the json file by running the following command:

```
tsm settings import -f <filename>.json
```

7. (Optional). Apply pending changes. At a command prompt, run:

```
tsm pending-changes apply
```

8. Restart Tableau Server. At a command prompt, run:

```
tsm restart
```

9. Restore repository data. See Restore from a backup.
10. (Optional). Repopulate TSM group membership. Add users to groups with this command:

```
sudo usermod -G <group_name> -a <username>
```

To restore a Tableau Server cluster

1. On the initial node, Install and Initialize TSM. If your organization used non-default system user accounts, as described in an early section of this topic, then you must specify the users during this step.
2. On the initial node, Activate and Register Tableau Server.
3. (Optional). On the initial node, Configure Local Firewall.
4. On the initial node verify LDAP (optional), and initialize Tableau Server. See Configure Initial Node Settings.
5. On the initial node, run `tsm topology nodes get-bootstrap-file --`

```
file <path\file>.json.
```

6. Copy the bootstrap.json file to all additional nodes in the cluster.

7. On each additional node in the cluster:

- a. Install the Tableau Server package.
- b. Navigate to the scripts directory.
- c. Initiate communication between initial node and the additional node:

```
sudo ./initialize-tsm -b <path-to-bootstrap>.json -u  
<admin-user-on-first-node> --accepteula
```

8. On the initial node, run `tsm topology list-nodes -v` and ensure that the node names have not changed from exported topology settings. *If the node names have changed, topology settings should be manually updated with new names, or the processes should be manually configured.*

9. The Cluster Controller process is required on every node and needs to be added explicitly. From the initial node, add an instance of the Cluster Controller to each additional node, where `<nodeID>` is the ID of one of the additional nodes. Add the process to each node separately. In this example we are adding the Cluster Controller to nodes 2 and 3:

```
tsm topology set-process -n node2 -pr clustercontroller -c  
1  
tsm topology set-process -n node3 -pr clustercontroller -c  
1  
tsm pending-changes apply
```

10. From the initial node, Deploy a Coordination Service Ensemble . The ensemble configuration must match your previous configuration.

11. On the initial node, import topology and configuration data. Copy the topology and

configuration json backup file to the computer. Import the json file by running the following command:

```
tsm settings import -f <filename>.json
```

12. On the initial node, apply pending changes. At a command prompt, run:

```
tsm pending-changes apply
```

13. On the initial node, restart Tableau Server. At a command prompt, run:

```
tsm restart
```

14. On the initial node, restore repository data. See [Restore from a backup](#).

15. On the initial node, repopulate TSM group membership. Add users to groups with this command:

```
sudo usermod -G <group_name> -a <username>
```

Restore other functionality

If the previous server was configured with the following features, then you will need to re-enable and reconfigure them on the restored server:

- Authentication solutions: OpenID, external SSL, and trusted authentication. See [Authentication](#).
- Site customizations: See [tsm customize](#).
- Enable access to PostgreSQL repository: See [tsm data-access repository-access enable](#).

Reencrypt Extracts After Restore

Optionally, if you are using the extract encryption at rest feature, after the backup is restored, you can reencrypt the extracts using different encryption keys. See [Extract Encryption at Rest](#).

Tableau Server on Linux Administrator Guide

Run `tabcmd reencryptextracts <site-name>` to reencrypt extracts on a given site. For more information, see `reencryptextracts`. Run this command on every site where you are storing encrypted extracts. Depending on the number of encrypted extracts on the site, this operation could consume significant server processing load. Consider running this operation outside of business hours.

Back up Tableau Server data

Regularly backing up Tableau Server is an important step in proper administration and maintenance of your server. You can use the `tsm maintenance restore` command to restore Tableau Server backups created using `tabadmin backup` and `tsm maintenance backup`. Database backups made in other ways, and virtual machine snapshots are not valid sources for restoring Tableau Server, so it is critical that you have an up-to-date backup.

You can only restore from a backup that has the same type of identity store as the running server. For example, a backup from a server using local authentication can be restored to a Tableau Server initialized with local authentication, but a backup from a server using Active Directory authentication cannot be restored to a server initialized with local authentication.

Looking for Tableau Server on Windows? See [Back Up Tableau Server Data](#).

Tableau Server data includes data extract files, as well as the Tableau PostgreSQL database, which contains workbook and user metadata, and configuration data. When you use TSM to create a backup, all of this data is saved in a single file with a `.tsbak` extension. If you are running a distributed installation of Tableau Server data from all the nodes is backed up.

The frequency of your backups depends on your environment, including how much use your server gets and how much and frequently the content and users change. Any changes or updates that happen after your backup will be lost if there is a system failure and you need to restore Tableau Server. The more activity there is, the more often you need to back the server up.

In addition to regular backups, you should always create a backup before upgrading to a new version of Tableau Server.

To help protect against data loss, after you create the backup, you should store the .tsbak file on a computer that is not a part of your Tableau Server installation.

Disk Space Usage for Backup

The free disk space required to create a backup varies depending on the amount of data in the Tableau Server repository and file store services, and their collocation with the tabadmincontroller service. During backups, the background tasks for cleaning up old extracts are temporarily paused. This means that, for the duration of the backup, extract refreshes will leave extra files in place, adding to disk space usage. If your backup takes a long time, or if your organization uses many extracts that are regularly updated, this can result in a significant amount of temporary disk space usage. These temporary files will be removed after the backup is complete.

The following table lists the disk space requirements for backup based on whether the node hosts the repository, file store, controller, or some combination of them.

Repository	File Store	Controller	Disk Space Required
			<p>3x repository data + 250 MB</p> <p>To obtain an estimate of the repository data, check the size of <code><data directory>/pgsql/data/base</code> directory.</p> <p>To obtain the exact size of the repository data, open the backup file and use the size of the <code>workgroup.pg_dump</code> file.</p>
			1.5x file store data

			To obtain an estimate of file store data (extracts, flows, etc.), check the size of <code><data directory>/dataengine</code> directory.
		✓	3x repository data + 250 MB + 2.5x file store data
✓	✓		3x repository data + 250 MB + 1.5x file store data
	✓	✓	3x repository data + 250 MB + 1.5x file store data
✓		✓	3x repository data + 250 MB + 2.5x file store data
✓	✓	✓	3x repository data + 250 MB + 1.5x file store data

Optimizing Tableau Server Backup

There are several ways you can maximize backup efficiency. Your environment can impact how effective each of these is, so test with your data to see what works best.

Optimizing with topology configurations:

- Co-locating File Store on the same node as the Administration Controller can reduce the length of time it takes to back up Tableau Server by reducing or eliminating the need to transfer data between nodes during the backup process. This is especially true if your organization uses many extracts.
- Co-locating the repository (pgsql) with the Administration Controller node can also help to reduce back up time, but the time savings is less significant than that of the File Store.

The Administration Controller is usually on the initial node, unless you have had an initial node failure and moved the controller to another node.

Optimizing with backup strategies:

Backup is a resource intensive process. If possible, doing your backups during off peak hours is a generally a good strategy. But this however, depends on your requirements and how often Tableau Server data is updated and what your restore requirements. For a detailed explanation of backup and disaster recovery, see [Tableau Server Disaster Recovery](#). Here are some backup strategies and adopt them to your requirements

- **Type of storage:** Solid State disks are recommended in general for backups. SSD helps make your backups faster and complete sooner compared to traditional spinning disks.
- **Backup compression:** You have the option of running your backups with or without compression. When you do your backup with compressions, your backup size will be comparatively smaller, but you may see a slower performance. So if your goal is more focused on speed, choose the `--skip-compression` option:

Use the `--skip-compression` option when backing up Tableau Server. This creates the backup without using compression, and results in a larger backup file but can reduce the amount of time it takes for the backup to complete. For more information, see `tsm maintenance backup`.

- **Snapshot backup:** This option is only available if you have configured your Tableau Server with External File Store. Although the performance of snapshot backups depend on the type of network attached storage, in general snapshot backups are faster than the traditional Tableau Server backups. For more information see, [Tableau Server External File Store](#).

Create a backup using the TSM command line interface (CLI)

Use the `tsm maintenance backup` command to create a backup of the data managed by Tableau Server. This data includes data extract files and the Tableau PostgreSQL database, which contains workbook and user metadata.

Note: When backing up Tableau Server on Linux to a network location, the unprivileged user must have write access to the network share where the backup files are written or the backup will fail.

To back up server configuration data, use the `tsm settings` command. When you use the `tsm maintenance backup` command, the current date is appended to the backup file:

```
tsm maintenance backup -f <backup_file> -d
```

For more information, see `tsm maintenance backup`.

Create a pre-upgrade backup

You should always create a backup before upgrading Tableau Server. You can create a backup while Tableau Server is running and minimize the amount of time the server is unavailable during upgrade. The process for creating a pre-upgrade backup is the same as for creating regular backups, with one additional consideration for distributed installations.

Note: Uninstall Tableau Server from any nodes that you are not including in your new installation to avoid conflicts between the older nodes and the new installation.

Backups during upgrades

During a Tableau Server upgrade, a temporary backup of the database is created to allow for migrations that occur as part of upgrades. This is done during the upgrade and in most cases has no noticeable impact to the upgrade process. In certain special cases there can be additional impacts:

- Major version postgres updates—If an upgrade includes a major version update to the database used for the Tableau repository, the internal upgrade backup is done without compression to save time. This requires additional temporary disk space during the

upgrade process.

Tableau Server versions that include a major version database update; 2020.4.

Scheduling and Managing Backups

Beginning in 2020.4.0 you can use tsm commands to schedule a backup. You need to do this from the command line (there is no TSM UI to schedule backups). The `tsm maintenance backup` command allows you to create and update backup schedules. The `tsm schedules` commands give you the ability to view, delete, pause, resume, and update schedules.

To schedule a backup:

1. Open a command prompt as administrator on the initial node (the node where TSM is installed).
2. Run the following command:

```
tsm maintenance backup -f <backup-file> -sr <recurrence> -st <time-to-run> -sd <days-to-run> -sn <schedule-name>
```

For example, to create a backup schedule named "monthly-backup" that runs on the 15th of each month at 2 am and generates a file called `<yyyy.mm.dd.hh.mm>`-`ts-mid_month_backup.tsbak`:

```
tsm maintenance backup -f ts-mid_month_backup -sr monthly -st 02:00 -sd 15 -sn monthly-backup
```

To view a scheduled backup:

1. Open a command prompt as administrator on the initial node (the node where TSM is installed).
2. Run the following command:

```
tsm schedules list
```

Tableau Server on Linux Administrator Guide

You can sort the schedules by scheduled run time, earliest to latest, or by name using the `--next-run` or `--schedule-name` options. You can alternately display details for a single schedule using the `--schedule-id` option. When you view a single schedule you see additional details about it, including when it was created, how many times it has run, and specific options used when it is run. Job options are shown in JSON format as "Job args".

To update a scheduled backup:

1. Open a command prompt as administrator on the initial node (the node where TSM is installed).
2. Run the following command:

```
tsm schedules update --schedule-id <ID> --schedule-time  
<time-to-run> --schedule-recurrence <frequency> --schedule-  
days <day-to-run>
```

Note: To add or change a name, use the `tsm-maintenance-backup` command.

To suspend or resume a backup schedule:

1. Open a command prompt as administrator on the initial node (the node where TSM is installed).
2. Run one of the following commands:

- To suspend a schedule:

```
tsm schedules suspend --schedule-id <scheduleID>
```

- To resume a suspended schedule:

```
tsm schedules resume --schedule-id <scheduleID>
```

Script the backup process

If you back up often, you might want to create a script that performs the backup and related tasks for you. These tasks include:

- Clean up files and folders before running the backup.
- Running the backup itself.
- Copying the backup file to a separate computer for safekeeping.

This section discusses `tsm` commands you can use together to perform a backup and related tasks.

For additional information about scripting TSM commands, see the [Tableau Community Forums](#).

Remove log files and clear temporary folders

You can clean old Tableau Server log file and temporary files to reduce the time it takes to create a backup, and to ensure the backup file is as small as possible.

To clean log files older than a few days, run the following command:

```
tsm maintenance cleanup
```

Run the backup

Note: When backing up Tableau Server on Linux to a network location, the unprivileged user must have write access to the network share where the backup files are written or the backup will fail.

To create the backup, use the `tsm maintenance backup` command:

```
tsm maintenance backup --file <backup_file> --append-date
```

Note the following about the command:

Tableau Server on Linux Administrator Guide

- Add `--append-date` to the command to include the date in the backup file name.
- The backup file is assembled in a temporary location in the data directory and then written to the directory defined in the `TSM basefilepath.backuprestore` variable:

```
/var/opt/tableau/tableau_server-  
/data/tabsvc/files/backups/<filename>.tsbak
```

For more information about where backup files are written, and how to change that location, see [tsm File Paths](#). **Note:** Even when you change the backup location, the backup process uses a temporary location in the data directory to assemble the backup file.

Copy the backup file to another computer

As a best practice, after the backup is created, copy the backup file to another location that is separate from Tableau Server.

Restore from a backup

Use the `tsm maintenance restore` command to restore your Tableau Server data. You might do this if you had a system failure and need to restore your data, if you need to switch back to a previous version of Tableau Server (for example, if there is a problem with an upgrade), or if you are moving Tableau Server to new hardware. You can use the `tsm maintenance restore` command to restore Tableau Server backups created using `tabadmin backup` and `tsm maintenance backup`.

Limitations when restoring Tableau Server

- If you are restoring a backup created using `tabadmin backup`, *and* you used a custom asset key, you must save a copy of your `asset_keys.yml` file so you can include the file when doing your restore. For more information, see [Save your asset keys file before uninstalling Tableau Server for Windows 2018.1.x or earlier](#).
- Database backups made in other ways, and virtual machine snapshots are not valid sources for restoring Tableau Server.

- When you use `tsm maintenance restore` to restore your Tableau data, data extract files and the contents of the PostgreSQL database are overwritten with the content in the backup file (`.tsbak`). If you are running a distributed installation of Tableau Server, perform the restore on the node running the TSM Controller (this is usually the initial node).
- You can only restore from a backup that has the same type of identity store as the running server. For example, a backup from a server using local authentication can be restored to a Tableau Server initialized with local authentication, but a backup from a server using Active Directory authentication cannot be restored to a server initialized with local authentication.
- You can only restore a backup file to a version of Tableau Server version that is the same or newer than the version the backup was created on. You cannot restore to an older version of Tableau.
- During restore, the restore process will initiate a full re-indexing of the content and external assets managed by Tableau Server. This process consumes CPU resources which may be noticeable during backup and restore.

Restore Tableau Server from a backup file

Note: This operation includes steps that you may need to perform using the TSM command line.

1. (Optional) Copy the `.tsbak` file to the default file location.

The `restore` command expects a backup file in the directory defined in the TSM `basefilepath.backuprestore` variable. By default:

```
/var/opt/tableau/tableau_server/data/tabsvc/files/backups/
```

For more information about file paths and how to change them, see [tsm File Paths](#).

Note: If you are restoring a backup that was copied into the backups folder, ensure that the Run As service account, which can be found in TSM Web UI under **Security**, has at least Read access to the backup file. Otherwise, the restore process may not be able to unzip the backup file and the restore will fail.

2. Stop the server. At a command prompt, type:

```
tsm stop
```

3. Restore from a backup file. At a command prompt, type:

```
tsm maintenance restore --file <file_name>
```

In the above line, replace `<file_name>` with the name of the backup file you want to restore from.

Note: If you encounter errors when trying to restore from backup, see [Troubleshoot Tableau Server on Linux](#).

4. Restart the server:

```
tsm start
```

Server Maintenance

As an administrator, you will want to check the status of the server, analyze and monitor the activity on the server, manage scheduled tasks, or perform certain maintenance activities such as clearing saved data connection passwords. In addition, there are several settings that you may want to specify to customize the user experience for people using the server. You can do some of these tasks from the General page of the Status page and others from the Settings page.

View Server Process Status

You can view server process status can be by running a TSM CLI command or by accessing TSM Web UI or Admin pages on Tableau Server.

Viewing process status with TSM CLI

Run the following command:

```
tsm status -v
```

This command outputs all of the processes that are configured on the instance and their corresponding status.

Viewing process status in web UI

There are two locations in Tableau Server or Tableau Services Manager (TSM) where administrators can view the state of Tableau processes. You may be able to access one or both of these locations, depending on how your account and server are set up. Most of the process status information that displays is duplicated on both Status pages. This section explains each page, and identifies what is unique for each one.

Note The status pages displayed in both web UI locations show a subset of the total processes configured on a given node. To view all processes, you must run the TSM CLI command, `tsm status -v`.

- The Tableau Services Manager (TSM) status page is accessible in TSM and can be viewed by TSM administrators. You must be able to log into TSM to see this page. For information about signing into TSM, see [Sign in to Tableau Services Manager Web UI](#).
- The Tableau Server status page appears in the Tableau Server web UI and is accessible by Tableau Server administrators. This page includes Tableau Server processes, along with links to troubleshooting documentation if a process is not running

Tableau Server on Linux Administrator Guide

as expected. If you hover your mouse pointer over the status indicator for a process, a tooltip shows the node name and the port the process is running on. The Tableau Server status page does not show TSM processes. For information about signing into Tableau Server as an administrator, see [Sign in to the Tableau Server Admin Area](#).

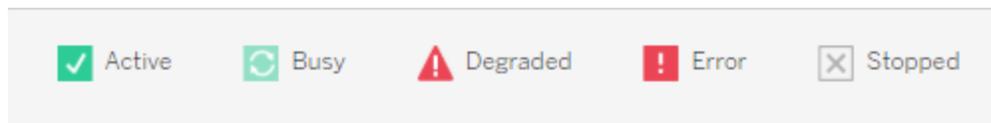
Tableau Services Manager (TSM) Status page

The TSM Status page shows you the state of server processes, including TSM Controller and License Server. These two processes do not display on the Tableau Server Status page.

 STATUS MAINTENANCE CONFIGURATION			
Process	node1	node2	node3
Gateway	✓	✓	✓
Application Server	✓	✓	✓ ✓
Interactive Microservice Container	✓	✓	✓
VizQL Server	✓ ✓ ✓ ✓	✓	✓ ✓
Cache Server	✓ ✓	✓	✓ ✓
Cluster Controller	✓	✓	✓
Search & Browse	✓		✓
Backgrounder	✓ ✓	✓ ✓	✓ ✓
Non-Interactive Microservice Container	✓	✓	✓
Data Server	✓ ✓	✓ ✓	✓
Data Engine	✓	✓	✓
File Store	✓	✓	✓
Repository	✓		✓
Tableau Prep Conductor			
Ask Data	✓	✓	✓
Elastic Server	✓		
TSM Controller	✓		
License Server	✓		

Refresh Status

Possible status indicators are listed at the bottom of the table:

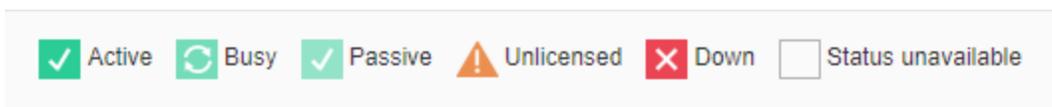


Server Status

Process Status
The real-time status of processes running in Tableau Server.

Process	OPENVM-QKRAFPE9	OPENVM-28P7I269	OPENVM-QKRAFPEL
Cluster Controller	✓	✓	✓
Gateway	✓	✓	✓
Application Server	✓	✓	✓
VizQL Server	✓✓✓✓	✓✓	✓✓
Cache Server	✓✓	✓✓	✓✓
Search & Browse	✓	✓	✓
Backgrounder	✓✓	✓✓	✓✓
Data Server	✓✓	✓✓	✓✓
Data Engine	✓	✓	✓
File Store	✓	✓	
Repository	✓		

Possible status indicators are listed at the bottom of the table:

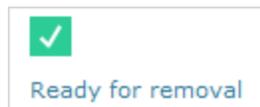


When Tableau Server is functioning properly, most processes will show as Active, Busy or Passive (Repository):

- **Active**—The process is functioning as intended. See File Store in Troubleshoot Server Processes for details on possible active states.
- **Busy**—The process is completing some task. See File Store and Repository in Troubleshoot Server Processes for more information.

- **Passive**—The repository is in passive mode
- **Unlicensed**—The process is unlicensed.
- **Down**—The process is down. The implications of this differ depending on the process.
- **Status unavailable**—Tableau Server is unable to determine the status of the process.

If there is additional information, a message displays below the status icon and links to appropriate documentation:



Note: Tableau Server is designed to be self-correcting. If a service or process stops responding or goes down, Tableau Server attempts to restart it. This can take 15 to 30 minutes to complete. Because of this, reacting immediately to service or process alerts can be counter-productive, especially in an installation with redundant services that can handle requests while one restarts.

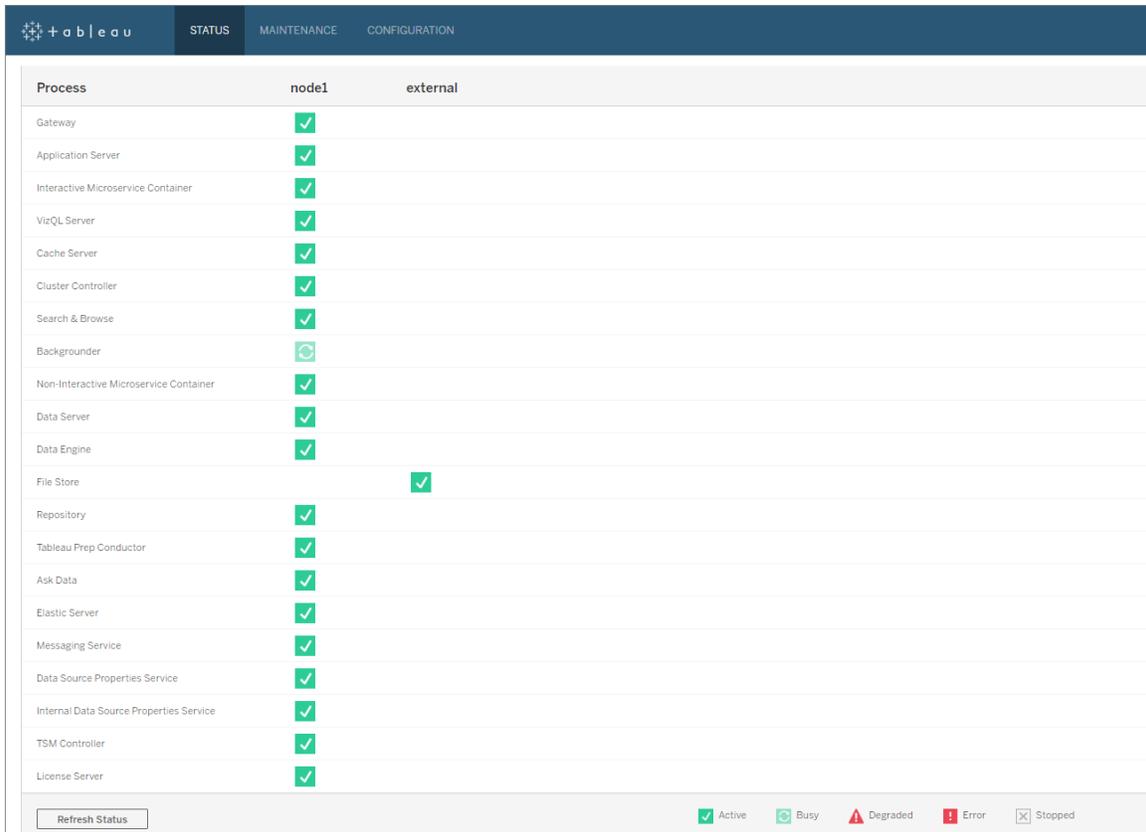
For more information about troubleshooting process status, see [Troubleshoot Server Processes](#).

External Node

Some processes can be configured external to Tableau Server. For example, File Store can be configured on a SAN or NAS, the repository can be deployed to an AWS RDS instance. In such cases, the Tableau Server Status page will show these processes on **External Node** with status **E**, and the Tableau Services Manager (TSM) status page will show these processes on **external** with a check mark to indicate that the process is configured externally.

Tableau Server Manager (TSM) status page showing File Store as configured external to Tableau Server:

Tableau Server on Linux Administrator Guide

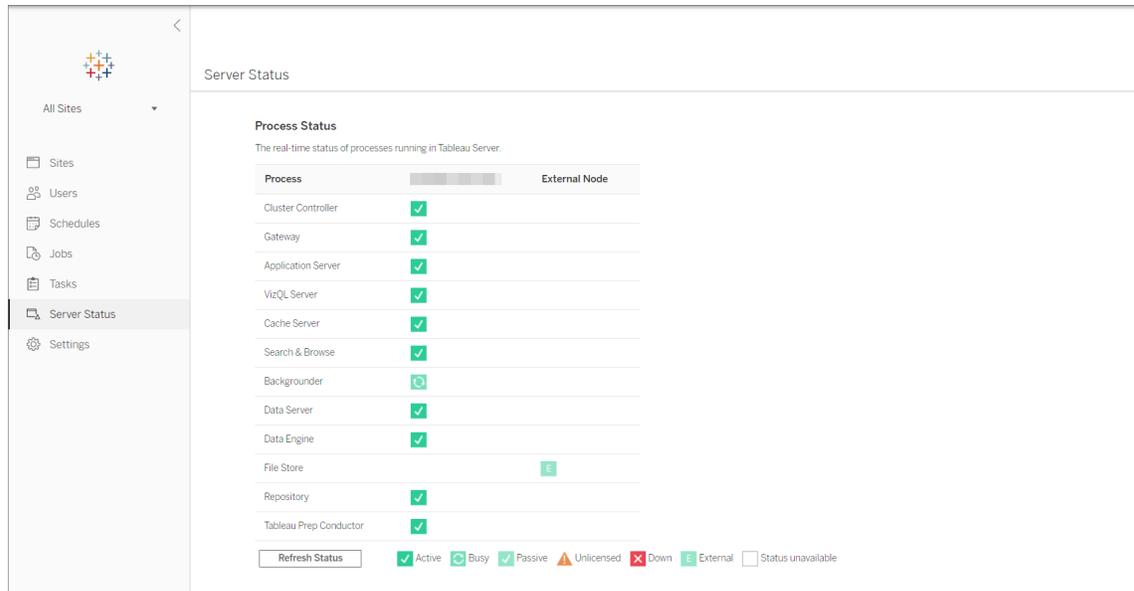


The screenshot shows the Tableau Server status page with the following data:

Process	node1	external
Gateway	✓	
Application Server	✓	
Interactive Microservice Container	✓	
VizQL Server	✓	
Cache Server	✓	
Cluster Controller	✓	
Search & Browse	✓	
Backgrounder	🔄	
Non-Interactive Microservice Container	✓	
Data Server	✓	
Data Engine	✓	
File Store		✓
Repository	✓	
Tableau Prep Conductor	✓	
Ask Data	✓	
Elastic Server	✓	
Messaging Service	✓	
Data Source Properties Service	✓	
Internal Data Source Properties Service	✓	
TSM Controller	✓	
License Server	✓	

Legend: ✓ Active 🔄 Busy ⚠ Degraded ! Error ✖ Stopped

Tableau Server status page showing File Store as configured external to Tableau Server:



Access Status Remotely

Note: The information in this article refers to the Tableau Server status page. For information about the Tableau Server status page and the TSM status page, see [View Server Process Status](#).

You must be a Tableau Server administrator to see the Server Status page, but you can grant remote access to other computers to allow access to a machine-readable (XML) version of the Status table by non-admin users and by computers other than the initial Tableau Server node. One reason you might do this is as part of a remote monitoring process.

To grant remote access to Tableau Server status:

1. Open a command prompt as an administrator and type the following:

```
tsm configuration set -k wgserver.systeminfo.allow_refer-
rer_ips -v <ip address>
```

In the above command, `<ip address>` is the IPv4 address of the computer for which you want to enable remote access to the Tableau Server status XML.

For example:

```
tsm configuration set -k wgserver.systeminfo.allow_refer-  
rer_ips -v 10.32.139.31
```

If you are enabling remote access for more than one computer, use commas to separate each IP address.

```
tsm configuration set -k wgserver.systeminfo.allow_refer-  
rer_ips -v 10.32.139.31,10.32.139.35
```

2. Commit the configuration change:

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

Now, users of computers with the IP addresses that have been added can view Tableau process status by entering the URL `http://<server-
>/admin/systeminfo.xml` in a browser or from a command line (for example, `curl http://jsmith/admin/systeminfo.xml`).

If Tableau Server has been configured to work with a load balancer or proxy server, use the hostname or IP address of the initial Tableau Server node to access the XML version of the status page.

For details on the XML that is returned, see [Get Process Status as XML](#).

Get Process Status as XML

To get a machine-readable version of the server process status, that is, a version of the status formatted in XML, use the following URL:

`http://my_tableau_server/admin/systeminfo.xml`

You must be signed in to Tableau Server to view the machine-readable process status, or have [enabled remote access](#).

The server returns a status report similar to the following:

```
<systeminfo xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
  <machines>
    <machine name="my_tableau_server">
      <repository worker="my_tableau_server:8060" status="Active"
preferred="false"/>
      <dataengine worker="my_tableau_server:27042" status-
s="Active"/>
      <applicationserver worker="my_tableau_server:8600"
status="Active"/>
      <apiserver worker="my_tableau_server:8000" status-
s="Active"/>
      <vizqlserver worker="my_tableau_server:9100"
status="Active"/>
      <dataserver worker="my_tableau_server:9700" status-
s="Active"/>
      <backgrounder worker="my_tableau_server:8250"
status="Active"/>
      <gateway worker="my_tableau_server:80" status="Active"/>
      <searchandbrowse worker="my_tableau_server:11000" status-
s="Active"/>
      <cacheserver worker="my_tableau_server:6379"
status="Active"/>
      <filestore worker="my_tableau_server:9345" status="Active"
pendingTransfers="0" failedTransfers="0" syncTimestamp="2015-
02-27T20:30:48.564Z"/>
      <clustercontroller worker="my_tableau_server:12012" status-
s="Active"/>
```

Tableau Server on Linux Administrator Guide

```
<coordination worker="my_tableau_server:12000" status-  
s="Active"/>  
</machine>  
</machines>  
<service status="Active"/>  
</systeminfo>
```

Status values in the XML

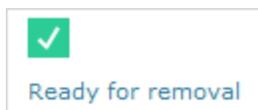
- **<process> worker** - The name of the node running the process and the port the process is using.
- **status** - The status of the process on the node. Possible values are: `Active`, `Passive`, `Unlicensed`, `Busy`, `Down`, `ReadOnly`, `ActiveSyncing`, `StatusNotAvailable`, `StatusNotAvailableSyncing`, `NotAvailable`, `DecommissionedReadOnly`, `DecomisioningReadOnly`, and `DecommissionFailedReadOnly`
- **pendingTransfers** - A count of the workbook or data source extracts the node needs to get to be fully synced. These represent items that were published to this file store node, and items that were published to other file store nodes and need to be copied to this node.
- **failedTransfers** - A count of the workbooks or data sources that did not transfer successfully to this file store node during the last automated job. The automated job normally runs about every 15 to 30 minutes, but may take longer when transferring a large number of extracts or large extracts.

Failed transfers do not necessarily indicate a problem with Tableau Server. The recurring automated job will normally transfer files that failed during the previous sync. Reasons for failed file transfers are listed in the logs.
- **syncTimestamp** - The time in UTC of the last automated job that ran and synchronized files.

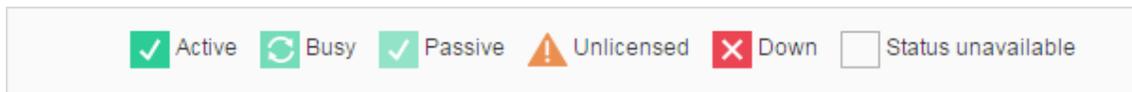
Troubleshoot Server Processes

Note: The information in this article refers to the Tableau Server status page. For information about the Tableau Server status page and the TSM status page, see [View Server Process Status](#).

When Tableau Server is functioning properly, processes will show as Active, Busy or Passive (Repository). If there is additional information, a message appears below the status icon:



Possible status indicators are:



Note: Tableau Server is designed to be self-correcting. If a service or process stops responding or goes down, Tableau Server attempts to restart it. This can take 15 to 30 minutes to complete. Because of this, reacting immediately to service or process alerts can be counter-productive, especially in an installation with redundant services that can handle requests while one restarts.

The following sections provide troubleshooting recommendations for status messages that you may see.

Cluster Controller

This message will only display if you have more than two nodes.

✘ Status: Down; Message: "Node degraded"

One or more of the following are true:

- Repository on the node is stopped.
- Node cannot respond to failover elsewhere on the cluster.
- If Tableau Server is configured for high availability and this is the active repository, failover to the second repository occurs.
- No status available for repository or file store on this node.

No action is necessary unless the cluster controller is regularly down or is down for an extended period of time.

If that occurs, take the following actions, in order, until the problem is resolved:

1. Check disk space. If disk space is limited, save the log files (use `tsm maintenance ziplogs`) in case you need them for Support, then remove unnecessary files.
2. Restart Tableau Server.
3. If Cluster Controller continues to show as down, save the log files (`tsm maintenance ziplogs`) and contact Support.

File Store

File Store status only reflects the state of the file store when the page was loaded.

An active status () with no message indicates that no extracts were being synchronized when the page was loaded. It is possible that the recurring "catch-all" job is running and synchronizing extracts.

Status: Busy; Message: "Synchronizing"

"Synchronizing," usually indicates that extracts were being synchronized across file store nodes when the page was loaded.

However, the "synchronizing" message is also returned following installation (both single-node and multi-node). After Tableau initializes the status should disappear within 15 or 20 minutes.

Status: Down; "Data Extracts unavailable"

On a single-node installation: "Data Extracts unavailable" indicates that existing extracts may be available but publish/refresh will fail. On multi-node installations, this message indicates that extract synchronization will fail for this node.

No action is necessary unless the file store is regularly down or is down for an extended period of time.

If that occurs, take the following actions, in order, until the problem is resolved:

1. Check disk space. If disk space is limited, save the log files (use `tsm maintenance ziplogs`) in case you need them for Support, then remove unnecessary files.
2. Restart Tableau Server.
3. If Cluster Controller continues to show as down, save the log files (`tsm maintenance ziplogs`) and contact Support.

Status: Busy; "Decommissioning"

This message indicates that this File Store is in read-only mode and that any unique files on this node are being replicated to other file store nodes.

To remove this node, wait until the status message changes to "Ready for removal".

Status: Active; "Ready for removal"

This message indicates that the File store is in read-only mode.

You can safely stop (`tsm stop`) the cluster and remove File Store processes, or remove entire node.

Status: Active; "Decommission failed"

This message indicates that the File store is in read-only mode, and that at least one unique file failed to replicate to another file store node.

To resolve a failed decommissioning:

Tableau Server on Linux Administrator Guide

1. Run the `tsm topology filestore decommission` command again.
2. Check disk space on other file store nodes. Decommissioning will fail if another file store node does not have enough space to store all the extracts.
3. Check the `tsm.log` file on the initial node and additional nodes for errors.
4. Stop Tableau Server (`tsm stop`) and then try running the `tsm topology filestore decommission` command again.
5. Put the file store node back into read/write mode (`tsm topology filestore recommission`), collect logs, and then contact Support.
6. With Support: copy and merge `extracts` directory from this file store node to the same directory on another file store node.

Repository

 **Status: Busy; Message: "Setting up"**

The "Setting up" message indicates one or more of the following states:

- Passive repository is being synchronized with active repository.
- Repository is not ready to handle failover.
- Repository may have gotten more than two minutes behind active repository and is being setup again (this is faster than waiting for a sync).
- Failover occurred and this former active repository is rejoining the cluster.

Wait until the repository status message changes to "Passive".

If this message does not appear, or if it is taking a long time:

1. Check disk space and free space if possible.
2. Check cluster controller logs for errors.
3. Restart node.

 **Status: Busy; Message: "Synchronizing"**

Repository is synchronizing, for example after a failover.

 **Status: Down; Message: n/a**

When the Repository shows a status of down and there is no message, then the Repository is in one of the following states:

- If the installation is configured for high availability, failover of the repository occurred.
- Processes are restarting with updated database connection configurations after failover.
- If another active repository is not available, Tableau Server is down.

Take these actions in order until a step resolves the problem:

1. Wait several minutes for cluster controller to attempt to restart.
2. Restart Tableau Server (`tsm restart`).
3. Check disk space. If disk space is limited, save the log files (use `tsm maintenance ziplogs`) in case you need them for Support, then remove unnecessary files.
4. Restart Tableau Server.
5. If repository continues to show as down, save the log files (`tsm maintenance ziplogs`) and contact Support.

 **Status: Passive; Message: n/a**

A passive status with no message indicates that the node is working as intended and that it is ready for failover if needed.

VizQL Server

 **Status: Unlicensed; Message: n/a**

For information about unlicensed status for a VizQL Server process, see [Handle an Unlicensed Server Process](#).

Clear Saved Data Connection Passwords

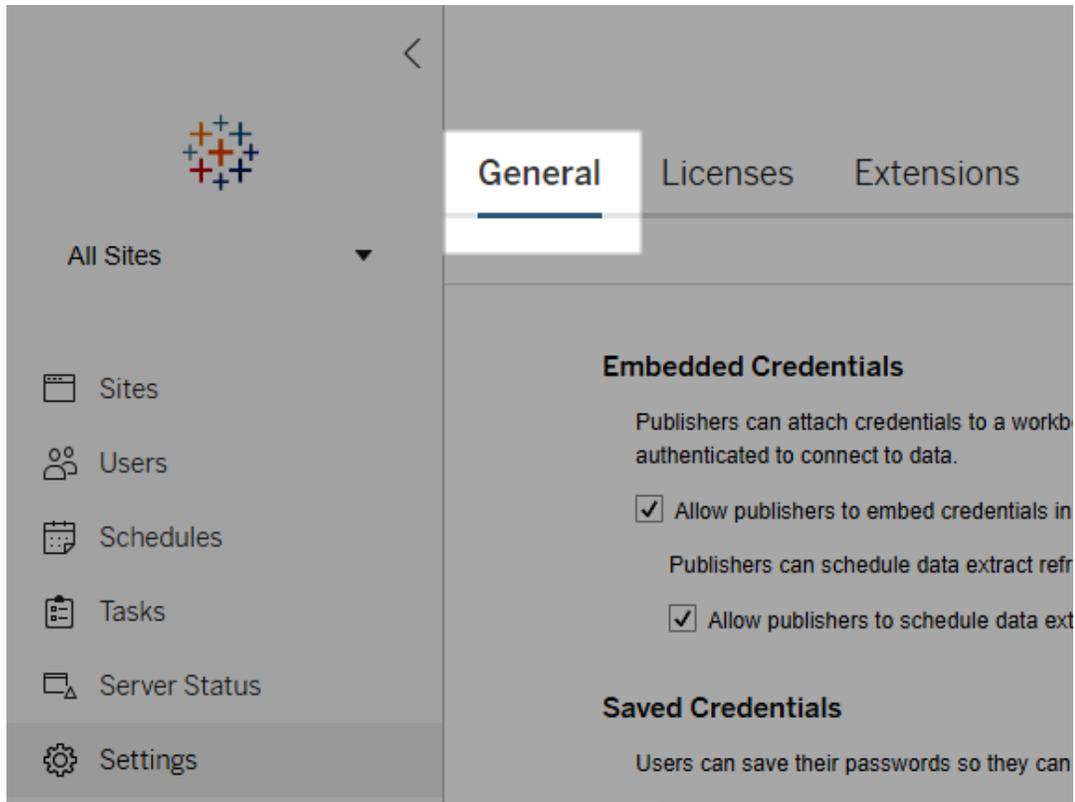
As the administrator, if you enable users to save data source passwords, server users can save data source passwords across multiple visits and browsers so they are not prompted for their credentials each time they connect to a data source.

You can reset the data source passwords for all Tableau Server users. Doing this forces them to sign in to the data sources the next time they visit a view that requires database

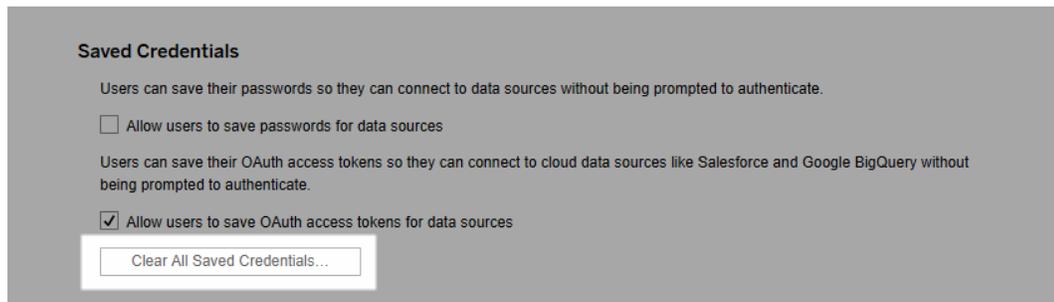
authentication. Server users can also clear their saved data connection passwords on an individual basis using their User Preferences page.

To clear saved data connection passwords for all server users:

1. In a single-site server, click **Settings > General**. On a multi-site server, select **Manage all sites**, then click **Settings > General**.

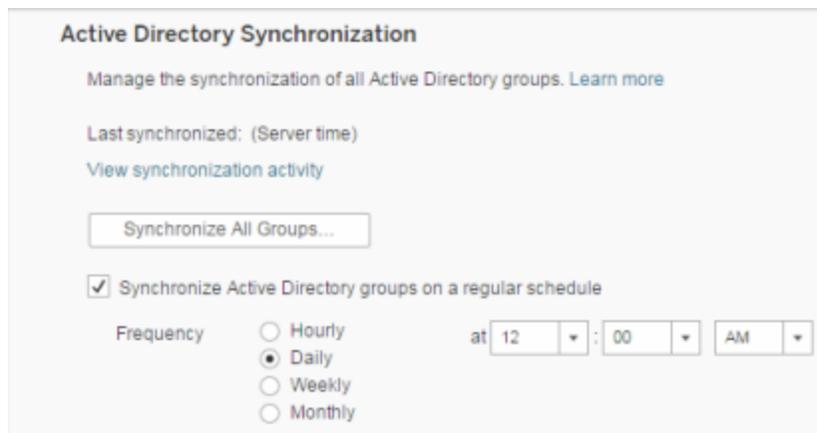


- Under Saved Credentials, click **Clear All Saved Credentials**.



Synchronize External Directory Groups on the Server

As a server administrator, you can synchronize all external directory (such as Active Directory) groups (that have been configured on Tableau Server) on a regular schedule or on-demand on the **General** tab of the **Settings** page for the server.



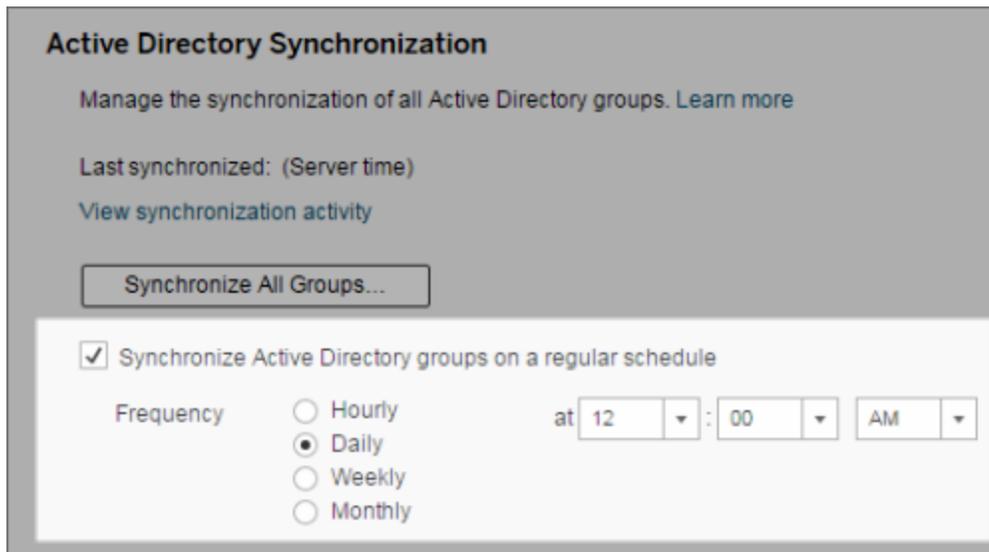
Note: In the context of user and group synchronization, Tableau Server configured with LDAP identity store is equivalent to Active Directory. Active Directory synchronization features in Tableau Server function seamlessly with properly configured LDAP directory solutions.

Before you begin

Before synchronizing groups as described in this topic, you must first import the external directory group into Tableau Server. See [Create Groups via Active Directory](#).

Synchronize external directory groups on a schedule

1. **Single-site:** Click **Settings**> **General**.
Multisite: In the site menu, click **Manage All Sites** and then click **Settings**> **General**.
2. Scroll down the page to **Active Directory Synchronization**, and then select **Synchronize Active Directory groups on a regular schedule**.



3. Select the frequency and time of synchronization.
4. Click **Save**.

Synchronize all external directory groups on demand

At any time, you can synchronize external directory (such as Active Directory) groups with Tableau Server to ensure that new users and changes in the external directory are reflected in all external directory groups on Tableau Server.

1. **Single-site:** Click **Settings> General**.

Multisite: In the site menu, click **Manage All Sites**, and then click **Settings> General**.

2. Under **Active Directory Synchronization**, click **Synchronize All Groups**.

Note: Beginning in versions 2021.1.23, 2021.2.21, 2021.3.20, 2021.4.15, 2022.1.11, 2022.3.3, a default time limit of 4 hours limits how long a scheduled group synchronization can take before it is canceled. You can change this time limit if your scheduled synchronization is of very large groups, or taking longer than this. For more information, see `Synchronize All Active Directory Groups on a Schedule and background.timeout.sync_ad_group`.

View synchronization activity

You can view the results of synchronization jobs in the **Background Tasks for Non Extracts** administrative view. **Queue Active Directory Groups Sync** is the task that queues and indicates the number of **Sync Active Directory Group** tasks to be run.

1. **Single-site:** Click **Status**.

Multisite: In the site menu, click **Manage All Sites** and then click **Status**.

2. Click the **Background Tasks for Non Extracts** link.
3. Set the **Task** filter to include **Queue Active Directory Groups Sync** and **Sync Active Directory Group**.

You can quickly navigate to this administrative view by clicking the **View synchronization activity** link in the **Settings** page for the server.

Set the minimum site role for users in an external directory group

In the **Groups - Details** page, you can set the minimum site role for group users to be applied during Active Directory synchronization.

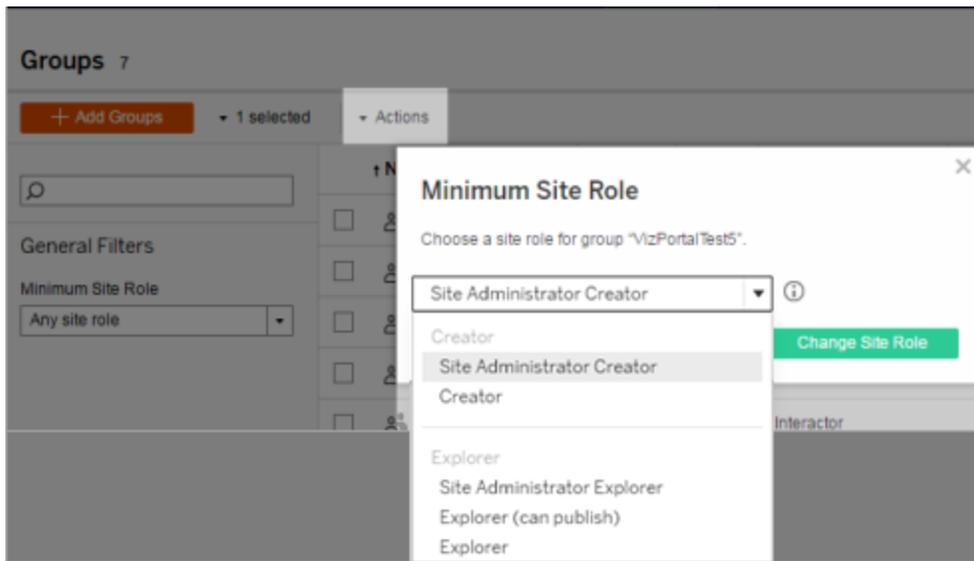
This setting does not run synchronization; instead, it sets the minimum site role to applied to the group every time synchronization runs. The result is that when you synchronize external directory groups, new users are added to the site with the minimum site role. If a user already exists, the minimum site role is applied if it gives the user more access in a site. If you don't set a minimum site role, new users are added as **Unlicensed** by default.

Note: A user's site role can be promoted but never demoted based on the minimum site role setting. If a user already has the ability to publish, that ability will always be maintained. For more information on minimum site role, see Site roles and Active Directory import and synchronization.

1. In a site, click **Groups**.
2. On the Groups page, select a group.

Click **Actions > Minimum Site Role**.

3. Select the minimum site role, and then click **Change Site Role**.



What happens when users are removed in the source external directory?

Users cannot be automatically removed from the Tableau Server through an external directory sync operation. Users that are disabled, deleted, or removed from groups in the external directory remain on Tableau Server so that administrators can audit and reassign the user's content before removing the user's account completely. For more information, see [Sync behavior when removing users from Active Directory](#).

Improving group synchronization performance

External directory synchronization is performed by the background process. The Background process is the same process that is used for managing and creating extracts, and is also used to generate subscription content. In large organizations with dynamic group membership and heavy extract usage, the external directory group synchronization process may be disruptive. We recommend running group synchronization during non-business hours.

By default, the Background process performs synchronization in a serial operation. This means that each group is synchronized, one after the other, in a single Background process. If you are running multiple instances of Background processes either on a single

Tableau Server or across a distributed deployment, consider enabling parallel processing for external directory synchronization. When parallel Backgrounder processing is enabled, the group synchronization is distributed across multiple Backgrounder processes for better performance.

To enable parallel backgrounder processing for group synchronization, open TSM CLI and enter the following commands:

```
tsm configuration set -k backgrounder.enable_parallel_adsync -v true  
  
tsm pending-changes apply
```

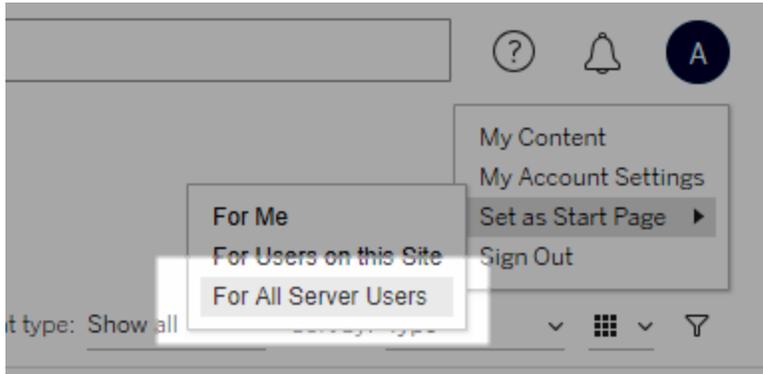
Set the Default Start Page for All Users

In a new deployment, when users sign in to the Tableau Server web authoring environment, they are taken to the Home screen, which displays a role-based welcome banner. Home also displays recent views, favorites, and the site's most-viewed content. As the server administrator, you can change users' default landing page at the server and site level. For example, you can show all workbooks, and when the user signs in, they see the workbooks they have access to.

To set the default start page for all users

1. Display the page or filtered view you want to be the default page users see when they sign in to the site.
2. Select your profile icon in the upper right area of the page.
3. To set the start page for:
 - All users on a site: click **Set as Start Page** and **For Users on this Site**.

- All users on the server: click **Set as Start Page** and **For All Server Users**.



User-set start pages and hierarchy

Users can set their own start pages from their profile icon, and can reset their start pages in their account settings (for information, search for “Access Your Profile and Account Settings” in the online Tableau Server Help for your operating system).

If a user sets their own start page, it will override any start page set by a server administrator. The next time that user signs in, they will land on the start page they've set. If a server administrator sets start pages for both a server and a site, users will default to the start page set for themselves (if any), then the site start page, then the server start page. If neither a user or an administrator has set a start page, users will default to Home.

Disable Automatic Client Authentication

After Tableau connected clients (Tableau Desktop, Tableau Mobile, Tableau Prep Builder, Bridge, and personal access tokens) successfully sign in to Tableau Server, they are automatically authenticated in the future. Connected client sessions are managed by OAuth access and refresh tokens.

By default, OAuth client tokens reset after a year. If a client token has not been used in 14 days, then it will expire. You can change these values by setting the `refresh_token.`

Tableau Server on Linux Administrator Guide

`absolute_expiry_in_seconds` and `refresh_token.idle_expiry_in_seconds` options. See `tsm configuration set Options`.

As a Tableau Server administrator can also disable OAuth automatic authentication for connected clients. In this case, session expiry is solely governed by Tableau Server session behavior, which manages Web authoring sessions. See 9. Verify session lifetime configuration.

Web authoring sessions are not considered a "connected client," and they do not use OAuth tokens.

To immediately disconnect connected clients from Tableau Server and require users to sign in every time they connect:

1. In the site menu, click **Manage All Sites**, and then click **Settings > General**.
2. Under **Connected Clients**, clear the option **Allow clients to automatically connect to Tableau Server**.
3. Click **Save**.

Remove Unneeded Files

As a best practice, you should regularly monitor disk space usage on your server. If the Tableau Server computer runs low on disk space, the impact to can be significant, including ultimately causing a failure. If you determine that space is getting low, you can archive any you want to save, and purge unneeded files, freeing up space for Tableau.

Monitoring disk space usage

There are several things you can do to monitor disk space usage:

- **Notifications:** You can configure Tableau Server to send notifications when disk space reaches predetermined levels. For more information, see [Configure Server Event Notification](#).
- **Administrative views:** You can use a pre-built administrative view to help monitor disk space usage. For more information, see [Server Disk Space](#).

Reducing disk space usage

To make more disk space available, you can take the following steps:

- **Archive log files:** Tableau Server generates log files when running. These can be helpful in troubleshooting issues, and when you are working with Tableau Support, but you do not need to leave them in place indefinitely. To save disk space without losing logs, you can archive them with the `tsm maintenance ziplogs` command, and then copy the ziplogs archive to a computer that is not part of the Tableau installation for safe keeping. For more information, see [Log File Snapshots \(Archive Logs\)](#).
- **Clean up unwanted files:** After archiving any logs you want to save, use the `tsm maintenance cleanup` command to remove log files older than seven days, temporary files, and optionally, rows from the `http_requests` table in the Tableau Server repository. You should run the cleanup command regularly. For more information about which files are removed, see [tsm maintenance cleanup](#).
- **Remove other files:** Over time Tableau Server can generate files that do not need to be left in place. In addition to the files mentioned above, be aware of files like old backups from previous versions. Tableau Server backup files have a `.tsbak` extension. We strongly recommend regularly backing up Tableau, and saving the backup files to a computer that is not part of the Tableau Server installation for safe keeping, but once you save the file in another location, you can delete it from the Tableau computer. For more information on backup files, including how to create them and where they are saved, see [Back up Tableau Server data](#).

Server Settings (General and Customization)

The following settings are available on the **General** and **Customization** pages in **Server - Settings**.

Many of these settings move from the Server Settings page to the Site Settings page when there is more than one site on the server. These are marked with "Moves to Site Settings on multi-site servers."

General

Setting	Description
<p>Site Name and ID</p> <p>Moves to Site Settings on multi-site servers.</p>	<p>Specifies the site name seen in the user interface and the ID seen in the site URL. (If you are editing the Default site, you cannot change the ID.)</p> <p>You can't change the “#/site” portion of the URL (for example, http://localhost/#/site/sales). In multi-site server environments, these segments appear in the URL for sites other than the Default site.</p>
<p>Storage</p> <p>Moves to Site Settings on multi-site servers.</p>	<p>Select either Server Limit or GB, and for the latter enter the number of gigabytes you want as a limit for storage space for published workbooks, extracts, and other data sources.</p> <p>If you set a server limit and the site exceeds it, publishers will be prevented from uploading new content until the site is under the limit again. Server administrators can track where the site is relative to its limit using the Max Storage and Storage Used columns on the Sites page.</p>
<p>Revision History</p> <p>Moves to Site Settings on multi-site servers.</p>	<p>Specifies the number of previous versions of workbooks, flows, and data sources that are stored on the server.</p>
<p>Tableau Prep Conductor</p> <p>Moves to Site Settings on multi-site servers.</p>	<p>Controls whether users with appropriate permissions can schedule and monitor flows. Tableau Prep Conductor is part of the Tableau Data Management Add-on. For more information, see About Tableau Prep Conductor.</p>
<p>Web Authoring</p>	<p>Controls whether browser-based authoring is enabled for the site. When web authoring for workbooks is dis-</p>

<p>Moves to Site Settings on multi-site servers.</p>	<p>abled, users can't create or edit published workbooks from the server web environment but instead must use Tableau Desktop to republish the workbook. When web authoring for flows is disabled, users can't create or edit published flows from the server web environment but instead must use Tableau Prep Builder to re-publish the flow.</p> <p>For more information, see Set a Site's Web Authoring Access and Functions in Tableau Online Help.</p>
<p>Managing Users</p> <p>Moves to Site Settings on multi-site servers.</p>	<p>Determines whether only server administrators can add and remove users and change their site roles, or whether site administrators can too.</p> <p>If you allow site administrators to manage users, specify how many users they can add to the site by selecting one of the following:</p> <ul style="list-style-type: none"> • Server Limit adds the number of available server seat licenses. For a server with core-based licensing, there is no limit. • Site Limit lets site administrators add users up to a limit you specify. • Site Role Limit lets site administrators add users of each site role up to the license limit you specify for the site. <p>For more information, see View Server Licenses.</p>
<p>Tableau Catalog</p> <p>Moves to Site Settings on multi-site servers.</p>	<p>Turns off Catalog capabilities when Tableau Server or a Tableau Online site is licensed with the Data Management Add-on. For more information, see Disable Catalog.</p>
<p>Workbook Performance after a Scheduled Refresh</p>	<p>Pre-computes recently viewed workbooks with scheduled refreshes to open them faster. For more inform-</p>

Moves to Site Settings on multi-site servers.	ation, see Configure Workbook Performance after a Scheduled Refresh .
Workbook Performance Metrics Moves to Site Settings on multi-site servers.	Lets site users collect metrics on how workbooks perform, such as how quickly they load To initiate recording, users must add a parameter to the workbook's URL. For more information, see Create a Performance Recording .
Managed Keychain Clean Up Moves to Site Settings on multi-site servers.	Lets site administrators manage saved credential keychains for OAuth connections on the site. For more information, see OAuth Connections .
Automatically Suspend Extract Refresh Tasks Moves to Site Settings on multi-site servers.	To save resources, Tableau can automatically suspend extract refresh tasks for inactive workbooks. This feature applies only to refresh schedules that run weekly or more often. For more information, see Automatically Suspend Extract Refreshes for Inactive Workbooks in Tableau Online Help.
Email Settings Moves to Site Settings on multi-site servers.	Specifies the From address and message footer seen in automatic emails for alerts and subscriptions.
User Visibility Moves to Site Settings on multi-site servers.	Controls what user and group names are visible to other users. For more information, see Manage User Visibility in Tableau Online Help.
Availability of Ask Data Moves to Site Settings on multi-site servers.	Controls whether Ask Data is enabled or disabled by default for data sources. Ask Data lets users query data using conversational language and automatically see visualizations. For more information, see Automatically Build Views with Ask Data in Tableau user Help.
Automatic Access to	Automatically grants users certain capabilities to

<p>Metadata about Databases and Tables</p> <p>Moves to Site Settings on multi-site servers.</p>	<p>external assets using derived permissions. For more information, see Turn off derived permissions in Tableau Online Help.</p>
<p>Sensitive Lineage Data</p> <p>Moves to Site Settings on multi-site servers.</p>	<p>Specifies whether sensitive lineage data should be obfuscated or filtered when users don't have the appropriate permissions to related metadata. For more information, see Sensitive lineage data.</p>
<p>Extract Encryption at Rest</p> <p>Moves to Site Settings on multi-site servers.</p>	<p>Lets you encrypt .hyper extracts while they are stored on Tableau Server. Server administrators can enforce encryption of all extracts on their site or allow users to encrypt all extracts associated with particular published workbooks or data sources. For more information, see Extract Encryption at Rest.</p>
<p>Tableau Mobile</p> <p>Moves to Site Settings on multi-site servers.</p>	<ul style="list-style-type: none"> • Enable offline favorites <p>Controls whether offline previews of users' favorite views are generated for display when users access the site on Tableau Mobile. For more information, see Manage Tableau Mobile Data on Devices in the Tableau Mobile Deployment Guide.</p> • Enable app lock <p>Requires a biometric method or device passcode for users to open the site on Tableau Mobile. For more information, see Enable App Lock for Added Security in the Tableau Mobile Deployment Guide.</p>
<p>Sharing</p>	<p>Allows users to share items directly with other users.</p>

Moves to Site Settings on multi-site servers.	When an item is shared, the recipients get a notification and the item is added to their Shared with Me page. If this is not enabled, users can only copy a link to share. For more information, see Share Web Content in Tableau user Help.
Data-Driven Alerts Moves to Site Settings on multi-site servers.	Lets users automatically receive emails when data reaches key thresholds. For more information, see Send Data-Driven Alerts in Tableau user Help.
Request Access Moves to Site Settings on multi-site servers.	Lets users send access requests to content or project owners. For more information, see Let Site Users Request Access to Content in Tableau Online Help.
Comments Moves to Site Settings on multi-site servers.	Controls whether users can add remarks in a Comments side pane for each view and @mention other Tableau users to notify them via email. For more information, see Comment on Views in Tableau user Help.
Metrics Content Type Moves to Site Settings on multi-site servers.	Controls whether metrics are available on the site. When enabled, users can create metrics from views and metrics appear as a content type. When disabled, metrics won't appear on the site or continue to sync; however, you can re-enable the feature to bring back previously created metrics. For more information, see "Set Up for Metrics" in Tableau Online Help or Tableau Server Help .
Site Time Zone for Extracts Moves to Site Settings on multi-site servers.	The default time zone for extract-based data sources in a site is Coordinated Universal Time (UTC). Site administrators can set a different time zone. For more information, see Set the Site Time Zone for Extracts in Tableau Online Help.
Run Now	Server administrators can use this setting to allow or

<p>On multi-site servers this appears on both Server Settings (at bottom of page) and Site Settings.</p>	<p>block users from manually running extract refreshes, flow runs, and subscriptions. This setting can be applied at the server level to include all the sites on Tableau Server or applied at the site level to include only specific sites.</p> <ul style="list-style-type: none"> • By default, this option is set to allow users to run jobs manually. Clear the check box to prevent users from running jobs manually. • This applies only to jobs that are manually initiated by a user from the web interface, REST API calls, or tabcmd commands. Jobs initiated from scheduled tasks will continue to run at the schedule time and will not be affected. • Select Run Now to allow users to change the connection type (Live/Extract) of data sources on the web.
<p>Recommendations for Views</p> <p>Moves to Site Settings on multi-site servers.</p>	<p>Controls whether recommendations show on the site and whether the names of users who have looked at recommended items show on recommendation tooltips.</p> <div data-bbox="678 1178 1373 1356" style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;"> <p>Note: If you use Tableau Server, your administrator can disable Recommendations.</p> </div>
<p>Cross-Database Joins</p> <p>Moves to Site Settings on multi-site servers.</p>	<p>Determines where the join process happens when joining data from multiple sources. For more information, see Combine Tables from Different Databases in Tableau user Help.</p>
<p>Web Page Objects</p> <p>Moves to Site Settings on multi-site servers.</p>	<p>Controls whether these dashboard objects can display target URLs. For more information, see Security for Web Page objects in Tableau user Help.</p>

<p>Manage Notifications</p> <p>Moves to Site Settings on multi-site servers.</p>	<p>Controls how site users can receive notifications for events such as extract jobs, flow runs, or when another user shares content with them or mentions them in a comment. Notifications can be seen in their Tableau site via the notification center or sent by email. When a notification is enabled, users can configure their notification preferences on their Account Settings page.</p>
<p>Language and Locale</p>	<p>Controls the language used for the server user interface and the locale used for views. Individual users can override this setting on their Account Settings page. Also, web browser settings are evaluated first to determine which language and locale should be used. For more information, see Language and Locale for Tableau Server.</p>
<p>Start Page</p>	<p>Links to the server's current default start page for all users. For more information on how to change the default start page, see Set the Default Start Page for All Users. Individual users will be able to override this setting (search for "Access Your Profile and Account Settings" in the Tableau Server Help for details).</p>
<p>Sign In Customization</p>	<p>Add a custom note to the server sign in page. The Sign In setting lets you add text. You can optionally add a URL to make the text a link. This note will also appear if a user receives a sign in error. Custom notes do not display on Tableau Mobile.</p> <p>To set a custom note, sign in to a site on Tableau Server. On the left-side navigation pane, select Manage all sites from the drop-down site list. Select Settings and add a message to Sign In Customization.</p> <p>For more information, see Customize Your Server.</p>

<p>Connected Clients</p>	<p>Controls whether mobile users must sign in and provide their credentials every time they connect to Tableau Server, or if users can connect with their devices to Tableau Server without providing credentials after they authenticate their device successfully the first time. For more information, see Disable Automatic Client Authentication.</p>
<p>Active Directory Synchronization</p> <p>Only appears when server is configured for AD identity store.</p>	<p>Controls the synchronization of all Active Directory groups in Tableau Server based on a schedule that you specify after you select the option Synchronize Active Directory groups on a regular schedule. For more information, see Synchronize External Directory Groups on the Server.</p>
<p>Guest Access</p>	<p>Allows users to view and interact with embedded views without having to sign in to a Tableau Server account. Permission can be assigned to the Guest User account to control the interactivity allowed for each view. This option is only available if you have a core-based server license.</p>
<p>Embedded Credentials in Content</p>	<ul style="list-style-type: none"> • Let publishers embed credentials in a data source, flow, or workbook <p>Allows publishers to attach passwords to published workbooks or flows that will automatically authenticate web users to connect to data sources. The passwords are attached to workbooks or flow inputs and are only accessible on the server. For example, when a workbook is opened in Tableau Desktop, users still need to enter a user name and password to connect to the data source. When this setting is turned off, all existing embedded passwords are saved but</p>

	<p>are not used for authentication. If you turn the setting back on, users don't have to re-embed the passwords.</p> <ul style="list-style-type: none"> • Let publishers schedule flow runs and extract refreshes <p>Allows publishers to assign workbooks or flows to schedules. This option is only available if Let publishers embed credentials in a data source, flow, or workbook is enabled. When this setting is enabled, Tableau Desktop users will see scheduling options in the Publish dialog box.</p>
<p>Saved Credentials for Data Sources</p>	<ul style="list-style-type: none"> • Let users save passwords for data sources <p>Allows users to choose "Remember my password" and save data source passwords across multiple visits, browsers, and devices. (By default, users can choose to "Remember my password until I sign out," which lets them save their password only for a single browser session.)</p> <p>As an administrator, you can clear all saved passwords at any time. In addition, users can clear their own saved passwords.</p> <ul style="list-style-type: none"> • Let users save OAuth access tokens for data sources <p>Allows users to store access tokens with their user preferences. Access tokens are provided by</p>

	<p>cloud data sources that support OAuth connections, and they are used instead of user names and passwords to grant access to the data.</p>
<p>Recommendations Training Schedule</p>	<p>Recommendations take two forms: recommendations for data sources and tables (for Tableau Desktop) and recommendations for views (for Tableau Server). Recommendations are based on the popularity of content and on content used by other users determined to be similar to the current user.</p> <p>The training schedule controls how often the server checks for new content and new usage information to keep the recommendations up to date. New content includes new and updated data sources and workbooks. New usage information includes information such as "Laura Rodriguez used the Food Catering data source" and "Henry Wilson accessed the Monthly Sales view."</p> <p>If you notice an impact on server performance, schedule this process to occur at a time when the server load is low. To track performance impact, look for the "Recommendations Trainer" or "View Recommendations Trainer" tasks in the Background Tasks for Non Extracts view.</p> <p>If you want to disable Recommendations, change the tsm configuration set option "recommendations.enabled" to false. If you want to disable only recommendations for views, change the option "recommendations.vizrecs.enabled" to false.</p>

Reset to Default Settings	Returns any server settings described here that have been changed since setup back to their original state.
OAuth Clients Registry	For a subset of connectors, you can register a custom OAuth client for the site to override an OAuth client that has been configured for the server. By registering a custom OAuth client, you enable new and existing connections to use the site-level OAuth client instead of the server-wide OAuth client. For more information, see Configure custom OAuth for a site .

Customization

Setting	Description
Welcome Banner	<p>Add a custom message to the welcome banner on the home page for all server users to see. The custom message can contain up to 240 characters of text and hyperlinks as well as one paragraph break. Administrators can also disable the default Tableau welcome banner for the server.</p> <p>For more information, see Customize Your Server.</p>

Stop or Restart the Tableau Server Computer

As a best practice, you should *always* stop Tableau Server before you stop or restart the computer it is running on. This is true whether you are running Tableau on virtual machines (VMs), or on dedicated hardware. You should never turn off a computer without first stopping Tableau Server. Shutting down the computer while Tableau is running can cause problems restarting Tableau Server, and may result in unexpected results.

To be safe, follow these steps, whether you have a scripted process to shut down your systems, or manually shut down your computers:

1. Stop Tableau Server.

You can do this either from the command line, using the `tsm stop` command, or from the TSM Web UI, by clicking **Tableau Server is running**, and selecting **Stop Tableau Server**.

Note: Some TSM processes will continue to run, even after you stop Tableau Server. This is normal, and you can go ahead and stop your computer. The running services are designed to shut themselves off when the computer is stopped.

2. Once Tableau is stopped, stop your computer.
3. When you are ready, restart your computer. This might be after you have completed planned maintenance, or after leaving the computer off for some extended period of down time.
4. Start Tableau Server.

You can do this either from the command line, using the `tsm start` command, or from the TSM Web UI, by clicking **Tableau Server is stopped**, and selecting **Start Tableau Server**.

TSM will start automatically when the computer starts, so you can run `tsm` commands even though Tableau Server is stopped.

tsm Command Line Reference

The topics in this section include reference content for Tableau Services Manager (TSM) command line interface (CLI) to support Tableau Server.

TSM is used to manage installation and configuration of Tableau Server. To learn more about TSM, see [Tableau Services Manager Overview](#).

You can automate the installation and configuration tasks supported by the TSM CLI using the TSM API. To learn more about the prerelease (Alpha) TSM API, see [Tableau Services Manager API](#).

Looking for tsm commands for Tableau Server on Windows? See [tsm Commands](#).

Using the tsm CLI

You can run tsm commands on the initial node (the node where TSM is installed), or on any additional node in the cluster.

To run tsm commands, you need to open a command prompt.

1. Open a command prompt with an account that is a member of the `tsmadmin` group on a node in the cluster.
2. Run the command you want. If you are running the command from a node other than the initial node, include the `-s` option to specify the URL of the initial node by name (not IP address), and include the TSM port, 8850.

To see the version of TSM and Tableau Server from the initial node:

```
tsm version
```

To see the version of TSM and Tableau Server from an additional node:

```
tsm version -s https://<initial_node_name>:8850
```

For example:

```
tsm version -s https://myTableauHost:8850
```

Authenticating with tsm CLI

Beginning in the 2019.2 release of Tableau Server, running tsm commands will not require you to enter a password if the following are true:

- The account you are running commands with is a member of the TSM-authorized group, by default, the `tsmadmin` group. The Tableau unprivileged user (by default, the `tableau user`) and root account may also run TSM commands.
- You are running commands locally on the Tableau Server that is running the Tableau Server Administration Controller service. By default, the Tableau Server Administration Controller service is installed and configured on the initial node in a distributed deployment.

Logging into tsm CLI locally

If you are running tsm commands on the local computer with user account that is a member of a TSM-authorized group, then you will not need to specify a password. In this case, just run the command, for example:

```
tsm version
```

Logging into tsm CLI remotely

If you are running TSM commands from a node in a cluster where the Tableau Server Administration Controller service is not running, then you must authenticate a session with the Tableau Server Administration Controller service on the remote computer before you can run commands. For example, run the following command:

```
tsm login -s <server_name> -u <account_name>
```

Where `<server_name>` is the name of the node where the Tableau Server Administration Controller service is running and `<account_name>` is an account that is a member of a TSM-authorized group.

After running this command, you will be prompted for a password. After the account has been authenticated, you can run TSM commands.

Viewing and adding accounts to the TSM-authorized group

The TSM-authorized group is created during server installation. By default, the TSM-authorized group that is named `tsmadmin`. If you created an alternative TSM-authorized group during installation, then substitute your group name for `tsmadmin` in the following code examples.

To view the user accounts in the `tsmadmin` group, run the following command:

```
grep tsmadmin /etc/group
```

To add a user account to the `tsmadmin` group:

```
sudo usermod -G tsmadmin -a <username>
```

Scripting and automating with tsm CLI

To run automation on a Tableau Server without a password in the script file, run the script on the initial node and with an account in the proper TSM-authorized group. See the "Authenticating" section above for more details.

Viewing help content in the shell

To view minimal help content from a command line, use the `tsm help category`.

Synopsis

```
tsm help [category] [command]
```

Commands

```
tsm help
```

Help for all tsm commands

```
tsm help <category>
```

Show help for a specific command category. For example, `tsm help authentication`.

```
tsm help <category> <command>
```

Show help for a specific command. For example, `tsm help authentication open-id`.

```
tsm help commands
```

List all top-level commands or categories.

Categories

tsm authentication

You can use the `tsm authentication` commands to enable, disable, and configure user authentication options for Tableau Server.

- **kerberos**
 - configure
 - disable
 - enable
- **list**
- **mutual-ssl**
 - configure
 - disable
 - enable
- **openid**
 - configure
 - disable
 - enable
 - get-redirect-url
 - map-claims

- **pat-impersonation**
 - disable
 - enable
- **saml**
 - configure
 - disable
 - enable
 - export-metadata
 - map-assertions
- **sitesaml**
 - disable
 - enable
- **sspi**
 - disable
 - enable
- **trusted**

tsm authentication kerberos <commands>

Enable, disable, and configure Kerberos user authentication on Tableau Server. See [Configure Kerberos](#).

Synopsis

```
tsm authentication kerberos configure --keytab-file <keytab_
file.keytab> [global options]
```

```
tsm authentication kerberos enable [global options]
```

```
tsm authentication kerberos disable [global options]
```

Options for kerberos configure

```
-kt, --keytab-file <keytab_file.keytab>
```

Required.

Specifies the service .keytab file used for requests to the KDC.

tsm authentication list

List the server's existing authentication-related configuration settings.

Synopsis

```
tsm authentication list [--verbose][global options]
```

Options

v, --verbose

Optional.

Show all configured parameters.

tsm authentication mutual-ssl <commands>

Enable, disable, and configure mutual SSL for user authentication on Tableau Server. To learn more about mutual SSL, see [Configure Mutual SSL Authentication](#).

Before you enable mutual SSL, you must enable and configure SSL for external communication. For information, see [Configure SSL for External HTTP Traffic to and from Tableau Server](#).

Synopsis

```
tsm authentication mutual-ssl configure [options]
[global options]
```

```
tsm authentication mutual-ssl disable [global options]
```

```
tsm authentication mutual-ssl enable [global options]
```

Tableau Server on Linux Administrator Guide

Options

`-cf, --ca-cert <certificate-file.crt>`

Optional.

Specifies the location and file name for the certificate file. The file must be a valid, trusted certificate from a Certificate Authority (for example, Verisign).

`-fb, --fallback-to-basic <true | false>`

Optional.

Specifies whether Tableau Server should accept user name and password for authentication if SSL authentication fails.

Default value is `false`, to indicate that when configured for mutual SSL, Tableau Server does not allow a connection when SSL authentication fails. However, Tableau Server accepts username and password authentication from REST API clients, even if this option is set to `false`.

`-m, --user-name-mapping <upn | ldap | cn>`

Optional.

Specifies the user name syntax (UPN, LDAP or CN) to retrieve from identity store or directory. The syntax must match the format for Subject or Subject Alternative Name on the user certificate.

`-rf, --revocation-file <revoke-file.pem>`

Optional.

Specifies the location and file name for the certificate revocation list file. This file can be a `.pem` or `.der` file.

tsm authentication openid <commands>

Enable, disable, and configure OpenID Connect (OIDC) user authentication on Tableau Server.

Synopsis

```
tsm authentication openid configure [options] [global options]
```

```
tsm authentication openid disable [global options]
```

```
tsm authentication openid enable [global options]
```

```
tsm authentication openid get-redirect-url [global options]
```

```
tsm authentication openid map-claims [options] [global options]
```

Options for openid configure

Note: Options must be set during initial configure, or during reconfigure. Individual options cannot be set separately.

```
-a, --client-authentication <string>
```

Optional.

Specifies custom client authentication method for OpenID Connect.

To configure Tableau Server to use the Salesforce IdP, set this value to `client_secret_post`.

```
-cs, --client-secret <string>
```

Optional.

Tableau Server on Linux Administrator Guide

Specifies the provider client secret. This is a token that is used by Tableau to verify the authenticity of the response from the IdP. This value is a secret and should be kept securely.

`-cu, --config-url <CONFIG-URL>`

Optional.

Specifies the provider configuration URL. The default value is constructed based on the name of the server (`gateway.public.host`), and the gateway port, if any (`gateway.public.port`). In addition, by default the protocol is set to `https://` if SSL is enabled for the server.

`-mf, --config-file <config-file.json>`

Optional.

Specifies the local path to the static OIDC discovery JSON document.

`-i, --client-id <CLIENT-ID>`

Optional.

Specifies the provider client ID that your IdP has assigned to your application.

`-id, --ignore-domain <true | false>`

Optional. Default: `false`

Set this to `true` if the following are true:

- You are using email addresses as usernames in Tableau Server
- You have provisioned users in the IdP with multiple domain names
- You want to ignore the domain name portion of the `email` claim from the IdP

Before you proceed, review the user names that will be used as a result of setting this option to `true`. User name conflicts may occur. In the case of a user name conflict, the risk of information disclosure is high. See Requirements for Using OpenID Connect.

`-if, --iframed-idp-enabled <true | false>`

Optional. Default: `false`

Specifies if IdP is allowed inside of an iFrame. The IdP must disable clickjack protection to allow iFrame presentation.

`-ij, --ignore-jwk <true | false>`

Optional. Default: `false`

Set this to `true` if your IdP does not support JWK validation. In this case, we recommend authenticating communication with your IdP using mutual TLS or another network layer security protocol.

`-r, --return-url <return-url>`

The URL of your server. This is typically is the public name of your server, such as `"http://example.tableau.com"`.

`-sn, --custom-scope-name <string>`

Optional.

Specifies a custom scope user-related value that you can use to query the IdP. See Requirements for Using OpenID Connect.

Options for openid map-claims

Use these options to change the default OIDC claims Tableau Server will use when communicating with your IdP. See Requirements for Using OpenID Connect.

Tableau Server on Linux Administrator Guide

`-i, --id <string>`

Optional. Default: `sub`

Change this value if your IdP does not use the `subclaim` to uniquely identify users in the ID token. The IdP claim that you specify should contain a single, unique string.

`-un, --user-name <string>`

Optional. Default: `email`

Change this value to the IdP claim that your organization will use to match user names as stored in Tableau Server.

tsm authentication pat-impersonation <commands>

Enable and disable personal access token impersonation for administrators on Tableau Server.

For more information about personal access token impersonation, see [Personal Access Tokens](#).

Synopsis

```
tsm authentication pat-impersonation disable [global options]
```

```
tsm authentication pat-impersonation enable [global options]
```

To view whether personal access token impersonation is enabled, run the following command:

```
tsm authentication list
```

tsm authentication saml <commands>

Configure Tableau Server to support single-sign on using the SAML 2.0 standard, enable or disable SAML for a site, map assertion attribute names between Tableau Server and the identity provider (IdP).

Available commands

```
tsm authentication saml configure [options] [global options]
```

```
tsm authentication saml disable [options] [global options]
```

```
tsm authentication saml enable [options] [global options]
```

```
tsm authentication saml export-metadata [options]
[global options]
```

```
tsm authentication saml map-assertions [options]
```

tsm authentication saml configure

Configure the SAML settings for the server. Specify the SAML certificate and metadata files, provide additional required information, set additional options.

If you are configuring SAML for the first time or have previously disabled it, you must run this command with `tsm authentication saml enable`. For more information, see [Configure Server-Wide SAML](#).

Synopsis

```
tsm authentication saml configure [options] [global options]
```

Options

```
-e, --idp-entity-id <id>
```

Required for initial SAML configuration; otherwise optional. IdP entity ID value.

Typically this is the same as the Tableau Server return URL (specified in the `--idp-return-url` parameter). The entity ID that you enter is used as a base for generating site-specific entity IDs. For example, if you enter the following:

```
http://tableau-server
```

A site configured for SAML might display the following entity ID:

`http://tableau-server/samlservice/public/sp/metadata?alias=48957410-9396-430a-967c-75bdb6e002a0`

To find a site's entity ID, go to the site's **Settings** page, and select the **Authentication** tab. When SAML is enabled, the entity ID is shown under the first step for configuring site-specific SAML, exporting metadata.

`-r, --idp-return-url <idp-return-url>`

Required for initial SAML configuration; otherwise optional. The SAML return URL configured in the IdP. This is typically the Tableau Server external URL; for example, `https://tableau-server`.

Notes

- `http://localhost` does not work for an external server.
- Adding a trailing slash to the URL (`https://tableau-server/`) is not supported.

`-i, --idp-metadata <idp-metadata.xml>`

Required for initial SAML configuration; otherwise optional. Provide the location and name of the XML metadata file you exported from the IdP's settings.

For example, `/var/opt/tableau/tableau_server/data/saml/<metadata-file.xml>`

`-cf, --cert-file <certificate.crt>`

Required for initial SAML configuration; otherwise optional. The location and file name for the x509 certificate file for SAML. For requirements for the certificate file, see **SAML Requirements**.

For example, `/var/opt/tableau/tableau_server/data/saml/<file.crt>`

`-kf, --key-file <certificate.key>`

Required for initial SAML configuration; otherwise optional. Location and name of the

key file that goes along with certificate.

For example, `/var/opt/tableau/tableau_server/data/saml/<file.key>`

`-a, --max-auth-age <max-auth-age>`

Optional. Default value is 7200 (2 hours).

The maximum number of seconds allowed between a user's authentication and processing of the AuthNResponse message.

`-d, --desktop-access <enable | disable>`

Optional. Default value is enable.

Use SAML to sign in to the server from Tableau Desktop. If single sign-on from Tableau client applications does not work with your IdP, you can set this to `disable`.

`-m, --mobile-access <enable | disable>`

Optional. Default value is enable.

Allow using SAML to sign in from older versions of Tableau Mobile app. Devices running Tableau Mobile app version 19.225.1731 and higher ignore this option. To disable devices running Tableau Mobile app version 19.225.1731 and higher, disable SAML as a client login option on Tableau Server.

`-so, --signout <enable | disable>`

Optional. Enabled by default.

Enable or disable SAML sign out for Tableau Server.

`-su, --signout-url <url>`

Optional. Enter the URL to redirect to after users sign out of the server. By default this is the Tableau Server sign-in page. You can specify an absolute or a relative URL.

Tableau Server on Linux Administrator Guide

Example

```
tsm authentication saml configure --idp-entity-id https://tableau-server --idp-metadata /var/opt/tableau/tableau_server-  
/data/saml/<metadata.xml> --idp-return-url https://tableau-server --cert-file /var/opt/tableau/tableau_server-  
/data/saml/<file.crt> --key-file /var/opt/tableau/tableau_server-  
/data/saml/<file.key>
```

tsm authentication saml enable and saml disable

Enable or disable server-wide SAML authentication. In this context, all sites and users that you enable for SAML go through a single identity provider.

Synopsis

```
tsm authentication saml enable [global options]  
  
tsm authentication saml disable [global options]
```

tsm authentication saml export-metadata

Export the Tableau Server .xml metadata file that you will use to configure the SAML IdP.

Synopsis

```
tsm authentication saml export-metadata [options]  
[global options]
```

Options

```
-f, --file [/path/to/file.xml]
```

Optional.

Specifies the location and file name in which the metadata will be written. If you don't include this option, `export-metadata` saves the file to the current directory, and names it `samlmetadata.xml`.

`-o, --overwrite`

Optional.

Overwrites an existing file of the same name specified in `-f`, or of the default name if `-f` is not included. If a file specified in `-f` exists, and `-o` is not included, the command does not overwrite the existing file.

tsm authentication saml map-assertions

Maps attributes between the IdP and Tableau Server. Provide the name that the IdP uses for the attribute specified in each argument.

Synopsis

```
tsm authentication saml map-assertions --user-name <user-name>
[global options]
```

Options

`-r, --user-name <user-name-attribute>`

Optional. The attribute in which the IdP stores the user name. On Tableau Server, this is the same as the display name.

`-e, --email <email-name-attribute>`

Do not use. This option is not supported by Tableau.

`-o, --domain <domain-name-attribute>`

Optional. The attribute in which the IdP stores the domain name. Use this option if you add users from a domain that's different from the domain of the Tableau Server

computer. For more information, see [When running multiple domains](#).

```
-d --display-name <display-name-attribute>
```

Do not use. This option is not supported by Tableau.

Example for saml map-assertions

```
tsm authentication saml map-assertions --user-name-  
e=<sAMAccountName> --domain=<FQDM> or tsm authentication saml  
map-assertions --user-name=jnguyen --domain=example.myco.com
```

tsm authentication sitesaml enable and sitesaml disable

Set the server to allow or disallow SAML authentication at the site level. Enabling site-specific SAML gives you access to the **Settings > Authentication** tab in the Tableau Server web UI. The **Authentication** tab contains the site-specific SAML configuration settings.

Use the `sitesaml enable` command with `saml configure` if you haven't yet configured the server to allow site-specific SAML. For more information, see [Configure Site-Specific SAML](#).

Synopsis

```
tsm authentication sitesaml enable [global options]
```

```
tsm authentication sitesaml disable [global options]
```

tsm authentication sspi <commands>

This command will only work on Tableau Server on Windows. If you attempt to enable SSPI on Tableau Server on Linux, an error will be returned.

Enable or disable automatic sign-in using Microsoft SSPI.

If you use Active Directory for authentication, you can optionally enable automatic logon, which uses Microsoft SSPI to automatically sign in your users based on their Windows user-name and password. This creates an experience similar to single sign-on (SSO).

Do not enable SSPI if you plan to configure Tableau Server for SAML, trusted authentication, a load balancer, or for a proxy server. SSPI is not supported in these scenarios.

Synopsis

```
tsm authentication sspi disable [global options]
```

```
tsm authentication sspi enable [global options]
```

As with all authentication commands, you must run `tsm pending-changes apply` after running this command.

tsm authentication trusted <commands>

Configure trusted authentication (trusted tickets) for user authentication on Tableau Server.

Synopsis

```
tsm authentication trusted configure [options] [global options]
```

Options

```
-th, --hosts <string>
```

Optional.

Specifies the trusted host names (or IPv4 addresses) of the web servers that will be hosting pages with Tableau content.

For multiple values, enter the names in a comma-separated list where each value is encapsulated in double-quotes.

For example:

Tableau Server on Linux Administrator Guide

```
tsm authentication trusted configure -th "192.168.1.101",  
"192.168.1.102", "192.168.1.103"
```

or

```
tsm authentication trusted configure -th "webserv1", "web-  
serv2", "webserv3"
```

`-t, --token-length <integer>`

Optional.

Determines the number of characters in each trusted ticket. The default setting of 24 characters provides 144 bits of randomness. The value can be set to any integer between 9 and 255, inclusive.

Global options

`-h, --help`

Optional.

Show the command help.

`-p, --password <password>`

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

`-s, --server https://<hostname>:8850`

Optional.

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port 8850, and use the server name not the IP address. For example `https://<tsm_hostname>:8850`. If no server is specified, `https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see [Connecting TSM clients](#).

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm configuration

You can use the `tsm configuration` commands to get, set, and update configuration key values.

"Unknown key" responses

Certain configuration keys will return an "Unknown key" response when you attempt to get their current value, or set a new value. If this happens, verify that you have the key spelled correctly, including proper capitalization. To change the value, use the `--force-keys` option on the `tsm configuration set` command. For a list of configuration keys you can change, see [tsm configuration set Options](#).

- `tsm configuration get`

- `tsm configuration list-dynamic-keys`
- `tsm configuration set`

`tsm configuration get`

View the current server configuration and topology.

Synopsis

```
tsm configuration get --key <config.key> [global options]
```

Option

`-k, --key`

Required.

Get the current value of the specified configuration key.

`tsm configuration list-dynamic-keys`

View all the configuration keys that can be configured dynamically (without restarting Tableau Server).

Synopsis

```
tsm configuration list-dynamic-keys [global options]
```

`tsm configuration set`

Set or import server configuration or topology.

Quotes around the `<config.key>` and the `<config_value>` are optional unless there are spaces, in which case you must use quotes around the key or value.

Note: After setting a configuration key value you must apply the pending configuration changes using `tsm pending-changes apply`. Until you do, the new value will not be used by Tableau or show up in the results of a `tsm configuration get` command. You can view pending changes using `tsm pending-changes list`. For more information, see `tsm pending-changes`.

Synopsis

```
tsm configuration set --key <config.key> --value <config_value>
[global options]
```

Options

`-k, --key <config.key>`

Required.

Configuration key.

`-v, --value <config_value>`

Required.

Configuration value.

`-d`

Optional.

Reset the configuration value to its default.

`-frc, --force-keys`

Optional.

Force a key to be added to configuration even if it did not previously exist.

Global options

`-h, --help`

Optional.

Show the command help.

`-p, --password <password>`

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

`-s, --server https://<hostname>:8850`

Optional.

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port `8850`, and use the server name not the IP address. For example `https://<tsm_hostname>:8850`. If no server is specified, `https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see [Connecting TSM clients](#).

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm configuration set Options

Below is a list of configuration options or keys that you can set with the `tsm configuration set` command. In many cases you can find out the current value of a configuration key with the `tsm configuration get` command.

This list is not intended to be an exhaustive list of Tableau Server configuration settings. It represents a subset of configuration keys that can be set by server administrators. Finally, some keys used internally by Tableau Server do not appear in this list.

Note: Configuration keys are case-sensitive.

Basic Use of tsm configuration keys

Setting a configuration key

```
tsm configuration set -k <config.key> -v <config_value>
```

In some cases, you must include the `--force-keys` option to set a configuration value for a key that has not been set before. For more information, see "Unknown key" responses.

After setting a configuration key value you must apply the pending configuration changes using `tsm pending-changes apply`. Until you do, the new value will not be used by Tableau or show up in the results of a `tsm configuration get` command. You can view pending changes using `tsm pending-changes list`. For more information, see `tsm pending-changes`.

Resetting a configuration key to default

To reset a configuration key back to its default value, use the `-d` option:

```
tsm configuration set -k <config.key> -d
```

Tableau Server on Linux Administrator Guide

Viewing the current value of a configuration key

To see what a configuration key is currently set to, use the `configuration get` command:

```
tsm configuration get -k <config.key>
```

In certain cases you cannot get a configuration value for a key that has not been set before. Instead the `tsm configuration get` command will return an "Unknown key" response. For more information, see "Unknown key" responses.

Configuration Keys

`adminviews.disabled`

Default value: `false`

Disables access to the Tableau Administrative views. By default, access to views is enabled (this option is set to "false").

`api.server.enabled`

Default value: `true`

Allows access to the [Tableau Server REST API](#). By default, this functionality is enabled.

`auditing.enabled`

Default value: `true`

Allows access to the PostgreSQL (Tableau Server's own database) historical auditing tables.

`backgrounder.default_run_now_priority`

Default value (integer): 0

This setting controls what priority is assigned to run now jobs, with 0 being the highest priority. Values should be specified should be in the range of 0 – 100.

`backgrounder.enable_parallel_adsync`

Note: Added in version 2018.3.6

Default value: `false`

Controls whether parallel processing of external directory group synchronization jobs is allowed when there are multiple backgrounders. By default a scheduled synchronization of external directory groups is handled serially, by a single backgrounder. Set this to `true` to enable parallel processing on multiple backgrounder instances.

`backgrounder.externalquerycachewarmup.enabled`

Default value: `true`

Controls the caching of workbook query results after scheduled extract refresh tasks.

`backgrounder.externalquerycachewarmup.view_threshold`

Default value: `2.0`

The threshold for caching workbook query results after scheduled extract refresh tasks. The threshold is equal to the number of views that a workbook has received in the past seven days divided by the number of refreshes scheduled in the next seven days.

The following two *backgrounder* command options determine how long a flow task can run before the flow background task is canceled. These two commands together determine the total timeout value for flow tasks.

`backgrounder.extra_timeout_in_seconds`

Default value: `1800`

The number of seconds beyond the setting in `backgrounder.querylimit` before a background job is canceled. This setting makes sure that a stalled job does not hold up subsequent jobs. The setting applies to processes listed in `backgrounder.timeout_tasks`. 1800 seconds is 30 minutes.

This does not apply to subscription jobs.

`backgrounder.default_timeout.run_flow`

Default value: 14400

The number of seconds for a flow run task is canceled. 14,400 seconds is 4 hours.

`backgrounder.failure_threshold_for_run_prevention`

Default value: 5

The number of consecutive failures of a subscription, extract, or flow run job before that job is suspended. Suspending continuously failing jobs helps preserve backgrounder resources for other jobs. To disable suspension of failing background tasks, set this to `-1`.

`backgrounder.log.level`

Note: Added in version 2020.3.0

Default value: `info`

The logging level for the backgrounder process. This is dynamically configurable, so if you are only changing this you do not have to restart Tableau Server. For more information, see [Change Logging Levels](#).

`backgrounder.querylimit`

Default value: 7200

Longest allowable time, in seconds, for completing a single extract refresh job. 7200 seconds = 2 hours.

Note: If a background job reaches this time limit, it may continue to run for an additional several minutes while being canceled.

This does not apply to subscription jobs.

`backgrounder.restrict_serial_collections_to_site_level`

Default value: `false`

In Tableau Server, you can schedule extract refreshes, subscriptions, or flows to run periodically. These scheduled items are referred to as tasks. The Backgrounder process initiates unique instances of these tasks to run them at the scheduled time. The unique instances of the tasks that are initiated as a result are referred to as jobs.

This setting affects schedules that are configured to run serially. By default, when a schedule is configured to run serially, all jobs using that schedule will run serially. When this setting is set to `true`, jobs running on different sites can run in parallel. Jobs for scheduled tasks on the same site will continue to run serially.

The example below illustrate this scenario:

Tableau Server includes a schedule named "Daily" to run jobs every day at 7 am. The "Daily" schedule is configured to run serially. Site "HR" and site "Payroll" each have multiple scheduled tasks that use the schedule, "Daily". When this setting is set to `true`, jobs for these scheduled tasks on Site "HR" can run in parallel with jobs on site "Payroll", whereas jobs on the same site will still only run serially.

`backgrounder.notifications_enabled`

Default value: `true`

Controls whether extract refresh and flow run alerts are enabled for all sites on the server. By default alerts are enabled. To disable the alerts for all sites on a server, set this to `false`.

Extract alerts can be enabled or disabled on a site basis by site administrators in site settings, or at the user level in user settings.

`backgrounder.sort_jobs_by_type_schedule_boundary_heuristics_milliSeconds`

Default value: `60000`

Controls the time window that identifies backgrounder jobs which are determined to have the same scheduled start time.

The backgrounder process orders work that is scheduled at the same time to be executed by job type, running the fastest category of jobs first: Subscriptions, then Incremental Extracts, then Full Extracts.

Jobs are batched to determine which jobs are scheduled at the “same time”. A value 60,000 milliseconds (the default) indicates jobs for schedules starting within a 1 minute window should be classified in the same batch and so are ordered by type within that batch.

`backgrounder.subscription_failure_threshold_for_run_prevention`

Default value: 5

Determines the number of consecutive subscription failures that must occur before alerting for a condition is suspended. When set to the default of 5, alerting is suspended after 5 consecutive subscription failures. A value of `-1` will allow notification email to continue indefinitely. This threshold is server-wide, so applies to all subscriptions defined on the server.

`backgrounder.subscription_image_caching`

Default value: `true`

Controls whether backgrounder will cache images that are generated for subscriptions. Cached images do not have to be regenerated each time so caching improves subscription performance. By default image caching is enabled. To disable image caching for all sites on a server, set this to `false`.

`backgrounder.timeout_tasks`

The default list represents all the possible values for this setting, and differs by version of Tableau Server:

Default value: The default value may be different, depending on your version of Tableau Server. To see the default value list for your version of Tableau, run the `tsm configuration get` command:

```
tsm configuration get -k backgrounder.timeout_tasks
```

The list of tasks that can be canceled if they run longer than the combined values in `backgrounder.querylimit` and `backgrounder.extra_timeout_in_seconds`. The list of tasks is delimited with commas. The default list represents all the possible values for this setting.

```
backgrounder.timeout.sync_ad_group
```

Version: Added in version 2021.1.23, 2021.2.21, 2021.3.20, 2021.4.15, 2022.1.11, 2022.3.3.

Default Value: 14400 seconds (4 hours)

This is the maximum allowable time, specified in seconds, for completing an Active Directory group sync. This applies to *scheduled* group synchronizations done by the backgrounder service and prevents long-running syncs from running indefinitely. This does not impact group synchronizations done using the Tableau Server UI or the REST API.

```
backgrounder.timeout.single_subscription_notify
```

Note: Added in version 2021.2

Default Value: 1800 30 minutes

This is the maximum allowable time specified in seconds for completing a single subscription job.

```
backup.zstd.thread_count
```

Note: Added in version 2021.1.0. This key is dynamically configurable. For more information, see [Tableau Server Dynamic Topology Changes](#)

Default value: 2

The number of threads that should be used when creating a backup.

Tableau Server on Linux Administrator Guide

Increasing this number can improve backup performance, but we recommend thread count not exceed the number of logical processors on the Tableau Server computer, up to four.

`basefilepath.backuprestore`

Default value: `/var/opt/tableau/tableau_server-
/data/tabsvc/files/backups/`

The location in which the `tsm maintenance backup` command creates the backup. This is also the location where the backup file must be when restored using the `tsm maintenance restore` command or the `tsm maintenance send-logs` command. For more information, see [tsm File Paths](#).

`basefilepath.log_archive`

Default value: `/var/opt/tableau/tableau_server/data/tabsvc/files/log-
archives/`

The location in which the `tsm maintenance ziplogs` command creates the zipped archive. For more information, see [tsm File Paths](#).

`basefilepath.site_export.exports`

Default value: `/var/opt/tableau/tableau_server-
/data/tabsvc/files/siteexports/`

The location in which the `tsm sites export` command creates the export file. For more information, see [tsm File Paths](#).

`basefilepath.site_import.exports`

Default value: `/var/opt/tableau/tableau_server-
/data/tabsvc/files/siteimports/`

The location in which the `tsm sites import` command expects the import file to be located. For more information, see [tsm File Paths](#).

`clustercontroller.log.level`

Note: Added in version 2020.3.0

Default value: `info`

The logging level for Cluster Controller. This is dynamically configurable, so if you are only changing this you do not have to restart Tableau Server. For more information, see [Change Logging Levels](#).

`clustercontroller.zk_session_timeout_ms`

Default value: `300000`

The length of time, in milliseconds, that Cluster Controller will wait for the Coordination Service (ZooKeeper), before determining that failover is required.

`dataAlerts.checkIntervalInMinutes`

Default value: `60`

The frequency, in minutes, at which Tableau Server checks to determine if data-alert conditions are true.

(The server also checks whenever extracts related to data alerts are refreshed.)

`dataAlerts.retryFailedAlertsAfterCheckInterval`

Default value: `true`

Determines how often Tableau Server rechecks failing data alerts. When set to `true`, the server rechecks failing alerts at the frequency defined by `dataAlerts.checkIntervalInMinutes`. When set to `false`, the server rechecks failing alerts every five minutes, more quickly notifying alert recipients if data conditions have changed, but reducing server performance.

(The server also checks whenever extracts related to data alerts are refreshed.)

Tableau Server on Linux Administrator Guide

`dataAlerts.SuspendFailureThreshold`

Default value: 350

Determines the number of consecutive data alert failures that must occur before alerting for a condition is suspended. When set to the default of 350, alerting is suspended after roughly two weeks of alerts. This threshold is server-wide, so applies to any data alert defined on the server.

`databaseservice.max_database_deletes_per_run`

Default value: null

Use this option to adjust the maximum number of embedded external assets (databases and tables) that can be deleted each time the background process, controlled by `features.DeleteOrphanedEmbeddedDatabaseAsset`, runs. If this option is left empty, the default maximum number of embedded external assets that can be deleted is 100.

For more information, see `features.DeleteOrphanedEmbeddedDatabaseAsset`.

This option was added beginning with Tableau Server version: 2021.2.

`dataserver.log.level`

Note: Added in version 2020.3.0

Default value: `info`

The logging level for Data Server. This is dynamically configurable, so if you are only changing this you do not have to restart Tableau Server. For more information, see [Change Logging Levels](#).

`elasticserver.vmopts`

Default value: `"-Xmx<default_value> -Xms<default_value>"`

The default value varies based on the amount of system memory. The JVM maximum heap size is scaled to be 3.125% of the total system RAM.

Controls the Elastic Server heap size. Because the default value scales automatically, use this option to override the default value only when absolutely necessary. Append the letter 'k' to the value to indicate kilobytes, 'm' for megabytes, or 'g' to indicate gigabytes. As a general rule, set initial heap size (`-Xms`) equal to the maximum heap size (`-Xmx`) to minimize garbage collections.

This option was added beginning with Tableau Server version: 2019.1.

`excel.shadow_copy_all_remote.enabled`

Default value: `false`

Controls whether Tableau Server creates a "shadow copy" of a shared Excel spreadsheet (`.xlsx` or `.xlsm`) that is being used as a live data source. When enabled, this option prevents Excel users from seeing a "Sharing Violation Error" and a message that the file is "currently in use." This option can have a performance impact with large Excel files. If Excel users do not need to edit the shared file, you do not need to enable this option.

Note: Tableau Server always attempts to create a shadow copy of a `.xls` file. This option does not change that behavior.

This option was added beginning with Tableau Server versions: 2019.1.5, 2019.2.1.

`features.ActiveMQ`

Default value: `true`

Controls whether Tableau Server uses the Apache ActiveMQ service (Tableau Server Messaging Service) for the internal messaging mechanism.

This option was added beginning with Tableau Server version: 2019.4.

`features.DeleteOrphanedEmbeddedDatabaseAsset`

Default value: `true`

Tableau Server on Linux Administrator Guide

Controls a background process, for Tableau Catalog (or Tableau Metadata API), that deletes embedded external assets (databases and tables) that are no longer associated with downstream Tableau content. This process runs everyday at 22:00:00 UTC (coordinated universal time) and can delete a maximum of 100 external assets each day until there are no remaining external assets without connections to downstream Tableau content. You can set this option to `false` to stop this process from running. Alternatively, you can also adjust the maximum number of external embedded assets that can be deleted using `dataservice.max_database_deletes_per_run`.

For more information see, [Troubleshoot missing content](#).

This option was added beginning with Tableau Server version: 2021.2.

`features.DesktopReporting`

Default value: `false`

Controls whether Desktop License Reporting is enabled on the server. When set to `false` (the default), no Administrative Views related to desktop licenses are available. Set this to `true` to enable license reporting and to make license usage and expiration Administrative Views visible on the Server Status page. **Note:** Desktop License Reporting must be enabled on the client (Tableau Desktop) in order for information to be reported to Tableau Server.

`features.MessageBusEnabled`

Default value: `true`

Controls whether Tableau Server uses the new internal messaging mechanism.

This option was added beginning with Tableau Server version: 2019.4.

`features.PasswordlessBootstrapInit`

Default value: `true`

Controls whether Tableau Server allows embedded credentials in bootstrap files. When enabled (the default), embedded credentials are included in the bootstrap file unless you

specify that they should not be included. Set this to `false` if credentials should never be included in any bootstrap file you generate. For more information on generating bootstrap files, see `tsm topology nodes get-bootstrap-file`.

This option was added beginning with Tableau Server version 2019.3.

`features.PasswordReset`

Default value: `false`

Applies only to servers that use local authentication. Set to `true` to let users reset their passwords with a "Forgot password" option on the sign-in page.

`filestore.empty_folders_reaper.enabled`

Note: Added in 2020.x (2020.1.14, 2020.2.11, 2020.3.6, 2020.4.2) and 2021.1.x. The default value was changed to `true` in 2021.2.

Default value: `true`

Enables the job that "reaps" (removes) empty Filestore folders.

`filestore_empty_folders_reap.frequency`

Note: Added in 2020.x (2020.1.14, 2020.2.11, 2020.3.6, 2020.4.2). This is not yet available in 2021.1.

Default value: `86400` (24 hours)

Specifies in minutes, how often to run the job that removes empty Filestore folders.

`filestore.log.level`

Note: Added in version 2020.3.0

Default value: `info`

Tableau Server on Linux Administrator Guide

The logging level for File Store. This is dynamically configurable, so if you are only changing this you do not have to restart Tableau Server. For more information, see [Change Logging Levels](#).

`filestore.reapemptyfoldersholdoffms`

Note: Added in 2020.x (2020.1.14, 2020.2.11, 2020.3.6, 2020.4.2). This is not yet available in 2021.1.

Default value: 300000 (5 minutes)

Specifies in milliseconds, the amount of time to wait before removing empty Filestore folders.

`floweditor.max_datafile_upload_size_in_kb`

Note: Added in version 2020.4

Default value: 1048576

For Tableau Prep flow web authoring, the maximum size of delimited text files (for example, CSV or TXT) that can be uploaded to Tableau Server.

`gateway.http.cachecontrol.updated`

Default value: `false`

The Cache-Control HTTP header specifies whether the client browser should cache content sent from Tableau Server. To disable caching of Tableau Server data on the client, set this option to `true`.

`gateway.http.hsts`

Default value: `false`

The HTTP Strict Transport Security (HSTS) header forces browsers to use HTTPS on the domain where it is enabled.

`gateway.http.hsts_options`

Default value: `"max-age=31536000"`

By default, HSTS policy is set for one year (31536000 seconds). This time period specifies the amount of time in which the browser will access the server over HTTPS.

`gateway.http.request_size_limit`

Default value: `16380`

The maximum size (bytes) of header content that is allowed to pass through the Apache gateway on HTTP requests. Headers that exceed the value set on this option will result in browser errors, such as HTTP Error 413 (Request Entity Too Large) or authentication failures.

A low value for `gateway.http.request_size_limit` can result in authentication errors. Single sign-on solutions that integrate with Active Directory (SAML and Kerberos) often require large authentication tokens in HTTP headers. Be sure to test HTTP authentication scenarios before deploying into production.

We recommend setting `tomcat.http.maxrequestsize` option to the same value that you set for this option.

`gateway.http.x_content_type_nosniff`

Default value: `true`

The X-Content-Type-Options response HTTP header specifies that the MIME type in the Content-Type header should not be changed by the browser. In some cases, where MIME type is not specified, a browser may attempt to determine the MIME type by evaluating the characteristics of the payload. The browser will then display the content accordingly. This process is referred to as "sniffing." Misinterpreting the MIME type can lead to security vulnerabilities. The X-Content-Type-Options HTTP header is set to 'nosniff' by default with this option.

`gateway.http.x_xss_protection`

Default value: `true`

The HTTP X-XSS-Protection response header is sent to the browser to enable cross-site scripting (XSS) protection. The X-XSS-Protection response header overrides configurations in cases where users have disabled XSS protection in the browser. The X-XSS-Protection response header is enabled by default with this option.

gateway.log.level

Note: Added in version 2020.3.0

Default value: `info`

The logging level for Gateway. This is dynamically configurable, so if you are only changing this you do not have to restart Tableau Server. For more information, see [Change Logging Levels](#).

gateway.public.host

Default value: `<hostname>`

The name (URL) of the server, used for external access to Tableau Server. If Tableau Server is configured to work with a proxy server or external load balancer, it is the name entered in a browser address bar to reach Tableau Server. For example, if Tableau Server is reached by entering `tableau.example.com`, the name for `gateway.public.host` is `tableau.example.com`.

gateway.public.port

Default value: `80` (`443` if SSL)

Applies to proxy server environments only. The external port the proxy server listens on.

gateway.slow_post_protection.enabled

Default value: `false`

Enabling this can provide some help in protecting against slow POST (Denial-of-Service) attacks by timing out POST requests that transfer data at extremely slow rates.

Note: This will not eliminate the threat of such attacks, and could have the unintended impact of terminating slow connections.

gateway.slow_post_protection.request_read_timeout

Default value: header=15-20,MinRate=500 body=10,MinRate=500

When enabled by the preceding option, `gateway.slow_post_protection.enabled`, this option sets the Apache `httpd` `ReadRequestTimeout`. The `httpd` directive is documented at [Apache Module `mod_reqtimeout`](#). The primary use of this option is as a defense the Slowloris attack. See the Wikipedia entry, [Slowloris \(computer security\)](#).

gateway.timeout

Default value: 7200

Longest amount of time, in seconds, that the gateway will wait for certain events before failing a request (7200 seconds = 2 hours).

gateway.trusted

Default value: IP address of proxy server machine

Applies to proxy server environments only. The IP address(es) or host name(s) of the proxy server.

gateway.trusted_hosts

Default value: Alternate names of proxy server

Applies to proxy server environments only. Any alternate host name(s) for the proxy server.

hyper.file_partition_size_limit

Default value: 0

When set to 0, the size is set to unlimited and will use all the disk space that is available.

This option is used to set the disk space limit for a query that spools to disk. If your disk space usage by the `spool.<id>.tmp` file is higher than where you need it to be for your environment, it means that queries are spooling and taking up disk space. Use this option to limit the amount of disk space that any one query can use. The `spool.<id>.tmp` file can be found in the temp folder of the user account running Tableau Server. You can specify this value in K(KB), M(MB), G(GB), or T(TB) units. For example, you can specify the size limit as 100G when you want to limit the disk space usage to 100 GB.

For more information about spooling see the Memory and CPU Usage section in Tableau Server Data Engine.

`hyper.global_file_partition_size_limit`

Default value: 0

When set to 0, the size is set to unlimited and will use all the disk space that is available.

This option is used to set the disk space limit for all queries that spool to disk. If your disk space usage by the `spool.<id>.tmp` file is higher than where you need it to be for your environment, it means that queries are spooling and taking up disk space. The `spool.<id>.tmp` file can be found in the temp folder of the user account running Tableau Server. Use this option to limit the amount of disk space in sum total that all queries use when spooling to disk. You can specify this value in K(KB), M(MB), G(GB), or T(TB) units. For example, you can specify the size limit as 100G when you want to limit the disk space usage to 100 GB. Tableau recommends that you start with this configuration when fine tuning your spooling limits.

For more information about spooling see the Memory and CPU Usage section in Tableau Server Data Engine.

`hyper.log_queries`

Default value: `true`

When set to true, query information is logged.

By default query information is logged. If however you find that the log files are too large for the amount of disk space available, you can set it to `false` to disable logging query information. Tableau recommends leaving this configuration set to `true`.

`hyper.log_query_cpu`

Default value: `false`

Use this setting to log how much time each query takes and the CPU usage.

`hyper.log_timing`

Default value: `false`

This setting is useful to find out more information about the queries, like compilation and parsing times. By default this setting is disabled. You can turn this by setting the value to `true` to collect more details about your queries. Note, however that this will increase the size of your data engine log files (`\logs\hyper`).

`hyper.log_troublesome_query_plans`

Default value: `true`

When set to `true`, logs query plans of query that are identified as problematic. Queries that are either canceled, running slower than 10 seconds, or if the queries are spooling to disk fall into this category. The information in the logs can be useful to troubleshoot problematic queries. You can change the setting to `false` if you are concerned about the size of the logs.

`hyper.memory_limit`

Default value: `80%`

Controls the maximum amount of memory used by Hyper. Specify the number of bytes. Append the letter 'k' to the value to indicate kilobytes, 'm' to indicate megabytes, 'g' to indicate gigabytes, or 't' to indicate terabytes. For example, `hyper.memory_limit='7g'`. Alternatively, specify the memory limit as a percentage of the overall available system memory. For example, `hyper.memory_limit='90%'`.

Tableau Server on Linux Administrator Guide

`hyper.memtracker_hard_reclaim_threshold`

Default value: 80%

This setting only applies to Windows. Hyper keeps decompressed and decrypted parts of the extract in memory to make subsequent accesses faster. This setting controls when worker threads will start writing this data out to a disk cache to reduce memory pressure. If given as a percentage, the value is interpreted as a percentage of the overall `hyper.memory_limit` setting. For example, `hyper.memtracker_hard_reclaim_threshold='60%'`. Absolute values can be specified as 'k' (kilobytes), 'm' (megabytes), 'g' (gigabytes), or 't' (terabytes). For example, `hyper.memtracker_hard_reclaim_threshold='10g'`. The value should be larger than the `hyper.memtracker_soft_reclaim_threshold`.

`hyper.memtracker_soft_reclaim_threshold`

Default value: 50%

This setting only applies to Windows. When interacting with a Hyper file, Hyper will write out some data for caching or persisting the data. Windows has the special behavior that it locks freshly written data into memory. To avoid swapping, we force out the data when Hyper reaches the configured limit for the reclaim threshold. When the soft reclaim threshold is reached, Hyper will try to reclaim cached data in the background to attempt to stay below the reclaim threshold. In situations where swapping would happen otherwise, triggering reclamation in Hyper can lead to a better outcome. Therefore, if your Tableau Server installation experiences a lot of swapping, this setting can be used to attempt to reduce the memory pressure.

Specify the number of bytes. Append the letter 'k' to the value to indicate kilobytes, 'm' to indicate megabytes, 'g' to indicate gigabytes, or 't' to indicate terabytes. Alternatively, specify the value as a percentage of the overall configured memory for Hyper. For example, `hyper.memtracker_soft_reclaim_threshold='20%'`.

`hyper.network_threads`

Default value: 150%

Controls the number of network threads used by Hyper. Specify either the number of network threads (for example, `hyper.network_threads=4`) or specify the percentage of threads in relation to the logical core count (for example, `hyper.network_threads='300%'`).

Network threads are used for accepting new connections and sending or receiving data and queries. Hyper uses asynchronous networking, so many connections can be served by a single thread. Normally, the amount of work that is done on network threads is very low. The one exception is opening databases on slow file systems, which can take a long time and block the network thread. If connection times are slow when you try to view or edit dashboards that use extracts and have not been used in a while and you frequently see “asio-continuation-slow” messages in the Hyper log and long “construct-protocol” times to Hyper in the Tableau log, try to increase this value.

`hyper.objectstore_validate_checksums`

Default value: `false`

A boolean setting that controls file integrity checks in Hyper. When set to `true`, Hyper will check the data in an extract file when it is first accessed. This allows silent corruption and corruption that would crash Hyper to be detected. In general, it is advisable to turn this setting on except for installations with very slow disks where it could cause performance regressions.

`hyper.query_total_time_limit`

Default value: `0` (which means unlimited)

Sets an upper bound on the total thread time that can be used by individual queries in Hyper. Append 's' to the value to indicate seconds, 'min' to indicate minutes, or 'h' to indicate hours. For example, to restrict all queries to a total time usage of 1500 seconds of total thread time: `hyper.query_total_time_limit='1500s'`. If a query runs longer than the specified limit, the query will fail and an error will be returned.

This setting allows you to automatically control runaway queries that would otherwise use too many resources. Hyper executes queries in parallel. For example, if a query executes

for 100 seconds and during this time is running on 30 threads, the total thread time would be 3000 seconds. The thread time of each query is reported in the Hyper log in the “query-end” log entries in the “total-time” field.

`hyper.session_memory_limit`

Default value: 0 (which means unlimited)

Controls the maximum memory consumption that an individual query can have. Specify the number of bytes. Append the letter 'k' to the value to indicate kilobytes, 'm' to indicate megabytes, 'g' to indicate gigabytes, or 't' to indicate terabytes. For example, `hyper.session_memory_limit='900m'`. Alternatively, specify the session memory limit as a percentage of the overall available system memory. For example, `hyper.session_memory_limit='90%'`.

Lowering this value can help when a query is using excessive amounts of memory and making other queries fail over a long period of time. By lowering the limit, the single big query would fail (or resort to spooling if spooling isn't turned off) and not have a negative impact on other queries.

`hyper_standalone.consistent_hashing.enabled`

Default value: `true`

Improves the chance that the extract for a query is already cached. If the node with the extract cached cannot support additional load, you will be routed to a new node and the extract will be loaded into cache on the new node. This results in better system utilization because extracts are only loaded into memory if there is load that justifies the need.

`hyper_standalone.health.enabled`

Default value: `true`

Switches the load balancing metric from random selection to picking the Data Engine (Hyper) node based on a health score that is made of up of a combination of current Hyper activity and system resource usage. Based on these values, the load balancer will pick the node that is most capable of handling an extract query.

`hyper.temp_disk_space_limit`

Default value: 100%

Sets the upper limit of disk space at which Hyper will stop allocating space for temporary files. This setting can help to stop the hard disk from filling up with temporary files from Hyper and running out of disk space. If disk space reaches this threshold, Hyper will attempt to recover automatically without administrator intervention.

Specify it as percentage of the overall available disk space to be used. For example, `hyper-temp_disk_space_limit='96%'`. When set to 100%, all of the disk space that is available can be used.

For Data Engine to start, the configured amount of disk space must be available. If not enough disk space is available, you will see a Data Engine log entry that says, “Disk limit for temporary files has been reached. Please free up disk space on the device. See the Hyper log for more information: No space left on device”.

`hyper.hard_concurrent_query_thread_limit`

Default value: 150%

Use this option to set the maximum number of threads Hyper should use for running queries. Use this when you want to set a hard limit on the CPU usage. Specify either the number of threads or specify the percentage of threads in relation to the logical core count. Hyper will most likely not use more resources than are configured by this setting but Hyper background and network threads are not affected by this setting (though they tend to not be CPU intensive).

It is important to consider that this setting controls the number of concurrent queries that can be executed. So, if you decrease this setting, the chance of queries needing to wait for currently running queries to complete increases, which may affect workbook load times.

`hyper.soft_concurrent_query_thread_limit`

Default value: 100%

Use this option to specify the number of threads that a single query can be parallelized across if sufficiently many threads are available given the `hyper.concurrent_query_thread_limit` setting. Specify either the number of threads or specify the percentage of threads in relation to the logical core count.

To illustrate this, here is a simplified example:

Let's say you set this value to 10 threads, this means queries can be parallelized up to 10 threads. If only 2 queries are running, the remaining 8 threads are used to parallelize the 2 queries.

The `hyper.hyper.concurrent_query_thread_limit`, and `hyper.hyper.soft_concurrent_query_thread_limit` options work together to give you some options to manage your CPU usage while maximizing available CPU resources to complete queries faster. If you don't want the Data Engine to use all the available CPU on the machine, change it to less than 100% to a percentage that is optimal for your environment. The soft limit is a way for you to limit CPU usage but allow it to go beyond the soft limit up to the hard limit if necessary.

Note: The `hyper.hyper.concurrent_query_thread_limit` and `hyper.hyper.soft_concurrent_query_thread_limit` options replace `hyper.num_job_worker_threads` and `hyper.num_task_worker_threads` options available in Tableau Server versions 2018.3 and earlier, and are deprecated in the current version. For information on the `hyper.num_job_worker_threads` and `hyper.num_task_worker_threads`, see [tsm configuration set Options](#).

`hyper.use_spooling_fallback`

Default value: `true`

When set to `true`, it allows spooling to disk when querying extracts exceeds set RAM usage (80% of installed RAM). In other words, it allows Hyper to execute a query using the disk if it exceeds RAM usage.

Tableau recommends that you use the default setting. You can turn this off by setting the value to `false` if you are concerned about disk usage. If you turn this setting off, queries that use more than 80% of installed RAM will be canceled. Spooling queries usually take substantially longer to finish.

For more information about spooling see the Memory and CPU Usage section in Tableau Server Data Engine.

`java.heap.size`

Default value: 128m

Size of heap for Tomcat (repository and solr). This generally does not need to change except on advice from Tableau.

`licensing.login_based_license_management.default_requested_duration_seconds`

Default value: 0

Set to the duration (in seconds) that a user's login-based license can be offline with no connection to Tableau Server before they are prompted to activate again. This duration is always refreshed when Tableau Desktop is in use and can connect to Tableau Server.

`licensing.login_based_license_management.enabled`

Default value: `true`

Set to `true` to enable login-based license management. Set to `false` to disable login-based license management.

Note: In order to use login-based license management, you must activate a product key that is enabled for login-based license management. You can use the `tsm licenses list` to see which product keys have login-based license management enabled.

Tableau Server on Linux Administrator Guide

licensing.login_based_license_management.max_requested_duration_seconds

Default value: 15552000

Set to the maximum duration (in seconds) that a user's login-based license can be offline with no connection to Tableau Server before they are prompted to activate Tableau again. The maximum value is 15552000 seconds (180 days). This duration is always refreshed when Tableau Desktop is in use and can connect to Tableau Server.

maestro.app_settings.sampling_max_row_limit

Default value: 1000000

Sets the maximum number of rows for sampling data from large data sets with Tableau Prep on the web.

maestro.input.allowed_paths

Default value: ""

By default, access to any directory will be denied, and only publishing to Tableau Server with content that is included in the tflx file is allowed.

A list of allowed network directories for flow input connections. You must enable Tableau Prep Conductor to schedule flows on your Tableau Server. For more information, see Tableau Prep Conductor.

The following rules apply and must be considered when configuring this setting:

- Paths should be accessible by Tableau Server. These paths are verified during server startup and at flow run time.
- Network directory paths have to be absolute and cannot contain wildcards or other path traversing symbols. For example `\\myhost\myShare*` or `\\my-host\myShare*` are invalid paths and would result in all the paths as disallowed. The correct way to safelist any folder under *myShare* would be `\\myhost\myShare` or `\\myhost\myShare\`.

Note: The `\\myhost\myShare` configuration will not allow `\\my-host\myShare1`. In order to safe list both of these folders one would have safe list them as `\\myhost\myShare; \\myhost\myShare1`.

- The value can be either `*` meaning that any path, including local (with the exception of some system paths configured using “`native_api.internal_disallowed_paths`”), or a list of paths, delimited by “`;`”.

Note: If a path is both on the flows allowed list and `internal_disallowed` list, `internal_disallowed` takes precedence.

Important:

This command overwrites existing information and replaces it with the new information you provided. If you want to add a new location to an existing list, you must provide a list of all the locations, existing and the new one you want to add. Use the following commands to see the current list of input and output locations:

```
tsm configuration get -k maestro.input.allowed_paths
tsm configuration get -k maestro.output.allowed_paths
```

For more information and details about configuring allowed directories for flow input and output connections, see [Safe list Input and Output Locations](#).

```
maestro.output.allowed_paths
```

Default value: ""

By default, access to any directories will be denied.

A list of allowed network directories for flow output connections. You must enable Tableau Prep Conductor to schedule flows on your Tableau Server. For more information, see [Tableau Prep Conductor](#).

The following rules apply and must be considered when configuring this setting:

- Paths should be accessible by Tableau Server. These paths are verified during server startup and at flow run time.
- Network directory paths have to be absolute and cannot contain wildcards or other path traversing symbols. For example `\\myhost\myShare*` or `\\my-host\myShare*` are invalid paths and would result in all the paths as disallowed. The correct way to safelist any folder under *myShare* would be `\\myhost\myShare` or `\\myhost\myShare\`.

Note: The `\\myhost\myShare` configuration will not allow `\\my-host\myShare1`. In order to safe list both of these folders one would have safe list them as `\\myhost\myShare; \\myhost\myShare1`.

- The value can be either `*` meaning that any path, including local (with the exception of some system paths configured using “`native_api.internal_disallowed_paths`”), or a list of paths, delimited by “;”.

Note: If a path is both on the flows allowed list and `internal_disallowed` list, `internal_disallowed` takes precedence.

For more information and details about configuring allowed directories for flow input and output connections, see [Safe list Input and Output Locations](#).

`maestro.sessionmanagement.maxConcurrentSessionPerUser`

Default value: 4

Sets the maximum number of flow web editing sessions that a user can have open at one time.

metadata.ingestor.blocklist

Default value: null

When configured, Tableau Catalog blocks specified content from being ingested. To specify which content to block, you must identify the blocklist values, which is a combination of both the site ID, content type, and content ID of the content you want to block, from the server “noninteractive” log files. Blocklist values must be separated by a comma.

Important: You should only use this option when directed to do so by Tableau Support.

For example, you can use the `tsm configuration set --force-keys -k metadata.ingestor.blocklist` to block ingestion of a combination of data sources, workbooks, and flows using the following command:

```
tsm configuration set --force-keys -k metadata.ingestor.blocklist -v "sites/1/datasources/289, sites/2/datasources/111, sites/1/workbooks/32, sites/3/workbooks/15, sites/1/flows/13, sites/1/flows/18"
```

To validate blocked content, review the server “noninteractive” log files for the following events:

- Skipping ingestion for
- Successfully updated blocklist to

For example:

```
Skipping ingestion for contentType [Workbook], contentId [sites/1/datasources/289], siteDisabled [false], swallowEvent [false], contentBlocked [true]
```

```
Skipping ingestion for contentType [Workbook], contentId [sites/3/workbooks/15], siteDisabled [false], swallowEvent [false], contentBlocked [true]
```

and

Tableau Server on Linux Administrator Guide

```
Successfully updated blocklist to: [sites/1/datasources/289,  
sites/1/workbooks/32, sites/2/datasources/111]
```

`metadata.ingestor.pipeline.throttleEventsEnable`

Default value: `false`

Controls whether indexing of new and updated content, also called eventing, is regulated across all sites on the server. By default, event throttling is turned off. To turn on event throttling, change this setting to `true` using the following command:

```
tsm configuration set -k metadata.in-  
gestor.pipeline.throttleEventsEnable -v true --force-keys
```

For more information about event throttling, see [Enable Tableau Catalog](#).

`metadata.ingestor.pipeline.throttleLimit`

Default value: `20`

When event throttling is enabled, this is the maximum number of new and updated content items that can be indexed during a specified period of time. Once the specified limit is reached for a specific item, indexing is deferred.

By default, the limit is set to `20` and can't be set to lower than `2`. You can use the following command to change the limit:

```
tsm configuration set -k metadata.in-  
gestor.pipeline.throttleLimit -v 25 --force-keys
```

Throttled events can be identified in the server "noninteractive" log files as `ingestor event flagged for removal by throttle filter`.

`metadata.ingestor.pipeline.throttlePeriodLength`

Default value: `20`

When event throttling is enabled, this is the period of time, in minutes, a specified maximum number of new and updated content items can be indexed. Once the specified time is reached, indexing of any additional new and updated content is deferred.

By default, the time is set to 30 minutes. You can use the following command to change the time:

```
tsm configuration set -k metadata.indexer.pipeline.throttlePeriodLength -v PT45M --force-keys
```

`metadata.query.limits.time`

Default value: 20

This is the longest allowable time, in seconds, for a Catalog or Metadata API query to run before a timeout occurs and the query is canceled. Tableau recommends incrementally increasing the timeout limit to *no more than* 60 seconds using the following command:

```
tsm configuration set -k metadata.query.limits.time -v PT30S --force-keys
```

Important: This option should be changed only if you see the error described here, Timeout limit and node limit exceeded messages. Increasing the timeout limit can utilize more CPU for longer, which can impact the performance of tasks across Tableau Server. Increasing the timeout limit can also cause higher memory usage, which can cause issues with the interactive microservices container when queries run in parallel.

`metadata.query.limits.count`

Default value: 2000

This is the number of objects (which can loosely map to the number of query results) that Catalog can return before the node limit is exceeded and the query is canceled. Tableau recommends incrementally increasing the timeout limit, to *no more than* 100,000 using the following command:

Tableau Server on Linux Administrator Guide

```
tsm configuration set -k metadata.query.limits.count -v 3000 --force-keys
```

Important: This option should be changed only if you see the error described here, Timeout limit and node limit exceeded messages. Increasing the node limit can cause higher memory usage, which can cause issues with the interactive microservices container when queries run in parallel.

metricservices.checkIntervallInMinutes

Default value: 60

Controls the interval, in minutes, between refreshes for metrics that rely on live data sources. A metric refreshes when the server checks for new data via the metric's connected view.

metricservices.failureCountToWarnUser

Default value: 10

Controls the number of consecutive refresh failures that must occur before the metric owner is warned. When set to the default of 10, a metric refresh must fail 10 times in a row before the owner is sent a notification about the failure.

metricservices.maxFailedRefreshAttempts

Default value: 175

Controls the number of consecutive refresh failures that must occur before a metric refresh is suspended.

mobile.deep_linking.on_prem.enabled

Default value: true

Controls whether links to Tableau Server are treated as deep links by the Tableau Mobile app. When set to `true`, links to supported content types open in the app. When set to `false`, links open in the mobile browser. For more information see, [Control deep linking for Tableau Mobile](#).

`monitoring.dataengine.connection_timeout`

Default value: `30000`

The length of time, in milliseconds, that Cluster Controller will wait for the data engine, before determining that a connection timeout occurred. The default is 30,000 milliseconds (30 seconds).

`native_api.connection.limit.<connection class>`

Set parallel query limit for the specified data source (connection class). This overrides the global limit for the data source.

`native_api.connection.globallimit`

Default value: `16`

Global limit for parallel queries. Default is 16 except for Amazon Redshift which has a default of 8.

`native_api.ExplainDataEnabled`

Default value: `true`

This option controls whether Explain Data is enabled or disabled for the server. For more information about Explain Data, see [Get Started with Explain Data](#) in the Tableau Help.

This option was added beginning with Tableau Server version: 2019.3.

`native_api.force_alternative_federation_engine`

Default value: `false`

Override the operation restrictions when joining data from a single file connection and a single SQL database connection. Set this option to `True` to force Tableau to process the join using the live database connection.

`native_api.ProtocolTransitionLegacyFormat`

Default value: `false`

Tableau Server on Linux Administrator Guide

Use the legacy name format for constrained delegation.

The name format was changed in version 10.1 to allow cross-domain protocol transition (S4U). If this causes problems with existing configurations and you don't need cross-domain protocol transition, configure Tableau Server to use the old behavior by setting this to `true`.

`native_api.unc_mountpoints`

Default value: none

Specifies UNC and FQDN path for shared Windows directories that are accessed by Tableau Server on Linux. Each path must also be referenced in a corresponding `auto.cifs` file. Separate each path by a semicolon, for example:

```
'//filesrv01/development;/mnt/filesrv01/development;filesrv01.example.lan/development;/mnt/filesrv01/development'
```

Subsequent updates to the `native_api.unc_mountpoints` value will overwrite the existing value. Therefore, each time you add a Windows share, you must include all shares in the updated value.

For more information, see the Community wiki topic, [Connecting to a Windows Shared Directory](#).

`native_api.InitializeQueryCacheSizeBasedOnWeights`

Default value: `True`

Controls whether the query cache size is initialized automatically based on the amount of available system memory. The query cache consists of the logical query cache, metadata cache, and native query cache. By default, this functionality is enabled.

`native_api.QueryCacheMaxAllowedMB`

The maximum size of the query cache in megabytes. This value varies based on the amount of system memory. The query cache consists of the logical query cache, metadata cache, and native query cache. Use the table below to determine your default value:

System Memory	Default Value for Tableau Server	Default Value for Tableau Desktop
64 GB and more	3200 MB	1600 MB
From 32 GB to 64 GB	2400 MB	1200 MB
From 16 GB to 32 GB	1600 MB	800 MB
16 GB and less	800 MB	400 MB

`native_api.LogicalQueryCacheMaxAllowedWeight`

Default value: 70

The weight of logical query cache size limit in the total query cache size.

`native_api.MetadataQueryCachMaxAllowedWeight`

Default value: 4

The weight of metadata query cache size limit in the total query cache size.

`native_api.NativeQueryCacheMaxAllowedWeight`

Default value: 26

The weight of native query cache size limit in the total query cache size.

`native_api.QueryCacheEntryMaxAllowedInPercent`

Default value: 60%

Specifies the maximum size of query results that can be put into the query cache. It is set as the percentage of the total query cache size. For example, if the logical query cache size is 100 MB and `native_api.QueryCacheEntryMaxAllowedInPercent` is set to 60 percent, then only query results that are smaller than 60 MB can be put into the logical query cache.

`nlp.concepts_shards_count`

Default value: 1

Note: The default shard count value is sufficient for most Tableau Server installations.

Controls the number of data shards for the Elastic Search Concepts index that stores field names, field synonyms, and analytical terms. The shard count partitions the search index to reduce total index size, which may improve the performance of Ask Data's semantic parser. Adjusting the shard count is another performance enhancement measure that you can take along with increasing the heap size through `elasticsearchserver.vmopts`.

Tableau recommends increasing the shard count by 1 for every 50 GB. To reduce the number of times you need to adjust the shard count, calculate the total index size by adding 50% to the current index. For example, if the total index size is less than 50 GB, then 1 shard is sufficient. Actual performance will vary depending on the server, the rate at which the index size grows, and other factors.

- 0 to 50 GB: 1
- 50 GB to 100 GB: 2
- 100 GB to 150 GB: 3

You can use the following command to increase the Concepts index shard count from default to 2:

```
tsm configuration set -k nlp.concepts_shards_count -v 2
```

```
nlp.values_shards_count
```

Default value: 1

Controls the number of data shards for the Elastic Search Values index that stores values, value synonyms, and aliases. The shard count partitions the search index to reduce total index size, which may improve the performance of Ask Data's semantic parser. Adjusting the shard count is another performance enhancement measure that you can take along with increasing the heap size through `elasticsearch.vmopts`.

Tableau recommends increasing the shard count by 1 for every 50 GB. To reduce the number of times you need to adjust the shard count, calculate the total index size by adding 50% to the current index. For example, if the total index size is less than 50 GB, then 1 shard is sufficient. Actual performance will vary depending on the server, the rate at which the index size grows, and other factors.

- 0 to 50 GB: 1
- 50 GB to 100 GB: 2
- 100 GB to 150 GB: 3

You can use the following command to increase the Values index shard count from default to 2:

```
tsm configuration set -k nlp.values_shards_count -v 2
```

```
nlp.defaultNewSiteAskDataMode
```

Default value: `disabled_by_default`

Use this option to set the initial value of the Ask Data Mode when a site is created. For more information see [Disable or Enable Ask Data for a Site](#).

Valid options are `disabled_by_default` and `disabled_always`.

Tableau Server on Linux Administrator Guide

This option was added beginning with Tableau Server versions: 2019.4.5, 2020.1.3.

`noninteractivecontainer.vmopts`

Default value: `"-XX:+UseConcMarkSweepGC -Xmx<default_value>g -XX:+ExitOnOutOfMemoryError"`

The default value varies based on the amount of system memory. The JVM maximum heap size is scaled to be 6.25% of the total system RAM.

This option controls the JVM maximum heap size for Tableau Catalog ingestion. Because the default value scales automatically, use this option to override the default value only when absolutely necessary by modifying the `-Xmx<default_value>g` argument. For example, you can use the following command to increase the max heap size to 2 GB:

```
tsm configuration set -k noninteractive.vmopts -v "-XX:+UseConcMarkSweepGC -Xmx2g -XX:+ExitOnOutOfMemoryError"
```

For more information, see [Memory for non-interactive microservices containers](#).

`pgsql.port`

Default value: 8060

Port that PostgreSQL listens on.

`pgsql.preferred_host`

Specifies the computer name of the node with the preferred repository installed. This value is used if the `--preferred` or `-r` option is specified with the `tsm topology failover-repository` command.

Example:

```
tsm configuration set -k pgsql.preferred_host -v "<host_name>"
```

Note: The `host_name` is case-sensitive and must match the node name shown in the output of `tsm status -v`.

`pgsql.verify_restore.port`

Default value: `8061`

Port used to verify the integrity of the PostgreSQL database. See `tsm maintenance backup` for more information.

`ports.blocklist`

Note: Added in version 2021.1

Default value: no ports blocked in the range used for automatic port assignment.

Used to specify ports within the port assignment range that should not be used by Tableau when dynamically assigning ports. This is useful when you know that another application is using a port within the range. Separate multiple ports with commas, for example:

```
tsm configuration set -k ports.blocklist -v 8000,8089, 8090
```

For more information on using the `ports.blocklist` key, see [Blocking specific ports within the range](#)

`recommendations.enabled`

Default value: `true`

Controls the recommendations feature, which powers recommendations for data sources and tables (for Tableau Desktop) and recommendations for views (for Tableau Server). Recommendations are based on the popularity of content and on content used by other users determined to be similar to the current user.

`recommendations.vizrecs.enabled`

Default value: `true`

Tableau Server on Linux Administrator Guide

Controls recommendations for views for Tableau Server users. This option is a child of `recommendations.enabled` and will have no effect if the parent option is set to false. When the parent option is set to true, and this option is set to false, data sources and tables will still be recommended to Tableau Desktop users, but recommendations for views on Tableau Server will be disabled.

`redis.max_memory_in_mb`

Default value: 1024

Specifies the size in megabytes of the cache server external query cache.

`refresh_token.absolute_expiry_in_seconds`

Default value: 31536000

Specifies the number of seconds for absolute expiry of refresh and access tokens. The tokens are used by clients (Tableau Mobile, Tableau Desktop, Tableau Prep, etc) for authentication to Tableau Server after initial sign-in. This configuration key also governs personal access token expiry. All refresh and access tokens are a type of OAuth token. To remove limits set to `-1`. To disable OAuth tokens, see [Disable Automatic Client Authentication](#).

`refresh_token.idle_expiry_in_seconds`

Default value: 1209600

Specifies the number of seconds when idle OAuth tokens will expire. The OAuth tokens are used by clients for authentication to Tableau Server after initial sign-in. To remove limits set to `-1`.

`refresh_token.max_count_per_user`

Default value: 24

Specifies the maximum number of refresh tokens that can be issued for each user. If user sessions are expiring more quickly than you expect, either increase this value or set it to `-1` to entirely remove token limits.

`rsync.timeout`

Default value: `600`

Longest allowable time, in seconds, for completing file synchronization (600 seconds = 10 minutes). File synchronization occurs as part of configuring high availability, or moving the data engine and repository processes.

`schedules.display_schedule_description_as_name`

Default value: `false`

Controls whether a schedule name displays when creating a subscription or extract refresh (the default), or the "schedule frequency description" name describing the time and frequency of the schedule displays. To configure Tableau Server to display timezone-sensitive names for schedules, set this value to `true`.

When true, the "schedule frequency description" is also displayed after the schedule name on the schedule list page.

`schedules.display_schedules_in_client_timezone`

Default value: `true`

Shows the "schedule frequency description" in the timezone of the user when true (uses the client browser timezone to calculate the "schedule frequency description").

`schedules.ignore_extract_task_priority`

Default value (boolean): `False`

This setting controls whether or not task priority is considered for determining the job rank which determines when to pull jobs off the queue. Setting this to `true` disables editing the task priority on tasks, and only schedule priority will be considered for determining the job rank.

Tableau Server on Linux Administrator Guide

`searchserver.connection_timeout_milliseconds`

Added in version 2019.1.

Default value, in milliseconds: 100000

Specifies, in milliseconds, the amount of time Search & Browse clients will wait to establish a connection to the Search & Browse server.

On especially busy Tableau Server computers, or if you see log errors "Failed zookeeper health check. Refusing to start SOLR." increase this value.

For more information, see Client session timeouts.

`searchserver.javamemopts`

Added in version 2019.1.

Default value: `-Xmx512m -Xms512m -XX:+ExitOnOutOfMemoryError -XX:-UsePerfData`

Determines JVM options for SOLR.

Of all configurable options, the maximum heap memory, configured by the `-Xmx` parameter, is the most important when tuning the searchserver. In most cases this should be set as high as is possible, up to 24 GB, based on available physical memory on the Tableau Server computer. To change only the max heap memory, specify the entire default string but only change the value for `-Xmx`.

Valid values for `-Xmx` depend on available memory on the Tableau Server computer, but cannot be greater than 24 GB. For more information, see Search & Browse Max Heap Memory.

`searchserver.startup.zookeeper_healthcheck_timeout_ms`

Added in version 2020.1.

Default value, in milliseconds: 300000

Specifies, in milliseconds, the amount of time Tableau Server should wait for a successful Zookeeper health check on startup.

On especially busy Tableau Server computers, or if you see log errors "Failed zookeeper health check. Refusing to start SOLR." increase this value.

For more information, see Zookeeper connection health check timeout at startup.

`searchserver.zookeeper_session_timeout_milliseconds`

Default value, in milliseconds: 100000

Specifies, in milliseconds, the amount of time Search & Browse clients will wait to establish a connection to the Coordination Service (Zookeeper).

For more information, see Client session timeouts.

`ServerExportCSVMaxRowsByCols`

Added in version 2020.3.

Default value: 0 (no limit)

Specifies the maximum number of cells of data that can be downloaded from View Data into a CSV file. By default, there is no limit. Specify the number of cells. For example to set a limit of 3 million:

```
tsm configuration set -k ServerExportCSVMaxRowsByCols -v
3000000
```

```
tsm pending-changes apply
```

`service.jmx_enabled`

Default value: `false`

Setting to `true` enables JMX ports for optional monitoring and troubleshooting.

`service.max_procs`

Default value: <number>

Maximum number of server processes.

`service.port_remapping.enabled`

Default value: `true`

Determines whether or not Tableau Server will attempt to dynamically remap ports when the default or configured ports are unavailable. Setting to `false` disables dynamic port remapping.

`session.ipsticky`

Default value: `false`

Makes client sessions valid only for the IP address that was used to sign in. If a request is made from an IP address different from that associated with the session token, the session token is considered invalid.

In certain circumstances—for example, when Tableau Server is being accessed by computers with known and static IP addresses—this setting can yield improved security.

Note: Consider carefully whether this setting will help your server security. This setting requires that the client have a unique IP address and an IP address that stays the same for the duration of the session. For example, different users who are behind a proxy might look like they have the same IP address (namely, the IP address of the proxy); in that case, one user might have access to another user's session. In other circumstances, users might have a dynamic IP address, and their address might change during the course of the session. If so, the user has to sign in again.

`sheet_image.enabled`

Default value: `true`

Controls whether you can get images for views with the REST API. For more information, see [REST API Reference](#).

ssl.ciphersuite

Default

value: HIGH:MEDIUM:!aNULL:!MD5:!RC4:!3DES:!CAMELLIA:!IDEA:!SEED

Specifies the cipher algorithms that are allowed for SSL.

For acceptable values and formatting requirements, see [SSLCipherSuite](#) on the Apache website.

ssl.client_certificate_login.blocklisted_signature_algorithms

Default value:

- Version 2020.4.0:

```
shalwithrsaencryption,
shalwithrsa
```

- Version 2020.4.1 and later:

```
shalwithrsaencryption,
shalwithrsa,
shalwithrsaandmgf1,
shalwithdsa,
shalwithecdsa
```

The default value blocks certificates with the SHA-1 signing algorithm. Specifies the client signing algorithms that are blocked for SSL. To disable blocking of all signature algorithms, run this key with an empty set of quotes.

For more information about this key, see the Knowledge Base article, [Mutual SSL Fails After Upgrading if Certificates Signed with SHA-1](#).

ssl.client_certificate_login.min_allowed.elliptic_curve_size

Default value: 256

Specifies the minimum elliptic curve size required for ECDSA client certificates that are authenticating with Tableau Server over mutual SSL. If a client presents an ECDSA client certificate that does not satisfy this minimum curve size, the authentication request will fail.

This option was introduced in Tableau Server version 2021.1.

`ssl.client_certificate_login.min_allowed.rsa_key_size`

Default value: `2048`

Specifies the minimum key size for RSA client certificates that are authenticating with Tableau Server over mutual SSL. If a client presents an RSA client certificate that does not satisfy this minimum key size, the authentication request will fail.

This option was introduced in Tableau Server version 2021.1.

`storage.monitoring.email_enabled`

Default value: `false`

Controls whether email notifications are enabled for server disk space monitoring. By default, email notifications are enabled. To enable notifications for disk space monitoring, set this to `true`.

SMTP must be configured for notifications to be sent. For details, see [Configure SMTP Setup](#).

`storage.monitoring.warning_percent`

Default value: `20`

Warning threshold of remaining disk space, in percentage of total disk space. If disk space falls below this threshold, a warning notification is sent.

`storage.monitoring.critical_percent`

Default value: `10`

Critical threshold of remaining disk space, in percentage of total disk space. If disk space falls below this threshold, a critical notification is sent.

`storage.monitoring.email_interval_min`

Default value: `60`

How often, in minutes, that email notifications should be sent when disk space monitoring is enabled and a threshold is crossed.

`storage.monitoring.record_history_enabled`

Default value: `true`

Determines whether free disk space history is saved and available to view in Administrative Views. To disable history storage for monitoring, set `storage.monitoring.record_history_enabled` to `false`.

`subscriptions.enabled`

Default value: `false`

Controls whether subscriptions are configurable system-wide. See [Set Up a Site for Subscriptions](#).

`subscriptions.timeout`

Default value: `1800`

Length of time, in seconds, for a view in a workbook subscription task to be rendered before the task times out. If this time limit is reached while a view is being rendered, the rendering continues, *but any subsequent view in the workbook is not rendered*, and the job ends in error. In the case of a single-view workbook, this value will never result in the rendering being halted due to a timeout.

`svcmonitor.notification.smtp.enabled`

Default value: `false`

Tableau Server on Linux Administrator Guide

Controls whether email notifications are enabled for server process events. By default notifications are sent when processes go down, fail over, or restart. To enable server process notifications, set this to `true`.

SMTP must be configured for notifications to be sent. For details, see [Configure SMTP Setup](#).

`svcmonitor.notification.smtp.mime_use_multipart_mixed`

Note: Added in version: 2020.1.8, 2020.2.5, 2020.3.1

Default value: `false`

Controls whether subscription HTML MIME attachments are sent as *multipart/related* (the default) or *multipart/mixed*.

To allow the iOS Mail application to properly open these attachments, set this to `true`.

`tabadmincontroller.auth.expiration.minutes`

Default value: `120`

Controls how long session cookies are valid. By default this is set to 120 minutes. This value also determines how long the embedded credentials in a node bootstrap file are valid. For more information, see [tsm topology nodes get-bootstrap-file](#).

`tdsservice.log.level`

Note: Added in version 2020.3.0

Default value: `info`

The logging level for the Data Source Properties service. This is dynamically configurable, so if you are only changing this you do not have to restart Tableau Server. For more information, see [Change Logging Levels](#).

`tomcat.http.maxrequestsize`

Default value: `16380`

The maximum size (bytes) of header content that is allowed to pass through the Apache gateway on HTTP requests. Headers that exceed the value set on this option will result in browser errors, such as HTTP Error 413 (Request Entity Too Large) or authentication failures.

A low value for `tomcat.http.maxrequestsize` may result in authentication errors. Single sign-on solutions that integrate with Active Directory (SAML and Kerberos) often require large authentication tokens in HTTP headers. Be sure to test HTTP authentication scenarios before deploying into production.

We recommend setting `gateway.http.request_size_limit` option to the same value that you set for this option.

`tomcat.https.port`

Default value: 8443

SSL port for Tomcat (unused).

`tomcat.server.port`

Default value: 8085

Port that tomcat listens on for shutdown messages.

`tomcatcontainer.log.level`

Default value: `info`

The logging level for microservices in the Interactive Microservice Container and Non-Interactive Microservice Container. This is dynamically configurable starting in version 2020.4, so if you are only changing this you do not have to restart Tableau Server. For more information, see [Change Logging Levels](#).

`tsm.log.level`

Default value: `info`

Logging level for TSM services. These logs include information that can be useful if you have problems with TSM services: Administration Agent, Administration Controller, Client File Service, Cluster Controller, Service Manager, and License Service. This configuration key does not change the logging level for Coordination Service or for maintenance processes. For more information, see [Change Logging Levels and Tableau Server Processes](#).

`tsm.controlapp.log.level`

Default value: `info`

Logging level for `control_<app>` services. These logs include information that can be useful if you are running into problems starting or reconfiguring a TSM or Tableau Server process. For more information, see [Change Logging Levels](#).

`usernotifications.reap_after_days`

Default value: `30`

Number of days after which a user notification will be deleted from the server.

`vizportal.adsync.update_system_user`

Default value: `false`

Specifies whether email addresses and display names of users are changed (even when changed in Active Directory) when an Active Directory group is synchronized in Tableau Server. To ensure that user email addresses and display names are updated during synchronization, set `vizportal.adsync.update_system_user` to `true`, and then restart the server.

`vizportal.commenting.delete_enabled`

Default value: `true`

When set to `true`, lets users delete comments on views. You can delete a comment if you created it, are the content owner, a project leader with an appropriate site role, or are an administrator. To learn which site roles are required for full project leader access, see [Project-level administration](#).

`vizportal.csv_user_mgmt.index_site_users`

Default value: `true`

Specifies whether indexing of site users is done user by user when importing or deleting users with a CSV file. When set to `true` (the default) indexing is done as each user is added or deleted. To delay the indexing of the site users until after the entire CSV file has been processed, set this to `false`.

`vizportal.log.level`

Default value: `info`

The logging level for vizportal Java components. Logs are written to `/var/opt/tableau/tableau_server/data/tabsvc/logs/vizportal/*.log`.

Set to `debug` for more information. Using the debug setting can significantly impact performance, so you should only use this setting when directed to do so by Tableau Support.

Note: Beginning with version 2020.4.0, this is dynamically configurable, so if you are only changing this you do not have to restart Tableau Server. For more information, see [Change Logging Levels](#).

`vizportal.openid.client_authentication`

Specifies custom client authentication method for OpenID Connect.

To configure Tableau Server to use the IdPs that require the `client_secret_post`, set this value to `client_secret_post`.

An example would be when connecting to the Salesforce IDP, which requires this.

`vizportal.openid.essential_acr_values`

Specifies a list of authentication context class reference (ACR) values to provide the OpenID Connect IdP as an essential claim request. The IdP is responsible for ensuring that authentication meets the expected criteria. If the `vizportal.openid.essential_acr_values` configuration key is populated, Tableau Server acts as the relying party and

will inspect the ACR claim in the token response. Tableau Server will only warn if the ACR claim doesn't match the expected configuration key value.

To set this option, enter the ACR values in order of preference, enclosed by double-quotes. You must separate multiple values by a comma and space, as in this example:

```
tsm configuration set -k vizportal.openid.essential_acr_values -  
v "value1, value2"
```

This option was added beginning with Tableau Server version: 2020.4

`vizportal.openid.voluntary_acr_values`

Specifies a list of authentication context class reference (ACR) values to provide the OpenID Connect IdP as a voluntary claim request. The IdP is responsible for ensuring that authentication meets the expected criteria. If the `vizportal.openid.voluntary_acr_values` configuration key is populated, Tableau Server acts as the relying party and will inspect the ACR claim in the token response. The authentication request will fail if the ACR claim is missing or the provided claim value doesn't match the expected configuration key value.

To set this option, enter the ACR values in order of preference, enclosed by double-quotes. You must separate multiple values by a comma and space, as in this example:

```
tsm configuration set -k vizportal.openid.voluntary_acr_values -  
v "value1, value2"
```

This option was added beginning with Tableau Server version: 2020.4

`vizportal.rest_api.cors.allow_origin`

Specifies the origins (sites) that are allowed access to the REST API endpoints on Tableau Server when `vizportal.rest_api.cors.enabled` is set to `true`. You can specify more than one origin by separating each entry with a comma (,).

```
tsm configuration set -k vizportal.rest_api.cors.allow_origin -v  
https://mysite, https://yoursite
```

If `vizportal.rest_api.cors.enabled` is `false`, the origins listed by this option are ignored. For more information, see [Enabling CORS on Tableau Server](#).

Note: You can use an asterisk (*) as a wild card to match all sites. This is not recommended as it allows access from any origin that has access to the server and can present a security risk. Do not use an asterisk (*) unless you fully understand the implications and risks for your site.

`vizportal.rest_api.cors.enabled`

Default value: `false`

Controls whether Tableau Server allows Cross Origin Resource Sharing (CORS). When set to `true`, the server allows web browsers to access the [Tableau REST API](#) endpoints. You can use this option and the REST API to create custom portals. By default, this functionality is not enabled. To specify which origins (sites) have access, use the `vizportal.rest_api.cors.allow_origin` option. Only the origins specified with this option are allowed to make requests to the Tableau Server REST API. For more information, see [Enabling CORS on Tableau Server](#).

`vizqlserver.allow_insecure_scripts`

Default value: `false`

Allows a workbook to be published to the server from Tableau Desktop, and to be opened from the server, even if the workbook contains SQL or R expressions that are potentially unsafe (for example, a SQL expression that could potentially allow SQL injection). When this setting is `false` (the default), publishing a workbook or opening it from the server results in an error message, and the workbook is blocked. Before you set this value to `true` review the Knowledge Base article, [Blocking or Allowing Insecure Scripts in Tableau Server](#).

`vizqlserver.browser.render`

Default value: `true`

Tableau Server on Linux Administrator Guide

Views under the threshold set by `vizqlserver.browser.render_threshold` or `vizqlserver.browser.render_threshold_mobile` are rendered by the client web browser instead of by the server. See [Configure Client-Side Rendering](#) for details.

`vizqlserver.browser.render_threshold`

Default value: 100

The default value represents a high level of complexity for a view displayed on a PC. Complexity factors include number of marks, headers, reference lines, and annotations. Views that exceed this level of complexity are rendered by the server instead of in the PC's web browser.

`vizqlserver.browser.render_threshold_mobile`

Default value: 60

The default value represents a high level of complexity for a view displayed on a tablet. Complexity factors include number of marks, headers, reference lines, and annotations. Views that exceed this level of complexity are rendered by the server instead of in the tablet's web browser.

`vizqlserver.clear_session_on_unload`

Default value: `false`

Determines whether or not VizQL sessions are kept in memory when a user navigates away from a view or closes their browser. The default value (`false`) keeps sessions in memory. To close VizQL sessions on leaving a view or closing a browser, set this to `true`.

`vizqlserver.force_maps_to_offline`

Note: Added in version 2020.4.0

Default value: `false`

Determines whether Tableau Server runs in offline mode for maps. This is useful in disconnected environments where access to the internet and the map server is restricted. To

enable offline mode for maps, set this value to `true`. For more information about installing and configuring Tableau Server in an environment without internet access, see [Install Tableau Server in a Disconnected \(Air-Gapped\) Environment](#).

`vizqlserver.geosearch_cache_size`

Default value: 5

Sets the maximum number of different geographic search locale/language data sets that can be loaded into server memory at the same time. When the server receives a geographic search request for locale/language data set that is not in memory, it will load the set into memory. If loading the data set will exceed the specified limit, the least recently used locale/language data set is cleared from memory so the requested one can be loaded. The minimum value is 1. Each cache takes approximately 60 MB in memory (so if you set this to 10, the memory usage would be 600 MB (60 * 10)).

`vizqlserver.initialsql.disabled`

Default value: `false`

Specify whether to ignore initial SQL statements for all data sources. Set this to `true` to ignore initial SQL:

```
tsm configuration set -k vizqlserver.initialsql.disabled -v true
```

`vizqlserver.log.level`

Default value: `info`

The logging level for VizQL Server Java components. Logs are written to `/var/opt/tableau/tableau_server/data/tabsvc/logs/vizqlserver/*.log`.

Set to `debug` for more information. Using the debug setting can significantly impact performance, so you should only use it when directed to do so by Tableau Support.

Note: Beginning with version 2020.3.0, this is dynamically configurable, so if you are only changing this you do not have to restart Tableau Server. For more information, see [Change Logging Levels](#).

`vizqlserver.NumberOfWorkbookChangesBetweenAutoSaves`

Default value: 5

Auto recover configuration for web authoring. Specifies the number of changes that a user must make to trigger auto save. Take care when changing this value. Auto recover functionality may impact the performance of web authoring and other viz-related operations on Tableau Server. We recommend tuning this value by making incremental adjustments over time.

`vizqlserver_<n>.port`

The port a VizQL server instance (specified by "<n>") is running on.

`vizqlserver.protect_sessions`

Default value: `true`

When set to `true`, prevents VizQL sessions from being reused after the original user signs out.

`vizqlserver.querylimit`

Default value: 1800

Longest allowable time for updating a view, in seconds. 1800 seconds = 30 minutes. This configuration option impacts VizQL Server and Data Server.

`vizqlserver.RecoveryAttemptLimitPerSession`

Default value: 3

Auto recover configuration for web authoring. The maximum number of attempts to recover the same session. Take care when changing this value. Auto recover functionality may

impact the performance of web authoring and other viz-related operations on Tableau Server. We recommend tuning this value by making incremental adjustments over time.

`vizqlserver.session.expiry.minimum`

Default value: 5

Number of minutes of idle time after which a VizQL session is eligible to be discarded if the VizQL process starts to run out of memory.

`vizqlserver.session.expiry.timeout`

Default value: 30

Number of minutes of idle time after which a VizQL session is discarded.

`vizqlserver.sheet_image_api.max_age_floor`

Default value: 1

The amount of time, in minutes, to cache images that are generated by the Query View Image method of the REST API. For more information, see the [REST API Reference](#) in the REST API help.

`vizqlserver.showdownload`

Default value: `true`

Controls the display of the **Tableau Workbook** option of the Download menu in views. When set to `false`, the Tableau Workbook option is unavailable.

`vizqlserver.showshare`

Default value: `true`

Controls the display of Share options in views. To hide these options, set to `false`.

Note: Users can override the server default by setting the "showShareOptions" JavaScript or URL parameter.

`vizqlserver.url_scheme_whitelist`

Specifies one or more URL schemes to allow (safe list) when using [URL actions](#) on views and dashboards. The schemes `http`, `https`, `gopher`, `mailto`, `news`, `sms`, `tel`, `tsc`, and `tsl` are allowed (safe listed) by default. This command can contain multiple comma and space-separated values, as in this example:

```
tsm configuration set -k vizqlserver.url_scheme_whitelist -v  
scheme1, scheme2
```

The values you specify overwrite previous settings. Therefore, you must include the full list of schemes in the `set` command. (You cannot amend the list of schemes by running the `set` command repeatedly.)

`vizqlserver.web_page_objects_enabled`

Default value: `true`

Controls whether Web Page objects in dashboards can display target URLs. To prevent web pages from appearing, set to `false`.

`vizqlserver.WorkbookTooLargeToCheckpointSizeKiB`

Default value: `1024`

Auto recover configuration for web authoring. Size limit (KB) for a workbook that will auto save. Workbooks larger than this value will not be auto-saved. Take care when changing this value. Auto recover functionality may impact the performance of web authoring and other viz-related operations on Tableau Server. We recommend tuning this value by making incremental adjustments over time.

webdataconnector.refresh.enabled

Deprecated. Use `tsm data-access web-data-connectors allow` instead.

Determines whether extract refreshes for web data connectors (WDCs) are enabled in Tableau Server. To disable refresh for all WDCs, set the value for this key to `false`, as shown below:

```
tsm configuration set --key webdataconnector.refresh.enabled --
value false
```

To learn more, see [Web Data Connectors in Tableau Server](#).

webdataconnector.whitelist.fixed

Deprecated. Use `tsm data-access web-data-connectors add` instead.

Specifies one or more web data connectors (WDCs) that can be used by to access data connections that are accessible over HTTP or HTTPS. This command is formatted as JSON data on a single line, with all double-quotes (") escaped using a backslash (\).

For example to add a San Francisco Film Locations WDC to the safe list:

```
tsm configuration set --key webdataconnector.whitelist.fixed --
value '{"\"https://tableau.data.world:443\": {\"properties\": {
\"secondary_whitelist\": [\"(https://data.world/)(.*)\"] } }
}'
```

To learn more, see [Web Data Connectors in Tableau Server](#).

webdataconnector.enabled

Deprecated. Use `tsm data-access web-data-connectors allow` instead.

Default value: `true`

When set to `true`, you can use `tsm` commands to manage web data connectors on the server.

Tableau Server on Linux Administrator Guide

`webdataconnector.whitelist.mode`

Default value: `mixed`

Determines how Tableau Server can run web data connectors. Supported modes are:

- `mixed`. Users can run connectors that are on an allowlist (safe list) of URLs. This mode originally also allowed users to run WDCs that had been imported. Importing WDCs is no longer supported.
- `fixed`. Users can run connectors that are on an allowlist (safe list) of URLs.
- `insecure`. Users can run any connector.

Important: Use the `insecure` option *only* for development and testing. Because connectors run custom code, running connectors that have not been vetted can pose a security threat.

`wgserver.audit_history_expiration_days`

Default value: `183`

Specifies the number of days after which historical events records are removed from the PostgreSQL database (the Tableau Server database).

`wgserver.change_owner.enabled`

Default value: `true`

Controls whether the ownership of a workbook, data source or project can be changed. Other options include `false` and `adminonly`.

`wgserver.clickjack_defense.enabled`

Default value: `true`

When set to `true`, helps prevent a malicious person from "clickjacking" a Tableau Server user. In a clickjack attack, the target page is displayed transparently over a second page, and

the attacker gets the user to click or enter information in the target page while the user thinks he or she is interacting with the second page.

For more information, see Clickjack Protection.

`wgserver.domain.accept_list`

This key was added beginning in version 2020.4.0 and replaces `wgserver.domain.whitelist`.

Default value: null

Allows connection from Tableau Server to secondary Active Directory domains. A secondary domain is one that Tableau Server connects to for user synchronization, but is a domain where Tableau Server is not installed. Tableau Server will attempt to connect to secondary domains for user and group synchronization. In some cases, Tableau Server may be unable to connect to the secondary domain, which will result in the error, "Domain not in accept list (errorCode=101015)."

Setting the `wgserver.domain.accept_list` option is required by a fix for the security vulnerability, [\[Important\] ADV-2020-003: Tableau Server Forced Authentication](#). As of February 2020, the fix for this vulnerability is included in all latest versions and maintenance releases of Tableau Server.

To set this option, enter the secondary domain enclosed by double-quotes. Multiple domains must be separated by a comma and a space. For example, `tsm configuration set -k wgserver.domain.accept_list -v "example.org, domain.com"`.

Wildcard functionality is not supported. For example, if Tableau connects to `sub1.example.org` and `sub2.example.org`, then both domains must be added.

Updating the `wgserver.domain.accept_list` option overwrites the existing value. Therefore, if you are adding a new domain to an existing set of domains stored in the value, include all existing domains with the new domain when you set the option. You can retrieve

Tableau Server on Linux Administrator Guide

the full list of existing domains by running `tsm configuration get -k wgserver.domain.accept_list`.

`wgserver.domain.ldap.domain_custom_ports`

Default value: `null`

Allows you to map child domains and their LDAP ports. Domain and port are separated by a colon (:) and each domain:port pair is separated by a comma (,) using this format: `FQDN1:port,FQDN2:port`

Example: `tsm configuration set -k wgserver.domain.ldap.domain_custom_ports -v childdomain1.lan:3269,childdomain2.lan:3269,childdomain3.lan:389`

`wgserver.domain.whitelist`

Important: This key has been deprecated as of version 2020.4.0. Use `wgserver.domain.accept_list` instead. To continue to use this key, use the `--force-keys` option when setting the value. *This key will be permanently disabled in a future release.* For instructions on using this key in version 2020.3.x or earlier, see the [Tableau Server Help for version 2020.3](#).

Default value: `null`

Allows connection from Tableau Server to secondary Active Directory domains. A secondary domain is one that Tableau Server connects to for user synchronization, but is a domain where Tableau Server is not installed. Tableau Server will attempt to connect to secondary domains for user and group synchronization. In some cases, Tableau Server may be unable to connect to the secondary domain, which will result in the error, "Domain not in whitelist (errorCode=101015)."

`wgserver.extended_trusted_ip_checking`

Default value: `false`

Enforces IP client matching for trusted ticket requests.

`wgserver.restrict_options_method`

Default value: `true`

Controls whether Tableau Server accepts HTTP OPTIONS requests. If this option is set to `true`, the server returns HTTP 405 (Method Not Allowed) for HTTP OPTIONS requests.

`wgserver.saml.blocklisted_digest_algorithms`

Default value: `SHA1`

Specifies the hashing algorithms that are not allowed for any relevant SAML certificate signatures or SAML assertion digest method or signature methods. When set, certificates or assertions that are signed & hashed with a blocklisted algorithm will be rejected and fail.

There are multiple places where SHA-1 could be used on both the Tableau and IdP side. For example:

- Certificates uploaded with TSM that are used by Tableau Server to sign the request that is sent to the IdP.
- Certificates in the IdP metadata used to verify the AuthnResponse (signature) received from the IdP using the public key in the Certificate.
- Incoming assertions signed and hashed with SHA-1 (DigestMethod set to SHA-1 and SignatureMethod set to SHA-1).

This option was introduced in Tableau Server 2021.1.

The default value was changed to `(SHA1` in Tableau Server 2021.2. For more information about upgrading to 2021.2 with SAML configured, see the Knowledge Base article, [Tableau Server Using SAML Authentication Fails to Start or Rejects Login After Upgrade to Tableau Server 2021.2](#).

`wgserver.saml.forceauthn`

Default value: `false`

When set to `true`, if the Tableau user session expires, Tableau Server will re-authenticate the user with the IdP. This option can also be used to ask the IdP to prompt the user for re-authentication, even if the user has an active IdP session.

This option was introduced in Tableau Server 2019.3.

`wgserver.saml.idpattribute.username`

Specifies the name of the attribute in which your SAML IdP stores user names. By default, this is set to `username`. If the attribute name that your IdP uses contains spaces, enclose it in quotation marks. For more information, see [Configure Server-Wide SAML](#) or [Configure Site-Specific SAML](#).

`wgserver.saml.iframe_idp.enabled`

Default value: `false`

Default of `false` means that when users select the sign-in button on an embedded view, the IdP's sign-in form opens in a pop-up window.

When you set it to `true`, and a server SAML user who is already signed in navigates to a web page with an embedded view, the user will not need to sign in to see the view.

You can set this to `true` only if the IdP supports signing in within an `iframe`. The `iframe` option is less secure than using a pop-up, so not all IdPs support it. If the IdP sign-in page implements clickjack protection, as most do, the sign-in page cannot display in an `iframe`, and the user cannot sign in.

If your IdP does support signing in via an `iframe`, you might need to enable it explicitly. However, even if you can use this option, it disables Tableau Server clickjack protection for SAML, so it still presents a security risk.

`wgserver.saml.maxassertiontime`

Default value: `3000`

Specifies the maximum number of seconds, from creation, that a SAML assertion is usable.

wgserver.saml.min.allowed.elliptic_curve_size

Default value: 256

This option was introduced in Tableau Server 2021.1 but did not include a default value. In 2021.2, the default value was set to 256.

This option specifies the minimum allowed ECDSA curve size for the certificate used for SAML authentication. If you upload a certificate that has an ECDSA curve size less than 256, TSM will log an error when you apply changes.

If you are upgrading to Tableau Server 2021.2 or later and your SAML certificate uses an ECDSA curve size less than 256, Tableau Server will not start after upgrading. We recommend uploading a new certificate with 256(or larger) ECDSA curve size before upgrading. Alternatively, you can run this command to set a lower ECDSA curve size on older versions (pre-2021.1) of Tableau Server before you upgrade. If you are running this command on a version prior to 2021.1, you must include the `--force-keys` option with the command. For more information about upgrading to 2021.2 with SAML configured, see the Knowledge Base article, [Tableau Server Using SAML Authentication Fails to Start or Rejects Login After Upgrade to Tableau Server 2021.2](#).

wgserver.saml.min.allowed.rsa_key_size

Default value: 2048

This option was introduced in Tableau Server 2021.1 but did not include a default value. In 2021.2, the default value was set to 2048.

This option specifies the minimum allowed RSA key length for the certificate used for SAML authentication. If you upload a certificate that has an RSA key length less than 2048, TSM will log an error when you apply changes.

To run SAML authentication with a 1024 RSA key length (not recommended), set this value to 1024.

If you are upgrading to Tableau Server 2021.2 or later and your SAML certificate uses a key length less than 2048, Tableau Server will not start after upgrading. We recommend uploading a new certificate with 2048 (or larger) key length before upgrading. Alternatively, you can run this command to set a lower key strength on older versions (pre-2021.1) of Tableau Server before you upgrade. If you are running this command on a version prior to 2021.1, you must include the `--force-keys` option with the command. For more information about upgrading to 2021.2 with SAML configured, see the Knowledge Base article, [Tableau Server Using SAML Authentication Fails to Start or Rejects Login After Upgrade to Tableau Server 2021.2](#).

`wgserver.saml.responseskew`

Default value: `180`

Sets the maximum number of seconds difference between Tableau Server time and the time of the assertion creation (based on the IdP server time) that still allows the message to be processed.

`wgserver.saml.sha256`

Default value: `true`

When set to `true`, Tableau Server will hash message signatures and digests with SHA-256 in SAML assertions to the IdP. Set this option to `false` only if your IdP rejects assertions containing SHA-256 hashed content.

`wgserver.session.apply_lifetime_limit`

Default value: `false`

Controls whether there is a session lifetime for server sessions. Set this to `true` to configure a server session lifetime.

`wgserver.session.idle_limit`

Default value: `240`

The number of minutes of idle time before a sign-in to the web application times out.

`wgserver.session.lifetime_limit`

Default value: 1440

The number of minutes a server session lasts if a session lifetime is set. The default is 1440 minutes (24 hours). If `wgserver.session.apply_lifetime_limit` is `false` (the default) this is ignored.

`wgserver.unrestricted_ticket`

Default value: `false`

Specifies whether to extend access to server resources for users authenticated by trusted tickets. Default behavior allows users to access views only. Setting this to `true` allows users with valid trusted tickets to access server resources (projects, workbooks, and so on) as if they had signed in using their credentials.

`workerX.gateway.port`

Default value: 80 (443 if SSL)

External port that Apache listens on for workerX (where a “worker” is the term used for subsequent server nodes in the cluster). `worker0.gateway.port` is Tableau Server’s external port. In a distributed environment, `worker0` is the initial Tableau Server node.

`workerX.vizqlserver.procs`

Default value: <number>

Number of VizQL servers.

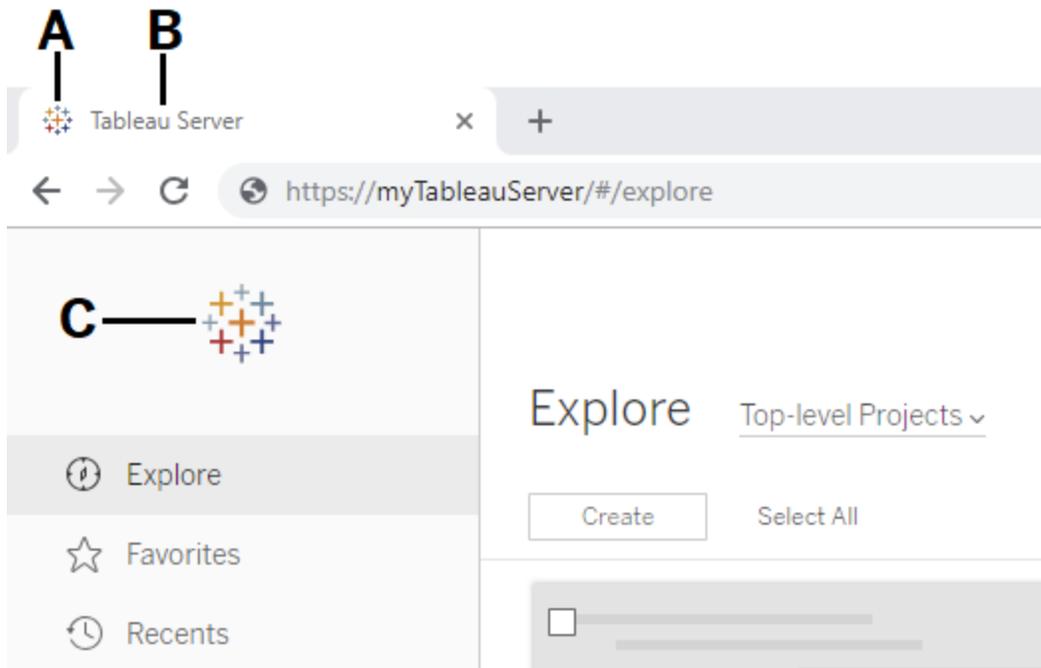
`zookeeper.config.snapCount`

Specifies the number of transactions necessary to cause the Coordination Service to create a snapshot of the logs. By default this value is 100,000 transactions. If your Coordination Service is not writing enough transactions to result in snapshots, the automatic cleanup of snapshots older than five days will not take place, and you may lose disk space to the trans-

action logs. By default transaction logs and snapshots are created in the Tableau data directory.

tsm customize

You can use the `tsm customize` command to customize the look and feel of the Tableau Server client browser experience.



Logo	Option flag	Minimum size/Maximum size, in pixels	Recommended size, in pixels
A - Window tab logo	cannot be changed	cannot be changed	cannot be changed
B - Server name	<code>--server-name</code>	does not apply	does not apply
C - Header logo	<code>--header-logo</code>	32 by 32 min, 160 by 160 max	48 by 48
Not shown - Sign in	<code>--signin-logo</code>	3000 by 3000 max	

Logo	Option flag	Minimum size/Maximum size, in pixels	Recommended size, in pixels
logo			
Header logo/Sign in logo	<code>--logo</code>	32 by 32 min, 160 by 160 max	48 by 48
Not shown - Logo shown when navigation pane is minimized	<code>--compact-logo</code>	32 by 32 max	32 by 32

The image files you use should be in GIF, JPEG, or PNG format.

The background colors on the header and sign in page are not the same. If you use the same image for both locations (if you use the `--logo` option, for example) your logo might look different depending on where it appears in the server interface.

As part of your disaster recovery plan, we recommend keeping a backup of the customization image files in a safe location off of the Tableau Server. The image files that you add to Tableau Server will be stored and distributed to other nodes by the Client File Service. However, the files are not stored in a recoverable format. See Tableau Server Client File Service.

Synopsis

```
tsm customize [options] [global options]
```

After you run the `customize` command, you must run the following command to apply changes:

```
tsm pending-changes apply
```

Options

Note: Use quotation marks around the path and image file if they include any spaces.

`--compact-logo "<path-to-logo>"`

Optional.

Specify a path to the image file that will be displayed when the navigation pane size is minimized. The maximum (and optimal) size is 32 by 32 pixels.

`--header-logo "<path-to-logo>"`

Optional.

Specify a path to the image file that will be displayed in the header only.

`--logo "<path-to-logo>"`

Optional.

Path to a single image file that will display for both the header and the sign-in window.

`--restore-defaults`

Optional.

Reset all customization options to default install state.

`--server-name <server_name>`

Optional.

Server name that appears in the browser tab, tooltips, and messages.

`--signin-logo "<path-to-logo>"`

Optional.

Specify a path to the image file that will be displayed for sign-in window only.

Global options

`-h, --help`

Optional.

Show the command help.

`-p, --password <password>`

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

`-s, --server https://<hostname>:8850`

Optional.

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port `8850`, and use the server name not the IP address. For example `https://<tsm_hostname>:8850`. If no server is specified, `https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see [Connecting TSM clients](#).

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm data-access

You can use the `tsm data-access` commands to configure data caching, enable or disable data repository access, enable SAML for single sign-on, and configure settings for Web Data Connectors (WDCs).

- caching
 - [data-access caching list](#)
 - [data-access caching set](#)
- repository
 - [repository-access disable](#)
 - [repository-access enable](#)
 - [repository-access list](#)
- set-saml-delegation
 - [set-saml-delegation configure](#)
 - [set-saml-delegation disable](#)
 - [set-saml-delegation enable](#)
- web-data-connectors
 - [web-data-connectors add](#)
 - [web-data-connectors allow](#)
 - [web-data-connectors delete](#)
 - [web-data-connectors list](#)

tsm data-access caching list

Displays data connection caching settings. To learn more about data connection caching settings, see [Configure Data Cache](#).

Synopsis

```
tsm data-access caching list [global options]
```

tsm data-access caching set

Sets data connection caching settings. To learn more about data connection caching settings, see [Configure Data Cache](#).

Synopsis

```
tsm data-access caching set [options] [global options]
```

Options

```
-r, --refresh-frequency
```

Optional.

Sets the frequency to refresh cached data with a new query to the underlying data source. You can specify a number to define the maximum number of minutes that data should be cached. You can also specify **low** to cache and reuse data for as long as possible, or **always** (equivalent to **0**) to refresh data each time that a page is loaded. If this option is not specified, it defaults to **low**.

tsm data-access repository-access disable

Disable external access to the Tableau PostgreSQL database for the default remote user. This will not disable access from localhost.

Synopsis

```
tsm data-access repository-access disable [options]
[global options]
```

Options

```
--repository-username <username>
```

Required.

The username, either **tableau** or **readonly**, with access to the data repository.

Tableau Server on Linux Administrator Guide

```
--request-timeout <timeout in seconds>
```

Optional.

Wait the specified amount of time for the command to finish. Default value is 1500 (25 minutes).

```
--ignore-prompt
```

Optional.

Suppress the prompt for restart and restart Tableau Server.

tsm data-access repository-access enable

Enables access to the Tableau PostgreSQL database.

By default, PostgreSQL traffic uses port 8060 (TCP). If you are running a local firewall, be sure to allow traffic for this port. To change the PostgreSQL port, see [Ports that are not dynamically mapped](#).

Synopsis

```
tsm data-access repository-access enable [options]  
[global options]
```

Options

```
--repository-password <password>
```

Required.

Sets (or changes) the password to access the data repository for the specified user-name.

```
--repository-username <username>
```

Required.

The username, either **tableau** or **readonly**, with access to the data repository.

```
--request-timeout <timeout in seconds>
```

Optional.

Wait the specified amount of time for the command to finish. Default value is 1500 (25 minutes).

```
--ignore-prompt
```

Optional.

Suppress the prompt for restart and restart Tableau Server.

tsm data-access repository-access list

Lists users who have access to the Tableau PostgreSQL database.

Synopsis

```
tsm data-access repository-access list [global options]
```

tsm data-access set-saml-delegation configure

Setup single sign-on for SAML SAP HANA so that Tableau Server functions as an Identity Provider (IdP) that provides single sign-on for users making SAP HANA data connections.

Synopsis

```
tsm data-access set-saml-delegation configure [options]  
[global options]
```

Options

```
-kf, --cert-key <cert-key>
```

Optional.

Tableau Server on Linux Administrator Guide

The SAML certificate key file.

`-cf, --cert-file <file-path>`

Optional.

The location of the SAML certificate file.

`-uf, --username-format <username-format>`

Optional.

Username format. Valid format keys are: 'username', 'domain_and_username', and 'email'.

`-uc, --username-case <username-case>`

Optional.

Username case. Valid case keys are: 'lower', 'upper', and 'preserve'.

tsm data-access set-saml-delegation disable

Disable single sign-on for SAML SAP HANA.

Synopsis

```
tsm data-access set-saml-delegation disable [global options]
```

tsm data-access set-saml-delegation enable

Enable single sign-on for SAML SAP HANA.

Synopsis

```
tsm data-access set-saml-delegation enable [global options]
```

tsm data-access web-data-connectors add

Add a web data connector (WDC) to the WDC safe list.

Synopsis

```
tsm data-access web-data-connectors add [options]
[global options]
```

Options

`-n, --name <name>`

Required.

The name for the WDC that will be displayed in the Tableau Server data source list. This name must be enclosed in single quotes (') or double quotes ("). Use double quotes (") if the name includes a space.

`-sec, --secondary <secondary-URL-1>, <secondary-URL-2>`

Optional.

A comma-delimited list of URLs that indicates which domains the connector can make requests to or receive data from, for example, external JavaScript libraries, REST APIs, or local files. Do not enclose the URLs in quotes. To add an entire domain to this secondary safe list, you can use a wildcard expression `.*` at the end of the URL, as shown in the following example: `https://www.example.com/*.*`

Note that on Windows, you could include parentheses (`.*`) around the wildcard, but it isn't necessary. On Linux, the parentheses will cause an error. Use `.*` as the wildcard.

Note that if you don't specify a secondary safe list, every request is allowed. For increased security, you should explicitly add the domains the WDC needs to access to a secondary safe list.

`--url <URL>`

Required.

The URL for the WDC (formatted as `<scheme>://<host>:<port>/<path>`, for example `https://www.tableau.com:443/example/`). For many WDCs the `<port>` value is 443, which is the default port used for HTTPS, but you can check the value for your connector by looking at the data source details on Tableau Server or Tableau Online. Note that you can't use a wildcard (`. *`) as part of the URL for the WDC.

tsm data-access web-data-connectors allow

Enable or disable WDC refreshes. Also, enable or disable the use of WDCs on Tableau Server.

Synopsis

```
tsm data-access web-data-connectors allow [options]
[global options]
```

Options

Use one or both options. At least one of `--refreshes` or `--type` is required.

```
-r, --refreshes <refreshes-allowed>
```

Optional if `--type` is specified.

Set to `false` to disallow WDC refreshes or `true` to allow WDC refreshes.

```
-t, --type <WDC-allowed>
```

Optional if `--refreshes` is specified.

Set to `none` to disallow the use of WDCs on Tableau Server (and omit WDCs from backups) or `all` to allow the use of WDCs.

tsm data-access web-data-connectors delete

Delete a specified WDC, or all WDCs, from the Tableau Server safe list.

Synopsis

```
tsm data-access web-data-connectors delete [options]
[global options]
```

Options

`--all`

Optional.

This option will delete all WDCs.

`--url <URL>`

Optional.

The URL for the WDC to delete.

tsm data-access web-data-connectors list

List all WDCs currently on the safe list.

Synopsis

```
tsm data-access web-data-connectors list [options]
[global options]
```

Options

Global options

`-h, --help`

Optional.

Show the command help.

`-p, --password <password>`

Tableau Server on Linux Administrator Guide

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

`-s, --server https://<hostname>:8850`

Optional.

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port `8850`, and use the server name not the IP address. For example `https://<tsm_hostname>:8850`. If no server is specified, `https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see [Connecting TSM clients](#).

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm email

Use the `tsm email` command to view and test your SMTP configuration.

For more information about configuring SMTP, see [Configure SMTP Setup](#).

tsm email test-smtp-connection

Run this command to test the SMTP connection. When run, TSM will attempt to establish a connection with the SMTP server that you have configured for Tableau Server. TSM will also return a connection status and the details of the SMTP configuration.

In some cases, the command will return a false-positive status. For example, if your Postfix SMTP server is set to require TLS, but Tableau Server is not configured for TLS, the connection is established and TSM will report a successful connection. However, in this scenario, Postfix actually rejects the email message after TSM has connected.

Synopsis

```
tsm email test-smtp-connection [global options]
```

Global options

`-h, --help`

Optional.

Show the command help.

`-p, --password <password>`

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

`-s, --server https://<hostname>:8850`

Optional.

Tableau Server on Linux Administrator Guide

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port `8850`, and use the server name not the IP address. For example `https://<tsm_hostname>:8850`. If no server is specified, `https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see [Connecting TSM clients](#).

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm initialize

You can use the `tsm initialize` command to initialize Tableau Server.

Note: You must apply or discard pending changes before running `tsm initialize` or the initialize will fail. Apply pending changes using the `tsm pending-changes apply` command. Discard any pending changes you do not want to apply using `tsm pending-changes discard`.

Synopsis

```
tsm initialize [options] [global options]
```

Options

`-r, --start-server`

Optional. Leave the server running after initialization is complete.

Global options

`-h, --help`

Optional.

Show the command help.

`-p, --password <password>`

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

`-s, --server https://<hostname>:8850`

Optional.

Use the specified address for Tableau Services Manager. The URL must start with

`https`, include port 8850, and use the server name not the IP address. For example

`https://<tsm_hostname>:8850`. If no server is specified,

`https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see [Connecting TSM clients](#).

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm jobs

You can use the `tsm jobs` commands to list, reconnect to, and cancel jobs.

- `cancel`
- `list`
- `reconnect`

tsm jobs cancel

Cancel a job on the server. Any job can be canceled before it starts running (when queued), but only backup and cleanup jobs can be canceled when they are already running.

Synopsis

```
tsm jobs cancel --id <jobID> [global options]
```

Options

`-i, --id <jobID>`

Required.

Id of the job to cancel.

tsm jobs list

List asynchronous jobs on the server.

Synopsis

```
tsm jobs list [--status <status>] [global options]
```

Options

```
-t, --status <status>
```

Optional.

Filter for jobs that match the given status.

tsm jobs reconnect

Reconnect to an asynchronous job to display its progress. If no job id is specified, it reconnects to the latest job.

Synopsis

```
tsm jobs reconnect[--id <jobID>] [global options]
```

Options

```
-i, --id <jobID>
```

Optional.

Specifies the id of the job that should be reconnected.

Global options

```
-h, --help
```

Optional.

Tableau Server on Linux Administrator Guide

Show the command help.

`-p, --password <password>`

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

`-s, --server https://<hostname>:8850`

Optional.

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port `8850`, and use the server name not the IP address. For example `https://<tsm_hostname>:8850`. If no server is specified, `https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see [Connecting TSM clients](#).

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm licenses

You can use the `tsm licenses` commands to manage server license tasks like activating or deactivating a Tableau Server product key on- or off-line, and getting associated files for offline activation or deactivation.

- `tsm licenses activate`
- `tsm licenses atr-configuration get`
- `tsm licenses atr-configuration set`
- `tsm licenses deactivate`
- `tsm licenses get-offline-activation-file`
- `tsm licenses get-offline-deactivation-file`
- `tsm licenses list`
- `tsm licenses refresh`

tsm licenses activate

Activates a Tableau Server product key.

Synopsis

```
tsm licenses activate --license-key <product-key>  
[global options]
```

Options

`-f, --license-file <file.tlf>`

Required if doing offline activation.

Specifies the license file (<file>.tlf) used for offline activation.

`-k, --license-key <product-key>`

Required if activating a valid product key.

Specifies the product key to use for online activation.

`-t, --trial`

Required if activating a trial license.

Activate a trial license.

tsm licenses atr-configuration get

View the Server authentication-to-run (ATR) duration. The ATR duration is the length of time that Tableau Server is authorized to run before the license must be renewed.

Synopsis

```
tsm licenses atr-configuration get --duration [global options]
```

Options

`--duration`

Displays the current authentication-to-run (ATR) in seconds. For example, 432000 (5 days).

tsm licenses atr-configuration set

Set the Server authentication-to-run (ATR) duration. The ATR duration is the length of time that Tableau Server is authorized to run before the license must be renewed.

Synopsis

```
tsm licenses atr-configuration set --duration <duration_in_seconds> [global options]
```

Options

`--duration <duration_in_seconds>`

Sets the authorization-to-run (ATR) duration (in seconds). For example, 432000 (5

days).

tsm licenses deactivate

Deactivates a Tableau Server product key either online or offline.

Synopsis

```
tsm licenses deactivate --license-key <product-key>
[global options]
```

Options

```
-f, --license-file <return_file.tlr>
```

Required if doing offline deactivation.

Specifies the license file (<file>.tlf) used for offline deactivation.

```
-k, --license-key <product-key>
```

Required if deactivating a product key.

Specifies the product key to use for online deactivation.

tsm licenses get-offline-activation-file

Generate an offline activate file to use for activating Tableau Server offline. To learn more, see [Activate Tableau Server Offline - Adding a License](#).

Synopsis

```
tsm licenses get-offline-activation-file --license-key
<product-key> --output-dir <path> [global options]
```

Options

```
-k, --license-key <product-key>
```

Tableau Server on Linux Administrator Guide

Required.

Specifies the product key to use for offline activation.

`-o, --output-dir <path>`

Required.

The location where the offline activation file should be saved. This location must exist.

tsm licenses get-offline-deactivation-file

Generate an offline deactivation file to use for deactivating Tableau Server offline. To learn more, see [Deactivate Tableau Server Offline](#).

Synopsis

```
tsm licenses get-offline-deactivation-file --license-key  
<product-key> --output-dir <path> [global options]
```

Options

`-k, --license-key <product-key>`

Required.

Specifies the product key to use for offline deactivation.

`-o, --output-dir <path>`

Required.

The existing location where the offline deactivation file should be saved.

tsm licenses list

Lists licenses that are activated on the Tableau Server deployment.

For example, a server with five Creator licenses, five Explorer licenses, 100 Viewer licenses, and a Data Management Add-on would provide command output similar to the following:

```

C:\Windows\system32>tsm licenses list
Number of product keys: 4
The following license keys will expire soon. Access renewal resources including information on how to renew your software or change your billing preferences here https://www.tableau.com/support/renew
TS90-06E2-8EF6-89EA-30EE  TS9R-3861-08B8-9E5A-C79D  TS4D-176C-EB48-3418-5E45  TSQJ-09B8-5CF8-F066-20AF
KEY      TYPE      CREATOR    EXPLORER    VIEWER      DATA MANAGEMENT ADD-ON    GUEST ACCESS LIC EXP    MAINT EXP    UPDATABLE    LBLM        SERVER MANAGEMENT ADD-ON
TS90-06E2-8EF6-89EA-30EE  Term      0           0           100         false         11/30/20    N/A         false        false        false
TS9R-3861-08B8-9E5A-C79D  Term      0           0           0           true          11/30/20    N/A         false        false        false
TS4D-176C-EB48-3418-5E45  Term      0           5           0           false         11/30/20    N/A         false        false        false
TSQJ-09B8-5CF8-F066-20AF  Term      5           0           0           false         11/30/20    N/A         false        false        false

```

The following fields are returned:

- **KEY:** A globally unique 16-character string that identifies the license.
- **TYPE:** Describes the type of license
 - **Term:** Term licenses map to a subscription schedule and must be renewed. The expiration date is listed under the LIC EXP field.
 - **Perpetual:** Perpetual licenses are purchased once and do not need to be renewed but must be refreshed to update the MAINT EXP or maintenance expiration date.
 - **Cores:** Core licenses are licenses that map to the number of cores on the computers running specific Tableau Server services. Core licensing allows for a guest user access to views on the server or embedded on other web servers. Core licenses also allow for unlimited Explorer and Viewer users.
- **CREATOR:** The number of Creator licenses issued to the Tableau Server deployment.
- **EXPLORER:** The number of Explorer licenses issued to the Tableau Server deployment.
- **VIEWER:** The number of Viewer licenses issued to the Tableau Server deployment.
- **DATA MANAGEMENT ADD-ON:** Tableau Server is licensed for the Data Management Add-on (`True/False`). See About Data Management Add-on.
- **GUEST ACCESS:** Tableau Server is licensed for a Guest User. See Guest User. The ability to leverage a Guest User requires Core licensing. See TYPE field.
- **LIC EXP:** The date that the license expires and Tableau Server will stop working. Term licenses expire. See TYPE field. Visit the Tableau [Customer Portal](#) to refresh licenses.
- **MAINT EXP:** Applies only to legacy perpetual licenses (TYPE = Perpetual). For Term licenses, this field will output, `N/A`. MAINT EXP displays the date that the maintenance contract for the Tableau Server deployment expires. To update the license maintenance key see Refresh Expiration Date for the Product Key. Visit the Tableau

[Customer Portal](#) to view maintenance purchase history and to purchase additional maintenance.

- **UPDATABLE:** Specifies whether the license is an updatable subscription license (`True/False`).
- **LBLM:** Specifies if login-based license management (LBLM) is enabled for the Tableau Server deployment (`True/False`). When enabled, LBLM allows users to log into Tableau Server to license their instance of Tableau Desktop or Prep, rather than entering a product key. For more information about LBLM, see [Login-based License Management](#).
- **SERVER MANAGEMENT ADD-ON:** Tableau Server is licensed for the Server Management Add-on (`True/False`). For more information about the Server Management Add-on, see [About Tableau Advanced Management](#).

Synopsis

```
tsm licenses list [global options]
```

tsm licenses refresh

Update the maintenance expiration date of all product keys on Tableau Server.

Synopsis

```
tsm licenses refresh [global options]
```

Global options

`-h, --help`

Optional.

Show the command help.

`-p, --password <password>`

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

```
-s, --server https://<hostname>:8850
```

Optional.

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port `8850`, and use the server name not the IP address. For example `https://<tsm_hostname>:8850`. If no server is specified, `https://<localhost | dnsname>:8850` is assumed.

```
--trust-admin-controller-cert
```

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see [Connecting TSM clients](#).

```
-u, --username <user>
```

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm login

Use the `tsm login` command to log in to Tableau Services Manager from a remote node.

If the account you are logged in as is a member of the TSM-authorized group, you do not need to provide credentials to run commands when running `tsm` CLI locally. For more information, see [Authenticating with tsm CLI](#).

Tableau Server on Linux Administrator Guide

Synopsis

```
tsm login [global options]
```

Global options

`-h, --help`

Optional.

Show the command help.

`-p, --password <password>`

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

`-s, --server https://<hostname>:8850`

Optional.

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port `8850`, and use the server name not the IP address. For example `https://<tsm_hostname>:8850`. If no server is specified, `https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see [Connecting TSM clients](#).

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm logout

You can use the `tsm logout` command to log out of Tableau Services Manager (TSM).

Synopsis

```
tsm logout [global options]
```

Global options

`-h, --help`

Optional.

Show the command help.

`-p, --password <password>`

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

`-s, --server https://<hostname>:8850`

Optional.

Tableau Server on Linux Administrator Guide

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port `8850`, and use the server name not the IP address. For example `https://<tsm_hostname>:8850`. If no server is specified, `https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see [Connecting TSM clients](#).

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm maintenance

You can use the `tsm maintenance` commands to manage server maintenance tasks like creating regular backups or restoring Tableau Server from a previously created backup.

- `tsm maintenance backup`
- `tsm maintenance cleanup`
- `tsm maintenance metadata-services`
 - `tsm maintenance metadata-services disable`
 - `tsm maintenance metadata-services enable`
 - `tsm maintenance metadata-services get-status`
- `tsm maintenance reindex-search`
- `tsm maintenance reset-searchserver`
- `tsm maintenance restore`

- `tsm maintenance send-logs`
- `tsm maintenance snapshot-backup`
 - `tsm maintenance snapshot-backup complete`
 - `tsm maintenance snapshot-backup prepare`
 - `tsm maintenance snapshot-backup restore`
- `tsm maintenance validate-resources`
- `tsm maintenance ziplogs`

tsm maintenance backup

Creates a backup of the data managed by Tableau Server. This data includes the Tableau PostgreSQL database (the repository) which contains workbook and user metadata, and extract (.tde or .hyper) files. This data does not include configuration data. See [Perform a Full Backup and Restore of Tableau Server](#).

Note: Do not use this command on Tableau Server installations with External File Store. See [Backup and Restore with External File Store](#).

Optimizing with topology configurations:

- Co-locating File Store on the same node as the Administration Controller can reduce the length of time it takes to back up Tableau Server by reducing or eliminating the need to transfer data between nodes during the backup process. This is especially true if your organization uses many extracts.
- Co-locating the repository (pgsql) with the Administration Controller node can also help to reduce back up time, but the time savings is less significant than that of the File Store.

The Administration Controller is usually on the initial node, unless you have had an initial node failure and moved the controller to another node.

The backup file is assembled in a temporary location in the data directory and then written to the directory defined in the `TSMbasefilepath.backuprestore` variable:

Tableau Server on Linux Administrator Guide

```
/var/opt/tableau/tableau_server-  
/data/tabsvc/files/backups/<filename>.tsbak
```

For more information about where backup files are written, and how to change that location, see [tsm File Paths](#). **Note:** Even when you change the backup location, the backup process uses a temporary location in the data directory to assemble the backup file.

Synopsis

```
tsm maintenance backup --file <backup_file> [options]  
[global options]
```

Options

```
-f, --file <backup_file>
```

Required.

For more information about backing up the repository data, see [Back up Tableau Server data](#) for more information.

```
-d, --append-date
```

Optional.

Append the current date to the end of the backup file name.

Note: Backups created using a schedule automatically include a date/time at the beginning of the backup file name. If you also use the `-d` option your file name will include the date twice. For more information about scheduling backups, see [Scheduling and Managing Backups](#).

```
-i, --description <string>
```

Optional.

Include the specified description of the backup file.

`--ignore-prompt`

Optional. Added in version 2020.2.

Back up without prompting, even if the File Store is not on the same node as the Administration Controller (usually the initial node). Use this prompt if automating backups (for example, with scripts).

`-k, --skip-verification`

Optional.

Do not verify the integrity of the database backup.

`--multithreaded`

Optional. Added in version 2021.1

By default, a single thread is used when creating a Tableau Server backup. When this option is specified, a backup is created using multiple threads. Two threads are used by default, when this option is specified. You can change the number of threads used by setting the `backup.zstd.thread_count` configuration key.

`--override-disk-space-check`

Optional.

Attempt to create a backup even when there is a low disk space warning.

`-po, --pg-only`

Optional.

Generates only the repository backup.

Tableau Server on Linux Administrator Guide

`--request-timeout <timeout in seconds>`

Optional.

Number of seconds to wait for the command to finish. Default value is 86400 (1440 minutes).

`-sd, --schedule-days <day[, day]>`

Optional. Added in version 2020.4.

Days on which to run the schedule. Use 1-7 for weekly schedule (1 for Monday, 7 for Sunday), 1-31 for monthly schedules (if a month does not include the specified day, the last day of the month is used). Separate multiple values with commas.

`-si, --schedule-id <ID>`

Optional. Added in version 2020.4.

Specify the ID of an existing schedule you want to update.

`--skip-compression`

Optional.

Create a backup without using compression. This results in a larger backup file but can reduce the amount of time it takes to complete the backup. If using this in a multi-node installation, we strongly recommend you have a File Store instance configured on your initial node.

`-sn, --schedule-name <name>`

Optional. Added in version 2020.4.

Specify the name for a schedule you are creating or updating.

`-sr, --schedule-recurrence <frequency>`

Optional. Added in version 2020.4.

Frequency of schedule recurrence. Valid options are "daily", "weekly", or "monthly".

`-st, --schedule-time <HH:MM>`

Optional. Added in version 2020.4.

The time a schedule should be run, in 24-hour format: HH:MM.

Examples

This example creates a backup called `ts_backup-<yyyy-mm-dd>.tsbak` in the `/var/opt/tableau/tableau_server/data/tabsvc/files/backups/` directory:

```
tsm maintenance backup -f ts_backup -d
```

This example creates a recurring weekly backup schedule named "weekly-saturday-backup" that runs every Saturday at noon and creates a backup called `<yyyy.m-m.dd.hh.mm>-ts_saturday_backup.tsbak`:

```
tsm maintenance backup -f ts_saturday_backup -sr weekly -st
12:00 -sd 6 -sn weekly-saturday-backup
```

For more details on managing scheduled backups, see [Scheduling and Managing Backups](#).

tsm maintenance cleanup

By default, deletes log files older than one day, and temporary files. Command options can modify which files are deleted and retention length.

The impact of this command depends on whether Tableau Server is running. If the server is running most old files and `http_requests` table entries can be deleted, but any files in use (locked by the operating system) cannot be deleted, so temporary files and active log files are not removed. To delete temporary files and current log files, you must stop the server before running this command.

Tableau Server on Linux Administrator Guide

If you are running Tableau Server on a distributed deployment, run this command on the node that is running the Administration Controller (also referred to as the *TSM Controller*) process. By default, the controller is on the initial node in the cluster.

Note: This command was added in Tableau Server version 10.5.1 and some options were added in version 2018.1.

Synopsis

```
tsm maintenance cleanup [options] [global options]
```

Options

`-a, --all`

Optional.

Perform all cleanup operations with default retention values. Equivalent to running the `cleanup` command with the following options: `-l -t -r -q -ic`.

`--http-requests-table-retention <# of days>`

Optional.

Default: 7 days

Specify the number of days of `http_requests` table entries that should be retained. Use this option with the `-q` option to delete entries older than the specified number of days are deleted. This option specifies table entry retention age but does not trigger actual deletion of table entries. The `-q` option triggers deletion of entries.

`-ic, --sheet-image-cache`

Optional. Added in version 2019.4.

Clear the image cache. This cache can contain images for offline previews, snapshots for subscription email messages, and subscription pdfs, as well as any images requested from the publish rest API endpoint (see [rest_api_ref.htm](#) for more information).

`-l, --log-files`

Optional.

Delete log files that are older than the `retention-period`. Files in the subdirectories under `data/tabsvc/logs` will be deleted.

`--log-files-retention <# of days>`

Optional.

Default: 1 (24 hours)

Delete logs older than this number of days. This command does not apply to temporary files.

`-q, --http-requests-table`

Optional.

Delete old `http_requests` table entries. Tableau Server must be running for table entries to be deleted. This option is ignored if Tableau Server is stopped. This option can be used alone to specify deletion of entries older than the default retention period (7 days), or together with the `--http-requests-table-retention` to specify a non-default retention period.

Note: Deleting `http_requests` table entries permanently removes data that is available to custom administrative views. Be sure removing this data will not impact any custom views you need.

`-r, --redis-cache`

Optional.

Tableau Server on Linux Administrator Guide

Clear the Redis cache.

```
--request-timeout <timeout in seconds>
```

Optional.

Wait the specified amount of time for the command to finish.

```
-t, --temp-files
```

Optional.

Delete all files and subdirectories in the following directories:

- `/var/opt/tableau/tableau_server/data/tabsvc/temp`: Only directories that are storing files for expired (not running) sessions are deleted.
- `/var/opt/tableau/tableau_server/data/tabsvc/httpd/temp`
- `/var/opt/tableau/tableau_server/temp`

Examples

This example cleans up all log files older than 2 days old:

```
tsm maintenance cleanup -l --log-files-retention 2
```

tsm maintenance metadata-services disable

Note: Command added in version 2019.3.

Use the `tsm maintenance metadata-services disable` command to disable the Tableau Metadata API.

Disabling the Metadata API stops continuous ingestion and indexing of information about the content on Tableau Server, deletes the index of information about the content published to Tableau Server and assets associated with that content, and disables the ability to both query the Metadata API and access Tableau Catalog.

Running this command stops and starts some services used by Tableau Server, which causes certain functionality, such as Recommendations, to be temporarily unavailable to your users.

Synopsis

```
tsm maintenance metadata-services disable
```

Option

--ignore-prompt

Optional.

Dismiss the confirmation prompt when disabling the Metadata API.

tsm maintenance metadata-services enable

Note: Command added in version 2019.3.

Use the `tsm maintenance metadata-services enable` command to enable the Tableau Metadata API for Tableau Server.

If Tableau Server is licensed with the Data Management, enabling the Metadata API enables Tableau Catalog.

When enabling the Metadata API, information about the content on Tableau Server is ingested and then indexed to the Metadata API Store. The Metadata API can be used to query schema, lineage, and user managed metadata about the content published to Tableau Server. After the Metadata API is enabled, metadata is continuously ingested and indexed until the Metadata API is disabled.

When running this command, keep the following in mind:

- This command stops and starts some services used by Tableau Server, which causes certain functionality, such as Recommendations, to be temporarily unavailable to your users.

Tableau Server on Linux Administrator Guide

- A new index of metadata is created and replaces the previous index every time this command is used.

For more information about the Tableau Catalog, see, [About Tableau Catalog](#).

Synopsis

```
tsm maintenance metadata-services enable
```

Option

```
--ignore-prompt
```

Optional.

Dismiss the confirmation prompt when enabling the Metadata API.

tsm maintenance metadata-services get-status

Note: Command added in version 2019.3.

Use the `tsm maintenance metadata-services get-status` command to get status information on Metadata Services.

Status on Metadata Services indicates if the Metadata API Store has been initialized or if the Tableau Metadata API is running or not.

Synopsis

```
tsm maintenance metadata-services get-status
```

tsm maintenance reindex-search

Use the `tsm maintenance reindex-search` command to rebuild the search index.

Synopsis

```
tsm maintenance reindex-search [options] [global options]
```

Option

```
--request-timeout <timeout in seconds>
```

Optional.

Wait the specified amount of time for the command to finish.

tsm maintenance reset-searchserver

Resets the search server to a clean state, deleting search information and rebuilding the search index.

Synopsis

```
tsm maintenance reset-searchserver [options] [global options]
```

Option

```
--request-timeout <timeout in seconds>
```

Optional.

Wait the specified amount of time for the command to finish.

tsm maintenance restore

Restore Tableau Server using the specified backup file. Restoring a backup file does not restore any configuration data. See [Perform a Full Backup and Restore of Tableau Server](#).

You can only restore from a backup that has the same type of identity store as the running server. For example, a backup from a server using local authentication can be restored to a Tableau Server initialized with local authentication, but a backup from a server using Active Directory authentication cannot be restored to a server initialized with local authentication.

Synopsis

```
tsm maintenance restore --file <file_name> [--restart-server]
[global options]
```

Options

`-f, --file <file_name>`

Required.

Specifies the backup file to restore from.

The `restore` command expects a backup file in the directory defined in the TSM `basefilepath.backuprestore` variable. By default:

```
/var/opt/tableau/tableau_server/data/tabsvc/files/backups/
```

For more information about file paths and how to change them, see [tsm File Paths](#).

`-ak, --asset-key-file <file_name>`

Optional.

Specify this option only if you are restoring from assets that were created by `tabadmin` on Tableau Server (versions 2018.1 and earlier).

Name of asset key file to restore from. The asset key file is created by the `tabadmin assetkeys` command. The file must be in the predefined backup/restore location on the server.

`-k, --skip-identity-store-verification`

Optional. Specify this option only if you are restoring from a backup file that was created by `tabadmin` on Tableau Server (versions 2018.1 and earlier).

Do not use this key in an attempt to change identity store type from Tableau Server that created original backup file. To change the identity store, see [Changing the Identity Store](#).

`-po, --pg-only`

Optional.

Restores only the repository.

`-r, --restart-server`

Optional.

Restart the server after the restore.

`--request-timeout <timeout in seconds>`

Optional.

Wait the specified amount of time for the command to finish.

tsm maintenance send-logs

Upload the specified file to Tableau and associate it with a support case. To successfully upload files to Tableau, your Tableau Server must be able to communicate with the send-logs server at <https://report-issue.tableau.com>.

Synopsis

```
tsm maintenance send-logs --case <case_number> --email <contact_email> --file <path/to/file> [global options]
```

Options

`-c, --case <case_number>`

Required.

Support case number.

`-e, --email <contact_email>`

Required.

Contact email.

Tableau Server on Linux Administrator Guide

`-f, --file <path/to/file>`

Required.

Specifies the location and name of the log file archive to send.

`--request-timeout <timeout in seconds>`

Optional.

Wait the specified amount of time for the command to finish.

tsm maintenance snapshot-backup complete

Note: Command added in version 2020.1 and only available when Tableau Server is configured for External File Store.

Complete the snapshot backup process on Tableau Server. Run this after you have taken a snapshot backup of your external storage.

The *tsm maintenance snapshot-backup prepare* and the *tsm maintenace snapshot-backup complete* commands are used to create a backup of Tableau Server data for Tableau Server installations that are configured with External File Store. For more information, see [Backup and Restore with External File Store](#)

Synopsis

```
tsm maintenance snapshot-backup complete [options]
[global options]
```

Options

`--request-timeout <timeout in seconds>`

Optional.

Wait the specified amount of time for the command to finish.

tsm maintenance snapshot-backup prepare

Note: Command added in version 2020.1 and only available when Tableau Server is configured for External File Store.

Prepares for snapshot backup. Once the preparation step is complete, you may take a snapshot backup of your network storage.

The *tsm maintenance snapshot-backup prepare* and the *tsm maintenace snapshot-backup complete* commands are used to create a backup of Tableau Server data for Tableau Server installations that are configured with External File Store. For more information, see [Backup and Restore with External File Store](#)

Synopsis

```
tsm maintenance snapshot-backup prepare [options]
[global options]
```

Options

```
--include-pg-backup
```

Optional.

A backup of the Repository is made and copied to the network share. This is only applicable to deployments where both **External Repository and External File Store** and configured. For more information, see [Backup and Restore with External File Store](#).

```
--request-timeout <timeout in seconds>
```

Optional.

Wait the specified amount of time for the command to finish.

tsm maintenance snapshot-backup restore

Note: Command added in version 2020.1 and only available when Tableau Server is configured for External File Store.

Restores the repository backup from the storage snapshot to Tableau Server.

For more information, see [Backup and Restore with External File Store](#).

Synopsis

```
tsm maintenance snapshot-backup restore [options]
[global options]
```

Options

```
--request-timeout <timeout in seconds>
```

Optional.

Wait the specified amount of time for the command to finish.

tsm maintenance validate-resources

Validate workbooks and data sources for a site. Use this command before migrating a site, to detect issues with site resources such as workbooks and data sources that will cause a site import to fail. Some resource problems can be corrected by republishing from local sources. Other problems might require assistance from Tableau Support.

Synopsis

```
tsm maintenance validate-resources --site-id <site ID>
[global options]
```

Options

```
-id,--site-id <site ID>
```

Required.

ID for the site whose resources you are validating.

`-r, --repair`

Optional.

Attempt to repair invalid resources. Those that cannot be repaired are noted in output.

`--request-timeout <timeout in seconds>`

Optional.

Wait the specified amount of time for the command to finish.

tsm maintenance ziplogs

Use the `ziplogs` command to create an archive of Tableau Server log files.

Note: If you cannot run the `ziplogs` command successfully, you can manually zip the Tableau Server logs. For more information, see [Troubleshoot Tableau Server on Linux](#).

Synopsis

```
tsm maintenance ziplogs [options] [global options]
```

Options

`-a, --all`

Optional.

Include all files except PostgreSQL data.

Tableau Server on Linux Administrator Guide

`-d, --with-postgresql-data`

Optional.

Include the PostgreSQL data folder if Tableau Server is stopped or PostgreSQL dump files if Tableau Server is running.

`--enddate <mm/dd/yyyy>`

Optional. Added in version 2019.3.

The last date of log files to be included. This option must be used with `--startdate` and cannot be used with `--minimumdate`. If this option is not specified, up to two days of logs will be included.

`-f, --file <name>`

Optional.

Specify a name for the zipped archive file. If no name is provided the archive is created as `logs.zip`. The file is written to the directory defined in the TSM `basefilepath.log_archive` variable. By default:

```
/var/opt/tableau/tableau_server/data/tabsvc/files/log-archives/
```

For more information about file paths and how to change them, see [tsm File Paths](#).

`-i, --description <string>`

Optional.

Include the specified description of the archive file.

`-l, --with-latest-dump`

Optional.

When any service crashes, Tableau Server generates a dumpfile. Set this option to include the most recent service crash dumpfile. If you do not set this option, then no dumpfile will be included in the resulting ziplog.

`-m, --minimumdate <mm/dd/yyyy>`

Optional.

Earliest date of log files to be included. If not specified, a maximum of two days of log files are included. Format of date should be "mm/dd/yyyy". This option cannot be used with `--startdate` and `--enddate` or `--all`.

`--nodes`

Optional. Added in version 2020.3.

Specify the nodes for which to create a backup. If not specified, backups are created for all nodes and saved to the initial node. Separate nodes with a comma. For example, to gather logs for nodes 2 and 4:

```
tsm maintenance ziplogs --nodes node2,node4
```

`-o, --overwrite`

Optional.

For an overwrite of an existing ziplog file. If a file by the same name already exists and this option is not used, the ziplogs command will fail.

By default the file is written to:

```
/var/opt/tableau/tableau_server/data/tabsvc/files/log-archives/
```

For more information about file paths and how to change them, see [tsm File Paths](#).

`--request-timeout <seconds>`

Optional.

Tableau Server on Linux Administrator Guide

Number of seconds to wait for the command to finish. Default value is 7200 (120 minutes).

`--startdate <mm/dd/yyyy>`

Optional. Added in version 2019.3.

The earliest date of log files to be included. This option must be used with `--enddate` and cannot be used with `--minimumdate`. If this option is not specified, up to two days of logs will be included.

Global options

`-h, --help`

Optional.

Show the command help.

`-p, --password <password>`

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

`-s, --server https://<hostname>:8850`

Optional.

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port 8850, and use the server name not the IP address. For example

`https://<tsm_hostname>:8850`. If no server is specified,
`https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see [Connecting TSM clients](#).

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm pending-changes

Use the `tsm pending-changes` commands to apply, discard, or view pending configuration and topology changes to Tableau Server.

Passwords and secrets that you enter during TSM configuration are encrypted after you save them. Secrets remain encrypted until, during, and after you apply pending changes. For more information about secret storage, see [Manage Server Secrets](#).

- `tsm pending-changes apply`
- `tsm pending-changes discard`
- `tsm pending-changes list`

tsm pending-changes apply

Use the `tsm pending-changes apply` command to apply pending configuration and topology changes to Tableau Server.

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt.

Synopsis

```
tsm pending-changes apply [global options]
```

Options

```
-iw, --ignore-warnings
```

Optional.

Ignore warning level constraints.

```
--ignore-prompt
```

Optional.

Suppress the prompt for restart. This only suppresses the prompt. The restart behavior is unchanged.

```
--request-timeout <timeout in seconds>
```

Optional.

Wait the specified amount of time for the command to finish. Default value is 1800 (30 minutes).

`tsm pending-changes discard`

Use the `tsm pending-changes discard` command to discard pending configuration and topology changes to Tableau Server.

Synopsis

```
tsm pending-changes discard [options] [global options]
```

Options

```
--config-only
```

Optional.

Discard only pending configuration changes.

```
--topology-only
```

Optional.

Discard only pending topology changes.

tsm pending-changes list

Lists pending configuration and topology changes to Tableau Server. Any changes that do not require a server restart will be listed as not requiring a restart. If none of the pending changes require a restart, a message displays saying the changes do not require a server restart. If any change in the list requires a restart, the entire list of pending changes will result in a restart. For more information on dynamic configuration or topology changes, see [What's New and Changed in Tableau Server](#) and [Tableau Server Dynamic Topology Changes](#).

Synopsis

```
tsm pending-changes list [options] [global options]
```

Options

```
--config-only
```

Optional.

Tableau Server on Linux Administrator Guide

List only pending configuration changes.

`--topology-only`

Optional.

List only pending topology changes.

Global options

`-h, --help`

Optional.

Show the command help.

`-p, --password <password>`

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

`--password 'my password'`

`-s, --server https://<hostname>:8850`

Optional.

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port `8850`, and use the server name not the IP address. For example `https://<tsm_hostname>:8850`. If no server is specified, `https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see [Connecting TSM clients](#).

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm register

You can use the `tsm register` command to register Tableau Server. This command lets you either display the registration file template (using the `--template` option) or provide the path to a completed registration file (using the `--file` option). You must use one of these two options when calling the `tsm register` command.

Synopsis

```
tsm register --template | --file <registration-filename>
[global options]
```

Options

`--file <registration-filename>`

Required.

Path to the file that contains the registration data.

`--template`

Required.

Display registration file template.

Global options

`-h, --help`

Optional.

Show the command help.

`-p, --password <password>`

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

`-s, --server https://<hostname>:8850`

Optional.

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port `8850`, and use the server name not the IP address. For example `https://<tsm_hostname>:8850`. If no server is specified, `https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see [Connecting TSM clients](#).

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm reset

Use the `tsm reset` command to clear the initial admin user so that you can enter a new one. After you run `tsm reset` you must rerun the `tabcmd initialuser` command to create a new initial admin. The new name cannot be the same username as the previous admin user.

If your organization is using Active Directory or LDAP for the Tableau identity store, then the account and password you specify must match an account in the directory.

Synopsis

```
tsm reset[option] [global options]
```

Option

```
-d, --delete-all-sessions
```

Optional.

Delete all active user sessions when the server is reset.

```
--request-timeout <timeout in seconds>
```

Optional.

Wait the specified amount of time for the command to finish. Default value is 1800 (30 minutes).

Global options

```
-h, --help
```

Optional.

Tableau Server on Linux Administrator Guide

Show the command help.

`-p, --password <password>`

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

`-s, --server https://<hostname>:8850`

Optional.

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port `8850`, and use the server name not the IP address. For example `https://<tsm_hostname>:8850`. If no server is specified, `https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see [Connecting TSM clients](#).

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm restart

You can use the `tsm restart` command to restart Tableau Server. The command stops the server if necessary, and then starts it.

Synopsis

```
tsm restart [global options]
```

Option

```
--request-timeout <timeout in seconds>
```

Optional.

Wait the specified amount of time for the command to finish. Default value is 1800 (30 minutes).

Global options

```
-h, --help
```

Optional.

Show the command help.

```
-p, --password <password>
```

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

```
-s, --server https://<hostname>:8850
```

Optional.

Tableau Server on Linux Administrator Guide

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port `8850`, and use the server name not the IP address. For example `https://<tsm_hostname>:8850`. If no server is specified, `https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see [Connecting TSM clients](#).

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm schedules

Note: Commands added in 2020.4.0. In this initial release, only backups can be managed with the `tsm schedules` commands.

You can use the `tsm schedules` commands to manage scheduled jobs. To create a scheduled backup job, use the `tsm maintenance backup` command. For details, see [tsm maintenance backup](#). For more details on managing scheduled backups, see [Scheduling and Managing Backups](#).

- `tsm schedules delete`
- `tsm schedules list`

- `tsm schedules resume`
- `tsm schedules suspend`
- `tsm schedules update`

tsm schedules delete

Delete the specified schedule.

Synopsis

```
tsm schedules delete [-si <scheduleID> | -sn <scheduleName>]
[global options]
```

Options

`-si, --schedule-id <scheduleID>`

Required if `--schedule-name` is not used.

Id of the schedule to delete.

`-sn, --schedule-name <scheduleName>`

Required if `--schedule-id` is not used.

Name of the schedule to delete.

tsm schedules list

List schedules on the server.

Synopsis

```
tsm schedules list [--next-run | --schedule-id <scheduleID> | -
-schedule-name <scheduleName>] [global options]
```

Tableau Server on Linux Administrator Guide

Options

`-nr, --next-run`

Optional.

Sort the schedules by their "next run" time, earliest to latest.

`-si, --schedule-id <scheduleID>`

Optional.

Id of the schedule to list the details of.

`-sn, --schedule-name <scheduleName>`

Optional.

Sort the schedules by their names, in alphabetical order.

tsm schedules resume

Resume the specified suspended schedule.

Synopsis

```
tsm schedules resume [-si <scheduleID> | -sn <scheduleName>]  
[global options]
```

Options

`-si, --schedule-id <scheduleID>`

Required if `--schedule-name` is not used.

Id of the schedule to resume.

`-sn, --schedule-name <scheduleName>`

Required if `--schedule-id` is not used.

Name of the schedule to resume.

tsm schedules suspend

Suspend the specified schedule.

Synopsis

```
tsm schedules suspend [-si <scheduleID> | -sn <scheduleName>]
[global options]
```

Options

`-si, --schedule-id <scheduleID>`

Required if `--schedule-name` is not used.

Id of the schedule to suspend.

`-sn, --schedule-name <scheduleName>`

Required if `--schedule-id` is not used.

Name of the schedule to suspend.

tsm schedules update

Update the specified schedule.

Synopsis

```
tsm schedules update[-si <scheduleID> -st <time_to_run> -sr
<recurrence_frequency> -sd <day-or-days>] [global options]
```

Tableau Server on Linux Administrator Guide

Options

`-si, --schedule-id <scheduleID>`

Required.

Id of the schedule to update.

`-sr, --schedule-recurrence <frequency>`

Required.

Recurrence frequency of the schedule. Valid options are "daily", "weekly", or "monthly".

`-st, --schedule-time <HH:MM>`

Required.

The time a schedule should be run, in 24-hour format: HH:MM.

`-sd, --schedule-days <day[, day]>`

Optional.

Days on which to run the schedule. For weekly schedules, use 1-7 where 1 is Monday, and 7 is Sunday. For monthly schedules, use 1-31. If a day doesn't exist for a specific month (30 for February, for example) the last valid day of the month is used. Separate multiple values with commas.

Global options

`-h, --help`

Optional.

Show the command help.

`-p, --password <password>`

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

`-s, --server https://<hostname>:8850`

Optional.

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port `8850`, and use the server name not the IP address. For example `https://<tsm_hostname>:8850`. If no server is specified, `https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see [Connecting TSM clients](#).

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm security

Use the `tsm security` commands to configure Tableau Server support for external (gateway) SSL or repository (Postgres) SSL. Repository SSL configuration includes the

option to enable SSL over direct connections from Tableau clients—including Tableau Desktop, Tableau Mobile, and web browsers—to the repository.

- `tsm security custom-cert`
 - `tsm security custom-cert add`
 - `tsm security custom-cert delete`
 - `tsm security custom-cert list`
- `tsm security external-ssl`
 - `tsm security external-ssl disable`
 - `tsm security external-ssl enable`
 - `tsm security external-ssl list`
- `tsm security kms`
 - `tsm security kms set-mode aws`
 - `tsm security kms set-mode azure`
 - `tsm security kms set-mode local`
 - `tsm security kms status`
- `tsm security maestro-rserve-ssl`
 - `tsm security maestro-rserve-ssl disable`
 - `tsm security maestro-rserve-ssl enable`
- `tsm security maestro-tabpy-ssl`
 - `tsm security maestro-tabpy-ssl disable`
 - `tsm security maestro-tabpy-ssl enable`
- `tsm security regenerate-internal-tokens`
- `tsm security repository-ssl`
 - `tsm security repository-ssl disable`
 - `tsm security repository-ssl enable`
 - `tsm security repository-ssl get-certificate-file`
 - `tsm security repository-ssl list`
- As of the 2020.2 release, to configure Rserve and TabPy analytics extensions, use the Tableau Server admin pages. See [Configure Connections with Analytics Extensions](#).

Prerequisites

Before you configure SSL, you must acquire certificates, and then copy them to the computer that runs the Tableau Server gateway process. Additional preparation is required for enabling direct connections from clients. To learn more, see the following articles:

Configure SSL for External HTTP Traffic to and from Tableau Server

Configure SSL for Internal Postgres Communication

For information about mutual (two-way) SSL, see [Configure Mutual SSL Authentication and `tsm authentication mutual-ssl` commands](#).

`tsm security custom-cert add`

Adds a custom CA certificate to Tableau Server. This certificate is optionally used to establish trust for TLS communication between a SMTP server and Tableau Server.

If a custom certificate already exists, this command will fail. You can remove the existing custom certificate using the `tsm security custom-cert delete` command.

Note: The certificate that you add with this command may be used by other Tableau Server services for TLS connections.

As part of your disaster recovery plan, we recommend keeping a backup of the certificate file in a safe location off of the Tableau Server. The certificate file that you add to Tableau Server will be stored and distributed to other nodes by the Client File Service. However, the file is not stored in a recoverable format. See [Tableau Server Client File Service](#).

Synopsis

```
tsm security custom-cert add --cert-file <file.crt>  
[global options]
```

Options

`-c, --cert-file <file.crt>`

Required. Specify the name of a certificate file in valid PEM or DER format.

tsm security custom-cert delete

Removes the server's existing custom certificate. Doing this allows you to add a new custom certificate.

Synopsis

```
tsm security custom-cert delete[global options]
```

tsm security custom-cert list

List details of custom certificate.

Synopsis

```
tsm security custom-cert list[global options]
```

tsm security external-ssl disable

Removes the server's existing SSL configuration settings and stops encrypting traffic between external clients and the server.

Synopsis

```
tsm security external-ssl disable [global options]
```

tsm security external-ssl enable

Enable and specify certificate and key files for SSL over external HTTP communication.

Synopsis

```
tsm security external-ssl enable --cert-file <file.crt> --key-  
file <file.key> [options] [global options]
```

Options

```
--cert-file <file.crt>
```

Required. Specify the name of a valid PEM-encoded x509 certificate with the extension .crt.

```
--key-file <file.key>
```

Required. Specify a valid RSA or DSA private key file, with the extension .key by convention.

```
--chain-file <chainfile.crt>
```

Specify the certificate chain file (.crt)

A certificate chain file is required for Tableau Desktop on the Mac. In some cases, a certificate chain file may be required for Tableau Mobile.

Some certificate providers issue two certificates for Apache. The second certificate is a chain file, which is a concatenation of all the certificates that form the certificate chain for the server certificate.

All certificates in the file must be x509 PEM-encoded and the file must have a .crt extension (not .pem).

```
--passphrase
```

Optional. Passphrase for the certificate file. The passphrase you enter will be encrypted while at rest.

Note: If you create a certificate key file with a passphrase, you cannot reuse the SSL certificate key for SAML.

```
--protocols <list protocols>
```

Optional. List the Transport Layer Security (TLS) protocol versions you want to allow or disallow.

TLS is an improved version of SSL. Tableau Server uses TLS to authenticate and encrypt connections. Accepted values include protocol versions supported by Apache. To disallow a protocol, prepend the protocol version with a minus (-) character.

Default setting: "all, -SSLv2, -SSLv3"

This default explicitly does not allow clients to use SSL v2 or SSL v3 protocols to connect to Tableau Server. However, we recommend that you also disallow TLS v1 and TLS v1.1.

Before you deny a specific version of TLS, verify that the browsers from which your users connect to Tableau Server support TLS v1.2. You might need to preserve support for TLSv1.1 until browsers are updated.

If you do not need to support TLS v1 or v1.1, use the following command to allow TLS v1.2 (using the value `all`), and explicitly deny SSL v2, SSL v3, TLS v1, and TLS v1.1.

```
tsm security external-ssl enable --cert-file file.crt --key-file  
file.key --protocols "all -SSLv2 -SSLv3 -TLSv1 -TLSv1.1"
```

tsm security external-ssl list

Displays a list of settings related to the configuration of gateway external SSL. The list includes the names of the certificate files in use, but not their location.

Synopsis

```
tsm security external-ssl list [global options]
```

tsm security kms set-mode aws

Set the KMS mode to AWS.

You will need the full ARN string from AWS KMS. This string is in the "General configuration" section of the AWS KMS management pages. The ARN is presented in this format: `arn:aws:kms:<region>:<account>:key/<CMK_ID>`, for example, `arn:aws:kms:us-west-2:867530990073:key/1abc23de-fg45-6hij-7k89-110mn1234567`.

For more information, see [AWS Key Management System](#).

Synopsis

```
tsm security kms set-mode aws --key-arn "<arn>" --aws-region
"<region>" [global options]
```

Options

`--key-arn`

Required. The `--key-arn` option takes a direct string copy from the ARN in the "General configuration" section of the AWS KMS management pages.

`--aws-region`

Required. Specify a region as shown in the Region column in the [Amazon API Gateway table](#).

Example

For example, if your AWS KMS instance is running in `us-west-2` region, your account number is `867530990073`, and your CMK key is `1abc23de-fg45-6hij-7k89-110mn1234567`, then the command would be:

Tableau Server on Linux Administrator Guide

```
tsm security kms set-mode aws --aws-region "us-west-2" --key-arn  
"arn:aws:kms:us-west-2:867530990073:key/1abc23de-fg45-6hij-7k89-  
110mn1234567"
```

tsm security kms set-mode azure

Set the KMS mode to Azure Key Vault.

Note: The KMS mode will display as "Azure Key Vault" when you run `tsm security kms status`, but you set it as "azure".

You will need the name of the Azure key vault and the name of the key in Azure.

For more information, see [Azure Key Vault](#).

Synopsis

```
tsm security kms set-mode azure --key-name "<key_name>" --vault-  
name "<vault_name>" [global options]
```

Options

`--key-name`

Required. The name of the asymmetric key stored in the Azure Key Vault.

`--vault-name`

Required. Name of the Azure Key Vault.

Example

For example, if your Azure Key Vault is named `tabsrv-keyvault` and your key is `tab-srv-sandbox-key01`, then the command would be:

```
tsm security kms set-mode azure --key-name "tabsrv-sandbox-
key01" --vault-name "tabsrv-keyvault"
```

tsm security kms set-mode local

Set or reset the KMS mode to local. Local is the default KMS mode. For more information, see Tableau Server Key Management System.

Synopsis

```
tsm security kms set-mode local [global options]
```

tsm security kms status

View the status of KMS configuration. The status returned includes:

- Status: OK indicates that the KMS is accessible by Tableau, or by the controller node if a multi-node installation.
- Mode: Local, AWS, or Azure Key Vault. Indicates what KMS mode is being used.
- Encrypt and decrypt master encryption key:

KMS stores a collection of master extract keys (MEKs). Each MEK has:

- An ID, for example, 8ddd70df-be67-4dbf-9c35-1f0aa2421521
- Either a “encrypt or decrypt key” or “decrypt-only key” status. If a key is “encrypt or decrypt”, Tableau Server will encrypt new data with it. Otherwise, the key will only be used for decryption
- A creation timestamp, for example, "Created at: 2019-05-29T23:46:54Z."
- First transition to encrypt and decrypt: a timestamp indicating when the key became an encrypt or decrypt key.
- Transition to decrypt-only: a timestamp indicating when the key transitioned to decrypt-only.

Other values returned depend on the KMS mode.

When the KMS mode is AWS, the following is returned:

Tableau Server on Linux Administrator Guide

- The ARN (ID) of the customer master key (CMK) .
- The region the CMK is in.
- The ID of the root master key (RMK) in use. The RMK is a key that is encrypted by the CMK. Tableau Server decrypts the CMK by making calls to AWS KMS. The RMK is then used to encrypt/decrypt the master extract key (MEK). The RMK can change, but there will be only one at a time.

When the KMS mode is Azure Key Vault, the following is returned:

- Vault name: The name of the Azure key vault.
- Azure Key Vault key name: The name of the key in the vault.

Synopsis

```
tsm security kms status [global options]
```

tsm security maestro-rserve-ssl disable

Disable the Rserve connection.

For more information, see [Use R \(Rserve\) scripts in your flow](#).

tsm security maestro-rserve-ssl enable

Configure a connection between an Rserve server and Tableau Server version 2019.3 or later.

For more information, see [Use R \(Rserve\) scripts in your flow](#).

Synopsis

```
tsm security maestro-rserve-ssl enable --connection-type <maestro-rserve-secure | maestro-rserve> --rserve-host <Rserve IP address or host name> --rserve-port <Rserve port> --rserve-username <Rserve username> --rserve-password <Rserve password> --rserve-connect-timeout-ms <Rserve connect timeout>
```

Options

`--connection-type`

Select `maestro-rserve-secure` to enable a secure connection or `maestro-rserve` to enable an unsecured connection. If you select `maestro-rserve-secure`, specify the certificate file path in the command line.

`--rserve-host`

Host

`--rserve-port`

Port

`--rserve-username`

Username

`--rserve-password`

Password

`--rserve-connect-timeout-ms`

The connect timeout in milliseconds. For example `--rserve-connect-timeout-ms 900000`.

`tsm security maestro-tabpy-ssl disable`

Disable the TabPy connection.

For more information, see [Use Python scripts in your flow](#).

tsm security maestro-tabpy-ssl enable

Configure a connection between a TabPy server and Tableau Server version 2019.3 or later.

For more information, see [Use Python scripts in your flow](#).

Synopsis

```
tsm security maestro-tabpy-ssl enable --connection-type <maestro-tabpy-secure | maestro-tabpy> --tabpy-host <TabPy IP address or host name> --tabpy-port <TabPy port> --tabpy-username <TabPy username> --tabpy-password <TabPy password> --tabpy-connect-timeout-ms <TabPy connect timeout>
```

Options

`--connection-type`

Select `maestro-tabpy-secure` to enable a secure connection or `maestro-tabpy` to enable an unsecured connection. If you select `maestro-tabpy-secure`, specify the certificate file `-cf<certificate file path>` in the command line.

`--tabpy-host`

Host

`--tabpy-port`

Port

`--tabpy-username`

Username

`--tabpy-password`

Password

`--tabpy-connect-timeout-ms`

The connect timeout in milliseconds. For example `--tabpy-connect-timeout-ms 900000`.

`tsm security regenerate-internal-tokens`

This command performs the following operations:

1. Stops Tableau Server if it is running.
2. Generates new internal SSL certificates for Postgres repository the search server.
3. Generates new passwords for all of the internally managed passwords.
4. Updates all Postgres repository passwords.
5. Generates a new encryption key for asset key management and encrypts the asset key data with the new key.
6. Generates a new encryption key for configuration secrets (master key) and encrypts the configuration with it.
7. Reconfigures and updates Tableau Server with all of these secrets. In a distributed deployment, this command also distributes the reconfiguration and updates across all nodes in the cluster.
8. Regenerates a new master key, adds it to the master keystore file, and then creates new security tokens for internal use.
9. Starts Tableau Server.

If you plan to add a node to your cluster after you have run this command, then you will need to generate a new node configuration file to update the tokens, keys, and secrets that are generated by this command. See [Install and Configure Additional Nodes](#).

For more information about internal passwords see [Manage Server Secrets](#).

Synopsis

```
tsm security regenerate-internal-tokens [options]
[global options]
```

Options

```
--request-timeout <timeout in seconds>
```

Optional.

Wait the specified amount of time for the command to finish. Default value is 1800 (30 minutes).

tsm security repository-ssl disable

Stop encrypting traffic between the repository and other server components, and stop support for direct connections from Tableau clients.

Synopsis

```
tsm security repository-ssl disable [global-options]
```

tsm security repository-ssl enable

Enables SSL and generates the server's .crt and .key files used for encrypted traffic between the Postgres repository and other server components. Enabling this also gives you the option to enable SSL over direct connections from Tableau clients to the server.

Synopsis

```
tsm security repository-ssl enable [options] [global options]
```

Options

```
-i, --internal-only
```

Optional. When set to `--internal-only`, Tableau Server uses SSL between the repository and other server components, and it supports but does not require SSL for direct connections through **tableau** or **readonly** users.

If this option is not set, Tableau Server requires SSL for traffic between the repository and other server components, as well as for direct connections from Tableau clients (for connections through the **tableau** or **readonly** users).

When you specify this option, you must also complete the steps described in Configure Postgres SSL to Allow Direct Connections from Clients.

tsm security repository-ssl get-certificate-file

Get the public certificate file used for SSL communication with the Tableau repository. SSL must be enabled for repository communication before you can retrieve a certificate. The certificate file is distributed automatically to internal clients of the repository in the Tableau Server cluster. To enable remote clients to connect over SSL to the repository, you must copy the public certificate file to each client.

Synopsis

```
tsm security repository-ssl get-certificate-file [global-  
options]
```

Options

`-f, --file`

Required.

Full path and file name (with `.cert` extension) where the certificate file should be saved. If a duplicate file exists it will be overwritten.

tsm security repository-ssl list

Returns the existing repository (Postgres) SSL configuration.

Synopsis

```
tsm security repository-ssl list [global-options]
```

Global options

`-h, --help`

Optional.

Show the command help.

`-p, --password <password>`

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

`-s, --server https://<hostname>:8850`

Optional.

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port `8850`, and use the server name not the IP address. For example `https://<tsm_hostname>:8850`. If no server is specified, `https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see [Connecting TSM clients](#).

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm settings

You can use the `tsm settings` commands to export (get) and import (set) configuration values.

- `tsm settings export`
- `tsm settings import`

Important: The server configuration file referenced in this topic includes a copy of the master keystore file used for encrypting configuration secrets. We strongly recommend that you take additional measures to secure the node configuration file, using mechanisms as described in [Securing secrets for import and export operations](#).

The following files are not exported or imported with this command. You must manage these files manually:

- SAML certificate file
- SAML key file
- SAML IdP metadata file
- The custom certificate installed by `tsm security custom-cert add`
- `OpenID.static.file`
- `Kerberos.keytab` file
- LDAP Kerberos keytab file
- LDAP Kerberos conf file
- Mutual SSL certificate file
- Mutual SSL revocation file
- Customization header logo file

Tableau Server on Linux Administrator Guide

- Customization sign-in logo file
- Customization compact logo file

tsm settings export

Export the current server configuration and topology to a file.

Synopsis

```
tsm settings export --output-config-file <path/to/output_
file.json> [global options]
```

Options

`-f, --output-config-file <file>`

Required.

Specifies the location and name of the file created by this operation.

tsm settings import

Import server configuration or topology.

Synopsis

```
tsm settings import --import-config-file <path/to/import_
file.json> [global options]
```

Options

`-f, --import-config-file <FILE>`

Required.

Path to input file.

`--config-only`

Optional.

`--topology-only`

Optional.

`-frc, --force-keys`

Optional.

Force a key to be added to configuration even if it did not previously exist.

Global options

`-h, --help`

Optional.

Show the command help.

`-p, --password <password>`

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

`-s, --server https://<hostname>:8850`

Optional.

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port 8850, and use the server name not the IP address. For example

Tableau Server on Linux Administrator Guide

`https://<tsm_hostname>:8850`. If no server is specified,
`https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see [Connecting TSM clients](#).

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm sites

You can use the `tsm sites` commands to export an existing site for import to a new site (also referred to as site migration), and to import the new site. An `unlock` command is available in case an error leaves a site locked.

The `tsm sites` commands will use your local file store to hold the export and import data. If you are running a multinode Tableau cluster, then you must run the `tsm sites` commands on a Tableau Server that is running the Data Engine process. For information about the Data Engine process and the processes that require it, see [Tableau Server Processes](#).

For comprehensive steps for migrating a site, see [Export or Import a Site](#).

- `tsm sites export`
- `tsm sites import`
- `tsm sites import-verified`
- `tsm sites unlock`

tsm sites export

Export a specified Tableau Server site to a .zip file. You can export a site to archive its settings at a specific point in time, or to complete the first step of a site migration process.

Note: The `tsm sites import` and `tsm sites export` commands can leave a site in a locked state if an error occurs. To unlock a site, use the `tsm sites unlock` command.

Synopsis

```
tsm sites export --site-id <source-siteID> --file <export-file>
[options] [global options]
```

Options

`-f, --file <export-file>`

Required.

Specify the name of the file to which Tableau Server saves all of the site's information.

This file is generated to the directory defined in the TSM `basefilepath.site_export.exports` variable. By default:

```
/var/opt/tableau/tableau_server/data/tabsvc/files/siteexports
```

For more information about file paths and how to change them, see [tsm File Paths](#).

`-id, --site-id <source-siteID>`

Required.

The site ID for the site you are exporting. You can get the site ID from the URL when you're signed in to the site from a web browser. For information about locating the site ID, see [Prepare the Source and Target Sites](#).

Tableau Server on Linux Administrator Guide

`-ow, --overwrite`

Optional.

Overwrite an export file of the same name that already exists.

`--request-timeout <timeout in seconds>`

Optional.

Wait the specified amount of time for the command to finish. Default value is 43200 (720 minutes).

tsm sites import

This command uses the .zip file you created using `tsm sites export` to generate a set of .csv files that show how the exported source site settings will map to the new target site.

By default, the .zip file is generated and saved to the `siteexports` directory at:

```
/var/opt/tableau/tableau_server/data/tabsvc/files/siteexports
```

Before you use this command, you must copy the .zip file to the directory in which Tableau will expect it. This location is defined in the TSM `basefilepath.site_import.-exports` variable. By default, the import directory is:

```
/var/opt/tableau/tableau_server/data/tabsvc/files/siteimports
```

For more information about file paths and how to change them, see [tsm File Paths](#).

Note: The `tsm sites import` and `tsm sites export` commands can leave a site in a locked state if an error occurs. To unlock a site, use the `tsm sites unlock` command.

Synopsis

```
tsm sites import --file <export-file.zip> --site-id <target-siteID> [options] [global options]
```

Options

```
-f, --file <export-file.zip>
```

Required.

Name of the .zip file created by the `tsm sites export` process, and which you must copy to the import directory. By default:

```
/var/opt/tableau/tableau_server/data/tabsvc/files/siteimports
```

```
-id, --site-id <target-siteID>
```

Required.

The site ID for the new site you are importing to (the target site). For information about locating the site ID, see [Prepare the Source and Target Sites](#).

```
-c, --continue-on-ignorable-errors
```

Optional.

Continue site import if errors occur which can be ignored. These errors can indicate issues with the import of a specific workbook or data source.

```
-k, --no-verify
```

Optional.

Skip verification of mapping files.

```
-m, --override-schedule-mapper <mapping-file.csv>
```

Optional.

Schedule mapping file to override the normal mapping by name.

```
--request-timeout <timeout in seconds>
```

Optional.

Wait the specified amount of time for the command to finish. Default value is 7200 (120 minutes).

tsm sites import-verified

Specify the directory that contains an exported site's .csv mapping files, to import to a new site. This is the final step of a site migration process.

Synopsis

```
tsm sites import-verified --import-job-dir <importjob-directory>  
--site-id <target-siteID> [options] [global options]
```

Options

```
-id, --site-id <target-siteID>
```

Required.

The site ID for the new site you are importing to (the target site). For information about locating the site ID, see [Prepare the Source and Target Sites](#).

```
-w, --import-job-dir <importjob-directory>
```

Required.

The parent of the `mappings` directory that contains the .csv files from the exported (source) site. The name of this parent directory includes the import id and date and time. For example:

```
/var/opt/tableau/tableau_server-
/data/tabsvc/files/siteimports/working/import_ff00_
20180102022014457
```

```
--request-timeout <timeout in seconds>
```

Optional.

Wait the specified amount of time for the command to finish. Default value is 7200 (120 minutes).

tsm sites unlock

Use this command to unlock a site.

Options

```
-id, --site-id <target-siteID>
```

Required.

The site ID for the site you are unlocking. For information about locating the site ID, see [Prepare the Source and Target Sites](#).

```
-d, --desired-state <state to leave unlocked site in>
```

Optional.

The state the site should be left in after it is unlocked. Options are "active" and "suspended". The default is "active" if not specified.

For example:

```
tsm sites unlock -id mysite -d suspended
```

```
--request-timeout <timeout in seconds>
```

Optional.

Wait the specified amount of time for the command to finish. Default value is 300 (5 minutes).

Global options

`-h, --help`

Optional.

Show the command help.

`-p, --password <password>`

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

`-s, --server https://<hostname>:8850`

Optional.

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port `8850`, and use the server name not the IP address. For example `https://<tsm_hostname>:8850`. If no server is specified, `https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see [Connecting TSM clients](#).

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm start

You can use the `tsm start` command to start Tableau Server. If the server is already running this command does nothing.

Synopsis

```
tsm start [option][global options]
```

Option

```
--request-timeout <timeout in seconds>
```

Optional.

Wait the specified amount of time for the command to finish. Default value is 1800 (30 minutes).

Global options

```
-h, --help
```

Optional.

Show the command help.

```
-p, --password <password>
```

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

Tableau Server on Linux Administrator Guide

```
--password 'my password'
```

```
-s, --server https://<hostname>:8850
```

Optional.

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port `8850`, and use the server name not the IP address. For example `https://<tsm_hostname>:8850`. If no server is specified, `https://<localhost | dnsname>:8850` is assumed.

```
--trust-admin-controller-cert
```

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see [Connecting TSM clients](#).

```
-u, --username <user>
```

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm status

You can use the `tsm status` command to display the status of Tableau Server and individual services (processes) that run as part of Tableau Server.

Synopsis

```
tsm status [global options]
```

Options

`-v, --verbose`

Optional.

Display status for every node in the Tableau Server cluster.

`tsm status` will return one of these potential statuses for a Tableau Server node:

- **RUNNING:** The node is running without error statuses for any service or process.
- **DEGRADED:** The node is running with one or more primary services - such as the repository - in an error state. If you have a single instance of the Messaging service and it fails,
- **ERROR:** All primary services or processes are in an error state on the node.
- **STOPPED:** The node is stopped, with no error statuses.

When running `tsm status` with the `--verbose` option, TSM will return a status for each individual service (process). Possible status messages include:

- `is running:` The service is running.
- `status is unavailable:` The status cannot be determined - such as when services are starting up.
- `is in a degraded state:` The service is running, but returning errors. This status indicates the service failed to install properly, has not been configured, or has failed in some way.
- `is in an error state:` The service is running, but returning errors. This status indicates the service failed to install properly, or has not been configured.
- `is synchronizing:` The File Store process is synchronizing with another instance of File Store.

Tableau Server on Linux Administrator Guide

- `is decommissioning`: The File Store process is being decommissioned.
- `is running (Active Repository)`: The active repository is running. This is the expected status.
- `is running (Passive Repository)`: The passive repository is running. This is the expected status when there are two repositories configured.
- `is stopped`: The service is stopped. This does not mean a service is in an error or problem state. Some services only run when needed (the Database Maintenance service for example).

Global options

`-h, --help`

Optional.

Show the command help.

`-p, --password <password>`

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

`-s, --server https://<hostname>:8850`

Optional.

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port 8850, and use the server name not the IP address. For example

`https://<tsm_hostname>:8850`. If no server is specified,
`https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see [Connecting TSM clients](#).

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm stop

You can use the `tsm stop` command to stop Tableau Server. If Tableau Server is already stopped, this command does nothing.

Synopsis

```
tsm stop [option] [global options]
```

Options

`--ignore-node-status <nodeID>`

Optional.

Ignore the status for the specified node or nodes when determining if the server has stopped. Useful if removing a bad node. Separate multiple nodes with commas.

Note: Option added in version 2020.1

`--request-timeout <timeout in seconds>`

Optional.

Wait the specified amount of time for the command to finish. Default value is 1800 (30 minutes).

Global options

`-h, --help`

Optional.

Show the command help.

`-p, --password <password>`

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

`-s, --server https://<hostname>:8850`

Optional.

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port `8850`, and use the server name not the IP address. For example `https://<tsm_hostname>:8850`. If no server is specified, `https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see [Connecting TSM clients](#).

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm topology

Use the `tsm topology` commands to prepare File Store nodes for safe removal or to put them back into read-write mode. You can also initiate a repository failover, get a list of nodes or ports, get the bootstrap configuration file required to add additional nodes to your cluster, remove nodes, configure external repository, and external File Store.

When making changes to topology, you need to also apply those pending changes. For more information, see `tsm pending-changes`.

- `cleanup-coordination-service`
- `deploy-coordination-service`
- `external-services`
 - `filestore`
 - `storage disable`
 - `storage enable`
 - `storage switch-share`
 - `list`
 - `repository`
 - `disable`
 - `enable`
 - `replace-host`
- `failover-repository`
- `filestore`
 - `decommission`
 - `recommission`

- `list-nodes`
- `list-ports`
- `nodes`
 - `get-bootstrap-file`
- `remove-nodes`
- `set-node-role`
- `set-ports`
- `set-process`
- `toggle-coordination-service`

`tsm topology cleanup-coordination-service`

Note: Beginning with version 2020.1.0, all coordination service ensemble commands require input for a "y/n" prompt confirming that a server restart will take place. To run these commands without input, include the `--ignore-prompt` option.

Use the `tsm topology cleanup-coordination-service` command to remove the non-production Tableau Server Coordination Service ensemble after you deploy a new ensemble. This command removes the old Coordination Service instances on all nodes in the non-production Coordination Service ensemble and is required after you deploy a new Coordination Service ensemble. To learn more about Coordination Service ensembles, see [Deploy a Coordination Service Ensemble](#) .

In version 2020.1.0 and later, the `tsm topology deploy-coordination-service` command also removes the old ensemble. There is no need to run this command separately unless the deployment fails.

Synopsis

```
tsm topology cleanup-coordination-service [option]
[global options]
```

Option

```
--request-timeout <timeout in seconds>
```

Optional.

Wait the specified amount of time for the command to finish. Default value is 2700 (45 minutes).

tsm topology deploy-coordination-service

Note: Beginning with version 2020.1.0, all coordination service ensemble commands require input for a "y/n" prompt confirming that a server restart will take place. To run these commands without input, include the `--ignore-prompt` option.

You can use the `tsm topology deploy-coordination-service` command to deploy the Tableau Server Coordination Service. This command deploys a Coordination Service ensemble, which is a set of Coordination Service instances that run on specified nodes in your server cluster. To learn more about Coordination Service ensembles, including how many nodes in your cluster should have a Coordination Service instance, see [Deploy a Coordination Service Ensemble](#) .

In version 2020.1.0 and later, the `tsm topology deploy-coordination-service` command also removes the old ensemble. There is no need to run the `cleanup-coordination-service` command separately.

Synopsis

```
tsm topology deploy-coordination-service --nodes
<nodeID,nodeID,...> [option] [global-options]
```

Options

```
-n, --nodes <nodeID,nodeID,...>
```

Required.

Tableau Server on Linux Administrator Guide

Node IDs of nodes to include in the new Coordination Service ensemble, separated by commas. You can specify 1, 3, or 5 Coordination Service nodes, depending on the total number of nodes in your cluster. For more information, see [The Coordination Service Quorum](#).

```
--request-timeout <timeout in seconds>
```

Optional.

Wait the specified amount of time for the command to finish. Default value is 2700 (45 minutes).

tsm topology external-services storage disable

Configure Tableau Server to run File Store locally. Use this command to disable External File Store and move the File Store data to your Tableau Server.

Synopsis

```
tsm topology external-services storage disable [options] [global options]
```

Options

```
-fsn <nodeID, nodeID,...>
```

Required

Specify the nodes that you want to configure File Store. You can specify more than one node. The data will be migrated to the first node in the list and then replicated to other nodes.

For more information, see [Reconfigure File Store](#) .

tsm topology external-services storage enable

Configure Tableau Server with External File Store. External File Store uses SAN or NAS to store File Store data.

Synopsis

```
tsm topology external-services storage enable [options] [global
options]
```

Options

-network-share

Required

Specify the name and path of the network share you want to use for your External File Store.

For more information, see [Reconfigure File Store](#) .

tsm topology external-services storage switch-share

Use this command to move your external services to a different network share. An example of this might be when your current network attached storage is at the end of life and you need to use a new network attached storage with new hardware. For more information, see [Re-Configure Tableau Server Repository](#).

Synopsis

```
tsm topology external-services filestore [option] [global
options]
```

Option

--network-share

Required

Specify the path to the new network share that you want to switch to.

tsm topology external-services list

Use the `tsm topology external-service-list` command to get a the service that is used for Tableau Server External Repository. For example, if you have configured Tableau Server

to use Amazon RDS, you will see the following message:

These externally configured services are in use by Tableau Server:

- pgsq1

Synopsis

```
tsm topology external-service list [global options]
```

Option

There are no options for this command.

tsm topology external-services repository disable -n nodeN

Use the `tsm topology external-services repository disable` command to stop using the external repository and reconfigure the installation to use a local repository. This will migrate the data to a local repository and configure Tableau Server to use the local repository.

Synopsis

```
tsm topology external-services repository disable -n nodeN
```

Option

`-n, --node-name <nodeID>`

Required.

Specifies the node ID of the node where the repository should be moved to.

Important: This does not stop or delete the RDS instance. For more information on how to delete an RDS instance, see [Deleting a DB Instance](#) on the AWS web site.

tsm topology external-services repository enable

Use the `tsm topology external-services repository enable` command to configure Tableau Server to use an external repository. This command can be used during installation of a new Tableau Server to configure the external repository. If this command is run on an already

existing and running Tableau Server, it will migrate the data from the local node to the external repository and configure Tableau Server to use the external repository after the migration is complete.

Synopsis

```
tsm topology external-services repository enable -f <file-name>.json -c <ssl certificate file>.pem
```

Options

--f <file name>

Required.

Full path and file name where the configuration file is saved. For more information, see [Re-Configure Tableau Server Repository](#).

--c <ssl certificate file>

Required for versions 2021.2 and 2021.2.1. Optional for versions 2021.2.2 and later.

For SSL configurations, download the certificate file and specify the file for use with this option.

1. **Amazon RDS:** See [Using SSL to Encrypt the Connection to a DB Instance](#).
2. **Azure Database:** See [Configure TLS connectivity for Azure Database for PostgreSQL](#).
3. **Stand-alone PostgreSQL Instance:** See [Configure SSL](#).

--no-ssl

Optional. This option is only available in versions 2021.2.2 and later.

This means SSL is not required when connecting to the External Repository. If you do not need to use encrypted connections, you must also configure the External Repository to allow unencrypted connections. When you use this option, connections will be encrypted if the external repository is configured to support TLS/SSL connections. Otherwise, Tableau Server will use non-encrypted connections.

tsm topology external-services repository replace-host

This command updates Tableau Server configuration settings to use the specified external repository. Use the `tsm topology external-services repository replace-host` command to reconfigure Tableau Server to use the new external repository immediately without moving data to it from your current external repository. You may need to manually migrate the data. You should only do this after you have fully evaluated and understand the impact of the potential data loss.

This command can be used in the following scenarios:

- Planned expiration of the SSL certificates used by RDS instances: RDS instances need to be updated with the new certificates, and Tableau Server needs to be configured to use the new certificate file to connect to the RDS instance.
- Disaster recovery: Use this to connect to a new RDS instance in disaster recovery scenarios. For more information, see [Create a PostgreSQL DB Instance on AWS Relational Database Service \(RDS\)](#)

Synopsis

```
tsm topology external-services repository replace-host -f <file-name>.json -c <ssl certificate file>.pem
```

Options

`-f <file name>`

Required.

Full path and file name where the configuration file is saved. For more information, see [Re-Configure Tableau Server Repository](#).

`-c <ssl certificate file>`

Optional.

The certificate file is the certificate to be imported to allow connections to the instance. For RDS, this is the CA cert used to sign the certificate of the instance. This is usually the latest root certificate `rds-ca-XXXX-root.pem` file. Use this parameter to update Tableau server if the certificate has changed on the RDS instance.

For more information, see [Using SSL/TLS to Encrypt a Connection to a DB Instance](#).

For more information on how to get the .pem file, see [Using SSL to Encrypt a Connection to a DB Instance](#).

`--ignore-prompt`

Optional.

Run this command without prompts.

`tsm topology failover-repository`

You can use the `tsm topology failover-repository` to manually initiate a repository failover from the current active repository to the second, passive repository.

The `tsm topology failover-repository` command is persistent. The failover repository remains the active repository until you issue the command again, or, if Tableau Server is configured for it, until automatic failover occurs. If you have a preferred active repository configured, use the `--preferred` option to switch back to that repository. For more information about configuring a preferred active repository, see [Tableau Server Repository](#). If Tableau Server is configured for high availability, failover of the repository is automatic when necessary. Use the `failover-repository` command to manually fail over the repository.

Synopsis

```
tsm topology failover-repository --preferred | --target <node_id> [global options]
```

Options

`-r, --preferred`

Required if `-t` or `--target` is not used.

Use the configured preferred node as the target for repository failover.

`--request-timeout <timeout in seconds>`

Optional.

Wait the specified amount of time for the command to finish. Default value is 1800 (30 minutes).

`-t, --target <node_id>`

Required if `-r` or `--preferred` is not used.

The node id of the target node onto which failover will occur. Find the node id by using the `tsm topology list-nodes` command.

tsm topology filestore decommission

You must use the `tsm topology filestore decommission` command to prepare a file store node or nodes for safe removal. This command puts the specified nodes into read-only mode and ensures there is no unique content on the specified nodes.

If decommissioning results in a single file store node, you must use the `--override` option or the decommission will fail.

Synopsis

```
tsm topology filestore decommission --nodes <nodeID,nodeID,...>  
[options] [global options]
```

Options

`-n, --nodes <nodeID,nodeID,...>`

Required.

List of one or more nodes to decommission, specified by node ID and separated by commas.

`--delete-filestore`

Optional.

Forces the removal of the file store, even if it has not been decommissioned. You should only use this option if the node the file store is on is in a error state and decommissioning cannot be done. Any unique files on the node will be permanently deleted.

`-o, --override`

Optional.

Overrides warnings or failures that would normally occur if removing the target File Store node would reduce the number of remaining file store nodes to one. This option cannot be used with the `--delete-filestore` option.

`--request-timeout <timeout in seconds>`

Optional.

Wait the specified amount of time for the command to finish. Default value is 1800 (30 minutes).

tsm topology filestore recommission

Use the `tsm topology filestore recommission` command to revert any decommissioned nodes back to read-write mode.

Tableau Server on Linux Administrator Guide

Synopsis

```
tsm topology filestore recommission --nodes <nodeID,nodeID,...>  
[global options]
```

Options

`-n, --nodes <nodeID,nodeID,...>`

Required.

List of one or more nodes to recommission, specified by node ID and separated by commas.

tsm topology list-nodes

Display the nodes in the cluster and (optionally) the services on each node.

Synopsis

```
tsm topology list-nodes [options] [global options]
```

Options

`-v, --verbose`

Optional.

Shows each node ID, the node role (for more information, see `set-node-role` below), the node address, and the processes on each node.

tsm topology list-ports

Display the ports in the cluster.

Synopsis

```
tsm topology list-ports [options] [global options]
```

Options

`--node-name <nodeID>`

Optional.

Specify the node to list ports for.

`--service-name`

Optional.

Specify the service to list ports for.

tsm topology nodes get-bootstrap-file

You can use the `tsm topology nodes get-bootstrap-file` command to get the bootstrap file that is required to add a new node to the cluster.

Important: The bootstrap file contains a copy of the master keystore file used for encrypting the configuration secrets. The file can also embedded credentials which are valid for a pre-determined amount of time (see `tabadmincontroller.auth.expiration.minutes`) and serve as a session cookie. We strongly recommend that you take additional measures to secure the bootstrap file.

The following command set provides an example method to encrypt the bootstrap file output. This method is similar to the encryption process described in more detail at [Securing secrets for import and export operations](#).

Note however, the method here must be passed as separate arguments with trailing `&& \` operators as follows:

```
mkfifo -m 600 /tmp/secure1 && \
tsm topology nodes get-bootstrap-file --file /tmp/secure1 && \
```

Tableau Server on Linux Administrator Guide

```
gpg --symmetric --batch --yes --passphrase-file ~/.secret-  
s/pgppassphrase.txt --cipher-algo AES256 --output encrypted.enc  
< /tmp/secure1 && \  
  
rm /tmp/secure1
```

Synopsis

```
tsm topology nodes get-bootstrap-file --file <path\file>.json  
[global options]
```

Options

```
-f, --file <file>
```

Required.

Full path and file name where the configuration file will be saved. If a duplicate file exists it will be overwritten.

```
-nec, --no-embedded-credential
```

Optional.

Added in version 2019.3.

By default embedded credentials are included in the bootstrap file. Use this option if credentials should not be included in the bootstrap file. Embedded credentials are temporary, and expire based on the value of the `tabad-mincontroller.auth.expiration.minutes` configuration key, by default 120 minutes.

Note: You can disable the ability to include embedded credentials at the server level, using a configuration option. For more information, see `features.PasswordlessBootstrapInit`.

tsm topology remove-nodes

Remove nodes from the cluster.

To complete removal of a node, you also must run the `tsm pending-changes apply` command. Some scenarios require that you move or redeploy processes before removing nodes. See [Remove a Node](#).

If you remove a node and want to re-add it to the cluster, you need to first run the `obliterate` script to clean Tableau off it, then reinstall the node using the normal process for adding a new node. For more information, see [Remove Tableau Server from Your Computer and Install and Configure Additional Nodes](#).

Note: To remove a node from a cluster it must have been configured with a process at some point in the past. If you are removing a node on which you've not configured any processes, then you must add a process on it, run `tsm pending-changes apply`, and then remove the node.

Synopsis

```
tsm topology remove-nodes --nodes <nodeID,nodeID,...>
[global options]
```

Options

```
-n, --nodes <nodeID,nodeID,...>
```

Required.

Specify the node or nodes to remove. If specifying multiple nodes, separate node IDs with a comma.

tsm topology set-node-role

Set the Backgrounder and Extract Queries node roles. This determines the type of tasks that will be performed on the nodes. The following node roles are useful if you have a multi-node cluster and requires Add-on licenses. For more information, see [Workload Management through Node Roles](#).

Note: Making configurations to node roles require a restart of the server and will require some downtime. For more information, see [tsm pending-changes](#).

Synopsis

```
tsm topology set-node-role [options] [global options]
```

Options

`-n, --nodes <nodeID,nodeID,...>`

Required.

List of one or more nodes to set node roles for, specified by node ID and separated by commas and without spaces between nodes.

`-r --role <all-jobs,flows,no-flows,extract-refreshes,subscriptions,extract-refreshes-and-subscriptions,no-extract-refreshes,no-subscriptions,no-extract-refreshes-and-subscriptions,extract-queries,extract-queries-interactive>`

Required

Sets the role for the nodes specified. The valid values for this option are:

- `all-jobs`: Backgrounder will run all types of jobs.
- `flows`: Backgrounder will run only flow run jobs.
- `no-flows`: Backgrounder will not run flow run jobs.

- **extract-refreshes:** Backgrounder will run only extract refresh jobs. This includes, incremental refreshes, full refreshes, encryption and decryption of all extracts including extracts that flow outputs generate.
- **subscriptions:** Backgrounder will run only subscription jobs.
- **extract-refreshes-and-subscriptions:** Backgrounder will run extract-refreshes, encryption and decryption of all extracts including extracts that flow outputs create, and subscription jobs.
- **no-extract-refreshes:** Backgrounder will run all jobs except extract-refreshes, extract encryption and decryption including extracts created from flow outputs.
- **no-subscriptions:** Backgrounder will run all jobs except subscriptions.
- **no-extract-refreshes-and-subscriptions:** Backgrounder will run all jobs except extract-refreshes, encryption and decryption of all extracts including extracts created from flow outputs, and subscriptions.
- **extract-queries:** The nodes selected will run as all-jobs and will prioritize the processing of extract queries.
- **extract-queries-interactive:** The nodes selected will run as all-jobs and will prioritize the processing of interactive extract queries, such as those that run when a user is looking at their screen and waiting for an extract-based dashboard to load. This is an advanced setting and it should only be used if the cluster has a heavy subscription and alert job workload that causes users to experience degraded performance on viz load times that run around the same time as scheduled loads.

tsm topology set-ports

Set the ports for a service instance.

Synopsis

```
tsm topology set-ports --node-name <nodeID> --port-name <port_
name> --port-value <port_value> [options] [global options]
```

Tableau Server on Linux Administrator Guide

Options

`-i, --instance <instance_id>`

Optional.

Specifies the instance id of the service. Defaults to 0 (zero) if not specified.

`-n, --node-name <nodeID>`

Required.

Specifies the node ID of the node.

`-pn, --port-name <port_name>`

Required.

The name of the port to be set, in this format: `service_name:port_type`. If no port type is specified, the primary port is assumed. For port name syntax, see [Dynamically mapped ports](#).

`-pv, --port-value <port_value>`

Required.

The port to set.

`-r, --restart`

Optional.

Suppress the prompt for restart and restart Tableau Server when necessary.

tsm topology set-process

Set the number of instances of a process on a node. If a node already has the specified process, the number is updated to match the specified count.

- You can only set one process at a time. If you specify more than one process, any process after the first one will be silently ignored.
- You must set a process one node at a time. If you specify more than one node, the command will display an "invalid node name" error.

When you update the number of processes on nodes, you also need to apply pending changes. In most cases this also requires a server restart (you will be prompted), but there are special cases where you can make dynamic topology changes without needing to restart the server. For more information, see [Tableau Server Dynamic Topology Changes](#).

Note: For a complete list of process names, see [Tableau Server Processes](#).

Synopsis

```
tsm topology set-process --count <process_count> --node
<nodeID> --process <process_name> [global options]
```

Options

`-c, --count <process_count>`

Required.

The process count (number of instances) to set.

`-n, --node <nodeID>`

Required.

Specifies the node ID of the node on which to set the process.

```
-pr, --process <process_name>
```

Required.

The name of the process to be set.

tsm topology toggle-coordination-service

Note: Beginning with version 2020.1.0, all coordination service ensemble commands require input for a "y/n" prompt confirming that a server restart will take place. To run these commands without input, include the `--ignore-prompt` option.

You can use the `tsm topology toggle-coordination-service` command to switch between coordination service ensembles. To learn more about Coordination Service ensembles, see [Deploy a Coordination Service Ensemble](#).

In version 2020.1.0 and later, the `tsm topology deploy-coordination-service` command also switches to the new ensemble. There is no need to run this command separately.

Synopsis

```
tsm topology toggle-coordination-service [option]  
[global options]
```

Option

```
--request-timeout <timeout in seconds>
```

Optional.

Wait the specified amount of time for the command to finish. Default value is 1800 (30 minutes).

Global options

`-h, --help`

Optional.

Show the command help.

`-p, --password <password>`

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

`-s, --server https://<hostname>:8850`

Optional.

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port `8850`, and use the server name not the IP address. For example `https://<tsm_hostname>:8850`. If no server is specified, `https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see [Connecting TSM clients](#).

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm user-identity-store

You can use the `tsm user-identity-store` commands to modify the settings of the identity store for Tableau Server after the initial configuration.

The initial configuration of the identity store is part of the installation process. See [Configure Initial Node Settings](#).

For introduction to identity store concepts, see [Identity Store](#).

For LDAP/Active Directory configuration reference table, see [External Identity Store Configuration Reference](#).

- [get-group-mappings](#)
- [get-user-mappings](#)
- [list](#)
- [set-connection](#)
- [set-group-mappings](#)
- [set-user-mappings](#)
- [verify-group-mappings](#)
- [verify-user-mappings](#)

tsm user-identity-store get-group-mappings [options]

Displays identity store group mappings.

Synopsis

```
tsm user-identity-store get-group-mappings [global options]
```

tsm user-identity-store get-user-mappings [options]

Displays identity store user mappings.

Synopsis

```
tsm user-identity-store get-user-mappings [global options]
```

tsm user-identity-store list [options]

Lists user-identity configuration.

Synopsis

```
tsm user-identity-store list [options] [global options]
```

Options

-v, --verbose

Optional.

Lists all configuration parameters.

tsm user-identity-store set-connection [options]

Sets identity store connection parameters.

Synopsis

```
tsm user-identity-store set-connection --kerbkeytab <kerb-  
keytab> [options] [global options]
```

Options

-b, --bind <username and password | Kerberos>

Optional.

Set LDAP bind type.

-d, --domain <domain>

Optional.

Tableau Server on Linux Administrator Guide

Domain name.

`-hn, --hostname <hostname>`

Optional.

The hostname of the LDAP server. You can enter a hostname or an IP address for this value. The host that you specify here will be used for user/group queries on the primary domain. In the case where user/group queries are in other domains, Tableau Server will query DNS to identify the appropriate domain controller.

`-kc, --kerbconfig <kerbconfig>`

Optional.

Kerberos configuration file path.

`-kp, --kerbprincipal <kerbprincipal>`

Optional.

Kerberos Principal.

`-kt, --kerbkeytab <kerbkeytab>`

Required.

Kerberos keytab file path.

`-l, --port <port>`

Optional.

Set LDAP Port value.

`-lp, --ldappassword <ldappassword>`

Optional.

LDAP Password.

`-lu, --ldapusername <ldapusername>`

Optional.

Set LDAP Username value.

`-n, --nickname <nickname>`

Optional.

NetBIOS name (nickname).

tsm user-identity-store set-group-mappings [options]

Sets identity store group mappings and configures LDAP directories that implement an arbitrary or custom schema.

Synopsis

```
tsm user-identity-store set-group-mappings [options]
[global options]
```

Options

`-b, --basefilter <groupbasefilter>`

Optional.

Set group BaseFilter value.

`-cn, --classnames <group_classnames>`

Optional.

Override default user classname values (contains "group" string) with the values you set here. You can provide multiple classnames separated by commas.

Tableau Server on Linux Administrator Guide

`-d, --description <description>`

Optional.

Group description.

`-e, --groupemail <groupemail>`

Optional.

Group email value.

`-m, --member <member>`

Optional.

Set the group members.

`-n, --groupname <groupname>`

Optional.

Name of the group.

tsm user-identity-store set-user-mappings [options]

Sets identity store user mappings and configures LDAP directories that implement an arbitrary or custom schema.

Synopsis

```
tsm user-identity-store set-user-mappings --certificate <certificate> [options] [global options]
```

Options

`-c, --certificate <certificate>`

Optional.

Users' certificate file location.

`-cn,--classnames <user_classnames>`

Optional.

Override default user classname values ("user" and "inetOrgPerson") with the values you set here. You can provide multiple classnames separated by commas.

`-dn,--displayname <displayname>`

Optional.

Display name of the user.

`-e,--email <email>`

Optional.

Users' email address.

`-jp,--jpegphoto <jpegfile>`

Optional.

Users' jpeg image location.

`-m,--memberof <groupname>`

Optional.

Group that the user is a member of.

`-t,--thumbnail <thumbnail>`

Optional.

Users' thumbnail location.

```
-ub, --basefilter <userbasefilter>
```

Optional.

Users' BaseFilter.

```
-uu, --ldapusername <ldapusername>
```

Optional.

User name.

tsm user-identity-store verify-group-mappings [options]

Validates configuration for LDAP group mapping.

Synopsis

```
tsm user-identity-store verify-group-mappings --verify <group_
name> [global options]
```

Options

```
-v, --verify <group_name>
```

Optional.

Name of group to search for.

tsm user-identity-store verify-user-mappings [options]

Validates configuration for LDAP user mapping.

Synopsis

```
tsm user-identity-store verify-user-mappings --verify <user_
name> [global options]
```

Options

`-v, --verify <user_name>`

Optional.

Name of user to search for.

Global options

`-h, --help`

Optional.

Show the command help.

`-p, --password <password>`

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

`-s, --server https://<hostname>:8850`

Optional.

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port `8850`, and use the server name not the IP address. For example `https://<tsm_hostname>:8850`. If no server is specified, `https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Tableau Server on Linux Administrator Guide

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see [Connecting TSM clients](#).

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm version

You can use the `tsm version` command to get the versions of TSM and Tableau Server.

Synopsis

```
tsm version [global options]
```

Global options

`-h, --help`

Optional.

Show the command help.

`-p, --password <password>`

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

`-s, --server https://<hostname>:8850`

Optional.

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port 8850, and use the server name not the IP address. For example `https://<tsm_hostname>:8850`. If no server is specified, `https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see [Connecting TSM clients](#).

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm File Paths

Certain `tsm` commands read files from or write files to default locations. These default locations are determined by `basefilepath` variables defined for each command. You can use `tsm` to view the current value of the variables, and to change the locations.

Default locations for files

During the `tsm maintenance backup`, `restore`, `send-logs`, and `ziplogs` processes, and the `tsm sites export` and `sites import` processes, Tableau Server uses default locations for the files created or used by these commands.

For details on disk space requirements for backing up Tableau Server, see [Disk Space Usage for Backup](#).

By default:

Tableau Server on Linux Administrator Guide

- **tsm maintenance commands:**
 - **backup**—The backup `.tsbak` file is created in a temporary location in the data directory on the initial node and then saved in:

```
/var/opt/tableau/tableau_server-  
/data/tabsvc/files/backups
```

- **restore**—The restore process restores a backup file from:

```
/var/opt/tableau/tableau_server-  
/data/tabsvc/files/backups
```

- **send-logs**—The send-logs sends the logs file from:

```
/var/opt/tableau/tableau_server-  
/data/tabsvc/files/backups
```

- **ziplogs**—The ziplogs file is generated in:

```
/var/opt/tableau/tableau_server/data/tabsvc/files/log-  
archives
```

- **tsm sites**

- **export**—The export `.zip` file is generated to the following directory:

```
/var/opt/tableau/tableau_server-  
/data/tabsvc/files/siteexports
```

- **import**—During the import process, Tableau Server looks for files in:

```
/var/opt/tableau/tableau_server-  
/data/tabsvc/files/siteimports
```

Get the current file location

You can see the current file location for a specific command using `tsm configuration get`:

- For `tsm` maintenance commands:

- backup, restore, and send-logs:

```
tsm configuration get -k basefilepath.backuprestore
```

- ziplogs:

```
tsm configuration get -k basefilepath.log_archive
```

- For `tsm` sites commands:

- export

```
tsm configuration get -k basefilepath.site_export.-
exports
```

- import

```
tsm configuration get -k basefilepath.site_import.-
exports
```

Change the current file location

You can change the expected file locations using the `tsm configuration set` command to update the `basefilepath` variables. For details about specific base file paths, see `tsm configuration set` Options.

Changing a `basefilepath` variable does not move existing files from the original directory to the new directory. If you want existing backup, restore, log files, or site export or import files to reside in the new directory you specify, you must move them manually. You are responsible for creating the new location and for setting the correct permissions to allow

tsm access to any files that will be placed there, and to the directory structure containing those files. For more information about permissions and tsm, see Files and Permissions in TSM.

The `tsm maintenance backup` command assembles the backup in a temporary location in the data directory before saving the backup file to the location specified by the `basefilepath.backuprestore` variable. Changing the `basefilepath` does not impact where the `tsm maintenance backup` command assembles the backup file.

You cannot specify a `basefilepath.backuprestore` location at the root of a drive. It must be in a folder.

- For tsm maintenance commands:
 - To change the backup, restore, or send-logs directory, run the following command:

```
tsm configuration set -k basefilepath.backuprestore -v  
"/new/directory/path"
```

- To change the ziplogs directory:

```
tsm configuration set -k basefilepath.log_archive -v  
"/new/directory/path"
```

- For tsm sites commands:
 - To change the sites export directory:

```
tsm configuration set -k basefilepath.site_export.-  
exports -v "/new/directory/path"
```

- To change the sites import directory:

```
tsm configuration set -k basefilepath.site_import.-  
exports -v "/new/directory/path"
```

After you change a default file location you need to do the following:

1. Apply pending changes:

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

2. Stop Tableau Server:

```
tsm stop
```

3. Restart the TSM Controller (as *tableau* system account):

```
sudo su -l tableau -c "systemctl --user restart tabad-  
mincontroller_0.service"
```

4. Wait several minutes for the controller to restart. You can confirm the controller has restarted with this command:

```
tsm status -v
```

When you can run that command and the Tableau Server Administration Controller is listed as 'running' the controller has restarted.

5. Start Tableau Server:

```
tsm start
```

Migrate from Tabadmin to the TSM CLI

The Tableau Services Manager (TSM) command-line interface (CLI) replaces the tabadmin CLI in Tableau Server on Linux, and in Tableau Server on Windows version 2018.2. This page maps tabadmin commands to TSM commands to help you to migrate to the TSM CLI.

To learn more about the TSM CLI, see [tsm Command Line Reference](#).

Looking for tabadmin commands for Tableau Server on Windows version 2018.1 and earlier? See [tabadmin Commands](#).

Tabadmin commands with a corresponding TSM CLI command

The following table shows which tabadmin commands correspond to commands available in the TSM CLI.

Command Description	Tabadmin Command(s)	Comparable TSM CLI Command
Activate a license	<code>tabadmin activate --activate</code>	<code>tsm licenses activate</code>
Deactivate licenses	<code>tabadmin activate --return</code>	<code>tsm licenses deactivate</code>
Activate a trial license	<code>tabadmin activate --trial</code>	<code>tsm licenses activate -trial</code>
Create a backup of the data managed by Tableau Server	<code>tabadmin backup</code>	<code>tsm maintenance backup</code> A backup created using TSM does not include any server configuration data. There is no option to include server configuration data.

Clear the server cache	<code>tabadmin clearcache</code>	<code>tsm maintenance cleanup -r</code>
Clean up temporary files and old log files	<code>tabadmin cleanup</code>	<code>tsm maintenance cleanup</code> Note: This command was added in version 10.5.1
Update the server configuration with any changes you've made	<code>tabadmin configure</code>	<code>tsm pending-changes apply</code>
Customize the server name and logos	<code>tabadmin customize</code>	<code>tsm customize</code>
Enable access to the repository	<code>tabadmin dbpass</code>	<code>tsm data-access repository-access enable</code>
Disable access to the repository	<code>tabadmin dbpass --disable</code>	<code>tsm data-access repository-access disable</code>
Set a file store instance to read-only mode	<code>tabadmin decommission</code>	<code>tsm topology filestore decommission</code>
Delete one or more Web Data Connectors (WDCs) from Tableau Server	<code>tabadmin delete_web-dataconnector</code>	<code>tsm data-access web-data-connectors delete</code> To learn more, see Web Data Connectors in Tableau Server.
Add a Web Data Connector (WDC) to	<code>tabadmin import_web-dataconnector</code>	<code>tsm data-access web-data-connectors add</code>

Tableau Server on Linux Administrator Guide

Tableau Server	and <code>tabadmin whitelist_web-dataconnector</code>	Note: TSM does not support importing WDCs, instead it lets you add WDCs to an allowlist. To learn more, see Web Data Connectors in Tableau Server .
List Web Data Connectors (WDCs) used by Tableau Server	<code>tabadmin list_web-dataconnectors</code>	<code>tsm data-access web-data-connectors list</code> To learn more, see Web Data Connectors in Tableau Server .
Export a site from Tableau Server	<code>tabadmin exportsite</code>	<code>tsm sites export</code>
Initiate a repository failover	<code>tabadmin fail-overrepository</code>	<code>tsm topology failover-repository</code>
Get a configuration option	<code>tabadmin get</code>	<code>tsm configuration get</code>
Get the OpenID redirect URL	<code>tabadmin get_openid_redirect_url</code>	<code>tsm authentication openid get-redirect-url</code>
Import site .csv files into Tableau Server	<code>tabadmin importsite</code>	<code>tsm sites import</code>
Import a site into Tableau Server using .csv files	<code>tabadmin importsite_verified</code>	<code>tsm sites import-verified</code>
Display license information for	<code>tabadmin licenses</code>	<code>tsm licenses list</code>

Tableau Server		Note: For more information about the output of this command, see View Server Licenses .
Move a file store from read-only mode to an active read/write state	<code>tabadmin recommission</code>	<code>tsm topology filestore recommission</code>
Regenerate internal security tokens	<code>tabadmin regenerate_internal_tokens</code>	<code>tsm security regenerate-internal-tokens</code>
Register Tableau Server	<code>tabadmin register</code>	<code>tsm register</code>
Rebuild the search index for Tableau Server	<code>tabadmin reindex</code>	<code>tsm maintenance reindex-search</code>
Reset the Tableau Server administrator account	<code>tabadmin reset</code>	<code>tsm reset</code> Note: Added in version 2018.1.
Stop and restart all Tableau Server processes	<code>tabadmin restart</code>	<code>tsm restart</code>
Restore from a Tableau Server backup file	<code>tabadmin restore</code>	<code>tsm maintenance restore</code> The restore command does not restore any server configuration data. This is true whether you are

Tableau Server on Linux Administrator Guide

		using a backup created with TSM or a backup created with tabadmin.
Set a configuration option	tabadmin set	tsm configuration set
Activate or suspend a site	tabadmin sitestate	tsm sites unlock
Start all Tableau Server processes	tabadmin start	tsm start
Get the status of Tableau Server and server processes	tabadmin status	tsm status
Stop all Tableau Server processes	tabadmin stop	tsm stop
Create an archive (.zip) file with Tableau Server log files	tabadmin ziplogs	tsm maintenance ziplogs The default behavior of the ziplogs command has changed: with tsm, the command collects up to the last two days of log files by default. The tabadmin ziplogs command collected up to seven days of log files. For more information, see tsm maintenance ziplogs.

Tabadmin commands with no corresponding TSM CLI command

The following table lists the tabadmin commands for which a comparable TSM CLI command is not available.

Command Description	Tabadmin Command	Notes
Add or remove a user from the system administrator group	<code>tabadmin administrator</code>	You can use the Tableau Server REST API Add User to Group and Remove User from Group methods to add or remove a user from the system administrator group.
Create a new key to encrypt sensitive information stored in the repository	<code>tabadmin assetkeys</code>	Use the <code>tsm security regenerate-internal-tokens</code> command to create or regenerate secrets and master keys.
Specify whether Tableau Server starts at system start-up time	<code>tabadmin autostart</code>	Tableau Server returns to the state it was in prior to a system restart. If it was running, it will restart. If it was stopped it will be stopped after the system starts.
Identify a second server node for backup	<code>tabadmin fail-overprimary</code>	TSM does not have primary nodes, so a TSM equivalent to this command is not needed.
Manage credentials for delegated data access on Tableau Server	<code>tabadmin manage_global_credentials</code>	We recommend that you use Kerberos delegation to Apache Impala for global credential management. To learn more, see Kerberos and Enable Kerberos Delegation for Hive/Impala in the

		Tableau Community.
Reset the password for a Tableau Server account	<code>tabadmin passwd</code>	If your server uses local authentication, you can use the Tableau Server REST API Update User method to reset the password for a user account.
Reset binding between Tableau Server user ID and Open ID Connect identity provider	<code>tabadmin reset_openid_sub</code>	
Determine whether your environment meets the minimum requirements to run Tableau Server	<code>tabadmin validate</code>	
Verify that a backup of the Tableau Server repository will restore successfully	<code>tabadmin verify_database</code>	The <code>tsm maintenance backup</code> command automatically verifies that a backup will restore correctly unless you use the <code>--skip-verification</code> parameter.
Prepare VizQL processes for fast load times after a Tableau Server restart	<code>tabadmin warmup</code>	The <code>tabadmin warmup</code> command is no longer necessary, as Tableau Server is now optimized to automatically provide fast load times after a server restart.

Entity Definitions and Templates

You can use the entity definitions and templates with the `tsm settings import` command to configure various Tableau Server settings.

Configuration File Example

This article provides an example of a complete JSON configuration file, with `gatewaySettings` and `identityStore` entities specified. In addition, a configuration key sets the gateway timeout to 900 seconds.

Your configuration file will look different depending on the options you need to set.

You might set multiple `.json` configuration files during installation. To set the values for each file in Tableau Server, you run the following command, once for each configuration file:

```
tsm settings import -f path-to-file.json
```

After you set the configuration files, run `tsm pending-changes apply` to apply the changes from all of the `.json` files you've set.

```
{
  "configEntities": {
    "gatewaySettings": {
      "_type": "gatewaySettingsType",
      "port": 80,
      "publicHost": "localhost",
      "publicPort": 80
    },
    "identityStore": {
      "_type": "identityStoreType",
```

```
        "type": "local",
        "domain": "example.lan",
        "nickname": "EXAMPLE"
    }
},
"configKeys": {
    "gateway.timeout": "900"
}
}
```

Entities vs keys

As shown in the example above, there are two classes of configuration parameters: `configEntities` and `configKeys`.

configEntities

Certain types of configuration are done through entity sets that map to specific scenarios, such as the identity store and gateway configurations. When you pass a set of `configEntities` with the `tsm settings import -f path-to-file.json` command, TSM validates the configuration. If values passed are invalid, TSM will provide an error. This enables you to make changes during the configuration process, rather than experience a configuration failure at initialization or run time.

Entities can be set only by including a `configEntities` block in a `.json` file.

Important: All files that are referenced in `configEntities` must be located on the local computer. Do not specify UNC paths.

configKeys

Entities cover only a small portion of the configuration values that can be set. Hundreds of keys correspond to parameters stored in `.yaml` files. Tableau Server uses these parameters to store all of the configuration information for all services.

You can set individual keys with the `tsm configuration` command. But during deployment, setting them along with other configuration scenarios in JSON files, as shown above, is more convenient.

Unlike `configEntities`, `configKeys` are not validated.

Note: We do not recommend setting parameters that are not documented in `tsm configuration set Options`.

gatewaySettings Entity

You must configure the gateway settings for the Tableau Server computer.

Use the configuration file template below to create a json file. After you have filled in the options with the appropriate values, pass the json file and apply settings with the following commands:

```
tsm settings import -f /path/to/file.json  
  
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

Gateway settings

The gateway settings in the template below specify the HTTP settings for Tableau Server. We recommend using SSL/TLS. Tableau Server is hard-coded to use port 443 for SSL/TLS. Therefore, if you enable SSL, you do not need to update the `gatewaySettings` entity.

Tableau Server on Linux Administrator Guide

Configuration template

Use this template to configure the gateway settings.

Important: All entity options are case sensitive.

For more explanation about configuration files, entities, and keys see Configuration File Example.

```
{
  "configEntities": {
    "gatewaySettings": {
      "_type": "gatewaySettingsType",
      "port": 80,
      "sslRedirectEnabled": true,
      "publicHost": "localhost"
    }
  }
}
```

Configuration file reference

This table includes all of the options that can be included with the "gatewaySettings" entity set.

`_type`

Required.

Value: "gatewaySettingsType"

Do not change.

`port`

Specifies HTTP port. Default is port 80.

`sslRedirectEnabled`

Options: `true` or `false`.

`publicHost`

Specifies host name for http/s service.

`trustedIPs`

Specifies trusted IP addresses that communicate with Tableau Server. Trusted IP addresses include upstream proxy servers and servers that are used for trusted authentication with Tableau Server. See [Configuring Proxies for Tableau Server and Add Trusted IP Addresses or Host Names to Tableau Server](#).

If you are running Tableau Server in a cluster then all other nodes of the cluster will automatically be included in the corresponding configuration file that this entity updates. Therefore, if you specify a new value for `trustedIPs`, then you must include the IP addresses for the other nodes in the value.

This option takes a list of strings, which requires passing each IP or host in quotes, separated by a comma (no space) and within brackets. For example:

```
["192.168.1.101", "192.168.1.102", "192.168.1.103"] or ["web-  
serv1", "webserv2", "webserv3"].
```

`trustedHosts`

Specifies trusted IP addresses that communicate with Tableau Server. Typically, this value contains a list of upstream proxy servers. The values in `trustedHosts` are used to determine client request targets.

If you are running Tableau Server in a cluster then all other nodes of the cluster will automatically be included in the corresponding configuration file that this entity updates. Therefore, if you specify a new value for `trustedIPs`, then you must include the IP addresses for the other nodes in the value.

This option takes a list of strings, which requires passing each IP or host in quotes, separated by a comma (no space) and within brackets. For example:

```
["192.168.1.101", "192.168.1.102", "192.168.1.103"] or ["web-  
serv1", "webserv2", "webserv3"].
```

identityStore Entity

Tableau Server requires an identity store to store user and group information. Review Authentication and Identity Store topics before configuring the identity store for the first time. After you have installed the identity store on Tableau Server, you cannot change it without reinstalling the server.

Important: All entity options are case sensitive.

Before you begin

Review the following information:

- If you will not be using the local identity store, then you will be using some version of LDAP. In this case, work with your directory/LDAP administrator to configure Tableau Server for your LDAP schema and bind requirements.
- Tableau Server configuration is optimized for Active Directory. If you are installing into Active Directory, we recommend configuring the identity store with Configure Initial Node Settings.
- LDAP bind is independent of user authentication. For example, you can configure Tableau Server to use simple bind to authenticate to the LDAP directory and then configure Tableau Server to authenticate users with Kerberos after installation.
- Do not connect to LDAP with simple bind over an unsecured connection. By default, LDAP with simple bind sends data in cleartext. Use LDAPS to encrypt traffic with simple bind. See [Configure Encrypted Channel to LDAP External Identity Store](#).
- To use Kerberos authentication for the LDAP bind with Tableau Server service, then you'll need a keytab file for GSSAPI bind, as described in the sections below. See also, [Understanding Keytab Requirements](#). In the context of Kerberos, GSSAPI bind is all you need during the base installation of Tableau Server. After you install the server, you can then [configure Kerberos for user authentication](#) and [Kerberos](#)

[delegation to data sources](#).

- In this topic, we make the distinction between *LDAP* (the protocol for connecting to directory services) and an *LDAP server* (an implementation of a directory service). For example, `slapd` is an LDAP server that is part of the OpenLDAP project.
- Validate the LDAP configuration before initializing the server, see [Configure Initial Node Settings](#).
- Import JSON configuration files only as part of the initial configuration. If you need to make LDAP changes after you have imported the JSON configuration file and initialized Tableau Server, do not attempt to re-import the JSON file. Rather, make individual key changes with native `tsm` commands or with `tsm configuration set`. See [External Identity Store Configuration Reference](#).

Configuration templates

The JSON templates in this section are used to configure Tableau Server with different identity store scenarios. Unless you're configuring a local identity store, you will need to select and edit a configuration file template that is specific to your LDAP environment.

Consider using the [Tableau Identity Store Configuration Tool](#) to help generate your LDAP JSON configuration file. The tool itself is not supported by Tableau. However, using a JSON file created by the tool instead of creating a file manually does not change the supported status of your server.

Select an identity store configuration template to edit:

- Local
- LDAP - Active Directory
- OpenLDAP - GSSAPI Bind
- OpenLDAP - Simple Bind

For more explanation about configuration files, entities, and keys see [Configuration File Example](#).

Tableau Server on Linux Administrator Guide

Local

Configure local as the identity store type if your organization does not already have an Active Directory or LDAP server for user authentication. When you select local as the identity store type, you use Tableau Server to create and manage users.

An alternative way to configure Tableau Server for local identity store is to run Setup GUI and select "Local" during the installation process. See [Configure Initial Node Settings](#).

```
{
  "configEntities": {
    "identityStore": {
      "_type": "identityStoreType",
      "type": "local"
    }
  }
}
```



Important

The LDAP configuration templates below are examples. The templates, as presented, will not configure LDAP connectivity in your organization. You must work with your directory administrator to edit the LDAP template values for a successful deployment.

Additionally, all files that are referenced in configEntities must be located on the local computer. Do not specify UNC paths.

LDAP - Active Directory

Tableau Server configuration is optimized for Active Directory. If you are installing into Active Directory, configure the identity store with [Configure Initial Node Settings](#).

An encrypted connection to Active Directory is required. See [Configure Encrypted Channel to LDAP External Identity Store](#).

If for some reason you are unable to configure the identity store to communicate with Active Directory with TSM web interface, use this JSON template to configure Tableau Server to connect to Active Directory. This template uses GSSAPI (Kerberos) bind to authenticate Tableau Server service to Active Directory. Tableau Server includes support for Active Directory schema. Therefore, if you set the `"directoryServiceType"` option to `"activedirectory"` then you do not need to provide schema info in the `"identityStoreSchemaType"` option.

If you are installing Tableau Server for Linux into Active Directory, and the computer where you are installing Tableau Server is already joined to the domain, then the computer will already have a Kerberos configuration file and a keytab file. Strictly speaking, you can use these files for GSSAPI bind, but we don't recommend using them. Instead, contact your Active Directory administrator and request a keytab specifically for the Tableau Server service.

```
{
  "configEntities":{
    "identityStore": {
      "_type": "identityStoreType",
      "type": "activedirectory",
      "domain": "your-domain.lan",
      "nickname": "YOUR-DOMAIN-NICKNAME",
      "directoryServiceType": "activedirectory",
      "bind": "gssapi",
      "kerberosKeytab": "<path to local key tab file>",
      "kerberosConfig": "/etc/krb5.conf",
      "kerberosPrincipal": "your-principal@YOUR.DOMAIN"
    }
  }
}
```

We recommend binding to Active Directory with GSSAPI. However, you can connect with simple bind and LDAPS. To connect with simple bind, change `bind` to `simple`, remove the three Kerberos entities, and add the `port/sslPort`, `username`, and `password` options. The following example shows Active Directory with simple bind json.

```
{
  "configEntities":{
    "identityStore": {
      "_type": "identityStoreType",
      "type": "activedirectory",
      "domain": "your-domain.lan",
      "nickname": "YOUR-DOMAIN-NICKNAME",
      "directoryServiceType": "activedirectory",
      "hostname": "optional-ldap-server",
      "sslPort": "636",
      "bind": "simple",
      "username": "username",
      "password": "password"
    }
  }
}
```

OpenLDAP - GSSAPI bind

Use the template below to configure OpenLDAP with GSSAPI bind. Do not use this template if your organization is running Active Directory. If you are installing into Active Directory, use the template above, LDAP - Active Directory.

In most cases, organizations that use OpenLDAP with GSSAPI (Kerberos) will use a keytab file to store credentials. In the following example, a keytab file is used for authentication credentials.

However, you can provide credentials through the `username` and `password` entities.

You can also specify both a keytab and a username and password pair. In this case, Tableau Server will attempt to use the keytab, but if authentication fails for any reason it will fallback and use the username and password credentials.

```
{
  "configEntities":{
    "identityStore": {
      "_type": "identityStoreType",
      "type": "activedirectory",
      "domain": "your-domain.lan",
      "nickname": "YOUR-DOMAIN-NICKNAME",
      "directoryServiceType": "openldap",
      "bind": "gssapi",
      "kerberosKeytab": "<path to local key tab file>",
      "kerberosConfig": "/etc/krb5.conf",
      "kerberosPrincipal": "your-principal@YOUR.DOMAIN",
      "identityStoreSchemaType": {
        "userBaseFilter": "(objectClass=inetOrgPerson)",
        "userUsername": "user",
        "userDisplayName": "displayname",
        "userEmail": "email",
        "userCertificate": "certificate",
        "userThumbnail": "thumbnail",
        "userJpegPhoto": "photo",
        "groupBaseFilter": "(objectClass=groupofNames)",
        "groupName": "groupname",
        "groupEmail": "groupemail",
        "groupDescription": "groupdescription",
        "member": "member",
        "distinguishedNameAttribute": "",
        "serverSideSorting": "",
        "rangeRetrieval": "",
        "userClassNames": ["inetOrgPerson","someClass2"],
        "groupClassNames": ["groupOfU-
niqueNames1","groupOfUniqueNames2"]
      }
    }
  }
}
```

```
    }  
  }  
}
```

OpenLDAP - Simple bind

```
{  
  "configEntities":{  
    "identityStore": {  
      "_type": "identityStoreType",  
      "type": "activedirectory",  
      "domain": "my.root",  
      "nickname": "",  
      "hostname": "optional-ldap-server",  
      "port": "389",  
      "directoryServiceType": "openldap",  
      "bind": "simple",  
      "username": "cn=username,dc=your,dc=domain",  
      "password": "password",  
      "identityStoreSchemaType": {  
        "userBaseFilter": "(objectClass=inetOrgPerson)",  
        "userUsername": "user",  
        "userDisplayName": "displayname",  
        "userEmail": "email",  
        "userCertificate": "certificate",  
        "userThumbnail": "thumbnail",  
        "userJpegPhoto": "photo",  
        "groupBaseFilter": "(objectClass=groupofNames)",  
        "groupName": "groupname",  
        "groupEmail": "groupemail",  
        "groupDescription": "groupdescription",  
        "member": "member",  
        "distinguishedNameAttribute": "",  
      }  
    }  
  }  
}
```

```

        "serverSideSorting": "",
        "rangeRetrieval": "",
        "userClassNames": ["inetOrgPerson", "someClass2"],
        "groupClassNames": ["groupOfU-
niqueNames1", "groupOfUniqueNames2"]
    }
}
}
}

```

Configuration template reference

Shared identity store options

`type`

Where you want to store user identity information. Either `local` or `activedirectory`. (If you want to connect to any LDAP server, select `activedirectory`.)

`domain`

The domain of the computer where you installed Tableau Server.

`nickname`

The nickname of the domain. This is also referred to as the NetBIOS name in Windows environments.

The `nickname` option is required for all LDAP entities. If your organization does not require a nickname/NetBIOS, then pass a blank key, for example: `"nickname": ""`.

LDAP GSSAPI bind options

`directoryservicetype`

The type of directory service that you want to connect to. Either `activedirectory` or `openldap`.

`kerberosConfig`

Tableau Server on Linux Administrator Guide

The path to the Kerberos configuration file on the local computer. If you are installing into Active Directory, we don't recommend using the existing Kerberos configuration file or keytab file that may already be on the domain-joined computer. See Identity Store.

`kerberosKeytab`

The path to the Kerberos keytab file on the local computer. It is recommended that you create a keytab file with keys specifically for Tableau Server service and that you do not share the keytab file with other applications on the computer. For example, on Linux, you might place the keytab file in the `/var/opt/tableau/keytab` directory.

`kerberosPrincipal`

The service principal name for Tableau Server on the host machine. The keytab must have permission for this principal. Do not use the existing system keytab at `/etc/krb5.keytab`. Rather, we recommend that you register a new service principal name. To see principals in a given keytab, run the `klist -k` command. See Understanding Keytab Requirements.

LDAP simple bind options

`directoryservicetype`

The type of directory service that you want to connect to. Either `activedirectory` or `openldap`.

`hostname`

The hostname of the LDAP server. You can enter a hostname or an IP address for this value. The host that you specify here will be used for user/group queries on the primary domain only. If user/group queries are in other domains (not in the primary domain), Tableau Server will not use this value, but instead will query DNS to identify the appropriate domain controller.

`port`

Use this option to specify the non-secure port of the LDAP server. Plaintext is usually 389.

`sslPort`

Use this option to enable LDAPS. Specify the secure port of the LDAP server. LDAPS is usually port 636. To use LDAPS you must also specify hostname option. See [Configure Encrypted Channel to LDAP External Identity Store](#).

`username`

The user name that you want to use to connect to the directory service. The account that you specify must have permission to query the directory service. For Active Directory, enter the username, for example, `jsmith`. For LDAP servers, enter the distinguished name (DN) of the user that you want to use to connect. For example, you might enter `cn=username,dc=your-local-domain,dc=lan`.

`password`

The password of the user that you want to use to connect to the LDAP server.

Shared LDAP options

The following options can be set for generic LDAP, OpenLDAP, or Active Directory implementations.

`bind`

The way that you want to authentication communication from the Tableau Server service to the LDAP directory service. Enter `gssapi` for GSSAPI (Kerberos).

`domain`

In Active Directory environments, specify the domain where Tableau Server is installed, for example, "example.lan".

For non-AD LDAP: the string you enter for this value is displayed in the "Domain" column of user management tools. You can enter an arbitrary string, but the key cannot be blank.

`root`

LDAP only. Do not specify for Active Directory.

Tableau Server on Linux Administrator Guide

If you do not use a dc component in the LDAP root or you want to specify a more complex root you need to set the LDAP root. Use the "o=my,u=root" format. For example, for the domain, `example.lan`, the root would be "o=example,u=lan".

`membersRetrievalPageSize`

This option determines the maximum number of results returned by an LDAP query. For example, consider a scenario where Tableau Server is importing an LDAP group that contains 50,000 users. Attempting to import such a large number of users in a single operation is not a best practice. When this option is set to 1500, Tableau Server imports the first 1500 users in the first response. After those users are processed, Tableau Server requests the next 1500 users from the LDAP server, and so forth. We recommend that you modify this option only to accommodate the requirements of your LDAP server.

`identityStoreSchemaType` options

If you configure an LDAP connection to an LDAP server, you can enter schema information specific to your LDAP server in the `identityStoreSchemaType` object.

Important If you are connecting to Active Directory ("`directoryServiceType`": "`activedirectory`"), then do not configure these options.

`userBaseFilter`

The filter that you want to use for users of Tableau Server. For example, you might specify an object class attribute and an organization unit attribute.

`userUsername`

The attribute that corresponds to user names on your LDAP server.

`userDisplayName`

The attribute that corresponds to user display names on your LDAP server.

`userEmail`

The attribute that corresponds to user email addresses on your LDAP server.

`userCertificate`

The attribute that corresponds to user certificates on your LDAP server.

`userThumbnail`

The attribute that corresponds to user thumbnail images on your LDAP server.

`userJpegPhoto`

The attribute that corresponds to user profile images on your LDAP server.

`groupBaseFilter`

The filter that you want to use for groups of users of Tableau Server. For example, you might specify an object class attribute and an organization unit attribute.

`groupName`

The attribute that corresponds to group names on your LDAP server.

`groupEmail`

The attribute that corresponds to group email addresses on your LDAP server.

`groupDescription`

The attribute that corresponds to group descriptions on your LDAP server.

`member`

The attribute that describes the list of users in a group.

`distinguishedNameAttribute`

The attribute that stores the distinguished names of users. This attribute is optional, but it greatly improves the performance of LDAP queries.

`serverSideSorting`

Whether the LDAP server is configured for server-side sorting of query results. If your LDAP server supports server-side sorting, set this option to `true`. If you are unsure whether your LDAP server supports this, enter `false`, as misconfiguration may cause errors.

`rangeRetrieval`

Whether the LDAP server is configured to return a range of query results for a request. This means that groups with many users will be requested in small sets instead of all at once. LDAP servers that support range retrieval will perform better for large queries. If your LDAP server supports range retrieval, set this option to `true`. If you are unsure whether your LDAP server supports range retrieval, enter `false`, as misconfiguration may cause errors.

`groupClassNames`

By default Tableau Server looks for LDAP group object classes containing the string “group”. If your LDAP group objects do not fit the default class name, override the default by setting this value. You can provide multiple classnames separated by commas. This option takes a list of strings, which requires passing each class in quotes, separated by a comma (no space) and within brackets. For example:

```
["basegroup", "othergroup"].
```

`userClassNames`

By default Tableau Server looks for LDAP user object classes containing the string “user” and “inetOrgPerson”. If your LDAP user objects do not use these default class names, override the default by setting this value. You can provide multiple classnames separated by commas. This option takes a list of strings, which requires passing each class in quotes, separated by a comma (no space) and within brackets. For example:

```
["userclass1", userclass2"].
```

Importing the JSON file

After you have finished editing the JSON file, pass the file and apply settings with the following commands:

```
tsm settings import -f path-to-file.json
```

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

kerberosSettings Entity

Before you configure Kerberos authentication, review Kerberos Requirements.

Use the configuration file template below to create a json file. After you have filled in the options with the appropriate values, pass the json file and apply settings with the following commands:

```
tsm settings import -f /path/to/file.json
```

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

Configuration template

Use this template to configure Kerberos settings.

Important: All entity options are case sensitive.

For more explanation about configuration files, entities, and keys see Configuration File Example.

After you have finished with the initial configuration of Kerberos authentication, use the `tsm authentication kerberos <commands>` sub-category to set additional values.

```
{
  "configEntities": {
    "kerberosSettings": {
      "_type": "kerberosSettingsType",
      "enabled": "true",
      "keytabFile": "/path/to/keytab_file"
    }
  }
}
```

Configuration file reference

The following list includes all of the options that can be included with the `"kerberosSettings"` entity set.

Option

Value

`enabled`

Options: `true` or `false`.

Enables Kerberos authentication.

`keytabFile`

Required.

Path to valid Kerberos keytab file.

`dBClasses`

Comma-separated list of database classes for global credentials. May be required for connecting to Cloudera data sources.

mutualSSLSettings Entity

Before you configure mutual SSL, review [Configure SSL for External HTTP Traffic to and from Tableau Server](#).

The `mutualSSLSettings` entity combines both SSL and mutual SSL configuration. Mutual SSL requires that external SSL has been enabled and properly configured.

The TSM entities use JSON and key-value pairs. Use the configuration file template below to create a `.json` file. Provide values for the appropriate keys for your environment, and then pass the `.json` file to Tableau Server with the following commands:

```
tsm settings import -f <path-to-file.json>

tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

Configuration template

Use this template to configure mutual SSL settings.

Important: All entity options are case sensitive.

For more explanation about configuration files, entities, and keys see [Configuration File Example](#).

```
{
  "configEntities": {
    "mutualSSLSettings": {
```

Tableau Server on Linux Administrator Guide

```
    "_type": "mutualSSLSettingsType",
    "sslEnabled": true,
    "proxyLogin": false,
    "clientCertRequired": true,
    "caCertFile": "required",
    "keyFileName": "required",
    "keyPassphrase": "",
    "chainFile": "",
    "revocationFile": "",
    "redirect": false,
    "fallbackToPassword": true,
    "protocols": "",
    "cipherSuite": "",
    "forceHttpsForPublicEmbed": false
  }
}
```

Configuration file reference

sslEnabled

Enable SSL. This is a prerequisite to enabling mutual SSL.

clientCertRequired (MutualSSL)

Set to true to enable mutual SSL authentication. Set to false to disable.

caCertFile (MutualSSL)

Required.

Specify the CA-issued certificate file for two-way SSL. The file path must be readable by Tableau Server.

`certFileName`

Specify the file that contains the concatenation of PEM encoded CA certificates that form the certificate chain for the server certificate.

Alternatively the referenced file can be the same as `caCertFile` when the CA certificates are directly appended to the server certificate for convenience.

`keyFileName`

If the key is not combined with the certificate, use this configuration key to point to the key file. If you have both an RSA and a DSA private key, you can configure both in parallel (for example, to also allow the use of DSA ciphers).

`keyPassphrase`

Optional. Passphrase for the certificate file. The passphrase you enter will be encrypted while at rest.

Note: If you create a certificate key file with a passphrase, you cannot reuse the SSL certificate key for SAML.

`revocationFile`

Specifies the file path for an SSL CA Certificate Revocation List (.crl) file.

`Redirect`

Default: true. Specifies whether Tableau Server should redirect http requests as https requests to the appropriate endpoint.

`clientCertMapping` (MutualSSL)

Specifies the method for retrieving the user name from the certificate.

Accepted values: ldap, upn, cn

Tableau Server on Linux Administrator Guide

- For a server using local authentication, the default setting is `upn` (User Principal Name).
- When Tableau Server authentication is configured for Active Directory (AD), the default is `ldap` (Lightweight Directory Access Protocol). This tells the server to go to AD to validate the user, and it ignores the names inside the certificate.

You can set `cn` for either authentication type to use the CN in the Subject DN in the certificate.

For more information, see [Mapping a Client Certificate to a User During Mutual Authentication](#).

`fallbackToPassword` (MutualSSL)

Set to true to give users the option to sign in to Tableau Server through their user name and password if mutual SSL authentication fails. Set to false to disallow this fallback option.

`protocols`

List the Transport Layer Security (TLS) protocol versions you want to allow or disallow.

Default value: `"all -SSLv2 -SSLv3"`

However, we recommend the using the following setting:

`"all -SSLv2 -SSLv3 -TLSv1 -TLSv1.1"`

For more information, see `tsm security external-ssl enable`. For general information, see the [Apache online documentation](#).

`cipherSuite`

List ciphers to allow or disallow for SSL.

Default value:

`"HIGH:MEDIUM:!aNULL:!MD5:!RC4:!3DES:!CAMELLIA:!IDEA:!SEED"`

See the [OpenSSL ciphers](#) page for cipher list format. Use caution when changing this option. The default values disallow ciphers that are no longer considered adequately secure.

`proxyLogin`

Default: `false`. Indicates that Tableau Server uses a proxy for SSL on sign-in only. It controls the protocol the server reports to Tableau Desktop for sign-in APIs.

`forceHTTPForPublicEmbed`

Default value: `false`. Forces the code for embedded views to use SSL.

openIDSettings Entity

Before you configure OpenID authentication, review [Requirements for Using OpenID Connect](#).

Use the configuration file template below to create a json file. After you have filled in the options with the appropriate values, pass the json file and apply settings with the following commands:

```
tsm settings import -f path-to-file.json
```

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

Configuration template

Use this template to configure OpenID settings.

Important: All entity options are case sensitive.

For more explanation about configuration files, entities, and keys see Configuration File Example.

After you have finished with the initial configuration of OIDC, use the tsm authentication openid <commands> sub-category to set additional values.

```
{
  "configEntities": {
    "openIDSettings": {
      "_type": "openIDSettingsType",
      "enabled": true,
      "clientId": "required",
      "clientSecret": "required",
      "configURL": "required if staticFile value is not set",
      "staticFile": "required if configURL value is not set",
      "externalURL": "required"
    }
  }
}
```

Configuration file reference

The following list includes all of the options that can be included with the "openIDSettings" entity set.

`_type`

Required.

Do not change.

`enabled`

Required.

Set to `true`.

`clientId`

Required.

Specifies the provider client ID that your IdP has assigned to your application. For example, "laakjwdlnaoiloadjkwaha".

`clientSecret`

Required.

Specifies the provider client secret. This is a token that is used by Tableau to verify the authenticity of the response from the IdP. This value is a secret and should be kept securely.

For example, "fwahfkjaw72123=".

`configURL`

Required.

Specifies provider configuration URL. If you do not specify a configuration URL, then delete this option and specify a path and file name for `staticFile` instead.

`staticFile`

Required.

Specifies the local path to the static OIDC discovery JSON document. If you do not specify a static file, then delete this option and specify a url for `configURL` instead.

`externalURL`

Required.

Tableau Server on Linux Administrator Guide

The URL of your server. This is typically is the public name of your server, such as `http://example.tableau.com`.

`connectionTimeout`

Optional.

Specifies connection timeout span in seconds. Default is 10.

`readTimeout`

Optional.

Specifies read timeout span in seconds. Default is 30.

`ignoreDomain`

Set this to `true` if the following are true:

- You are using email addresses as usernames in Tableau Server
- You have provisioned users in the IdP with multiple domain names
- You want to ignore the domain name portion of the `email` claim from the IdP

Before you proceed, review the user names that will be used as a result of setting this option to `true`. User name conflicts may occur. In the case of a user name conflict, the risk of information disclosure is high. See Requirements for Using OpenID Connect.

`ignoreJWK`

Set this to `true` if your IdP does not support JWK validation. In this case, we recommend authenticating communication with your IdP using mutual TLS or another network layer security protocol. Default is `false`.

`customScope`

Specifies a custom scope user-related value that you can use to query the IdP. See Requirements for Using OpenID Connect.

`idClaim`

Change this value if your IdP does not use the `subclaim` to uniquely identify users in the ID token. The IdP claim that you specify should contain a single, unique string.

`usernameClaim`

Change this value to the IdP claim that your organization will use to match user names as stored in Tableau Server.

`clientAuthentication`

Specifies custom client authentication method for OpenID Connect.

To configure Tableau Server to use the Salesforce IdP, set this value to `client_secret_post`.

`iFramedIDPEnabled`

Set to `true` to allow IdP displayed in an iFrame. The IdP must disable clickjack protection to allow iFrame presentation.

samlSettings Entity

This article contains a template and reference for configuring server-wide SAML on Tableau Server, using a configuration file with keys and values for the `samlSettings` entity. This information supplements the SAML configuration steps in Configure Server-Wide SAML.

To create a SAML configuration template and apply it to Tableau Server, you complete the following steps:

1. Review the following two sections that describe the template and how it's structured (Template categories and definitions and `samlSettings` configuration template).
2. Paste the JSON code shown in the template into a new text file, and save it using a

.json extension.

3. Use the SAML configuration entity reference to help you provide values where required.
4. Add optional key/value pairs specific to your environment. For example, if your SAML certificate key file requires a passphrase, you will need to specify the password in the `wgserver.saml.key.passphrase` parameter using the `tsm configuration set` command.
5. Pass the configuration file to Tableau Server.

Template categories and definitions

The template uses placeholders for each key value. These placeholders are categorized as follows:

- **Required:** Attributes with the `"required"` value must be replaced with valid data before you run the configuration command. Review the configuration file reference for valid values.
- **Hard-coded:** Attribute names that are prepended with an underscore (`_`), for example `"_type"` hold hard-coded values. Do not change these values.
- **Default values:** Attributes that are set to a value that is not `"required"` are default values. These are required attributes that you can change as appropriate for your environment.
- **Empty sets:** Values that are empty (`" "`) can be passed as they are, or you can provide a value for your installation.

Important: All entity options are case sensitive.

samlSettings configuration template

Paste this code into a text file and customize it for your environment, using the reference below.

```
{
  "configEntities": {
    "samlSettings": {
      "_type": "samlSettingsType",
      "enabled": true,
      "returnUrl": "required",
      "entityId": "required",
      "certFile": "required",
      "keyFile": "required",
      "idpMetadataFile": "required",
      "idpDomainAttribute": "",
      "idpUsernameAttribute": "required"
    }
  }
}
```

SAML configuration entity reference

The following table includes all of the options you can include with the "samlSettings" entity set.

idpMetadataFile

Required. The path and file name for the XML file generated by the IdP. The XML metadata must include the user name attribute (assertion).

If you completed the steps described in Configure Server-Wide SAML the value you enter here would be:

```
/var/opt/tableau/tableau_server/data/saml/<metadata-file.xml>
```

enabled

true | false

Tableau Server on Linux Administrator Guide

Required. Indicates whether SAML authentication is enabled. Do not set this option to `true` before setting other required SAML configuration options.

`returnURL`

This is typically the external URL that Tableau Server users enter in their browser to access the server, such as `https://tableau_server.example.com`. This value is used to create the ACS URL attribute when configuring the IdP.

`entityId`

Required. Service provider (in this case, Tableau Server) entity ID value.

Identifies your Tableau Server configuration to the IdP. We recommend that you enter the same value as the `returnURL` option.

`idpUsernameAttribute`

Required. In the IdP metadata, find the attribute that is used to specify user name values, and enter the name of that attribute. Default is `username`.

`certFile`

Required. Enter the location and file name of the x509 certificate (.crt) file for SAML. For example:

```
/var/opt/tableau/tableau_server/data/saml/<file.crt>
```

For more information, see [SAML Requirements and Configure Server-Wide SAML](#).

`keyFile`

Required. Specify the location of the private key (.key) file that accompanies the certificate file. For example:

```
/var/opt/tableau/tableau_server/data/saml/<file.key>
```

Note: If you are using a RSA PKCS#8 key that requires a passphrase, you must set the passphrase using a `configKey` entity (see [Configuration File Example](#)) or with `tsm configuration set`. The key for the passphrase using these methods is `wgserver.saml.key.passphrase`. The value must be a non-null string.

`idpDomainAttribute`

Optional. For organizations that use LDAP or Active Directory, this value specifies which SAML attribute Tableau Server will reference to determine the domain name. For example, if your IdP specifies the domain name in the `domain` attribute, then you would specify `domain` for this value.

If you do not provide a value for this key, the value used depends on the Tableau Server identity store setting:

- For local identity store, the `idpDomainAttribute` value is ignored.
- For Active Directory or LDAP identity stores, Tableau uses the FQDN from the configuration setting `wgserver.domain.default`.

To get the value for `wgserver.domain.default`, you can run the following command:

```
tsm configuration get --key wgserver.domain.default
```

`desktopNoSAML`

true | false

Optional. Allow users to use SAML authentication when they sign in from Tableau Desktop.

By default this is not set, so the effective behavior is equivalent to setting it to false. If single sign-on from Tableau client applications does not work with your IdP, you can set this to true to disable SAML authentication through Tableau Desktop.

Tableau Server on Linux Administrator Guide

appNoSAML

true | false

Optional. Allow using SAML to sign in from older versions of Tableau Mobile app. Devices running Tableau Mobile app version 19.225.1731 and higher ignore this option. To disable devices running Tableau Mobile app version 19.225.1731 and higher, disable SAML as a client login option on Tableau Server.

logoutEnabled

true | false

Optional. Enables single logout for users who have logged on with SAML. The default is `true`.

The IdP configuration metadata must include a single logout endpoint with POST binding.

This setting applies only for server-wide SAML

When set to `false`, Tableau Server will not attempt single logout.

logoutUrl

Optional. Enter the URL to redirect to after users sign out of the server. Setting this option requires that `logoutEnabled` is set to `true`.

By default this is the Tableau Server sign-in page. You can specify an absolute or a relative URL.

maxAuthenticationAge

Optional. Specifies the maximum number of seconds allowed between a user's authentication with the IdP and processing of the AuthNResponse message. The default value is 7200 (2 hours).

To optimize session length use the same timeout value as is set on the IdP.

`maxAssertionTime`

Optional. Specifies the maximum number of seconds, from creation, that a SAML assertion is usable. Default value is 3000 (50 minutes).

`sha256Enabled`

`true | false`

Optional. The type of signature Tableau Server will use when sending messages to the IdP. When set to `true`, Tableau Server will sign messages with the SHA 256 signature algorithm. When set to `false`, Tableau Server will sign messages with SHA 1. Default is `true`.

This option sets the signature algorithm to the following messages that Tableau Server signs:

- AuthnRequest messages when `signRequests` is enabled.
- LogoutRequest messages if `logoutEnabled` is enabled.

`signRequests`

`true | false`

Optional. Specifies whether Tableau Server will sign the AuthnRequests that are sent to the IdP. Signed requests are not always necessary for all IdPs. We recommend signing requests to ensure the most secure option when configuring SAML. To verify whether your IdP accepts signed request, inspect the IdP metadata: if `wantAuthnRequestsSigned` is set to `true`, then your IdP will accept signed requests.

Default value is `true`. To disable signed requests, set this option to `false`.

`acceptableAuthnContexts`

Optional. Sets the `AuthNContextClassRef` SAML attribute. This optional attribute enforces validation of certain authentication "contexts" in IdP initiated flows. Set a comma-separated set of values for this attribute. When this attribute is set, Tableau Server validates that the SAML response contains at least one of the values listed. If

Tableau Server on Linux Administrator Guide

the SAML response does not contain one of the configured values, authentication will be rejected, even if the user has successfully authenticated with the IdP.

Leaving this option blank will result in default behavior: any successfully authenticated SAML response will result in a user being granted a session within Tableau Server.

`iFramedIdpEnabled`

`true | false`

Optional. Default value is `false`, meaning that when users select the sign-in button on an embedded view, the IdP's sign-in form opens in a pop-up window.

When you set it to `true`, and a server SAML user who is already signed in navigates to a web page with an embedded view, the user will not need to sign in to see the view.

You can set this to `true` only if the IdP supports signing in within an `iframe`. The `iframe` option is less secure than using a pop-up, so not all IdPs support it. If the IdP sign-in page implements clickjack protection, as most do, the sign-in page cannot display in an `iframe`, and the user cannot sign in.

If your IdP does support signing in via an `iframe`, you might need to enable it explicitly. However, even if you can use this option, it disables Tableau Server clickjack protection for SAML, so it still presents a security risk.

Pass the configuration file to Tableau Server

After you have provided an appropriate value for each entity you include in the configuration template, use the following commands to pass the `.json` file and apply settings to Tableau Server.

```
tsm settings import -f path-to-file.json
```

```
tsm pending-changes apply
```

See also

After you complete the initial SAML configuration, use `tsm authentication mutual-ssl <commands>` to set additional values.

For the command-line reference for configuring SAML, see `tsm authentication saml <commands>`.

sapHanaSettings Entity

Use the `sapHanaSettings` entity to configure SAML delegation for SAP HANA.

Review [Configure SAP HANA SSO](#) before you continue.

Use the configuration file template below to create a json file. After you have filled in the options with the appropriate values, pass the json file and apply settings with the following commands:

```
tsm settings import -f path-to-file.json
```

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

SAP HANA SAML settings

The settings in the template below specify the settings for Tableau Server in a SAML delegation scenario with SAP HANA.

Configuration template

Use this template to configure trusted authentication settings.

All that are referenced in `configEntities` must be located on the local computer. Do not specify UNC paths.

Important: All entity options are case sensitive.

For more explanation about configuration files, entities, and keys see Configuration File Example.

```
{
  "configEntities": {
    "sapHanaSettings": {
      "_type": "sapHanaSettingsType",
      "enabled": "true",
      "usernameFormat": "username",
      "usernameCase": "preserve",
      "certFile": "path-to-cert_file",
      "keyFile": "path-to-key_file"
    }
  }
}
```

Configuration file reference

This table includes all of the options that can be included with the "gatewaySettings" entity set.

enabled

Required.

Values: true or false

usernameFormat

Values: username, domain_and_username, or email

Specifies user name credential format.

usernameCase

Values: lower, upper, or preserve

Specifies user input name case.

`certFile`

Specifies file path and name for the certificate file on the local computer.

For example, `"/var/opt/tableau/tableau_server/data/saml/saml_certificate.crt"`.

`keyFile`

Specifies file path and name for the certificate key on the local computer.

For example, `"/var/opt/tableau/tableau_server/data/saml/saml_key.der"`.

shareProductUsageDataSettings Entity

Before you configure this entity, see [Product usage data](#) to better understand the behavioral and usage data that Tableau collects. (Your confidential database values are never included.)

Use the configuration file template below to create a json file. After you have filled in the options with the appropriate values, pass the json file and apply settings with the following commands:

```
tsm settings import -f /path/to/file.json
```

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

Configuration template

Use this template to configure product usage data settings.

Important: All entity options are case sensitive.

For more explanation about configuration files, entities, and keys see Configuration File Example.

```
{
  "configEntities": {
    "shareProductUsageDataSettings": {
      "_type": "shareProductUsageDataSettingsType",
      "enabled": "true"
    }
  }
}
```

Configuration file reference

The following list includes all of the options that can be included with the "shareProductUsageDataSettings" entity set:

`_type`

Required value: `shareProductUsageDataSettingsType`

`enabled`

Options: `true` or `false`

The default, `true`, shares product usage data from your server with Tableau.

trustedAuthenticationSettings Entity

Before you configure trusted authentication, review Trusted Authentication.

Use the configuration file template below to create a json file. After you have filled in the options with the appropriate values, pass the json file and apply settings with the following commands:

```
tsm settings import -f /path/to/file.json
```

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

Configuration template

Use this template to configure trusted authentication settings.

Important: All entity options are case sensitive.

For more explanation about configuration files, entities, and keys see Configuration File Example.

After you have finished with the initial configuration of trusted authentication, use the `tsm authentication trusted <commands> sub-category` to set additional values.

```
{
  "configEntities": {
    "trustedAuthenticationSettings": {
      "_type": "trustedAuthenticationSettingsType",
      "trustedHosts": ["webserv1", "webserv2", "webserv3"]
    }
  }
}
```

Configuration file reference

The following list includes all of the options that can be included with the `"trustedAuthenticationSettings"` entity set.

`trustedHosts`

Required.

IP address or host names of web servers that request trusted tickets from Tableau Server.

This option takes a list of strings, which requires passing each IP or host in quotes, separated by a comma (no space) and within brackets. For example:

```
["192.168.1.101", "192.168.1.102", "192.168.1.103"] or ["web-serv1", "webserv2", "webserv3"].
```

The values you specify overwrite previous settings. Therefore, you must include the full list of hosts when you configure this value.

`tokenLength`

Optional.

The value can be set to any integer between 9 and 255, inclusive.

Determines the number of characters in each trusted ticket. The default setting of 24 characters provides 144 bits of randomness. This option is ignored unless `useV2Tickets` is set to `true`, which is not a recommended best practice.

`logLevel`

Optional.

```
all | debug | info | warn | error | fatal | off
```

Default: `info`

Specifies logging level for processes related to creating and redeeming trusted tickets. See [Change Logging Levels](#).

`timeoutInSeconds`

Optional.

Default: 180

Specifies the length of time (in seconds) to invalidate trusted tickets after they are created.

`tryCount`

Optional.

Integer.

Default: 10

Specifies the number of times to attempt to create a trusted ticket entry.

`use9DigitToken`

Optional.

`true` | `false`

Default: `false`

When set to `true`, tickets are 9 digits long (as in version 8.0 and earlier) and the option `tokenLength` is ignored. This option is intended for temporary support of legacy code.

Warning: Setting this option to `true` severely and negatively impacts the security strength of trusted ticket authentication.

`useV2Tickets`

Optional.

`true | false`

Default: `false`

Specifies whether Tableau Server should return a legacy URL format for trusted ticket requests. The legacy URL format includes a 24 character, Base64-encoded string. Beginning with Tableau Server 10.3, the URL that is returned has been updated and includes a Base64-encoded UUID and a 24 character secure random string. Only set option this to `true` if you have deployed trusted tickets with custom code that requires the legacy URL format. We recommend instead, updating your custom code to accept the new URL format.

□

web-data-connector-settings Entity

This entity is used to manage web data connector (WDC) settings. To learn more about using WDCs in Tableau Server, see [Web Data Connectors in Tableau Server](#) and [tsm data-access](#).

Use the configuration file template below to create a `.json` file. After you have filled in the options with the appropriate values, pass the `.json` file and apply settings with the following commands:

```
tsm settings import -f /path/to/file.json
```

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see [tsm pending-changes apply](#).

Web data connector settings

The web data connector (WDC) settings in the template below specify whether WDCs are enabled, whether refresh of WDCs is enabled, and the primary and secondary safe lists. The safe lists indicate which WDC URLs are approved for use in your Tableau Server installation, and the domains or URLs that a connector can send requests to and receive requests from.

Configuration template

Use this template to configure the WDC settings.

Important: All entity options are case sensitive.

For more explanation about configuration files, entities, and keys see Configuration File Example.

Single WDC

```
{
  "configEntities": {
    "web-data-connector-settings": {
      "_type": "webDataConnectorSettingsType",
      "refreshEnabled": true,
      "whitelist": {
        "https://www.example.com:443/wdc/": {
          "secondaryWhitelist": [
            "https://www.example.com/*.*",
            "https://www.coolapi.com/*.*"
          ]
        }
      },
      "enabled": true
    }
  }
}
```

```
}
```

Multiple WDCs

```
{  
  "configEntities": {  
    "web-data-connector-settings": {  
      "_type": "webDataConnectorSettingsType",  
      "refreshEnabled": true,  
      "whitelist": {  
        "https://www.example.com:443/wdc/": {  
          "secondaryWhitelist": [  
            "https://www.example.com/*.*"  
          ]  
        },  
        "https://www.mysite.com:443/coolwdc/": {  
          "secondaryWhitelist": [  
            "https://www.mysite.com/*.*",  
            "https://www.coolapi.com/*.*",  
          ]  
        }  
      },  
      "enabled": true  
    }  
  }  
}
```

Configuration file reference

This table includes all of the options that can be included with the `web-data-connector-settings` entity set.

`_type`

Required.

Value: `webDataConnectorSettingsType`

Do not change.

`refreshEnabled`

Set to `false` to disable refresh of WDCs. Defaults to `true`.

`whitelist`

Required.

Can contain one or more matching sets of safe lists and secondary safe lists (one set per WDC). The first URL provided is the safe list, where you specify the WDC URL and port, formatted as follows:

```
<scheme>://<host>:<port>/<path>
```

For many WDCs the `<port>` value is 443, which is the default port for HTTPS, but you can check the value for your connector by looking at the data source details on Tableau Server or Tableau Online.

`secondaryWhitelist`

Required.

Specifies the domains or URLs that a connector can send requests to and receive requests from, for example, external JavaScript libraries, REST APIs, or local files. To add an entire domain to this secondary safe list, you can use a wildcard expression `.*` at the end of the URL, as shown in the following example:

```
https://www.example.com/*.*
```

`enabled`

Set to `false` to disable use of WDCs. Defaults to `true`.

tabcmd

Tableau provides the `tabcmd` command-line utility which you can use to automate site administration tasks on your Tableau Server site. For example, creating or deleting users, projects, and groups.

This utility is included with Tableau Server, and is automatically installed on the server nodes. You can also run it from other computers, even computers that are not part of your Tableau Server installation, but to do so you need to download the `tabcmd` installer from the Tableau website. For more information, see [Install tabcmd](#) below.

Install tabcmd

When Tableau Server or Tableau Online is upgraded to a new version, if an updated version of `tabcmd` is required, you can download it from the [Tableau Server Releases](#) page on the Tableau website.

For Tableau Server, we recommend you download the version that matches your server version. For Tableau Online, we recommend you always download the latest version to avoid issues caused by version incompatibilities. In either case, using an out of date version of `tabcmd` can cause errors and unpredictable results.

1. Open a web browser and go to the [Tableau Server Releases](#) page. Go to this page even if you use Tableau Online.
2. If you're using:
 - **Tableau Online**, select the latest Tableau *Server* release.
 - **Tableau Server (Windows or Linux)**: select the release that matches your server version.

In either case, if the expanded information shows maintenance releases, select the latest maintenance release or the one that matches your server version.



This takes you to the release notes page, called Resolved Issues, where you can read about security improvements and resolved issues.

3. Scroll to the **Download Files** section under the resolved issues, select the `tabcmd` download link that is compatible with the computer on which you'll run the `tabcmd` commands.

Download Files

Windows

- [TableauServerTabcmd-64bit-2020-1-3.exe \(93 MB\)](#)
- [TableauServer-64bit-2020-1-3.exe \(1540 MB\)](#)

Linux

- [tableau-tabcmd-2020-1-3.noarch.rpm \(10 MB\)](#)
- [tableau-tabcmd-2020-1-3_all.deb \(10 MB\)](#)
- [tableau-server-2020-1-3.x86_64.rpm \(1647 MB\)](#)
- [tableau-server-2020-1-3_amd64.deb \(1649 MB\)](#)

The remaining steps refer to this computer as “the `tabcmd` computer.”

4. Save the installer to the `tabcmd` computer, or a location accessible from that computer (a mounted drive, for example).
5. Complete the installation steps as appropriate for the operating system of the `tabcmd` computer:

• Windows

By default `tabcmd` is installed to `C:\Program Files\Tableau\Tableau Server\<version>\extras\Command Line Utility`. You can change this during installation and recommend that you install `tabcmd` to a folder named `tabcmd` at the root of the `C:\` drive (`C:\tabcmd`). This can make it easier to locate and run, and will accommodate some limitations with the Windows operating system if you add the `tabcmd` directory to the Windows PATH.

Note The `tabcmd` Setup program does not add the `tabcmd` directory to the Windows PATH variable. You can add it manually, or you can include the full path to `tabcmd` each time you call it.

You can install `tabcmd` in two ways on Windows:

- Double-click the installer to follow the steps in the UI:
 - a. Accept the license agreement.
 - b. If you want to install to a non-default location, click **Customize** and type or browse to the location you want to install `tabcmd` to.
 - c. Click **Install**.

If you are prompted by Windows Defender Firewall or User Account Control, click **Allow access**.

- Run the installer from a command prompt:
 - a. Open a command prompt as administrator on the `tabcmd` computer.
 - b. Navigate to the directory where you copied the `tabcmd` installer.

c. Install tabcmd:

```
tableau-setup-tabcmd-tableau-<version_code>-
x64.exe /quiet ACCEPTTEULA=1
```

To install to a non-default location:

```
tableau-setup-tabcmd-tableau-<version_code>-
x64.exe /quiet ACCEPTTEULA=1 INSTALLDIR-
R="<path\to\install\directory>"
```

For example:

```
tableau-setup-tabcmd-tableau-<version_code>-
x64.exe /quiet ACCEPTTEULA=1 INSTALLDIR-
R="C:\tabcmd"
```

For a complete list of command line options you can use with the tabcmd installer, run the installer with a `/?`. For more information on tabcmd installer command line options, see [Install Switches and Properties for tabcmd \(Windows\)](#).

The tabcmd Setup program creates logs in `C:\Users->s\<user>\AppData\Local\Temp` you can use if you have problems installing tabcmd. The logs use the naming convention `Tableau_Server_Command_Line_UTILITY_(<version_code>)_#####.log`.

• Linux

Note: To run tabcmd on a Linux computer, you must have Java 8 (also called Java 1.8) installed. On RHEL-like systems this will be installed as a dependency when you install tabcmd. On Ubuntu systems, you need to install Java 8 (1.8) separately if it is not already installed.

As of July 2022, Debian distributions are no longer supported. For more information, see [this Tableau Community post](#).

- a. Log on as a user with sudo access to the tabcmd computer.
- b. Navigate to the directory where you copied the `.rpm` or `.deb` package that you downloaded.

- On RHEL-like distributions, including CentOS, run the following command:

```
sudo yum install tableau-tabcmd-<version>.noarch.rpm
```

- On Ubuntu, run the following command:

```
sudo apt-get install ./tableau-tabcmd-<version>_all.deb
```

To uninstall tabcmd from a Linux computer, see the documentation for the Linux variety you are running.

6. (Optional) Add the fully qualified location where tabcmd is installed to your system path to allow you to run tabcmd commands without changing to that location, or specifying the location with each command. Steps to do this depend on the type and version of your operating system. For more information, see [PATH_\(variable\)](#).

How to use tabcmd

The basic steps for using tabcmd are as follows:

1. Open the Command Prompt as an administrator.
2. On a Windows computer, if you installed tabcmd on a computer other than the initial

node, change to the directory where you installed `tabcmd`.

On a Linux computer, you do not need to change to the install directory.

3. Run the `tabcmd` command.

When you use `tabcmd`, you must establish an authenticated server session. The session identifies the server or Tableau Online site and the user running the session. You can start a session first, and then specify your command next, or you can start a session and execute a command all at once.

Important: If you are using `tabcmd` to perform more than one task, you must run tasks one after another (serially), rather than at the same time (in parallel).

Commands (such as `login`) and the options (such as `-s`, `-u`, etc.) are not case sensitive, but the values you provide (such as `User@Example.com`) are case sensitive.

Examples

The following command demonstrates starting a session with the Tableau Server named *tabserver.mycompany.com*:

```
tabcmd login -s http://tabserver.mycompany.com -u admin -p  
mypassword
```

The next example shows a command that deletes a workbook named *Sales_Workbook*:

```
tabcmd delete "Sales_Workbook"
```

Here's how to accomplish all of the above with one command—note that you do not need `login` here:

```
tabcmd delete "Sales_Workbook" -s http://t-  
abserver.mycompany.com -u admin -p mypassword
```

Tableau Server on Linux Administrator Guide

A Tableau Server can run multiple sites. When a workbook is on the Default site of a multi-site server you don't need to specify Default, the above command is sufficient. However, if the command applies to something on a site other than Default, you need to specify the site ID for that site (see login). Here's the same command for a workbook that's on the West Coast Sales site (site ID `wsales`):

```
tabcmd delete "Sales_Workbook" -s http://tabserver.mycompany.com  
-t wsales -u admin -p mypassword
```

The options `-s`, `-t`, `-u`, and `-p` are among the `tabcmd` global variables, which can be used with any command.

For more information, see [tabcmd Commands](#).

Status messages and logs

When a command is successful, `tabcmd` returns a status code of zero. A full error message for non-zero status codes is printed to **stderr**. In addition, informative or progress messages may be printed to **stdout**.

A full log named **tabcmd.log** that includes debugging, progress, and error messages is written to `<home_dir>/tableau/tabcmd/`.

tabcmd Commands

Looking for Tableau Server on Windows? See [tabcmd Commands](#).

You can use the following commands with the `tabcmd` command line tool:

- `addusers` (to group)
- `createextracts`
- `creategroup`
- `createproject`
- `createsite`
- `createsiteusers`

createusers
decryptextracts
delete *workbook-name* or *datasource-name*
deleteextracts
deletegroup
deleteproject
deletesite
deletesiteusers
deleteusers
editdomain
editsite
encryptextracts
export
get *url*
initialuser
listdomains
listsites
login
logout
publish
publishsamples
reencryptextracts
refreshextracts
removeusers
reset_openid_sub
runschedule
set
syncgroup
upgradethumbnails
validateidpmetadata
version

addusers *group-name*

Adds users to the specified group.

Example

```
tabcmd addusers "Development" --users "users.csv"
```

Options

`--users`

Add the users in the given `.csv` file to the specified group. The file should be a simple list with one user name per line. User names are not case sensitive. The users should already be created on Tableau Server.

For more information, see [CSV Import File Guidelines](#).

`--[no-]complete`

When set to `complete` this option requires that all rows be valid for any change to succeed. If not specified, `--complete` is used.

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** [Configure Mutual SSL](#)
- **Linux:** [Configure Mutual SSL](#)

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or ") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Tableau Server on Linux Administrator Guide

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, `tabcmd` (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

createextracts

Creates extracts for a published workbook or data source.

Options

`-d, --datasource`

The name of the target data source for extract creation.

`--embedded-datasources`

A space-separated list of embedded data source names within the target workbook. Enclose data source names with double quotes if they contain spaces. Only available when creating extracts for a workbook.

`--encrypt`

Create encrypted extract.

`--include-all`

Include all embedded data sources within target workbook. Only available when creating extracts for workbook.

`--parent-project-path`

Path of the project that is the parent of the project that contains the target resource. Must specify the project name with `--project`.

`--project`

The name of the project that contains the target resource. Only necessary if `--workbook` or `--datasource` is specified. If unspecified, the default project 'Default' is used.

`-u, -url`

Tableau Server on Linux Administrator Guide

The canonical name for the resource as it appears in the URL.

`-w, -workbook`

The name of the target workbook for extract creation.

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** [Configure Mutual SSL](#)
- **Linux:** [Configure Mutual SSL](#)

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or ") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, `tabcmd` (the client) does not validate the server's SSL certificate.

Tableau Server on Linux Administrator Guide

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

`creategroup` *group-name*

Creates a group. Use `addusers` (for local groups) and `syncgroup` (for Active Directory groups) commands to add users after the group has been created.

Example

```
tabcmd creategroup "Development"
```

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token

remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** [Configure Mutual SSL](#)
- **Linux:** [Configure Mutual SSL](#)

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID,

surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or ") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, tabcmd (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

--

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

`createproject` *project-name*

Creates a project.

Example

```
tabcmd createproject -n "Quarterly_Reports" -d "Workbooks showing quarterly sales reports."
```

Options

`-n, --name`

Specifies the name of the project that you want to create.

`--parent-project-path`

Specifies the name of the parent project for the nested project as specified with the `-n` option. For example, to specify a project called "Nested" that exists in a "Main" project, use the following syntax: `--parent-project-path "Main" -n "Nested"`.

`-d, --description`

Specifies a description for the project.

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `-`

Tableau Server on Linux Administrator Guide

`-password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** [Configure Mutual SSL](#)
- **Linux:** [Configure Mutual SSL](#)

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or ") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, tabcmd (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the

command. By default, the process will wait until the server responds.

--

Specifies the end of options on the command line. You can use -- to indicate to `tabcmd` that anything that follows -- should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use -- in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

`createsite` *site-name*

Creates a site.

Examples

Create a site named `West Coast Sales`. A site ID of `WestCoastSales` will be automatically created, the site will have no storage quota limit, and site administrators will be able to add and remove users:

```
tabcmd createsite "West Coast Sales"
```

Create a site named `West Coast Sales` with a site ID of `wsales`:

```
tabcmd createsite "West Coast Sales" -r "wsales"
```

Prevent site administrators from adding users to the site:

```
tabcmd createsite "West Coast Sales" --no-site-mode
```

Set a storage quota, in MB:

```
tabcmd createsite "West Coast Sales" --storage-quota 100
```

Options

`-r, --url`

Used in URLs to specify the site. Different from the site name.

`--user-quota`

Maximum number of users that can be added to the site.

`--[no-]site-mode`

Allows or denies site administrators the ability to add users to or remove users from the site.

`--storage-quota`

In MB, the amount of workbooks, extracts, and data sources that can be stored on the site.

`--extract-encryption-mode`

The extract encryption mode for the site can be **enforced**, **enabled** or **disabled**. For more information, see Extract Encryption at Rest.

`--run-now-enabled`

Allow or deny users from running extract refreshes, flows, or schedules manually. **true** to allow users to run tasks manually or **false** to prevent users from running tasks manually. For more information, see Server Settings (General and Customization).

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `-`

Tableau Server on Linux Administrator Guide

`-password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** [Configure Mutual SSL](#)
- **Linux:** [Configure Mutual SSL](#)

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or "") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, tabcmd (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the

command. By default, the process will wait until the server responds.

--

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

`createsiteusers filename.csv`

Adds users to a site, based on information supplied in a comma-separated values (CSV) file. If the user is not already created on the server, the command creates the user before adding that user to the site.

The CSV file must contain one or more user names and can also include (for each user) a password, full name, license type, administrator level, publisher (yes/no), and email address. For information about the format of the CSV file, see [CSV Import File Guidelines](#).

As an alternative to including administrator level and publisher permissions in the CSV file, you can pass access level information by including the `--role` option and specifying the site role you want to assign users listed in the CSV file.

By default, users are added to the site that you are logged in to. To add users to a different site, include the global `--site` option and specify that site. (You must have permissions to create users on the site you specify.)

If the server contains multiple sites, you cannot add server (system) administrators through the `createsiteusers` command. Use `createusers` instead. If you specify the `Server-Administrator` site role for the `--role` option, the command returns an error. If the CSV file includes `System` as value for administrator, the value is ignored and the user is assigned the `Unlicensed` license type.

If the server contains only one site (the default site), you can specify `system` for the administrator value for a user, or even assign the `ServerAdministrator` site role using the `--role` option, if you want all users in the CSV file to be server administrators.

By default, this command creates users using a synchronous operation (it waits for all operations to complete before proceeding). You can use the `--no-wait` option to specify an asynchronous operation.

Local authentication

If the server is configured to use local authentication, the information in the CSV file is used to create users.

Active Directory authentication

If the server is configured to use Active Directory authentication, user information is imported from Active Directory, and password and friendly name information in the CSV file is ignored. Further, if a user is specified in the CSV file but no corresponding user exists in Active Directory, the user is not added to Tableau Server. For Active Directory users, because the user name is not guaranteed to be unique across domains, you must include the domain as part of the user name. You can specify this as either `domain\username` or `username@domain.com`; however, we recommend using the `domain\username` format. For more information, see [User Management in Deployments with External Identity Stores](#).

Example

```
tabcmd createsiteusers "users.csv" --role "Explorer"
```

Options

`--admin-type`

Deprecated. Use the `--role` option instead.

`--[no-]complete`

Deprecated. Default error behavior: if there are more than 3 errors within a ten-row

span, then the command will fail.

`--no-publisher`

Deprecated. Use the `--role` option instead.

`--nowait`

Do not wait for asynchronous jobs to complete.

`--publisher`

Deprecated. Use the `--role` option instead.

`--role`

Specifies a site role for all users in the `.csv` file. When you want to assign site roles using the `--role` option, create a separate CSV file for each site role.

Valid values are: `ServerAdministrator`, `SiteAdministratorCreator`, `SiteAdministratorExplorer`, `SiteAdministrator`, `Creator`, `ExplorerCanPublish`, `Publisher`, `Explorer`, `Interactor`, `Viewer`, **and** `Unlicensed`.

The default is `Unlicensed` for new users and unchanged for existing users. Users are added as unlicensed also if you have a user-based server installation, and if the `createsiteusers` command creates a new user, but you have already reached the limit on the number of licenses for your users.

Note: On a multi-site Tableau Server, if you want to assign the `ServerAdministrator` site role using the `--role` option, use the `createusers` command instead of `createsiteusers`.

`--silent-progress`

Do not display progress messages for the command.

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** [Configure Mutual SSL](#)
- **Linux:** [Configure Mutual SSL](#)

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

Tableau Server on Linux Administrator Guide

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or ") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, `tabcmd` (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not

need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

`createusers` *filename.csv*

Create users in Tableau Server, based on information supplied in a comma-separated values (CSV) file.

The CSV file must contain one or more user names and can also include (for each user) a password, full name, license type, administrator level, publisher (yes/no), and email address. For information about the format of the CSV file, see [CSV Import File Guidelines](#).

As an alternative to including administrator level and publisher permissions in the CSV file, you can pass access level information by including the `--role` option and specifying the site role you want to assign users listed in the CSV file.

If the server has only one site (the default site), the user is created and added to the site. If the server has multiple sites, the user is created but is not added to any site. To add users to a site, use `createsiteusers`.

Tableau Server on Linux Administrator Guide

If you have a user-based server installation, and if the command creates a new user but you have already reached the limit on the number of licenses for your users, the user is added as an unlicensed user.

Local authentication

If the server is configured to use local authentication, the information in the CSV file is used to create users.

Active Directory authentication

If the server is configured to use Active Directory authentication, user information is imported from Active Directory, and password and friendly name information in the CSV file is ignored. Further, if a user is specified in the CSV file but no corresponding user exists in Active Directory, the user is not added to Tableau Server. For Active Directory users, because the user name is not guaranteed to be unique across domains, you must include the domain as part of the user name. You can specify this as either `domain\username` or `username@domain.com`; however, we recommend using the `domain\username` format. For more information, see [User Management in Deployments with External Identity Stores](#).

Example

```
tabcmd createusers "users.csv" --role "ServerAdministrator"

tabcmd createusers "users.csv"
```

Options

`--admin-type`

Deprecated. Use the `--role` option instead.

`--[no-]complete`

Deprecated. Default error behavior: if there are more than 3 errors within a ten-row

span, then the command will fail.

`--no-publisher`

Deprecated. Use the `--role` option instead.

`--nowait`

Do not wait for asynchronous jobs to complete.

`--publisher`

Deprecated. Use the `--role` option instead.

`-r, --role`

Specifies a site role for all users in the `.csv` file. When you want to assign site roles using the `--role` option, create a separate CSV file for each site role.

Valid values are: `ServerAdministrator`, `SiteAdministratorCreator`, `SiteAdministratorExplorer`, `SiteAdministrator`, `Creator`, `ExplorerCanPublish`, `Publisher`, `Explorer`, `Interactor`, `Viewer`, **and** `Unlicensed`.

On a multi-site server, the command does not assign the user to a site. Therefore, the only site roles the command can successfully assign are `ServerAdministrator` and `Unlicensed`. If you specify any other site role, the command assigns the `Unlicensed` role.

On a single-site server, the user is created and added to the default site using the role that you specify.

If you have a user-based server installation, and if the command creates a new user but you have already reached the limit on the number of licenses for your users, the user is added as an unlicensed user.

`--silent-progress`

Do not display progress messages for the command.

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** [Configure Mutual SSL](#)
- **Linux:** [Configure Mutual SSL](#)

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or ") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, `tabcmd` (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not

Tableau Server on Linux Administrator Guide

need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

decryptextracts

Decrypt all extracts on a site. If no site is specified, extracts on the default site will be decrypted. For more information, see [Extract Encryption at Rest](#).

Depending on the number and size of extracts, this operation may consume significant server resources. Consider running this command outside of normal business hours.

Example

```
tabcmd decryptextracts "West Coast Sales"
```

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is

stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** [Configure Mutual SSL](#)
- **Linux:** [Configure Mutual SSL](#)

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Tableau Server on Linux Administrator Guide

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or ") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, `tabcmd` (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

--

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

delete workbook-name or datasource-name

Deletes the specified workbook or data source from the server.

This command takes the name of the workbook or data source as it is on the server, not the file name when it was published.

Example

```
tabcmd delete "Sales_Analysis"
```

Options

`-r, --project`

The name of the project containing the workbook or data source you want to delete. If not specified, the "Default" project is assumed.

`--parent-project-path`

Specifies the name of the parent project for the nested project as specified with the `-r` option. For example, to specify a project called "Nested" that exists in a "Main" project, use the following syntax: `--parent-project-path "Main" -r "Nested"`.

`--workbook`

The name of the workbook you want to delete.

Tableau Server on Linux Administrator Guide

`--datasource`

The name of the data source you want to delete.

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** [Configure Mutual SSL](#)
- **Linux:** [Configure Mutual SSL](#)

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or ") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, `tabcmd` (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

Tableau Server on Linux Administrator Guide

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

deleteextracts

Deletes extracts for a published workbook or data source.

Options

`-d, --datasource`

The name of the target data source for extract deletion.

`--embedded-datasources`

A space-separated list of embedded data source names within the target workbook. Enclose data source names with double quotes if they contain spaces. Only available when deleting extracts for a workbook.

`--encrypt`

Create encrypted extract.

`--include-all`

Include all embedded data sources within target workbook.

`--parent-project-path`

Path of the project that is the parent of the project that contains the target resource. Must specify the project name with `--project`.

`--project`

The name of the project that contains the target resource. Only necessary if `--workbook` or `--datasource` is specified. If unspecified, the default project 'Default' is used.

`-u, -url`

The canonical name for the resource as it appears in the URL.

`-w, -workbook`

The name of the target workbook for extract deletion.

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `-password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

Tableau Server on Linux Administrator Guide

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** [Configure Mutual SSL](#)
- **Linux:** [Configure Mutual SSL](#)

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or """) or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, `tabcmd` (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows

Tableau Server on Linux Administrator Guide

how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

`deletegroup` *group-name*

Deletes the specified group from the server.

Example

```
tabcmd deletegroup "Development"
```

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** [Configure Mutual SSL](#)
- **Linux:** [Configure Mutual SSL](#)

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or "") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, `tabcmd` (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

`deleteproject` *project-name*

Deletes the specified project from the server.

Using `tabcmd`, you can specify only a top-level project in a project hierarchy. To automate tasks you want to perform on a project within a parent project, use the equivalent Tableau [REST API](#) call.

Example

```
tabcmd deleteproject "Designs"
```

Option

```
--parent-project-path
```

Specifies the name of the parent project for the nested project as specified with the command. For example, to specify a project called "Designs" that exists in a "Main" project, use the following syntax: `--parent-project-path "Main" "Designs"`.

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

```
-h, --help
```

Displays the help for the command.

```
-c, --use-certificate
```

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** [Configure Mutual SSL](#)
- **Linux:** [Configure Mutual SSL](#)

```
-s, --server
```

The Tableau Server URL, which is required at least once to begin session.

Tableau Server on Linux Administrator Guide

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or ") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, `tabcmd` (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

`deletesite` *site-name*

Deletes the specified site from the server.

Example

```
tabcmd deletesite "Development"
```

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `-`

Tableau Server on Linux Administrator Guide

`-password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** [Configure Mutual SSL](#)
- **Linux:** [Configure Mutual SSL](#)

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or ") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, tabcmd (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the

command. By default, the process will wait until the server responds.

--

Specifies the end of options on the command line. You can use -- to indicate to `tabcmd` that anything that follows -- should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use -- in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

`deletesiteusers filename.csv`

Removes users from from the site that you are logged in to. The users to be removed are specified in a file that contains a simple list of one user name per line. (No additional information is required beyond the user name.)

By default, if the server has only one site, or if the user belongs to only one site, the user is also removed from the server. On a Tableau Server Enterprise installation, if the server contains multiple sites, users who are assigned the site role of **Server Administrator** are removed from the site but are not removed from the server.

If the user owns content, the user's role is change to **Unlicensed**, but the user is not removed from the server or the site. The content is still owned by that user. To remove the user completely, you must change the owner of the content and then try removing the user again.

If the user was imported from Active Directory, the user is removed from the site and possibly from the server. However, the user is not deleted from Active Directory.

Example

```
tabcmd deletesiteusers "users.csv"
```

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** [Configure Mutual SSL](#)
- **Linux:** [Configure Mutual SSL](#)

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

Tableau Server on Linux Administrator Guide

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or ") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, `tabcmd` (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not

need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

`deleteusers filename.csv`

Deletes the users listed in the specified comma-separated values (`.csv`) file.

The `.csv` file should contain a simple list of one user name per line.

Example

```
tabcmd deleteusers "users.csv"
```

Options

`--[no-]complete`

When set to `--complete` this option requires that all rows be valid for any change to succeed. If not specified, `--complete` is used.

Tableau Server on Linux Administrator Guide

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** [Configure Mutual SSL](#)
- **Linux:** [Configure Mutual SSL](#)

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or ") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, `tabcmd` (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

editdomain

Note: As a best practice, you should back up Tableau Server before you edit the domain. The domain is saved in the Tableau repository, and if it is incorrectly changed, administrators may not be able to sign in.

Changes the nickname or full domain name of an Active Directory domain on the server. A domain “nickname” is the Windows NetBIOS domain name.

You can modify the nickname for any domain the server is using. In general, you can modify the full domain name for any domain except the one that you used to sign in. However, if the user name that you are currently signed in with exists in both the current domain and the new domain, you can modify the full name for the current domain.

To ensure that Tableau Server can connect to other Active Directory domains, you must also specify secondary domains that Tableau Server connects to by setting the `wgserver.domain.whitelist` option with TSM. For more information about secondary domains and configuring the connection, see `wgserver.domain.whitelist`.

Review [User Management in Deployments with External Identity Stores](#) to understand how multiple domains, domain name mapping, and user names interact with Tableau Server.

To see a list of domains, use [listdomains](#).

Examples

```
tabcmd editdomain --id 2 --nickname "new-nickname"
```

```
tabcmd editdomain --id 3 --name "new-name"
```

Options

`--id`

The ID of domain to change. To get a list of domain IDs, use [listdomains](#).

`--name`

The new name for the domain.

`--nickname`

The new nickname for the domain.

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `-password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

Tableau Server on Linux Administrator Guide

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** [Configure Mutual SSL](#)
- **Linux:** [Configure Mutual SSL](#)

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or """) or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, `tabcmd` (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows

how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

editsite *site-name*

Changes the name of a site or its web folder name. You can also use this command to allow or deny site administrators the ability to add and remove users, or prevent users from running certain tasks manually. If site administrators have user management rights, you can specify how many users they can add to a site.

Examples

```
tabcmd editsite wc_sales --site-name "West Coast Sales"
```

```
tabcmd editsite wc_sales --site-id "wsales"
```

```
tabcmd editsite wsales --status ACTIVE
```

```
tabcmd editsite wsales --user-quota 50
```

Options

`--site-name`

The name of the site that's displayed.

`--site-id`

Used in the URL to uniquely identify the site.

`--user-quota`

Maximum number of users who can be members of the site.

`--[no-]site-mode`

Allow or prevent site administrators from adding users to the site.

`--status`

Set to `ACTIVE` to activate a site, or to `SUSPENDED` to suspend a site.

`--storage-quota`

In MB, the amount of workbooks, extracts, and data sources that can be stored on the site.

`--extract-encryption-mode`

The extract encryption mode for the site can be **enforced**, **enabled** or **disabled**. For more information, see Extract Encryption at Rest. Depending on the number and size of extracts, this operation may consume significant server resources.

`--run-now-enabled`

Allow or deny users from running extract refreshes, flows, or schedules manually. **true** to allow users to run tasks manually or **false** to prevent users from running tasks manually. For more information, see Server Settings (General and Customization).

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `-password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

Tableau Server on Linux Administrator Guide

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** [Configure Mutual SSL](#)
- **Linux:** [Configure Mutual SSL](#)

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or """) or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, `tabcmd` (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows

how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

encryptextracts

Encrypt all extracts on a site. If no site is specified, extracts on the default site will be encrypted. For more information, see [Extract Encryption at Rest](#).

Depending on the number and size of extracts, this operation may consume significant server resources. Consider running this command outside of normal business hours.

Example

```
tabcmd encryptextracts "West Coast Sales"
```

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** [Configure Mutual SSL](#)
- **Linux:** [Configure Mutual SSL](#)

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or "") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

Tableau Server on Linux Administrator Guide

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, `tabcmd` (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

export

Exports a view or workbook from Tableau Server and saves it to a file. This command can also export just the data used for a view. View data is exported at the summary level. To

export detail-level data, you must use the Tableau Server UI. For details, see [Download Views and Workbooks](#).

Note the following when you use this command:

- **Permissions:** To export, you must have the **Export Image** permission. By default, this permission is Allowed or Inherited for all roles, although permissions can be set per workbook or view.
- **Exporting data:** To export just the data for a view, use the `--csv` option. This exports the summary data used in a view to a `.csv` file.
- **Specifying the view, workbook, or data to export:**
 - Use part of the URL to identify what to export, specifically the `"workbook/view"` string as it appears in the URL for the workbook or view. Do not use the “friendly name,” and exclude the `:iid=<n>` session ID at the end of the URL.

For example, the Tableau sample view *Global Temperatures* in the *Regional* workbook has a URL similar to this: `<server_name>/#/views/Regional/GlobalTemperatures?:iid=3`

To export the *Global Temperatures* view, use the string `Regional/GlobalTemperatures`.

Do *not* use `Regional/Global Temperatures`, or `Regional/GlobalTemperatures?:iid=3`.

- If the server is running multiple sites and the view or workbook is on a site other than Default, Use `-t <site_id>`.
- To export a workbook, get the URL string by opening a view in the workbook, and include the view in the string you use.

In the above example, to export the *Regional* workbook, use the string `Regional/GlobalTemperatures`.

- To export a workbook, it must have been published with **Show Sheets as Tabs** selected in the Tableau Desktop Publish dialog box.

Note: The Tableau workbook that contains the **administrative views** cannot be exported.

- To filter the data you download, add a parameter filter using this format:

```
?<filter_name>=value
```

or, if filtering on a parameter and that parameter has a display name that matches the name of a measure or dimension:

```
?Parameters.<filter_name>=value
```

- **The saved file's format:** Your format options depend on what's being exported. A workbook can only be exported as a PDF using the `--fullpdf` argument. A view can be exported as a PDF (`--pdf`) or a PNG (`--png`).
- **The saved file's name and location** (optional): If you don't provide a name, it will be derived from the view or workbook name. If you don't provide a location, the file will be saved to your current working directory. Otherwise, you can specify a full path or one that's relative to your current working directory.

Note: You must include a file name extension such as `.csv` or `.pdf`. The command does not automatically add an extension to the file name that you provide.

- **Dashboard web page objects not included in PDF exports:** A dashboard can optionally include a web page object. If you are performing an export to PDF of a dashboard that includes a web page object, the web page object won't be included in the

PDF.

- **Non-ASCII and non-standard ASCII characters and PDF exports:** If you are exporting a view or workbook with a name that includes a character outside the ASCII character set, or a non-standard ASCII character set, you need to URL encode (percent-encode) the character.

For example if your command includes the city Zürich, you need to URL encode it as `Z%C3%BCrich`:

```
tabcmd export "/Cities/Sheet1?locationCity=Z%C3%BCrich" -
fullpdf
```

Clearing the Cache to Use Real-Time Data

You can optionally add the URL parameter `?:refresh=yes` to force a fresh data query instead of pulling the results from the cache. If you are using `tabcmd` with your own scripting and the `refresh` URL parameter is being used a great deal, this can have a negative impact on performance. It's recommended that you use `refresh` only when real-time data is required—for example, on a single dashboard instead of on an entire workbook.

Examples

Views

```
tabcmd export "Q1Sales/Sales_Report" --csv -f "Weekly-
Report.csv"
```

```
tabcmd export -t Sales "Sales/Sales_Analysis" --pdf -f "C:\T-
ableau_Workbooks\Weekly-Reports.pdf"
```

```
tabcmd export "Finance/InvestmentGrowth" --png
```

```
tabcmd export "Finance/InvestmentGrowth?:refresh=yes" --png
```

Workbooks

```
tabcmd export "Q1Sales/Sales_Report" --fullpdf
```

Tableau Server on Linux Administrator Guide

```
tabcmd export "Sales/Sales_Analysis" --fullpdf --pagesize  
tabloid -f "C:\Tableau_Workbooks\Weekly-Reports.pdf"
```

Options

`-f, --filename`

Saves the file with the given filename and extension.

`--csv`

View only. Export the view's data (summary data) in `.csv` format.

`--pdf`

View only. Export as a PDF.

`--png`

View only. Export as an image in `.png` format.

`--fullpdf`

Workbook only. Export as a PDF. The workbook must have been published with **Show Sheets as Tabs** enabled.

`--pagelayout`

Sets the page orientation (`landscape` or `portrait`) of the exported PDF. If not specified, its Tableau Desktop setting will be used.

`--pagesize`

Sets the page size of the exported PDF as one of the following: `unspecified`, `letter`, `legal`, `note folio`, `tabloid`, `ledger`, `statement`, `executive`, `a3`, `a4`, `a5`,

b4, b5, or quarto. Default is letter.

`--width`

Sets the width in pixels. Default is 800 px.

`--height`

Sets the height in pixels. Default is 600 px.

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** [Configure Mutual SSL](#)
- **Linux:** [Configure Mutual SSL](#)

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

Tableau Server on Linux Administrator Guide

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or ") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, `tabcmd` (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

get url

Gets the resource from Tableau Server that's represented by the specified (partial) URL. The result is returned as a file.

Note the following when you use this command:

- **Permissions:** To get a file, you must have the **Download/Web Save As** permission. By default, this permission is allowed or inherited for all roles, although permissions can be set per workbook or view.
- **Specifying a view or workbook to get:** You specify a view to get using the

`"/views/<workbookname>/<viewname>.<extension>"` string, and specify a workbook to get using the

`"/workbooks/<workbookname>.<extension>"` string. Replace `<workbookname>` and `<viewname>` with the names of the workbook and view as they appear in the URL when you open the view in a browser and replace `<extension>` with the type of file you want to save. Do not use the session ID at the end of the URL (`?:iid=<n>`) or the "friendly" name of the workbook or view.

For example, when you open a view *Regional Totals* in a workbook named *Metrics Summary*, the URL will look similar to this:

```
/views/MetricsSummary_1/RegionalTotals?:iid=1
```

Use the string `/views/MetricsSummary_1/RegionalTotals.<extension>` to get the view.

Use the string `/workbooks/MetricsSummary_1.<extension>` to get the workbook.

- **File extension:** The URL must include a file extension. The extension determines what's returned. A view can be returned in PDF, PNG, or CSV (summary data only) format. A Tableau workbook is returned as a TWB if it connects to a published data source or uses a live connection, or a TWBX if it connects to a data extract.

Note: If you are downloading a view to a PDF or PNG file, and if you include a `--filename` parameter that includes the `.pdf` or `.png` extension, you do not have to include a `.pdf` or `.png` extension in the URL.

- **The saved file's name and location** (optional): The name you use for `--filename` should include the file extension. If you don't provide a name and file extension, both will be derived from the URL string. If you don't provide a location, the file is saved to your current working directory. Otherwise, you can specify a full path or one that's

relative to your current working directory.

- **PNG size** (optional): If the saved file is a PNG, you can specify the size, in pixels, in the URL.

Clearing the cache to use real-time data

You can optionally add the URL parameter `? :refresh=yes` to force a fresh data query instead of pulling the results from the cache. If you are using `tabcmd` with your own scripting, using the `refresh` parameter a great deal can have a negative impact on performance. It's recommended that you use `refresh` only when real-time data is required—for example, on a single dashboard instead of on an entire workbook.

Examples

Views

```
tabcmd get "/views/Sales_Analysis/Sales_Report.png" --filename
"Weekly-Report.png"
```

```
tabcmd get "/views/Finance/InvestmentGrowth.pdf" -f
"Q1Growth.pdf"
```

```
tabcmd get "/views/Finance/InvestmentGrowth" -f "Q1Growth.pdf"
```

```
tabcmd get "/views/Finance/InvestmentGrowth.csv"
```

```
tabcmd get "/views/Finance/InvestmentGrowth.png?:size=640,480"
-f growth.png
```

```
tabcmd get "/views/Finance/InvestmentGrowth.png?:refresh=yes" -
f growth.png
```

Workbooks

```
tabcmd get "/workbooks/Sales_Analysis.twb" -f "C:\Tableau_Work-
books\Weekly-Reports.twb"
```

Tableau Server on Linux Administrator Guide

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** [Configure Mutual SSL](#)
- **Linux:** [Configure Mutual SSL](#)

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or ") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, `tabcmd` (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not

Tableau Server on Linux Administrator Guide

need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

initialuser

Create the initial administrative user on a server that does not have an initial administrative user defined.

Enclose values in single quotes.

Note: The `tabcmd initialuser` command does not require authentication to Tableau Server, but you must run the command on the initial server node.

Examples

```
tabcmd initialuser --username 'admin' --password 'password' --  
server http://localhost
```

```
tabcmd initialuser --username 'admin' --password 'password' --friendly 'Tableau Admin' --server http://localhost
```

To prompt for the password in the shell, do not include the `--password` parameter in the command. For example:

```
tabcmd initialuser --username 'admin' --server http://localhost
```

Options

`-f, --friendly`

Creates the initial administrative user with the display name.

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

Tableau Server on Linux Administrator Guide

- **Windows:** [Configure Mutual SSL](#)
- **Linux:** [Configure Mutual SSL](#)

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or "") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, `tabcmd` (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

listdomains

Displays a list of the Active Directory domains that are in use on the server, along with their nicknames and IDs. If the server is configured to use local authentication, the command returns only the domain name `local`.

Example

```
tabcmd listdomains
```

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** [Configure Mutual SSL](#)
- **Linux:** [Configure Mutual SSL](#)

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or ") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

Tableau Server on Linux Administrator Guide

`--no-certcheck`

When specified, `tabcmd` (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

listsites

Returns a list of sites to which the logged in user belongs.

Example

```
tabcmd listsites --username adam --password mypassword
```

Options

`--get-extract-encryption-mode`

The extract encryption mode for the site can be **enforced**, **enabled** or **disabled**. For more information, see [Extract Encryption at Rest](#).

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** [Configure Mutual SSL](#)
- **Linux:** [Configure Mutual SSL](#)

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

Tableau Server on Linux Administrator Guide

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or """) or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, `tabcmd` (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

login

Logs in a Tableau Server user.

Use the `--server`, `--site`, `--username`, `--password` global options to create a session.

Note: When you use the `tabcmd login` command, you cannot use SAML single sign-on (SSO), even if the server is configured to use SAML. To log in, you must pass the user name and password of a user who has been created on the server. You will have the permissions of the Tableau Server user that you're signed in as. For more information, see [Set Users' Site Roles and Permissions](#).

Tableau Server on Linux Administrator Guide

If you want to log in using the same information you've already used to create a session, just specify the `--password` option. The server and user name stored in the cookie will be used.

If the server is using a port other than 80 (the default), you will need to specify the port.

You need the `--site (-t)` option only if the server is running multiple sites and you are logging in to a site other than the Default site. If you do not provide a password you will be prompted for one. If the `--no-prompt` option is specified and no password is provided the command will fail.

Once you log in, the session will continue until it expires on the server or the `logout` command is run.

Example

Logs user jsmith in to the Tableau Server running on their local machine:

```
tabcmd login -s http://localhost -u jsmith -p password
```

Logs administrator in to the Sales site on sales-server:

```
tabcmd login -s http://sales-server -t Sales -u administrator -p password
```

```
tabcmd login -s http://sales-server:8000 -t Sales -u administrator -p password
```

Logs administrator in to the Sales site on sales-server using SSL, but does not validate the server's SSL certificate:

```
tabcmd login --no-certcheck -s https://sales-server -t Sales -u administrator -p password
```

Establishes a forward proxy and port for localhost:

```
tabcmd login --proxy myfwdproxyserver:8888 -s http://localhost  
-u jsmith -p password
```

Logs user jsmith in to the reverse proxy using SSL:

```
tabcmd login -s https://myreverseproxy -u jsmith -p password
```

Options

-s, --server

If you are running the command from a Tableau Server computer that's on your network, you can use `http://localhost`. Otherwise, specify the computer's URL, such as `http://bigbox.myco.com` or `http://bigbox`.

If the server is using SSL, you will need to specify `https://` in the computer's URL.

For Tableau Online, specify the URL `https://online.tableau.com`.

-t, --site

Include this option if the server has multiple sites, and you are logging in to a site other than the default site.

The site ID is used in the URL to uniquely identify the site. For example, a site named West Coast Sales might have a site ID of `west-coast-sales`.

-u, --username

The user name of the user logging in. For Tableau Online, the user name is the user's email address.

-p, --password

Password for the user specified for `--username`. If you do not provide a password you will be prompted for one.

Tableau Server on Linux Administrator Guide

`--password-file`

Allows the password to be stored in the given `filename.txt` file rather than the command line, for increased security.

`-x, --proxy`

Use to specify the HTTP proxy server and port (Host:Port) for the `tabcmd` request.

`--no-prompt`

Do not prompt for a password. If no password is specified, the `login` command will fail.

`--no-proxy`

Do not use an HTTP proxy server.

`--cookie`

Saves the session ID on login. Subsequent commands will not require a login. This value is the default for the command.

`--no-cookie`

Do not save the session ID information after a successful login. Subsequent commands will require a login.

`--timeout SECONDS`

The number of seconds the server should wait before processing the `login` command. Default: 30 seconds.

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--`

`password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** [Configure Mutual SSL](#)
- **Linux:** [Configure Mutual SSL](#)

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

Tableau Server on Linux Administrator Guide

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or "") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, `tabcmd` (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the

command. By default, the process will wait until the server responds.

--

Specifies the end of options on the command line. You can use -- to indicate to `tabcmd` that anything that follows -- should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use -- in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

logout

Logs out of the server.

Example

```
tabcmd logout
```

publish *filename.twb(x)*, *filename.tds(x)*, or *filename.hyper*

Publishes the specified workbook (`.twb(x)`), data source (`.tds(x)`), or extract (`.hyper`) to Tableau Server.

If you are publishing a workbook, by default, all sheets in the workbook are published without database user names or passwords.

The permissions initially assigned to the workbook or data source are copied from the project that the file is published to. Permissions for the published resource can be changed after the file has been published.

If the workbook contains user filters, one of the thumbnail options must be specified.

Example

Tableau Server on Linux Administrator Guide

```
tabcmd publish "analysis.twbx" -n "Sales_Analysis" --db-username  
"jsmith" --db-password "secret-password"
```

```
tabcmd publish "analysis_sfdc.hyper" -n "Sales Analysis"  
--oauth-username "user-name" --save-oauth
```

If the file is not in the same directory as tabcmd, include the full path to the file.

Example

```
tabcmd publish "\\computer\volume\Tableau Work-  
books\analysis.twbx" -n "Sales_Analysis" --db-username "jsmith"  
--db-password "secret-password"
```

```
tabcmd publish "\\computer\volume\Tableau Workbooks\analysis_sfd-  
c.hyper" -n "Sales Analysis" --oauth-username "username" --save-  
oauth
```

Options

`-n, --name`

Name of the workbook or data source on the server. If omitted, the workbook, data source, or data extract will be named after filename.

`-o, --overwrite`

Overwrites the workbook, data source, or data extract if it already exists on the server.

`-r, --project`

Publishes the workbook, data source, or data extract into the specified project. Publishes to the "Default" project if not specified.

`--parent-project-path`

Specifies the name of the parent project for the nested project as specified with the `-r`

option. For example, to specify a project called "Nested" that exists in a "Main" project, use the following syntax: `--parent-project-path "Main" -r "Nested"`.

`--db-username`

Use this option to publish a database user name with the workbook, data source, or data extract.

`--db-password`

Use this option to publish a database password with the workbook, data source, or extract.

`--save-db-password`

Stores the provided database password on the server.

`--oauth-username`

The email address of the user account. Connects the user through a preconfigured OAuth connection, if the user already has a saved access token for the cloud data source specified in `--name`. Access tokens are managed in user preferences.

For existing OAuth connections to the data source, use this option instead of `--db-username` and `--db-password`.

`--save-oauth`

Saves the credential specified by `--oauth-username` as an embedded credential with the published workbook or data source.

Subsequently, when the publisher or server administrator signs in to the server and edits the connection for that workbook or data source, the connection settings will show this OAuth credential as embedded in the content.

Tableau Server on Linux Administrator Guide

If you want to schedule extract refreshes after publishing, you must include this option with `--oauth-username`. This is analogous to using `--save-db-password` with a traditional database connection.

`--thumbnail-username`

If the workbook contains user filters, the thumbnails will be generated based on what the specified user can see. Cannot be specified when `--thumbnail-group` option is set.

`--thumbnail-group`

If the workbook contains user filters, the thumbnails will be generated based on what the specified group can see. Cannot be specified when `--thumbnail-username` option is set.

`--tabbed`

When a workbook with tabbed views is published, each sheet becomes a tab that viewers can use to navigate through the workbook. Note that this setting will override any sheet-level security.

`--append`

Append the extract file to the existing data source.

`--replace`

Use the extract file to replace the existing data source.

`--disable-uploader`

Disable the incremental file uploader.

`--restart`

Restart the file upload.

`--encrypt-extracts`

Encrypt extracts when you publish a workbook, data source, or extract to the server. For more information, see [Extract Encryption at Rest](#).

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `-password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** [Configure Mutual SSL](#)
- **Linux:** [Configure Mutual SSL](#)

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

Tableau Server on Linux Administrator Guide

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or ") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, `tabcmd` (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

publishsamples

Description

Publishes Tableau Sample workbooks to the specified project. Any existing samples will be overwritten.

Syntax

```
tabcmd publishsamples -n [project name] [Global options]
```

Example

Publish samples to the Inside Sales project on the Default site, as user `jsmith`.

Tableau Server on Linux Administrator Guide

```
tabcmd publishsamples -n "Inside Sales" -t "" -s localhost --  
username "jsmith" --password "secret-password"
```

Options

`-n, --name`

Required. Publishes the Tableau samples into the specified project. If the project name includes spaces, enclose the entire name in quotes.

`--parent-project-path`

Specifies the name of the parent project for the nested project as specified with the `-n` option. For example, to specify a project called "Nested" that exists in a "Main" project, use the following syntax: `--parent-project-path "Main" -n "Nested"`.

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** [Configure Mutual SSL](#)
- **Linux:** [Configure Mutual SSL](#)

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or "") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

Tableau Server on Linux Administrator Guide

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, `tabcmd` (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

reencryptextracts

Reencrypt all extracts on a site with new encryption keys. This command will regenerate the key encryption key and data encryption key. You must specify a site. For more information, see [Extract Encryption at Rest](#).

Depending on the number and size of extracts, this operation may consume significant server resources. Consider running this command outside of normal business hours.

Examples

```
tabcmd reencryptextracts "Default"
```

```
tabcmd reencryptextracts "West Coast Sales"
```

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `-password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

Tableau Server on Linux Administrator Guide

- **Windows:** [Configure Mutual SSL](#)
- **Linux:** [Configure Mutual SSL](#)

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or "") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, `tabcmd` (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

refreshextracts *workbook-name* or *datasource-name*

Performs a full or incremental refresh of extracts belonging to the specified workbook or data source.

Tableau Server on Linux Administrator Guide

This command takes the name of the workbook or data source as it appears on the server, not the file name when it was published. Only an administrator or the owner of the workbook or data source is allowed to perform this operation.

Note: This method will fail and result in an error if your Server Administrator has disabled the **RunNow** setting for the site. For more information, see [Tableau Server Settings](#).

Examples

```
tabcmd refreshextracts --datasource sales_ds
```

```
tabcmd refreshextracts --project "Sales External" --datasource sales_ds
```

```
tabcmd refreshextracts --project "Sales External" --parent-project-path "Main" --project "Sales External" --datasource sales_ds
```

```
tabcmd refreshextracts --workbook "My Workbook"
```

```
tabcmd refreshextracts --url SalesAnalysis
```

```
tabcmd refreshextracts --workbook "My Workbook" --addcalculations
```

```
tabcmd refreshextracts --datasource sales_ds --removecalculations
```

Options

```
--incremental
```

Runs the incremental refresh operation.

`--synchronous`

Adds the full refresh operation to the queue used by the Backgrounder process, to be run as soon as a Backgrounder process is available. If a Backgrounder process is available, the operation is run immediately. The refresh operation appears on the Background Tasks report.

During a synchronous refresh, `tabcmd` maintains a live connection to the server while the refresh operation is underway, polling every second until the background job is done.

`--workbook`

The name of the workbook containing extracts to refresh. If the workbook has spaces in its name, enclose it in quotes.

`--datasource`

The name of the data source containing extracts to refresh.

`--project`

Use with `--workbook` or `--datasource` to identify a workbook or data source in a project other than *Default*. If not specified, the Default project is assumed.

`--parent-project-path`

Specifies the name of the parent project for the nested project as specified with the `--project` option.

For example:

- To specify a project called "Nested" that exists in a "Main" project, use the following syntax:
- `--parent-project-path "Main" --project "Nested"`

Tableau Server on Linux Administrator Guide

- To specify a project called "Nested2" that is nested within the "Nested" project:
- `--parent-project-path "Main/Nested" --project "Nested2"`

`--url`

The name of the workbook as it appears in the URL. A workbook published as “Sales Analysis” has a URL name of “SalesAnalysis”.

`--addcalculations`

Use with `--workbook` to materialize calculations in the embedded extract of the workbook or `--datasource` to materialize calculations in the extract data source. Adds the operation to the queue used by the Backgrounder process. If a Backgrounder process is available, the operation runs immediately. This operation appears on the [Background Tasks for Extracts](#) administrative view.

`--removecalculations`

Use with `--workbook` or `--datasource` to remove calculations that were previously materialized. Adds the operation to the queue used by the Backgrounder process. If a Backgrounder process is available, the operation runs immediately. This operation appears on the [Background Tasks for Extracts](#) administrative view.

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** [Configure Mutual SSL](#)
- **Linux:** [Configure Mutual SSL](#)

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or "") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

Tableau Server on Linux Administrator Guide

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, `tabcmd` (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you

might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

reset_openid_sub

Clears OpenID Connect identifiers (sub values) that have already been associated with Tableau Server identities. See [Changing IdPs in Tableau Server for OpenID Connect](#).

Example

```
tabcmd reset_openid_sub --target-username jsmith
```

Options

`--target-username`

Clears sub value for the specified individual user.

`--all`

Clears sub values for all users.

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h`, `--help`

Displays the help for the command.

Tableau Server on Linux Administrator Guide

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** [Configure Mutual SSL](#)
- **Linux:** [Configure Mutual SSL](#)

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or """) or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, `tabcmd` (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows

Tableau Server on Linux Administrator Guide

how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

`removeusers` *group-name*

Removes users from the specified group.

Example

```
tabcmd removeusers "Development" --users "users.csv"
```

Options

`--users`

Remove the users in the given `.csv` file from the specified group. The file should be a simple list with one user name per line.

`--[no-]complete`

Requires that all rows be valid for any change to succeed. If not specified `--complete` is used.

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h`, `--help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** [Configure Mutual SSL](#)
- **Linux:** [Configure Mutual SSL](#)

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or "") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

Tableau Server on Linux Administrator Guide

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, `tabcmd` (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you

might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

`runschedule` *schedule-name*

Runs the specified schedule.

This command takes the name of the schedule as it is on the server.

This command is not available for Tableau Online.

Note: This method will fail and result in an error if your Server Administrator has disabled the **RunNow** setting for the site. For more information, see [Tableau Server Settings](#).

Example

```
tabcmd runschedule "5AM Sales Refresh"
```

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `-password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Tableau Server on Linux Administrator Guide

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** [Configure Mutual SSL](#)
- **Linux:** [Configure Mutual SSL](#)

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or ") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, `tabcmd` (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

Tableau Server on Linux Administrator Guide

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

set *setting*

Enables the specified setting on the server. Details about each setting can be seen on the Maintenance page on the server.

Use an exclamation mark in front of the setting name to disable the setting. You can enable or disable the following settings:

- `allow_scheduling`
- `embedded_credentials`
- `remember_passwords_forever`

Example

```
tabcmd set embedded_credentials
```

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** [Configure Mutual SSL](#)
- **Linux:** [Configure Mutual SSL](#)

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or ") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Tableau Server on Linux Administrator Guide

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, `tabcmd` (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

syncgroup *group-name*

Synchronizes a Tableau Server group with an Active Directory group. If the Tableau Server group does not already exist, it is created and synchronized with the specified Active Directory group.

If the group name itself includes an "@" (other than as the domain separator) you need to refer to the symbol using the hex format "\0x40".

Example

```
tabcmd syncgroup "Development"
```

```
tabcmd syncgroup "Dev\0x40West"
```

Note: If you synchronize a group that you are a member of, changes that you make using this command do not apply to your user. For example, if you use this command to remove the administrator right from users in a group that you are a member of, you are still an administrator when the command finishes.

Options

```
--grant-license-mode <grant-license-mode>
```

Specifies whether a role should be granted on sign in. Default is `on-sync`. Valid values are `on-login`, `on-sync`. If no value is specified, `on-sync` is assumed and the default role will be granted when the group is synchronized. For more information, see [Modifying user roles with Grant role on sign in](#).

```
--no-publisher
```

Deprecated. Use the `--role` option instead.

```
--overwritesiterole
```

Tableau Server on Linux Administrator Guide

Allows a user's site role to be overwritten with a less privileged one when using `--role`. By default, a user site role can be promoted when using `--role`, but cannot be demoted. Because the `--overwritesiterole` option will demote user site roles, use it with caution.

`--publisher`

Deprecated. Use the `--role` option instead.

`-r, --role`

Specifies a site role for users in the group. The default is `Unlicensed`.

Valid values are: `SiteAdministratorCreator`, `SiteAdministratorExplorer`, `SiteAdministrator`, `Creator`, `ExplorerCanPublish`, `Publisher`, `Explorer`, `Interactor`, `Viewer`, `Unlicensed`.

`--silent-progress`

Do not display progress messages for the command.

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** [Configure Mutual SSL](#)
- **Linux:** [Configure Mutual SSL](#)

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or "") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Tableau Server on Linux Administrator Guide

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, `tabcmd` (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

upgradethumbnails

Starts and stops the Upgrade Thumbnails job. To learn more, see Upgrade Thumbnails Job.

Examples

To start the Upgrade Thumbnail job:

```
tabcmd upgradethumbnails --server <serverURL>
```

To stop the in progress Upgrade Thumbnail job:

```
tabcmd upgradethumbnails --server <serverURL> --stop
```

Options

`--stop`

When specified, stops the in progress Upgrade Thumbnails job. If this option is not specified, the Upgrade Thumbnail job will be started.

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h`, `--help`

Displays the help for the command.

Tableau Server on Linux Administrator Guide

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** [Configure Mutual SSL](#)
- **Linux:** [Configure Mutual SSL](#)

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or """) or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, `tabcmd` (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows

how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

validateidpmetadata

Identifies Tableau Server sites that are configured with IdPs using the insecure digest algorithm, SHA-1. This command also identifies IdPs that are using certificates with an insufficient RSA key size or elliptic curve size.

Note: This command is only available for site-specific SAML. For more information, see [Configure Site-Specific SAML](#).

Options

```
--digest-algorithms <ALGORITHMS>
```

A space-separated list of digest algorithms. Legal values are `sha1` and `sha256`. If not specified, server uses values from server configuration setting, `wgserver.saml.blocklisted_digest_algorithms`.

```
--min-allowed-elliptic-curve-size <SIZE>
```

If not specified, server uses values from server configuration setting, `wgserver.saml.min.allowed.elliptic_curve_size`.

```
--min-allowed-rsa-key-size <SIZE>
```

If not specified, server uses values from server configuration setting, `wgserver.saml.min.allowed.rsa_key_size`.

```
--site-names <SITENAMES>
```

A space-separated list of site names on which to perform certificate validation. If not specified, then all sites are inspected.

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** [Configure Mutual SSL](#)
- **Linux:** [Configure Mutual SSL](#)

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

Tableau Server on Linux Administrator Guide

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or ") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, `tabcmd` (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

version

Displays the version information for the current installation of the `tabcmd` utility.

Example

```
tabcmd version
```

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `-password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

Tableau Server on Linux Administrator Guide

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** [Configure Mutual SSL](#)
- **Linux:** [Configure Mutual SSL](#)

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either

an empty string with single or double quotes (" or ") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, tabcmd (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

Install Switches and Properties for `tabcmd` (Windows)

You can use the following switches when installing the Tableau Server Command Line Utility (`tabcmd`) version 2019.4.0 or later from the command line on Windows.

Note: There are no equivalent switches for the Linux version of the `tabcmd` installer.

Switch	Description	Comments
<code>/install /repair /uninstall /layout "<directory>"</code>	Run Setup to either install, repair, or uninstall <code>tabcmd</code> , or with <code>/layout</code> , create a complete local copy of the installation bundle in the directory specified.	Default is to install, displaying UI and all prompts. If no directory is specified on a fresh install, <code>C:\Program Files\Tableau\Tableau Server-<version>\extras\Command Line Utility</code> is assumed.
<code>/passive</code>	Run Setup	

	with minimal UI and no prompts.	
<code>/quiet /silent</code>	Run Setup in unattended, fully silent mode. No UI or prompts are displayed.	Note: Use either <code>/silent</code> or <code>/quiet</code> , not both.
<code>/norestart</code>	Run Setup without restarting Windows, even if a restart is necessary.	Note: In certain rare cases, a restart cannot be suppressed, even when this option is used. This is most likely when an earlier system restart was skipped, for example, during installation of other software.
<code>/log "<logfile>"</code>	Log information to the specified file and path. By default log files are created in the user's <code>%TEMP%</code> folder with a naming con-	If no file location is specified, the log file is written to the user's TEMP folder (<code>C:\Users\<username>\AppData\Local\Temp</username></code>). Check this log file for errors after installation. Example: <code><Setup file> /silent /log "C:\Tableau\Logs\tabcmd-Install"</code> <code>ACCEPTTEULA=1</code>

Tableau Server on Linux Administrator Guide

	<pre>vention of Tableau_ Server_ Command_ Line_util- ity_<ver- sion_ code>.log.</pre>	
--	---	--

Properties	Description	Comments
ACCEPTEULA=1 0	Accept the End User License Agreement (EULA). Required for quiet, silent, and passive install. 1 = accept the EULA, 0 = do not accept the EULA.	<p>If not included when using <code>/passive</code>, <code>/silent</code> or <code>/quiet</code>, Setup fails silently.</p> <p>If included but set to 0, Setup fails.</p>

<pre>INSTALLDIR- ="<code><path\to\i- installation\directory></code>"</pre>	<p>Install tabcmd to the spe- cified non- default install location.</p>	<p>Specifies the location to install tabcmd. If not used, tabcmd is installed to C:\Program Files\Tableau Server\<code><version_ code></code>\ex- tras\Command Line Utility</p> <p>Example: <code><Setup file> /silent INSTALLDIR- R="C:\tabcmd"</code></p>
--	---	--

Troubleshooting

You can use the following topics to troubleshoot and resolve issues with Tableau Server.

Troubleshoot Tableau Server on Linux

Follow the suggestions in this topic to resolve common issues with Tableau Server. For additional troubleshooting steps based on process status viewed on the Status page, see [Troubleshoot Server Processes](#).

The following table displays the default locations of the installation, data, logs, and scripts directories:

Directory	Default location
Installation:	<code>/opt/tableau/tableau_server</code>
Data:	<code>/var/opt/tableau/tableau_server/data</code>
Logs:	<code>/var/opt/tableau/tableau_server/data/tabsvc/logs/</code>
Scripts:	<code>/opt/tableau/tableau_server- /packages/scripts.<version_code>/</code>

General Troubleshooting Steps

Many Tableau Server issues can be addressed or tested with one or more of these basic steps:

Clean install

Install Tableau Server on Linux on a computer that has never had Tableau installed on it. If you are reusing a computer or VM that has had a previous version of Tableau Server installed, follow the steps in [Remove Tableau Server from Your Computer](#) to clean Tableau off your computer before you install the new version.

If you run into problems installing Tableau Server you may need to entirely remove Tableau from your computer and do a clean install. See [Remove Tableau Server from Your Computer](#) for details.

Disk space

Make sure there is enough disk space on each computer running Tableau Server. Limited disk space can cause a failure to install, a failure to upgrade, or problems running Tableau Server.

Remove old log files

If you are running out of disk space you can clean up old Tableau Server log files. These can take up space and as a best practice you should remove them regularly.

Version 10.5.x

If you have version 10.5.1 and higher, run this command at a terminal prompt to clean up log files you do not need:

```
tsm maintenance cleanup
```

Version 10.5.0

If you are running version 10.5.0 of Tableau Server on Linux, the cleanup command is not available and you need to run these commands at a terminal prompt:

```
sudo find /var/opt/tableau/tableau_server/data/tabsvc/temp/* -  
mtime +2 -type f -delete
```

```
sudo find /var/opt/tableau/tableau_server/data/tabsvc/logs/* -  
mtime +2 -type f -delete
```

Important: The Linux file system makes it possible to delete files that are open and if you do this the Tableau processes may not be able to recreate the files. This will result in empty log files. To fix this you can stop Tableau Server, restart the TSM Controller, and restart Tableau again:

1. Stop Tableau Server:

```
tsm stop
```

2. Restart the TSM Controller:

```
sudo systemctl restart tabadmincontroller_0.service
```

3. Wait several minutes for the controller to restart. You can confirm the controller has restarted with this command:

```
tsm status -v
```

Tableau Server on Linux Administrator Guide

When you can run that command and the Tableau Server Administration Controller is listed as 'running' the controller has restarted.

4. Start Tableau Server:

```
tsm start
```

Manually gather logs

If you cannot run `tsm maintenance ziplogs` for any reason (for example, if you have a critical failure before you run `tsm initialize`), you can manually collect and zip the logs by running these commands in a terminal window:

```
cd /var/opt/tableau/tableau_server/data/tabsvc/  
  
cp /var/opt/tableau/tableau_server/logs/app-install.log logs  
  
cp ~/.tableau/tsm/tsm.log logs  
  
tar -czvf ~/logs.tar.gz logs
```

This creates a file called `logs.tar.gz` in your home directory. You can upload or send this file to Tableau.

Restart server

Restart Tableau Server. Issues related to indexing and processes not fully started can be resolved by restarting Tableau Server in a controlled way. To restart Tableau Server, use the `tsm restart` command. This will stop all the processes associated with Tableau Server and then restart them.

Edit installation and configuration files using Linux

You should edit or create any files used to install or configure Tableau Server on Linux using a Linux operating system. Files created using Microsoft Windows will cause errors in Tableau Server on Linux installation and configuration because Linux operating systems end files with a line-feed (LF) character, whereas Windows ends files with a carriage-return character and a line-feed character (CR LF). Non-Linux (CR LF) file endings can cause errors

during Automated Installation of Tableau Server if they appear in the `config.json`, `reg_template.json` or `secrets` files used by the automated installer. Non-Linux (CR LF) file endings can also cause errors during registration or when configuring identity store settings or gateway settings.

Check systemd logs

If Tableau Server will not start, and you do not find anything useful in the Tableau logs (see Work with Log Files for more information) you can check the systemd logs for messages related to the TSM Service starting and stopping. The logs are stored at `/var/log/messages` (RHEL-like distros) or `/var/log/syslog` (Ubuntu). We recommend using the `journalctl` command to search and parse the systemd logs.

Installing Tableau Server

Install fails due to hardware requirements

Tableau Server cannot install if the computer you are installing on does not meet the minimum hardware requirements. For details on requirements, see Before you install....

Install fails due to timeouts

If you install Tableau Server on a computer with limited resources, for example, a computer that just meets the minimum hardware requirements, you can run into problems where `tsm` commands timeout due to slow response. You can specify a longer timeout by using the global `--request-timeout` option on all `tsm` commands. For more information on the `--request-timeout` option, see for example, `tsm initialize`.

Install fails with "Failed to initialize the instance of the temporary database"

Tableau Server on Linux only supports UTF-8 character encoding. If your Linux locale is missing the UTF-8 encoding, your installation can fail with an error similar to this one:

```
Failed to initialize the instance of the temporary database
```

Tableau Server on Linux Administrator Guide

To check if your locale is using UTF-8 encoding, run the `localectl` command at a command prompt. The resulting output should look something like this (your locale may be different):

```
[tableauserver-centos1a ~]$ localectl
System Locale: LANG=en_US.UTF-8
[tableauserver-centos1a ~]$
```

If the `LANG` value does not include `.UTF-8` then you need to run `localectl` to add it:

```
sudo localectl set-locale LANG=<your_locale>.UTF-8
```

Note: In some cases `localectl` may not complete (timeout) if your version of `systemd` is old. Updating `systemd` may fix this problem and allow you to set the UTF-8 encoding. On RHEL-like systems, use this command to update `systemd`: `sudo yum update systemd`

Installation fails on a virtual machine in Parallels

Parallels is currently not supported. If you install Tableau Server on a Linux virtual machine in Parallels, the install might fail.

Tableau Server doesn't start

If Tableau Server does not start or is running in a degraded state, run the `tsm restart` command. This will shut down any processes that are running, and restart Tableau Server.

Cannot start Tableau Server after installation

Tableau Server might not start if your computer's hostname changes after installation. One of the main reasons why the hostname might change is if you use the [cloud-init](#) package on CentOS. If you use the cloud-init package, *reboot the computer* where you want to install Tableau Server before you begin the installation process. Alternatively, you can fix the hostname without rebooting by running the following command:

```
sudo hostnamectl set-hostname `hostnamectl --static`
```

The cloud-init package is commonly used to initialize new virtual machines, configure SSH public key authentication, and more. For example, some CentOS images use cloud-init, and cloud-init is commonly used in OpenStack deployments. However, the version of cloud-init included by default in the CentOS 7.x repositories (cloud-init 0.7.5-10.el7.centos.1) has a **known issue** that prevents your computer from displaying its Fully Qualified Domain Name (FQDN) along with its hostname until after it restarts.

Because the Tableau Server installation process uses your computer's hostname to configure server processes and generate TLS certificates, Tableau Server might not start if it is configured to use a hostname without the FQDN.

To determine if your computer is displaying the correct hostname, run the `hostnamectl` command. In the following example, the command displays a transient hostname which indicates that it will not return the FQDN and must be restarted.

```
$ hostnamectl
   Static hostname: server01.example.com
Transient hostname: server01
[...]
```

Alternatively, in the following example, the command displays the correct hostname and FQDN:

```
$ hostnamectl
   Static hostname: server01.example.com
[...]
```

Cannot create initial administrator account with multiple Active Directory (AD) domains

When you create the initial administrator account on Tableau Server, you might see the following error if you selected AD as the authentication type:

```
Failed to authenticate username and password
```

This occurs when Tableau Server attempts to connect to multiple AD domains. For example, you might see this error if you install Tableau Server on a computer that is part of one domain and you attempt to authenticate AD users that are part of another domain.

Tableau Server on Linux Administrator Guide

Fonts

Tableau Server uses the fonts installed on the system to render workbooks based on the fonts used when a workbook was created. When a font is not available, Tableau Server will use the closest equivalent based on font families; this is true for both Windows and Linux Servers. On Linux Servers missing fonts may be more obvious because Linux ships with fewer included fonts than Windows and OS/X systems do. This matters because many workbooks are authored in Tableau Desktop on Windows or on Mac.

Tableau Server on Linux ships with the following fonts:

- Arial
- Courier
- Georgia
- Times New Roman
- Verdana
- Trebuchet MS
- Tableau Font

Workbooks which use fonts other than these may appear differently than expected when viewed on Tableau Server on Linux, due to missing fonts. To resolve this issue, install the appropriate fonts onto all nodes in your Tableau Server installation.

Support for Asian character sets

If you see empty boxes where you expect to see Asian characters in workbooks that are displayed on Tableau Server, then you should install the language-appropriate font packages in your Linux environment.

Initializing Tableau Server

TSM initialization fails because the `tableau` user account exists but is not a member of the group `tableau`

When you install and initialize Tableau Services Manager (TSM) and Tableau Server, the initialization script (`initialize-tsm`) creates the users and groups needed to run, or confirms that the existing ones are configured with the required characteristics. By default the

script creates a user called `tableau` and adds it to a group called `tableau`. If a `tableau` user already exists but is not part of the `tableau` group, the script fails with a warning.

If this happens you can fix the conflict by using a `--unprivileged-user` flag to specify a different user, and the user will be created and added to the `tableau` group.

For example, to specify a user named `tableauserver`, you would run the script from the `/opt/tableau/tableau_server/packages/scripts.<version_code>` directory using this command:

```
sudo ./initialize-tsm --unprivileged-user="tableauserver" --
accepteula
```

For a complete list of options that can be used with the `initialize-tsm` script, use the `-h` option:

```
sudo ./initialize-tsm -h
```

Error initializing Tableau Server on unsupported system locale

If you attempt to install Tableau Server on a computer with a locale that is not one of the 11 supported locales, you will get an error during installation.

Tableau Server will run on a system using one of the following locales:

`de_DE`, `en_GB`, `en_US`, `es_ES`, `fr_FR`, `it_IT`, `ja_JP`, `ko_KR`, `pt_BR`, `zh_CN`, or `zh_TW`.

Any other locale will generate the error.

Error initializing Tableau Server when `en_US.utf8` is not included in locale list

If you attempt to install Tableau Server on a computer that does not have `en_US.utf8` in the locale list, the initialization will fail with an error. To see if `en_US.utf8` is included, type `locale -a` at a shell prompt.

Tableau Server on Linux Administrator Guide

If `en_US.utf8` is not listed, you can add `en_US` to the locale list by typing `sudo locale-gen en_US.UTF-8` at a shell prompt on Ubuntu, or `sudo localedef -i en_US -f UTF-8` at a shell prompt on RHEL-like distributions.

Error: status 10 - initializing Tableau Server when data directory path includes a period

If you attempt to install Tableau Server and specify a data directory with a path that includes a period ("."), initialization will fail with errors including:

```
Connection timed out
```

and

```
ERROR: TSM services returned status 10
```

To avoid this issue, choose a data directory that does not include a period in its path.

Error initializing Tableau Server after reinstallation

If you uninstall and reinstall Tableau Server, you can encounter an error initializing Tableau Server. For example, you might see the following error:

```
ERROR com.tableau.tabadmin.webapp.asyncjobs.JobStepRunner - Running step WaitForConfigure failed
com.tableau.tabadmin.webapp.exceptions.ServiceFailedStateException
```

This error occurs when artifacts remain from a previous installation that cause services to fail to start. To prevent this error, use the `tableau-server-obliterate` script in the `/opt/tableau/tableau_server/packages/scripts.<version_code>` folder. For more information about completely removing Tableau Server, see [Remove Tableau Server from Your Computer](#).

Activating Tableau Server

Tableau Server license activation fails

In certain cases activation of the Tableau product key using the `tsm licenses activate -k <product_key>` command fails with an error:

```
License Server not available
```

This can happen if your computer is unable to connect through TCP port 443 to the Tableau licensing server at `licensing.tableau.com`.

To resolve this you need to configure your network and/or host-based firewalls to allow access to that address and port, or activate Tableau offline. For more information, see [Activate Tableau Server Offline - Adding a License](#).

Reindexing Tableau Server Search & Browse

Problems that can be solved by reindexing Search & Browse

Symptoms of an index that needs to be rebuilt include:

- A blank list of sites when a user attempts to log in
- A blank list of projects when a user tries to select a project
- Missing content (workbooks, views, dashboards)
- Unexpected or inaccurate alerts (for example, an "refresh failed" alert on a workbook that does not include an extract)

If you see any of these behaviors, rebuild the Search & Browse index using the `tsm maintenance reindex-search` command.

Restarting Tableau Server

Restarting Tableau Server or applying changes fails

If one of the Tableau Server services fails, you might see an error when you attempt to restart the server or apply configuration changes.

To see if a failed service is causing the error, type the following command:

Tableau Server on Linux Administrator Guide

```
tsm status -v
```

To find out why a service failed, view the `tabadminagent` and `tabadmincontroller` log files in the data directory. For example, a service might fail because of concurrency issues or port configuration issues. Please include any issues you encounter in your feedback.

As a workaround, you can attempt to resolve the failure by removing and re-adding the service in TSM. Once the service has started, you can retry your previous configuration change or retry restarting the server with the `tsm restart` command.

Error restarting Tableau Server after adding or configuring a node

If you add a or configure the node without a Gateway process, Tableau Server might fail to restart and you could see errors like these:

```
ERROR : com.tableau.tabadmin.configuration.PortConfigurationExtractor -  
Unable to find port config key worker1.gateway.port
```

and

```
Message: Missing port configuration value for key 'worker1.gateway.port'
```

These errors appear in the `gateway.log` file and occur when a Tableau Server node is configured with either an Application Server or VizQL Server but without a Gateway. A Gateway process is required if either Application Server or VizQL Server is running on a node.

Backup/Restore

Problems related to restoring a backup created by Tableau Server can be the result of permissions issues. Proper permissions are necessary for both the file that TSM is restoring, and the location of the file. When TSM handles the backup, it puts the file in a default location and sets permissions appropriately. You can run into permission problems if you are restoring a backup that was copied to your Linux server, or a backup from a non-default location on

your server. For details about using a non-default location, including how to change the location, see [tsm File Paths](#).

Errors may include:

```
Server Was Denied Access to File
```

or

```
Restoring the backup '<backup>.tsbak' was unsuccessful
```

or

```
Comparing authentication methods failed
```

The Tableau Server backup and restore processes must have:

- Read permission—The processes need to access the `.tsbak` backup file directly.
- Execute permission—The processes also require execute permissions to the directory structure in which the `.tsbak` file is placed.

When TSM creates a backup in the default location, it sets the permissions it needs. If you copy a file to the Linux server, or move it to a non-default directory, the permissions may not allow the TSM processes proper access. You need to verify that both the file, and the directory tree that contains it, allow access by the TSM user *tableau*. The file permissions must give the *tableau* user read access to the `.tsbak` file. You can do this by setting the group on the file to the *tableau* group, and giving that group read access. The directory permissions must give the *tableau* user read access. You can do this by setting the group on the directory to the *tableau* group, and giving that group read and execute access on the directories.

For detailed information about TSM and file permissions, see [Files and Permissions in TSM](#).

File locations

Changing basefilepath does not change the location of an existing file

Several tsm commands write files to default locations. You can change these default locations for each command using a tsm set command, but doing so does not move any existing files from the original location to the new one, and it does not create the new location. You are responsible for creating the new location, and for making sure it has the correct permissions to allow tsm access to any files in the location, and the entire directory structure that contains the files.

For more information about changing default locations for backup, restore, site import and export, and ziplogs files, see tsm File Paths.

For information about tsm permissions, see Files and Permissions in TSM.

TSM commands

TSM command line does not show progress for long-running tasks

If you run a tsm command such as restore or ziplogs that takes more than 2 hours to complete, the command will continue to run until completion on the server. To display the progress of the job, use the `tsm jobs reconnect` command.

Opening Firewall ports

Manually opening firewall ports on Ubuntu

The current version of Tableau Server does not support the `ufw` firewall that is used on Ubuntu. For customers that do not want to install `firewalld` on Ubuntu, another option is to manually open those ports. The following steps will confirm that `ufw` is running, and open TCP ports 8850 and 80 to connections from any source address:

1. Run the following command to confirm `ufw` is running:

```
sudo ufw status
```

If the result is `Status: inactive`, you will need to enable `ufw` and ensure that you can continue to connect via `ssh`, which is outside the scope of these release notes.

2. Run the following command to allow access to port 8850:

```
sudo ufw allow 8850
```

3. Run the following command to allow access to port 80:

```
sudo ufw allow 80
```

OpenID fails on first sign-in attempt

If you have configured Open ID Connect authentication for Tableau Server, the first sign-in attempt fails. To successfully log in, users must retry authentication after the initial failure.

Administrative views do not display

The Status tab of Tableau Server includes links to visualizations that display server metrics. These visualizations require the PostgreSQL driver to access the appropriate data from the Tableau Server repository. The PostgreSQL driver is not installed automatically, so if you have not installed the driver, the views will not display. For more information, see [Database Drivers](#).

Note: To use administrative views, the PostgreSQL driver must be installed on any node that runs the VizQL Server process.

Changing locale on view

When you change your user locale after opening a view, any subsequent attempt to open the view will fail with an "unexpected error." You can still open views that you have not previously opened.

To work around this issue, sign out of Tableau Server after changing your locale, and then sign back in. All views will display properly.

Work with Log Files

Tableau Server creates log files as a normal part of its activities. You may need to use the server log files when you are troubleshooting issues with Tableau Server or if Tableau Support requests logs from you to help you resolve an issue.

You can create a zipped log file archive using the `tsm maintenance ziplogs` command. The zipped archive contains copies of the logs you can unzip and look at, or send to Tableau Support. Once you have a copy of the archive, you can delete the archive from your server. For more information on log file archives, see [Log File Snapshots \(Archive Logs\)](#).

This collection of topics provides information about how to create log file archives, the contents of specific log files, and details about when and how you might want to look at a log.

Contents of Tableau Server Logs

Every Tableau Server process writes information about what it is doing to its own log file. Singly these give detailed information on the actions of each process. Taken together these log files contain detailed information about internal communication between components of Tableau Server while processing users' requests or performing automated tasks. Tableau Server logs only contain technical information useful for troubleshooting; the status of different components, actions taken by different processes, communication attempts, queries to the database (not including results), and timings of requests, for example.

Log files could contain some specific data such as names of database servers, as well as their IP addresses and ports, names or IP addresses of Tableau Server computers, and URLs and names of the workbooks and views accessed by users.

Log files do not contain any sensitive customer data such as passwords, results of the queries, or data shown on the views.

The `tsm maintenance ziplogs` command allows users to not only generate a zipped archive of log files, but also to include Tableau Server repository data if the `-d` option is specified. The repository contains metadata from Tableau Server (for example, usernames, groups, projects, permissions on Tableau Server, extract refresh schedules). The repository also includes layout and connection information for the workbooks, but does not have any data such as passwords, actual data from the database or data shown on the view.

Data displayed in views comes from extract files or databases, and is cached in memory. It is not saved in logs or, in the case of live connections, in separate files on Tableau Server computers. Extract files are stored on Tableau Server computers as `.hyper` files in the `dataengine` folder, but are never included in the zipped log archive.

Investigating Tableau Server Issues

The range and complexity of possible issues with Tableau Server means that there is no simple process you can use to investigate all problems, but a general approach would include these steps:

1. **Clean up** existing log files to reduce their size. For more information, see [Remove Unneeded Files](#).

Important: If there is a chance you will want to get help from Tableau Support troubleshooting an issue, be sure to create a zipped archive of your logs before cleaning them up. The clean up can delete important information Support may need. For details on creating log archives, see [Log File Snapshots \(Archive Logs\)](#).

2. **Set the appropriate logging level.** This is something that Tableau Support will instruct you on. For more information including impact of different log levels, see [Change Logging Levels](#).
3. **Reproduce the issue** you are troubleshooting so the logs capture the events related to the problem.
4. **Create an archive** of the logs. For more information see [Log File Snapshots \(Archive Logs\)](#).

Important: Use this archive when looking at the log files. You should not edit, move or delete any files directly on the server.

5. **Review the TSM Administration Controller log** (`/tabadmincontroller/tabadmincontroller_node<n>-<n>.log`) to understand any configuration or deployment done by TSM from the command line, Web UI, or API, including jobs started by TSM. Start with the controller log. This is where you'll get most useful information.

Note: The `tsm.log` is less wordy than the `tabadmincontroller_*.log` but can provide useful, complimentary troubleshooting information.

6. **Review the Apache logs** (`/httpd/access.####_##_##_##_##_#.log` and `/httpd/error.log`) for requests that may be related to the issue you are investigating.

The Apache logs will contain a fair amount of "noise" that does not apply to issues you are experiencing.

- If you find a request that seems to be related to your issue, search the `vizqlserver` directory for entries that include the unique request ID from the Apache logs.
 - Look for the response code and message associated with the request ID.
 - Search for the name of the workbook, view, dashboard, or data source that is related to your issue. Make sure to look for a relevant timestamp.
 - If you find a request that seems to be related to your issue, look at the response code associated with the request. (200s are good, 500s indicate problems.)
 - Locate the unique request ID associated with the request you've identified (the unique request ID is a 24 character alphanumeric string at the very end of the request).
7. **Review the log archive** further to search for other messages and possible errors.
 - Use the request ID from the Apache logs to search the `vizqlserver` folder of the log archive for files containing related log entries. Look for indications of a

problem (for example, error messages or long-running queries).

- The free, open source tool, Logshark can be a useful option for reviewing log archives. For more information, see [Troubleshooting Tableau Server](#) in the Tableau Blueprint.

8. Review script logging.

Tableau Server includes logs for most of the bash scripts that are included in the `scripts` directory at `/opt/tableau/tableau_server-
/packages/scripts.<version_code>/`. These logs are saved to the `/var/tmp` directory each time a script is run.

By default: `/var/opt/tableau/tableau_server/data/tabsvc/logs/`

9. Contact support

If you are not able to solve the issue yourself, or if requested by Tableau Support, send the zipped archive to Tableau.

Tableau Server Logs and Log File Locations

Tableau Server generates log files as a normal part of its functioning. Each service that runs as part of Tableau Server generates its own logs. These log files include information about what is happening on the server, what the service or process is doing, and what, if any errors or warnings are generated. The extent of information in the logs depends on which service is writing the logs, what the logging levels are set to, and what is happening on the server.

Looking for Tableau Server on Windows? See [Server Log File Locations](#).

Log files can be useful in helping to identify and fix issues that Tableau Server is having. In some cases, system administrators may be able to look at logs and find clues to what is happening, but in most situations the Tableau Server logs are most useful for Tableau Support. When you open a case with Support, you may be asked to send log files from your server.

Note: The specific directories and logs generated by Tableau Server depend on the version of server you are running, and which processes you have configured. New services and processes are added periodically to support new functionality. For details about processes or services you might find logs for, see [Tableau Server Processes](#).

Tableau Server log files on an active cluster

As a best practice you should not edit or delete log files in an active Tableau Server installation. Doing this can cause unexpected behavior or server downtime. Most Tableau Server logs are written to a location in the data directory. Some logs are written to other locations.

The easiest and safest way to gather and view server log files is to create a log archive, which is a zipped collection of logs from all nodes in a cluster. If you think you may need old logs for any reason, for example, to compare with new logs after doing an upgrade, or to send to Tableau Support when troubleshooting a server issue, create a zip archive, and move the archive to a safe location that is not part of your Tableau Server infrastructure. For more information about log files in a log archive, see [Server Log Files in a zipped archive](#).

Logs can take up a good deal of space, especially on a heavily used server. You can use the `tsm maintenance cleanup` command to remove logs you no longer want or need. but if you think you may need your existing logs, consider archiving them before cleanup.

Primary log locations on a working Tableau Server installation

Most of the Tableau Server logs are written to the data directory, `/var/opt/tableau/tableau_server/data/tabsvc/logs/`. Subdirectories are created for each instance of a service, with a name that includes the service name and the version code. For example:

```
/var/opt/tableau/tableau_server/data/tabsvc/logs/backgrounder
```

Configuration file locations on a working Tableau Server installation

In addition to logs for each service or process,, a `config` subdirectory contains configuration information about the service.

```
/var/opt/tableau/tableau_server/data/tabsvc/config/backgrounder
```

Tableau Support may ask you to gather some of these if you are working with them on a server issue. The contents can be analyzed by Support.

Logs that are not written in the primary location

A few logs are not part of the main set of logs, and are written to locations other than the normal log directories:

- The TSM log. The `tsm.log` file is located in `<home_dir>/tableau/tsm`.
- The install log. The `app-install.log` file is located in `/var/opt/tableau/tableau_server/logs`.
- The upgrade log. The `app-upgrade.log` file is located in `/var/opt/tableau/tableau_server/logs`.
- Bash script logs. Most Tableau Server bash scripts located in the `/scripts` directory (`/opt/tableau/tableau_server/packages/scripts.<version>`) generate their own logs. These are written to the `/var/tmp` directory each time a script is run.

Server Log Files in a zipped archive

You may want to look at Tableau Server log files, or need to send them to Tableau Support if you have a problem with your server. Use the `tsm maintenance ziplogs` command to create a zipped archive of log files from all nodes in your installation. By default, Tableau Server log file archives are gathered in a zip file called `logs.zip`, but you can specify a different file name when you create the archive. You can copy the archive from the server to a local computer and open it there, or send it to Tableau Support.

When you unzip the archive, a directory is created for each node in the cluster, and in that directory are sub-directories for each service or process using this naming convention:

```
<service_name>_<instance>.<version>.<build>
```

Tableau Server on Linux Administrator Guide

If there are multiple instances of a service on a node, there will be multiple directories for that service, one for each instance. For example, if you have two Backgrounders on a node, you will see directories like these:

```
backgrounder_0.<version>.<build>  
backgrounder_1.<version>.<build>
```

The specific directories and logs in the zip file depend on what version of Tableau Server you have, and which processes you have configured. For details about processes or services you might find logs for, see [Tableau Server Processes](#).

Log File Snapshots (Archive Logs)

Tableau Server includes functionality to generate a snapshot of log files for archival purposes. If you plan to clean up and delete old log files as part of regular server maintenance, you may consider archiving log files to an off-server storage location before deleting them.

Or, if you are working with Tableau Support on a case, the support engineer may request a server log file snapshot.

This topic describes:

- How to generate a log file snapshot
- How to send the snapshot directly to Tableau Support from the Tableau Server administration tools
- How to download the snapshot
- How to delete archived logs

Use the TSM web interface

1. Open TSM in a browser:

`https://<tsm-computer-name>:8850`. For more information, see [Sign in to Tableau Services Manager Web UI](#).

2. Click the **Maintenance**.
3. Generate a log file snapshot.

- a. On the Server Maintenance page, under Log Files, click **Generate Log File Snapshot**.

An options dialog displays:

- b. On the Options page, enter or select the options you want, including a **Description**, time **Range** of log files to be included, and the optional types of logs to be included (**Include Postgres Data**, **Include Recent Crash Dumps**), then click **Generate Log File Snapshot**.

The log file snapshot is saved to a fixed location on the computer where TSM and Tableau Server are installed. If you have a multi-node installation, the snapshot is saved to the initial node of the cluster. The location is specified by the `base-filepath_log_archives` variable.

By default the snapshot is saved to:

```
/var/opt/tableau/tableau_server/data/tabsvc/files/log-archives
```

You can find the current location by querying the `basefilepath_log_archives` setting, and change the location by specifying a new value for `base-filepath_log_archive`. For more information, see [tsm File Paths](#).

4. After you generate the snapshot, you can select it and upload it to Technical Support, download it to your local machine, or delete it:

Tableau Server on Linux Administrator Guide

Log Files

Generate a custom log file snapshot. After Tableau Server has generated the snapshot, you can download the snapshot or upload it to Tableau Technical Support. Generating

<input type="button" value="Generate Log File Snapshot"/>	<input type="button" value="Upload To Technical Support Case"/>	<input type="button" value="Download"/>	<input type="button" value="Delete"/>			
Description	Created	Range	Size	Stored in	Status	
<input type="radio"/>	Jun 7, 2018, 6:43:14 PM UTC	2 days	3.7 MB	node1	Succeeded	
<input type="radio"/>	Jun 7, 2018, 10:05:34 PM UTC	2 days	230 B	node1	Succeeded	
<input checked="" type="radio"/>	Logs generated at 6/19/2018, 7:37:26 AM	Jun 19, 2018, 2:45:53 PM UTC	2 days	69 MB	node1	Succeeded

5. To download a local copy of the snapshot or to delete it, select the snapshot under Log Files and then select the appropriate **Action**.

Uploading log snapshots for Tableau Support

1. Click the **Maintenance** tab.
2. Select the snapshot you want to send.

Log Files

Generate a custom log file snapshot. After Tableau Server has generated the snapshot, you can download the snapshot or upload it to Tableau Technical Support. Generating

<input type="button" value="Generate Log File Snapshot"/>	<input type="button" value="Upload To Technical Support Case"/>	<input type="button" value="Download"/>	<input type="button" value="Delete"/>			
Description	Created	Range	Size	Stored in	Status	
<input type="radio"/>	Jun 7, 2018, 6:43:14 PM UTC	2 days	3.7 MB	node1	Succeeded	
<input type="radio"/>	Jun 7, 2018, 10:05:34 PM UTC	2 days	230 B	node1	Succeeded	
<input checked="" type="radio"/>	Logs generated at 6/19/2018, 7:37:26 AM	Jun 19, 2018, 2:45:53 PM UTC	2 days	69 MB	node1	Succeeded

3. Click **Upload to Technical Support Case**.
4. In the dialog that displays, enter the **Support Case Number** and your **Contact**

Email Address, then click **Upload Snapshot**.

Upload To Technical Support Case
✕

Upload your log file snapshot to Tableau Technical Support. Upload time is dependent upon the network connection and size of the log file snapshot. To file a new support case, select the Info icon in the header and then select Support.

Support Case Number

Contact Email Address

You are about to upload the following log file snapshot:

Description	Logs generated at 6/19/2018, 7:37:26 AM
Created	Jun 19, 2018, 2:45:53 PM UTC
Range	3 days
Size	69 MB

[Tableau Software Privacy Policy](#)

Cancel
Upload Snapshot

Use the TSM CLI

You create a snapshot archive of Tableau Server log files using the `tsm maintenance ziplogs` command.

By default, this command creates a zip file containing all of the log files. If you are running a distributed installation of Tableau Server, perform this step from the initial node. Logs from all nodes will be included in the zip file.

Note: If you cannot run the `ziplogs` command successfully, you can manually zip the Tableau Server logs. For more information, see [Troubleshoot Tableau Server on Linux](#).

To create a log file snapshot:

Tableau Server on Linux Administrator Guide

1. On the initial node, open a terminal session.
2. Type the following command:

```
tsm maintenance ziplogs -l -f <filename>
```

where `<filename>` is name of the zipped archive file you want to create. Choose a unique name with no spaces. If an existing ziplog with the same file name already exists the creation of the file will fail unless you include the `-o` option to force an overwrite, delete the existing file, or specify a different name in the command.

You can specify a time range for the snapshot and you can also specify which types of logs to include. For more information, see `tsm maintenance ziplogs`.

The log file snapshot is saved to a fixed location on the computer where TSM and Tableau Server are installed. If you have a multi-node installation, the snapshot is saved to the initial node of the cluster. The location is specified by the `basefilepath_log_archives` variable.

By default the log file snapshot is saved to:

```
/var/opt/tableau/tableau_server/data/tabsvc/files/log-archives
```

You can find the current location by querying the `basefilepath.log_archives` setting:

```
tsm configuration get -k basefilepath.log_archive
```

and change the location by specifying a new value for `basefilepath.log_archive`:

```
tsm configuration set -k basefilepath.log_archive -v  
"<drive>:/new/directory/path"
```

For more information, see [tsm File Paths](#).

Sending log archives to Tableau Support

You can send log files to Tableau Support as a part of a customer support case (a customer support case number is required). Before sending a log file, use `tsm maintenance zip-logs` command to combine the log files into a single zip file archive. If you are creating the archive to send to Tableau Support, see the [Knowledge Base](#) for information about how to upload large files.

- In a terminal session, type the following command:

```
tsm maintenance send-logs -f <zip file name> -c <case number> -e <email address>
```

where `<case number>` is your support case number, `<email address>` is your contact email for this support case, and `<zip file name>` is the file name of your archive with `.zip` file extension.

Change Logging Levels

By default, Tableau Services Manager (TSM) and Tableau Server log events at the **Info** level. You can change this if you need to gather more information (if you are working with Tableau Support, for example).

As a best practice you should not increase logging levels except when troubleshooting an issue, as instructed by Support. You should only set a logging level to debug when investigating a specific issue. Changing log levels can have these impacts:

- Increasing the log level to `debug` or `trace` increases the amount of information being logged and can have a significant impact to performance. Reproduce the issue and then reset the logging level back to `info`.
- Setting the log level to `warn` or `error` can reduce the amount of information so much that it is not useful for Tableau Support.

Logging Levels

The following logging levels are listed in order of increasing amount of information logged:

Tableau Server on Linux Administrator Guide

- off
- fatal
- error
- warn
- info (the default)
- debug
- trace

Change Logging Levels

Set logging levels for TSM and Tableau Server processes using **tsm configuration set** configuration keys. The key you use depends on which component of TSM or Tableau Server you want to change the logging level for.

Dynamic log level configuration

In version 2020.2 we introduced dynamic configuration. The capability has been expanded in subsequent releases. If you are only changing logging levels for one or more of these components, and are running the appropriate version of Tableau, you can change the logging levels without restarting Tableau Server.

These logging levels are dynamically configurable, beginning with these versions:

- 2020.2 - tsm services (`tsm.log.level`) and control application services (`tsm-controllerapp.log.level`).
- 2020.3 - backgrounder (`backgrounder.log.level`), cluster controller (`cluster-controller.log.level`), data server (`dataserver.log.level`), file store (`filestore.log.level`), data source properties (`tdsservice.log.level`), and VizQL server (`vizqlserver.log.level`).
- 2020.4 - adds interactive microservice container (`tomcatcontainer.log.level`) and application server (`viz-portal.log.level`).

Configuration Keys for Changing Logging Levels

This table includes both dynamically configurable keys and those that are not dynamically configurable.

Configuration key	Location of affected logs (path begins with /var/- opt/tableau/tableau_server- /data/tabsvc/logs/)
<p>tsm.log.level</p> <p>Changes TSM logging levels for: clientfileservice, licenseservice, tabadminagent, tabadmincontroller, tabsvc</p>	<p>/<service>/<service>_node<n>-<instance>.log</p> <p>example: /clientfileservice/clientservice_node1-0.log</p>
<p>tsm.controlapp.log.level</p> <p>Changes TSM logging levels for: control applications</p>	<p>/<service>/control_<service>_node<n>-<instance>.log</p> <p>examples: /clientfileservice/control_clientservice_node1-0.log</p> <p>/filestore/control_filestore_node1-0.log</p>
<p><process>.native_api.-log.level</p> <p>Valid process names are backgrounder, vizportal, vizqlserver, data-server</p> <p>Note: These are not dynamically configurable.</p>	<p>/vizqlserver/*.txt</p>

Tableau Server on Linux Administrator Guide

<pre>backgrounder.log.level</pre> <p>Changes logging levels for: Backgrounder</p>	<pre>/backgrounder/*.log</pre>
<pre>clustercontroller.log.level</pre> <p>Changes logging levels for: Cluster Controller</p>	<pre>/clustercontroller/*.log</pre>
<pre>dataserver.log.level</pre> <p>Changes logging levels for: Data Server</p>	<pre>/dataserver/*.log</pre>
<pre>filestore.log.level</pre> <p>Changes logging levels for: File Store</p>	<pre>/filestore/*.log</pre>
<pre>gateway.log.level</pre> <p>Changes logging levels for: Gateway</p>	<pre>/gateway/*.log</pre>
<pre>tdsservice.log.level</pre> <p>Changes logging levels for: Data Source Properties service</p>	<pre>/tdsservice/*.log</pre>
<pre>tomcatcontainer.log.level</pre> <p>Changes logging levels for microservices in: Interactive Microservice Container and Non-Interactive Microservice Container</p>	<pre>/tomcatcontainer/*.log</pre>
<pre>vizportal.log.level</pre>	<pre>/vizportal/*.log</pre>

Changes logging levels for: Application Server	
<code>vizqlserver.log.level</code>	<code>/vizqlserver/*.log</code>
Changes logging levels for: VizQL Server	

For more information, see [tsm configuration set Options](#).

If you are only changing dynamically configurable logging levels, you do not need to stop or start the server (for more information, see [Dynamic log level configuration](#) above). If you are changing other logging levels, you need to stop Tableau Server before changing the logging levels, and restart it afterward. On a multi-node installation of Tableau Server, set logging levels from the initial node.

To change the logging level:

1. (Optional for dynamically configurable logging levels in 2020.2.0 and later) Stop Tableau Server by opening a command prompt and typing:

```
tsm stop
```

2. Set the logging level to by typing `tsm configuration set -k <config.key> -v <config_value>`

where `<config.key>` is one of the keys in the above table and `<config_value>` is a valid logging level.

Examples:

- `tsm configuration set -k backgrounder.native_api.log.level -v debug`
- `tsm configuration set -k tsm.log.level -v debug`

Tableau Server on Linux Administrator Guide

- `tsm configuration set -k tsm.controlapp.log.level -v debug`

3. Apply pending changes by running the `tsm pending-changes apply` command.

4. (Optional, only if server is stopped) Start Tableau Server by running the following command:

```
tsm start
```

Reset Logging Levels

After you reproduce the issue and gather the information related to the issue, reset the logging levels so there is no lingering performance impact and no additional disk space used up.

Reset the logging level back to its default (info) using the appropriate command with a `-d` option. You need to apply pending changes after resetting the level, and if you are resetting logging levels for Tableau Server processes, you must stop the server before making the change, and start it applying the pending changes.

Examples:

- `tsm configuration set -k backgrounder.native_api.log.level -d`
- `tsm configuration set -k tsm.log.level -d`

Troubleshoot Tableau Server Install and Upgrade

Follow the suggestions in this topic to resolve common issues with Tableau Server. For additional troubleshooting steps based on process status viewed on the Status page, see [Troubleshoot Server Processes](#).

General Troubleshooting Steps

Many Tableau Server issues can be addressed with some basic steps:

1. Make sure there is enough disk space on each computer running Tableau Server. Limited disk space can cause a failure to install, a failure to upgrade, or problems running Tableau Server.
2. Restart Tableau Server. Issues related to processes not fully started can be resolved by restarting Tableau Server in a controlled way. To restart Tableau Server, use the `tsm restart` command. This will stop all the processes associated with Tableau Server and then restart them.
3. Reindex Tableau Server. Issues related to indexing can be resolved by reindexing Tableau Server. To reindex Tableau Server, use the `tsm maintenance reindex-search` command. For more information, see Reindexing Tableau Server Search & Browse below.
4. Restart the computer on which Tableau Server is running. Some issues, such as those related to data source connectivity, can be resolved by restarting the server computer.

Common Tableau Server Install Issues

Installation logs location

The install log, `app-install.log`, is located in `/var/opt/tableau/tableau_server/logs`.

The upgrade log, `app-upgrade.log`, is located in `/var/opt/tableau/tableau_server/logs`.

Unable to log into TSM or Tableau Server (sign in screen redisplay after entering credentials)

Using Internet Explorer or Edge, if you enter your credentials into the TSM or Tableau Server sign-in screen and the page redisplay without signing you in, verify that the hostname or domain in your URL does not include an underscore (`_`). If the hostname or domain of the Tableau Server computer includes an underscore (`_`) Internet Explorer or Edge

Tableau Server on Linux Administrator Guide

browsers will not set a cookie, so the page will redisplay without signing you in. To work around this, use "localhost" or the IP address of the computer in the URL. For example:

```
https://localhost:8850.
```

For more information, see the [Tableau Knowledge Base](#).

Multiple install attempts fail

If you attempt to install Tableau Server and the install fails, any subsequent installation attempts are likely to fail unless you run the `tableau-server-obliterate` script to clean Tableau off the computer.

A failed install attempt can leave the computer in a state that causes subsequent attempts to also fail with errors that don't seem directly related to a previous install attempt. One possible error is:

```
Enabling and starting all services
+ services=(appzookeeper* tabadmincontroller* tabsvc* licenseservice*
fnlicenseservice* tabadminagent* clientfileservice*)
+ systemctl_user enable appzookeeper_0.service 'tabadmincontroller*'
'tabsvc*' 'licenseservice*' fnlicenseservice_0.service 'tabadminagent*'
'clientfileservice*'
++ id -ru a_tabadminpoc
+ local unprivileged_uid=222954
+ su -l a_tabadminpoc -c 'XDG_RUNTIME_DIR=/run/user/222954 systemctl --user
enable appzookeeper_0.service tabadmincontroller* tabsvc* licenseservice*
fnlicenseservice_0.service tabadminagent* clientfileservice*'
Failed to execute operation: No such file or directory
```

To fix this problem, run the `tableau-server-obliterate` script to clean up any leftover remnants of the previous install attempt and then restart the computer. For more information, see [Running the tableau-server-obliterate script](#).

Important: If you created a backup of Tableau (<file>.tsbak) you want to keep (for example, to restore to your new installation), copy that file to a safe location on another computer to guarantee it is not removed when you clean up your Tableau computer.

Install fails due to hardware requirements

Tableau Server cannot install if the computer you are installing on does not meet the minimum hardware requirements. The requirements apply to all computers on which you are installing Tableau Server. For details on minimum hardware requirements, see [Minimum Hardware Requirements and Recommendations for Tableau Server](#).

Install or upgrade fails due to CPU requirements

Beginning in version 2020.4.0 Tableau Server requires CPUs that support SSE4.2 and POPCNT instruction sets. You cannot install or upgrade Tableau Server 2020.4.0 or oater on computers that have CPUs which do not support these instruction sets.

You may see this error message when installing a new installation, or in preparation for upgrading an existing installation:

```
Your computer's processor doesn't meet the minimum requirements that Tableau requires to install the software. If you are using a VM, make sure Processor compatibility mode is off.
```

The SSE4.2 and POPCNT instruction sets have been common for more than 10 years and most newer CPUs support them, but if you get an error related to processor minimum requirements when attempting to install or upgrade Tableau Server on a Virtual Machine (VM), Processor compatibility mode may be enabled on the VM. To successfully install or upgrade Tableau on a VM, make sure the Processor compatibility mode is turned off.

Common Tableau Server Upgrade Issues

Maps do not display or display incompletely after upgrading

Beginning with Tableau version 2019.2, the internet access requirements changed for maps. If you are upgrading from version 2019.1.x or earlier to version 2019.2.x or later, and maps are not displaying as expected, confirm that your environment is configured to allow access on port 443 to `mapsconfig.tableau.com` and `api.mapbox.com`.

In version 2019.1.x or earlier, access was necessary to `maps.tableausoftware.com`.

For more details on internet access requirements, see [Communicating with the Internet](#).

Upgrade script error: "Tableau Server Version change validation failed."

When upgrading, if you run the `upgrade-tsm` script from the `scripts.<version_code>` directory for the earlier version, the upgrade will fail with an error:

```
Tableau Server Version change validation failed.  
Tableau Server <version> is already installed.
```

If you get this error, change to the `scripts.<version_code>` directory for the version you just installed and run the script from there.

Upgrade logs location

The upgrade log, `app-upgrade.log`, is located in `/var/opt/tableau/tableau_server/logs`.

Upgrade multi-node, initializing additional node fails with "Enter your credentials again" error

If you attempt to initialize an additional node when upgrading Tableau Server and see this error:

```
Enter your credentials again. The credentials you enter must  
provide administrative access to the computer where you gen-  
erated the configuration file.
```

this is an indication that the node is unable to connect to or communicate with the initial node.

This can happen for multiple reasons:

- The credentials you entered are not valid or you mistyped them. The credentials must be for a user who has administrative permissions on the computer where Tableau Server was first installed. You do not need to use the credentials of the user who created the bootstrap file but doing so will ensure you are using valid credentials.
- The local firewall of the computer you are trying to add is not allowing communication to the initial node. For more information, see [Local firewall configuration](#).

Upgrading fails due to lack of disk space

If there is not enough disk space for the Tableau Server Setup program to run and do the upgrade, the installation will fail. The amount of disk space required will depend on the size of your repository database and the number and size of your extracts.

To free up disk space:

1. Create a log archive snapshot using the `tsm maintenance ziplogs` command.

After you create the ziplogs file, save it to a safe location that is not part of your Tableau Server installation.

2. Clean up unnecessary files using the `tsm maintenance cleanup` command.
For more information, see [Remove Unneeded Files](#).

Upgrade fails on RebuildSearchIndex job

Beginning with version 2020.1.x, the final step in an upgrade is to rebuild the search index. At this point all services have been upgraded, so if this job fails, you can manually reset the search server by running the `tsm maintenance reset-searchserver` command. You do not need to obliterate and start over.

The error will be:

```
An error occurred while rebuilding search index.
```

To reset the search server :

Tableau Server on Linux Administrator Guide

1. On the initial node, open a terminal session.

This must be a new terminal session because the upgrade script updates system environment for the new version.

2. Rebuild the search index using the `tsm maintenance reset-searchserver` command.

Upgrade fails on 2020.4.0 or later

Beginning with version 2020.4.0, the Checkpoint Upgrade feature allows you to retry a failed upgrade. In general, this is most useful for experienced server administrators and IT professionals who are comfortable with Tableau Server log files and are willing to search through them. But the feature can help in all failed upgrades because it allows you to rerun the `upgrade-tsm` script, and the script is run from the last successful step, saving time. For those with experience, it may be possible to identify problems like disk space problems, or permissions issues, correct them, and rerun the upgrade.

If you are upgrading to version 2020.4.0 or later and the upgrade fails, the following steps may help you to complete the upgrade:

- Rerun the `upgrade-tsm` script. Upgrade failures are sometimes a result of timeouts during the upgrade process, and rerunning the script can allow the upgrade to get beyond intermittent or occasional timing issues. This is also a step that is safe to do, and easy. Rerunning the script will do no harm, and at worst, the upgrade will fail again at the same point, but without needing to go through any previous steps.

The script is located in the `\scripts` directory:

```
opt/tableau/tableau_server/packages/scripts.<version_code>/upgrade-tsm
```

If your Tableau Server upgrade isn't successful when you rerun the `upgrade-tsm` script, and you are comfortable with Tableau Server logs, you can take these additional troubleshooting steps:

- Look at the output of the script in the command window. Useful error messages may

help you identify the cause of the upgrade failure and give you some ideas for how to correct the issue.

- Look in the `app-upgrade.log` file. Any errors that are displayed at the command line will also appear in the `app-upgrade.log` file, often with more details.
- Look in the `tabadmincontroller.log` file. Upgrade problems that aren't easily identifiable in the above two instances are likely the result of an issue in a job. The `tabadmincontroller.log` file may have more information that helps you diagnose the issue.

Note: For information about log file locations, see [Tableau Server Logs and Log File Locations](#).

Common Settings Import Issues

Import of settings file causes "not present on any node" validation error due to missing services

If you are upgrading by installing a new version of Tableau Server and importing a settings file from an earlier version, you may encounter topology validation errors when running the `tsm settings import` command.

This can happen when you export a settings file from an older version of Tableau Server and import it into a new version, and new services have been added to Tableau between the two versions.

Errors will be similar to this (the specific service may be different):

```
>tsm settings import -f 20183-export.json
```

```
Pending topology set.
```

```
There are 1 topology validation errors/warnings.
```

```
Service 'elasticsearch' is not present on any node in the cluster.
```

```
Service: Elastic Server
```

To resolve this issue, add any missing services to Tableau Server:

1. For any service that generated a validation error, add the service with an instance count of 1.

For example, if the Elastic Server is not present in the cluster, set the process instance count to 1 using the service name that appears in the first line of the validation error message:

```
tsm topology set-process -n node1 -pr elasticserver -c 1
```

Repeat this step for each service that results in an error.

2. When you have no more warnings or errors, apply the pending changes:

```
tsm pending-changes apply
```

Your settings should be imported successfully.

Import of settings file causes "configuration value you specified does not match" error

If you are installing a new version of Tableau Server and import a settings file from an earlier version, you may encounter configuration validation errors when running the `tsm settings import` command. These can occur when a settings file includes a configuration value that has since been removed from Tableau.

The error will look similar to this (the configuration key may be different):

```
>tsm settings import -f 20183-export.json
Configuration error: At least one configuration value you specified does not match a known configuration key. This applies to the following keys: '[features.TsmConfigFileService]'
Use this parameter to override unknown key error: --force-keys
```

To resolve this issue, edit the settings file you are importing to remove the reference to the configuration key or keys in the error:

1. Copy the JSON settings file and save the copy for backup.
2. Open the JSON settings file in a plain text editor.
3. Locate and delete the entire line that includes the key. In this example, `features.TsmConfigFileService`:

```
"configKeys" : {
  "config.version" : 19,
  "tabadmincontroller.port" : "8850",
  "endpoints.enabled" : false,
  "endpoints.health.enabled" : true,
  "features.TsmConfigFileService" : true,
  "tableau_projects.language" : "en",
```

The above is an example of a small section of an exported settings file and is not intended to represent the entire contents of the file.

4. Save the settings file and import it again.

You may encounter additional errors related to topology validation. For information about solving those errors, see [Import of settings file causes "not present on any node" validation error due to missing services above](#).

"You cannot directly modify instances of the Coordination Service" error

This error can occur in two situations:

- When you import a Tableau Server settings file into an installation that has a different Coordination Service topology than the settings file does
- When you attempt to configure the Coordination Service using the `tsm topology set-process` command

If you see this error after importing a settings file:

The Tableau Server settings file has a different Coordination Service topology than the target server does. This can happen if you are upgrading Tableau Server by installing a new version and importing a settings file from an earlier version. If you have not explicitly

deployed a Coordination Service ensemble on the target server, it has a single instance of Coordination Service, on the initial node.

To correct this error you can take either correct the mismatch from the command line, or by editing the settings import file. You can also discard all pending changes, deploy the Coordination Service on the target computer to match the settings in the import file, and reimport the settings file.

To correct the mismatch from the command line, for each node that generates an error, use the `tsm topology set-process` command to revert the instance count of Coordination Service.

1. Run the `tsm pending-changes list` command. The output shows you which nodes have changes.
2. Find the node or nodes where the Coordination Service count is changed.

For example, if the settings file had a Coordination Service instance on node2, but the target system did not have any Coordination Service instance on that node, the count for node 2 would show as changed from 0 to 1 by the import of the settings file:

```
C:\Windows\system32>tsm pending-changes list
Configuration
There are no pending configuration changes.
Topology
node2:
    Coordination Service
        New Instance Count:1
        Old Instance Count:0
```

3. Use the `tsm topology set-process` command to set the count back to the "Old Instance" value.

For the example above:

```
tsm topology set-process -n node2 -c 0 -pr "Coordination
Service"
```

4. Once you have reset any Coordination Service instance count that was changed, apply pending changes:

```
tsm pending-changes apply
```

If you see the error when setting the process count for Coordination Service manually:

This error can also occur if you attempt to update the Coordination Service directly, using the `tsm topology set-process` command instead of the `tsm topology` commands for managing the Coordination Service. If you tried this:

1. Use the `tsm pending-changes discard` command to discard the pending changes.
2. Use the correct commands for configuring the Coordination Service. For more information, see [Deploy a Coordination Service Ensemble](#).

Starting Tableau Server

Tableau Server cannot determine if it fully started

In some instances Tableau Server may report that it could not determine if all components started properly on startup. A message displays: "Unable to determine if all components of the service started properly."

If you see this message after starting, verify that Tableau Server is running as expected by using a `tsm status -v` command.

If the status shows as running ("Status: RUNNING"), then the server successfully started and you can ignore the message. If the status is DEGRADED or STOPPED, see "Tableau Server doesn't start" in the next section.

Tableau Server doesn't start

If Tableau Server does not start or is running in a degraded state, run the `tsm restart` command from a command prompt. This will shut down any processes that are running, and restart Tableau Server.

Reindexing Tableau Server Search & Browse

Problems that can be solved by rebuilding Search & Browse index

Symptoms of an index that needs to be rebuilt include:

- A blank list of sites when a user attempts to log in
- A blank list of projects when a user tries to select a project
- Missing content (workbooks, views, dashboards)
- Unexpected or inaccurate alerts (for example, an "refresh failed" alert on a workbook that does not include an extract)

If you see any of these behaviors, reset and rebuild the Search & Browse index using the `tsm maintenance reset-searchserver` command.

Activating Tableau Server

Tableau Server license activation fails

In some instances Tableau Server license activation may fail. Error messages can range from a very generic one:

- `An error has occurred`

To more specific messages:

- `Function flxActCommonLicSpcPopulateFromTS returned error 50030, 71521,`
- `No license found for 'Tableau Server'`

To resolve this issue, try these solutions in the order listed:

Confirm you can access the licensing server

The Tableau licensing service was moved to a new data center on October 6, 2018. This means any environments that required special configuration (static IP safe listing for

example) to access `licensing.tableau.com` or `licensing.tableau.com` will need to be updated before you can activate, refresh, or deactivate a Tableau product key.

To test access, type the URL and the port of the licensing server in a browser:

```
https://licensing.tableau.com:443
```

and:

```
https://atr.licensing.tableau.com/_status/healthz
```

If you are able to access the server, a "Test success" message displays for the first server, and an "OK" message displays for the second.

Tableau Server needs to make a connection to the following internet locations for licensing purposes:

- `licensing.tableau.com:443`
- `atr.licensing.tableau.com:443`
- `s.ss2.us`
- `ocsp.rootg2.amazontrust.com`
- `ocsp.rootca1.amazontrust.com`
- `ocsp.sca1b.amazontrust.com`
- `crt.sca1b.amazontrust.com`
- `crt.rootca1.amazontrust.com`
- `ocsp.sca0a.amazontrust.com`
- `crt.sca0a.amazontrust.com`
- `ocsp.sca1a.amazontrust.com`
- `crt.sca1a.amazontrust.com`

Tableau Server on Linux Administrator Guide

- obsp.sca2a.amazontrust.com
- crt.sca2a.amazontrust.com
- obsp.sca3a.amazontrust.com
- crt.sca3a.amazontrust.com
- obsp.sca4a.amazontrust.com
- crt.sca4a.amazontrust.com
- crl.rootca1.amazontrust.com
- crl.rootg2.amazontrust.com
- crl.sca1b.amazontrust.com

Requests to the above domains may be on port 80 or 443. Port 80 is used for certificate validation (revocation, certificate chain, etc). Port 443 is used for SSL connections.

Verify the date and time

Verify the date and time on the initial Tableau Server computer is correct. If the clock is set to a time and date earlier than the current date, Tableau Server cannot be activated.

Force the product key to be read again

1. On the initial Tableau Server computer, log on as a user with sudo access.
2. Change to the Tableau Server bin directory. By default this is:

```
/opt/tableau/tableau_server/packages/bin.<version_code>/
```

3. Type the following commands:

```
tsm stop
```

```
./lmreread
```

```
tsm start
```

Send the contents of trusted storage to Tableau Support

If FlexNet Licensing Services is installed and running but you're still seeing an error, there might be a problem with the Tableau product key information. To resolve this issue, complete the following steps to create a file of the key information located in trusted storage.

1. On the initial Tableau Server computer, log on as a user with sudo access.
2. Type the following command:

```
serveractutil -view > <machine_name>-LicResults.txt
```

This creates the `<machine_name>-LicResults.txt` file in your current directory. If you don't have write permissions for that location and see an error, change to a location where you do have permission to create a file and run the command again.

3. Contact Tableau Support (<http://www.tableau.com/support/request>) and include the `<machine_name>-LicResults.txt` file that you created.

tabcmd Installation Problems

Installing tabcmd separately

tabcmd is automatically installed on the initial Tableau Server node when you install Tableau Server, but if you want to run it on another computer, you need to download and install tabcmd separately. For details, see [Install tabcmd](#).

Problems installing tabcmd on Linux

tabcmd requires Java 8 (also known as Java 1.8) to run properly. On RHEL-like systems, this will be installed as a dependency when installing tabcmd. On Ubuntu systems, you need to install Java 8 (1.8) separately if it is not already installed.

As of July 2022, Debian distributions are no longer supported. For more information, see [this Tableau Community post](#).

Tableau Server on Linux Administrator Guide

Java is not installed

If you see errors similar to this when installing `tabcmd`, confirm that Java 8 is installed on your Linux computer:

```
Cannot find 'java' in your PATH. Install 'java' and make sure it
is in your PATH to continue.
```

Incorrect version of Java is installed

If you see errors similar to these, confirm that Java 8 is installed:

```
Exception in thread "main" java.lang.Un-
supportedClassVersionError: com/tableausoftware/tabcmd/Tabcmd :
Unsupported major.minor version 52.0
```

or.

```
*** Uncaught exception NoClassDefFoundError: javax/xml-
l/bind/JAXBException
*** See the logs for the stacktrace.
```

systemd User Service Failures

You may receive one of the following errors when upgrading or when running `initialize-tsm` during a fresh installation:

- "Failed to get D-Bus connection: No such file or directory"
- "\$XDG_RUNTIME_DIR not found"
- "systemd unit user@<userID> is not running. Check /var/log/messages or /var/log/syslog."

Background

As of 2018.1, Tableau Server uses the `systemd` user service to manage processes. This means there is a `systemd` process that runs as the unprivileged user. By default, Tableau Server Setup creates an unprivileged account named `tableau`. The Tableau Server processes are spawned from the `systemd` process and not the system-wide `systemd` process, which runs as root.

Important: This troubleshooting note applies primarily to RHEL 7-based distros. However, if you see one of these errors, it's possible that the same issues exist on Ubuntu distros.

The `systemd` user service is not used as commonly as the normal `systemd` process manager. Red Hat disabled the `systemd` user service in RHEL 7 (and thereby all distros that come from RHEL, like CentOS, Oracle Linux 7, Amazon Linux 2). However, RedHat has assured Tableau that running the `systemd` user service is supported as long as the service is re-enabled.

Upgrading from Tableau Server on Linux 10.5

If you are upgrading from Tableau Server 10.5, check that the unprivileged user has a valid shell and home directory. For Tableau Server 10.5, Tableau deliberately created the unprivileged user with shell set to `/sbin/nologin` and home directory `"/"`. If the unprivileged user was created by `initialize-tsm`, then during upgrade to 2018.1 Tableau updates the shell and home directory.

However, if you created the unprivileged user during the initial installation of 10.5, then you will get an error when trying to upgrade.

To fix this, you must set the shell to `/sbin/nologin` and the home directory `"/"`, and then run upgrade again.

Fresh installation error troubleshooting

Verify that the `systemd` user service is running.

Check by running the command, `ps -fww $(pgrep -f "systemd --user")`

If the `systemd` user service is not running, then something prevented it from starting.

Follow this list to troubleshoot:

Tableau Server on Linux Administrator Guide

- Check the logs in `/var/log/messages`
- Run `journalctl`
- Verify that any customization that you may have done to your PAM configuration includes has not removed `pam_systemd.so`.

If the RHEL 7 PAM file, `/etc/pam.d/system-auth` is missing the following line:

```
-session optional pam_systemd.so
```

then it must be added back for Tableau Server to function.

- If `-session optional pam_systemd.so` is present in your PAM configuration, the user service cannot start, and the error message `$XDG_RUNTIME_DIR not found` is showing in `/var/log/messages`, do not attempt to set the environmental variable. In this scenario, the error is not accurate.

The real error is that the PAM module `pam_systemd.so` is unable to allocate the user session. The default configuration suppresses error messages from `pam_systemd.so`. To surface error messages and debug messages, change the line in `/etc/pam.d/system-auth` from `-session optional pam_systemd.so` to `session optional pam_systemd.so debug`. (Removing the leading hyphen will surface the error messages, and adding `debug` will surface more verbose logging.)

Now you can look in `/var/log/messages`, `/var/log/secure` and `/var/log/audit/audit.log` files to see error messages.

Example

You may see the following error message:

```
systemd-logind: Failed to mount per-user tmpfs directory /run-  
/user/0: Permission denied
```

In this case, searching the error online leads to the Redhat KB article, <https://access.redhat.com/solutions/2460611>.

The article recommends updating the `selinux-policy` package by running `sudo yum update selinux-policy`.

In some cases, upgrading from version 3.12.X to 3.13.X fixes a `$XDG_RUNTIME_DIR` not found problem. Be sure to run `sudo reboot` after updating the package.

Troubleshoot Server Sign in Problems

There are several different sign in options between Tableau Services Manager (TSM) and Tableau Server.

- **TSM**—If you are not able to sign into TSM, make sure you are using credentials for a user who has administrative rights to the computer where TSM is installed. This user may or may not also be a Tableau Server administrator. This is true whether you are signing in to the Web UI or the CLI. For more information, see [Sign in to Tableau Services Manager Web UI](#).
- **Tableau Server**—
 - Administrators: if you are signing into Tableau Server as an administrator, you must use credentials for a user who has an administrator role in Tableau Server. You create the initial administrator when you first install Tableau, but can add other users as administrators once Tableau is installed and running. For more information, see [Sign in to the Tableau Server Admin Area](#).
 - Non-administrative users: If you are signing into Tableau Server as a user, you need to use credentials for a user who has been added to Tableau Server. For more information, see [Sign in to Tableau Server or Online](#).

Note: If users with valid credentials are unable to sign into Tableau Server, make sure you have not added a node without applying pending changes. If you have a pending new node, signing into Tableau Server may not be possible.

Troubleshooting scenarios

Unable to log into TSM or Tableau Server (sign in screen redisplay after entering credentials)

Using Internet Explorer or Edge, if you enter your credentials into the TSM or Tableau Server sign-in screen and the page redisplay without signing you in, verify that the hostname or domain in your URL does not include an underscore (_). If the hostname or domain of the Tableau Server computer includes an underscore (_) Internet Explorer or Edge browsers will not set a cookie, so the page will redisplay without signing you in. To work around this, use "localhost" or the IP address of the computer in the URL. For example:
`https://localhost:8850.`

For more information, see the [Tableau Knowledge Base](#).

Troubleshoot Licensing

This topic includes instructions for troubleshooting issues related to Tableau Server licensing.

Handle an unlicensed server

Tableau offers two licensing models: role-based and core-based. To learn more about role-based and core-based licensing, see [Licensing Overview](#).

role-based licensing requires each active user account to be covered by a license. role-based licenses have a defined capacity, or number of users that they allow. Each user is assigned a unique user name on the server and is required to identify themselves when connecting to the server.

Core-based licensing has no constraints on the number of user accounts in the system, but it does restrict the maximum number of processor cores that Tableau Server can use. You can install Tableau Server on one or more machines to create a cluster, with the restriction that the total number of cores in all the machines does not exceed the number of cores you have licensed and that all of the cores on a particular machine are covered by the license.

Unlicensed role-based server

The most common reason for a server that has role-based licensing to be unlicensed is an expired product key or an expired maintenance contract.

Unlicensed core-based server

A core-based server can become unlicensed for a variety of reasons, such as an expired product key or when Tableau Server nodes running licensed processes cannot contact the Tableau Server node running the License Manager service. To learn more about licensed processes, see [Tableau Server Processes](#).

When the server is unlicensed you may not be able to start or administer the server. You can, however, manage your licenses using the `tsm licenses` command.

Unlicensed server administrator

All Tableau Server administrators require a user license. Tableau Server administrators will always consume the highest role available. If a Creator product key is activated, the Tableau Server Administrator(s) will take this role. If the highest role available on Tableau Server is an Explorer, the Server Administrator will take the Explorer role. If Creator licenses are added to the server, any existing Server Administrator accounts using Explorer licenses will automatically convert to use Creator licenses.

TSM administrator accounts do not require licenses.

If the license that the server administrator is using expires, then the account will become unlicensed and will be unable to sign in.

Verify the expiration date of your license(s) for the administrators on the server:

Tableau Server on Linux Administrator Guide

- Run `tsm licenses list`.
- Compare the date with the date displayed in the [Tableau Customer Portal](#).
- If the portal does not display the date that you expect, contact [Customer Success](#).
- To renew your license, visit the [Tableau renewal](#) web page.
- Run the `tsm licenses activate` command to activate a new license for the administrator account(s).

If the TSM date matches the portal date and the following refresh operation fails, contact [Tableau Support](#).

If the license for your administrator account has expired or will expire soon, you will need to activate a new license for the account. Alternatively, you can unlicense a non-administrator user to free a license for the server administrator account.

If a Tableau Server administrator is using a Creator, Explorer or Viewer license and their license expires, they will use another license of the same type, if available. If no license seats are available the user will become “unlicensed”.

Important: Do not restart Tableau Server until you have activated a new license or transferred a site role for the server administrator account.

Troubleshoot role-based licensing

This section provides information about resolving issues that can occur when adding the role-based Viewer, Explorer and Creator licenses to Tableau Server or Tableau Online, or when these licenses expire. The highest available license type is Creator, followed by Explorer, and finally Viewer. To learn more about role-based licensing, see [Licensing Overview](#).

A user or administrator is unlicensed due to license expiration

To avoid having users unexpectedly become unlicensed or move to another site role, you should always do one of the following before the license that they are currently using expires:

- Renew and activate a replacement license. If a user occupies a Creator, Explorer or viewer license and their license expires, they will use another license of the same type, if available.
- Change the site role of those users to allow the use of a license that is not due to expire.

To learn how site roles can be changed to require a different license, see [Set Users' Site Roles](#).

The reassignment of users to new licenses is governed by the following logic:

- When a Server Administrator user occupies a Creator license and their license expires (with no replacement licenses available), they are reassigned to an Explorer license if any Explorer licenses are available. This license reassignment occurs in order of most recent login. Server Administrators displace other users who might be currently using an Explorer license. If no Creator or Explorer licenses are available a Server Administrator becomes unlicensed.
- When a non-Server Administrator user occupies a Creator license and their license expires (with no replacement licenses available), they become unlicensed. To avoid having these users become unlicensed, change their site role prior to license expiration. This is especially important for users in the Site Administrator Creator site role, who must move to the Site Administrator Explorer site role before their Creator license expires to avoid losing Site Administrator capabilities.
- When a non-Server Administrator user occupies an Explorer or Viewer license and their license expires (with no replacement licenses available), they are upgraded to a higher license type, if licenses of that type are available. Specifically, the following occurs when a license expires:
 - Users who occupy an Explorer license will move to a Creator license, if available (with no change to site role).
 - Users who occupy a Viewer license will move to an Explorer license, if available. If no Explorer licenses are available, these users will move to a Creator license, if available (with no change to site role).
 - If no licenses are available at the higher license types, those users are moved to Unlicensed.

Users are reassigned to a new license as described above in order of most recent login, with lower license types reassigned first (first Viewer, then Explorer, and then Creator).

For example: Two users with a Viewer license, a user with the Creator license, and two Server Administrators with a Creator license all have their licenses expire. Four unexpired Explorer licenses are available for these users. In this situation, the following occurs in the order shown below:

1. The user with a Viewer license who logged in most recently is reassigned to an Explorer license.
2. The second user with a Viewer license is reassigned to an Explorer license.
3. The Server Administrator user with a Creator license who logged in most recently is reassigned to an Explorer license, and then the second Server Administrator with a Creator license is reassigned to the remaining Explorer license.
4. The user with the Creator license becomes unlicensed.

Server Administrator site role is unchanged when using a Creator license

Server Administrators gain Creator capabilities if Creator licenses are available in Tableau Server, with no change to their site role name. All other Tableau Server and Tableau Online users gain Creator licenses only if assigned to a site role that includes Creator in its name.

Licenses are not immediately available

When you add a role-based license to Tableau Server, those licenses become available to all users when you restart Tableau Server.

A user with a Viewer license cannot open Tableau Server or Tableau Online workbooks from Tableau Desktop

A user with a Viewer license who also has a separate Tableau Desktop license will be unable to open workbooks on Tableau Server or Tableau Online using Tableau Desktop. To open workbooks such using Tableau Desktop, that user will need an Explorer or Creator license on Tableau Server or Tableau Online.

Handle an Unlicensed Server Process

There are several status indicators on the Tableau Server Status page that help you understand the state of Tableau Server processes. An orange-color status box, "Unlicensed", indic-

ates that one of the server processes is unable to retrieve the Tableau Server license information.

In the image below, one of the VizQL processes is unlicensed:

Process Status
The real-time status of processes running in Tableau Server.

Process	Primary 10.32.139.21	Worker 10.32.139.22
Gateway	✓	✓
Application Server	✓	✓
API Server	✓	✓
VizQL Server	✓ ✓	⚠
Cache Server	✓ ✓	✓ ✓
Search & Browse	✓	✓
Backgrounder	✓	✓
Data Server	✓ ✓	✓ ✓
Data Engine	✓	⊘
File Store	✓	⊘
Repository	✓	⊘

Refresh Status ✓ Active 🔄 Busy ⊘ Passive ⚠ Unlicensed ✖ Down ⊘ Status unavailable

There may be several reasons why a process is unable to access licensing information. For example, there may be network issues preventing a process running on an additional node from communicating with the licensing service on the initial node. Or, the unlicensed process may be getting sent more requests than it can accept at a particular moment and can't handle the licensing request. The impact to users depends on which process is unable to confirm its license, and whether there are other instances of the process on one of the server nodes. In the case of the unlicensed VizQL process above, some users may be able to access views while others cannot.

To resolve the problem, **stop**, then **start** Tableau Server.

Tableau Services Manager (TSM) Command Timeout

When Tableau Server is configured with two instances of the repository and failover to the backup repository occurs, TSM attempts to restart the original repository so that it is available as a backup. If this cannot be done for any reason, subsequent TSM commands can fail due to timeouts while waiting for the original repository to recover.

Commands that can be impacted include:

- tsm maintenance restore
- tsm maintenance reindex-search
- tsm reset
- tsm security regenerate-internal-tokens
- tsm sites export
- tsm sites import

If any of these commands is failing, and you have a repository that is not recovering, remove the repository from the server topology, apply pending changes, and re-add it.

Troubleshooting Tableau Services Manager (TSM) Backup

Backup fails to start because services do not start

When you back up Tableau Server, one of the first steps taken is to confirm that key services are running, and, if they are not, to start them. If these services cannot be started:

- Active Repository
- File Store
- Cluster Controller

any attempt to back up Tableau Server will fail with one of the following errors:

```
An error occurred starting one or more of the following services: Active Repository, File Store, Cluster Controller.
```

```
One or more of the following services did not start in a timely fashion: Active Repository, File Store, Cluster Controller.
```

To successfully back Tableau Server up, make sure these processes can start.

Cookie Restriction Error

When a user signs in to Tableau Server, a session cookie is stored in their local browser. The stored cookie is how Tableau Server maintains that the signed in user has been authenticated and can access the server. Because the cookie is set with the same domain or sub-domain as the browser's address bar, it is considered a first-party cookie. If a user's browser is configured to block first-party cookies, they will be unable to sign in to Tableau Server.

When a user signs in to Tableau Server via an embedded view, or in an environment where trusted authentication has been configured, the same thing happens: a cookie is stored. In this case, however, the browser treats the cookie as a third-party cookie. This is because the cookie is set with a domain that's different from the one shown in the browser's address bar. If a user's web browser is set to block third-party cookies, authentication to Tableau Server will fail. To prevent this from occurring, web browsers must be configured to allow third-party cookies.

Troubleshoot Subscriptions

"The view snapshot in this email could not be properly rendered."

If you receive a subscription with this error message, there could be several reasons:

- **Missing credentials:** Some views are published with embedded credentials. You may receive the above error if the embedded credentials are now out-of-date, or if the view was republished without the embedded credentials.
- **Database temporarily down:** If the view has a live database connection and the database was temporarily down when the subscription was being generated, you might receive the above error.
- **Background process timeout:** By default, the background process that handles

subscriptions has a timeout value of 30 minutes per view for the rendering of a view. If rendering a view goes beyond this time limit, the next view in the workbook results in a failed job due to the timeout. In the majority of cases, this default is plenty of time.

However, if the background process is handling an extraordinarily large and complex dashboard, that may not be enough time. You can check the Background Tasks for Non Extracts admin view to see if that's the case. To increase the timeout threshold, use the `tsm configuration set subscriptions.timeout` command.

Can't see images in email

For images of content to display in a subscription email, users subscribed to views, in addition to **View** permissions, must also have **Download Image/PDF** permissions. For more information, see [Permissions](#).

Can't subscribe

If you can see a view on Tableau Server and it has a subscription icon (✉+) in the upper right corner, you can subscribe to it.

To subscribe to a view, Tableau Server needs to be correctly configured (described in [Manage Subscriptions](#)) and the view you're subscribing to must either have embedded credentials for its data source or not rely on credentials at all. Examples of the latter include a workbook that connects to an extract that isn't being refreshed, or a workbook whose data is in a file that was included with the workbook at publish time. Embedding credentials is a step that happens in Tableau Desktop (see the [Tableau Help](#) for details).

No subscription icon

It's possible to see a view but be unable to subscribe to it. This can happen because for several reasons:

- **No subscriptions have been scheduled:** If no subscriptions have been scheduled, the subscription icon will not appear. To set a schedule for subscriptions, see [Create or Modify a Schedule](#).

- **The view uses a live database connection:** Views with live database connections, where you're prompted for your database credentials when you first click the view, aren't available for subscription. A subscription includes a view (or workbook), data, and a schedule. To deliver the data required for the view, Tableau Server either needs embedded database credentials or data that doesn't require credentials. Where live database connections are concerned, Tableau Server doesn't have the credentials, only the individual users do. This is why you can only subscribe to views that either don't require credentials or have them embedded.
- **Tableau Server is configured for trusted authentication:** You may also be able to see a view but be unable to subscribe to it (no subscription icon) if Tableau Server is configured for trusted authentication. See [Ensure Access to Subscriptions](#) for more information.

Receiving invalid or "broken" subscriptions

If you configured subscriptions on test or development instances of Tableau Server in addition to your in-production instance, disable subscriptions on your non-production instances. Keeping subscriptions enabled on all instances can result in your users receiving subscriptions that appear to be valid, but which don't work, or receiving subscriptions even though they've unsubscribed from the view or workbook.

Missing PDF attachment

You can add a PDF attachment to your subscription if your administrator has it enabled. If the PDF attachment is missing from your subscription, it might be because the size of the PDF exceeds either the email server size limit or the maximum size limit set by server administrators. In Tableau Server, the maximum size limit for PDF attachments to subscriptions can be adjusted through the tsm configuration option `subscriptions.max_attachment_size_megabytes`. For more information, see [Configure Server Event Notification and Set Up a Site for Subscriptions](#).

Suspended Subscriptions

By default, a subscription is suspended after 5 consecutive subscription failures. To change the threshold number of subscription failures that can occur before they are suspended, use

the tsm configuration set option, `backgrounder.subscription_failure_threshold_for_run_prevention`. This sets the threshold for the number of consecutive failed subscriptions necessary before suspending the subscription. This is a server-wide setting.

Only Server administrators can configure the threshold number of subscription failures before a subscription is suspended. For information on setting this threshold, see [Set up a Server for Subscriptions](#).

By default, administrators are not emailed when a subscription is suspended, but can opt-in to suspension emails per site through **My Account Settings**.

Resume suspended subscriptions

Administrators and subscription owners can resume subscriptions in several ways:

- from My Subscription tab in Content Settings
- from the Subscriptions tab per workbook
- from the Subscriptions tab under Tasks (Server Admins only)

When a subscription is resumed, the alert failing count goes back to zero. The next evaluation of the subscription will occur at the next scheduled evaluation time.

Can't set subscription frequency to "When Data Refreshes"

You can set subscriptions to run when an extract refreshes if the workbook uses one published connection to an extract. When creating or modifying a subscription, you may not see a **Frequency** option if the workbook uses:

- More than one extract refresh
- A live data connection

Subscriptions not arriving ("Error sending email. Can't send command to SMTP host.")

You may see the above error in Windows Event Viewer if subscriptions aren't arriving and your SMTP server is using encrypted (TLS) sessions. To send subscriptions to an SMTP server that is configured with TLS, you must configure secure SMTP on Tableau Server. See [Configure SMTP Setup](#). (If you're experiencing this error, note that Tableau Server will still indicate that subscriptions are being sent in the [Background Tasks for Non Extracts](#) admin view.)

Custom scripts not working after upgrade to 8.1

To support better session management, starting with version 8.1, a hash tag (#) was added to the end of view URLs. If you had custom subscriptions scripting that generated views as PDFs or PNGs you may need to update your scripts to allow for the hash tag.

For example, prior to version 8.1, view URLs use this syntax: `http://tableauserver/views/SuperStore/sheet1`. To generate a view as a PNG, `.png` could be added to the end of the URL. For example, `http://tableauserver/views/SuperStore/sheet1.png`.

In versions 8.1, 8.2, or 8.3, view URLs use this syntax: `http://tableauserver/views/SuperStore/sheet1#1`. To generate a PNG, add `.png` before the hash tag. For example: `http://tableauserver/views/SuperStore/sheet1.png#1`

Custom scripts not working after upgrade to 9.0

In version 9.0, the session ID at the end of server URLs is indicated by an "iid" parameter, `:iid=<n>`. For example, `http://localhost/#/views/Sales2015/SalesMarginsByAreaCode?:iid=1`. This parameter replaces the hash tag "`#<n>`" used for the session ID in 8.x versions of Tableau Server.

If you use custom subscriptions scripts that generate views as PDFs or PNGs, you may need to update your scripts by removing the hash tag and number (`# <n>`), and inserting the `?:iid=` session ID parameter before the number.

Starting in version 9.0, view URLs use this syntax: `http://tableauserver/views/SuperStore/sheet1?:iid=2`.

To generate a PNG in version 9.0 and later, add `.png` before the session ID: `http://tableauserver/views/SuperStore/sheet1.png?:iid=2`

Server Administrator Reference

You can learn more about Tableau Server processes, ports, and accounts and permissions.

Tableau Server Processes

This topic describes the options for setting the process configuration. To configure Tableau Server processes, you need to specify which processes and how many instances should run on each node. You do this using the `tsm topology set-process` command. For more information, see [Changing the number of processes on a node](#).

Looking for Tableau Server on Windows? See [Tableau Server Processes](#).

Except where explicitly noted in the table below, applying changes in processes will stop Tableau Server if it is running when you apply those changes. After changes are applied, Tableau Server is returned to the state it was in before process configuration, so if the server was running, it will be restarted.

Important: Your process topology will depend on your organizational needs.

Licensed processes

Some of the processes that are installed as a part of Tableau Server are "licensed" processes. Licensed processes need a valid Tableau Server license in order to run. Other processes that are installed as a part of Tableau Server are not tied to a valid license. This has the following impact:

- Every licensed process needs to regularly contact the Tableau Server License Manager service that runs on the initial Tableau Server computer to verify they are licensed. If they cannot confirm there is a valid license, for example, if the initial node is not available, the process will not run and Tableau Server may not function properly or reliably.
- If you have a core-based Tableau Server license, the cores on any node with a licensed process will count against the total count of licensed cores.

Note: If you have the Data Management Add-on and a core-based license, you will need to understand how the licensed processes will count against the total count of licensed cores that come with each license. For more information, see [Licensing Tableau Prep Conductor for Tableau Server](#).

The "Licensed" column in the table below identifies those processes that require a valid license, and which impact the count of cores in core-based licenses.

Tableau Server Processes These processes have a status of <code>running</code> when Tableau Server is running, and <code>stopped</code> when Tableau Server is stopped.				
Name shown in	Name used with	Purpose	Notes	Licensed
<code>tsm status -v</code>	<code>tsm topology set-process</code>			

Tableau Server on Linux Administrator Guide

Application Server	<code>vizportal</code>	The Application Server (VizPortal) handles the web application, REST API calls, and supports browsing and searching.	When Application Server is installed, Data Engine is also installed, unless the node already has an instance of Data Engine. When the first instance of Application Server is installed on a node, the Interactive Container Service is also installed.	Yes
Ask Data	Cannot be configured manually.	The Ask Data service is used by the Ask Data feature.	Runs automatically on all nodes where Data Server is running.	No
Backgrounder	<code>backgrounder</code>	The Backgrounder runs server tasks, including extract refreshes, subscriptions, 'Run Now' tasks, and tasks initiated from <code>tabcmd</code> .	When Backgrounder is installed, Data Engine is also installed, unless the node already has an instance of Data Engine. When the first instance of Backgrounder is installed on a node, the Non-	Yes

			<p>Interactive Container Service is also installed.</p> <p>Backgrounder is a single-threaded process. You can add more instances of backgrounder to a node to expand the capacity of the node to run jobs in parallel.</p> <p>In most situations you can change the number of backgrounder instances on an existing node of a running server without causing a stop and restart the server. An exception is if you are adding backgrounder to an existing node that did not previously have backgrounder or any other process that also installs Data Engine. For more information, see Tableau Server Dynamic Topology</p>	
--	--	--	---	--

Tableau Server on Linux Administrator Guide

			Changes.	
Cache Server	<code>cacheserver</code>	<p>The Cache Server is a query cache distributed and shared across the server cluster. This in-memory cache speeds user experience across many scenarios. VizQL server, background, and data server (and application server to a lesser extent) make cache requests to the cache server on behalf of users or jobs. The cache is single-threaded, so</p>		No

		if you need better performance you should run additional instances of cache server.		
Cluster Controller	<code>cluster-controller</code>	The Cluster Controller is responsible for monitoring various components, detecting failures, and running fail-over when needed.	Required on each node. Not automatically installed.	No
Collections Added in 2021.2	<code>collections</code>	The Collections service provides meta data for the collections and favorites features.	The Collections service is installed on the first node where the Application (vizportal) is installed. For high availability, we recommend you install an instance of the Collections service on every node	No

			that has an instance of the Application Server installed. For more information, see Tableau Server Collections Service.	
Content Exploration Added in 2021.1	con- ten- texploration	The Content Exploration service extends search and browse capabilities for Tableau Server. It also depends on Elastic Server to do its operations.	The Content Exploration service is installed on the initial node. For high availability, we recommend you install an instance of the Content Exploration service on every node that has Application Server installed. For more information, see Tableau Server Content Exploration Service.	No
Data Engine	Cannot be configured manually.	The Data Engine creates data extracts and processes queries.	Automatically installed when you install File Store, VizQL Server, Application Server (VizPortal), Data Server, Prep Flow Authoring, or Background.	Yes

			<p>Note: When File Store is configured externally, Data Engine is no longer installed with File Store. For more information see Tableau Server External File Store.</p>	
Data Server	<code>dataserver</code>	The Data Server manages connections to Tableau Server data sources.	When Data Server is installed, Data Engine is also installed, unless the node already has an instance of Data Engine.	Yes
Data Source Properties Added in version 2020.1	<code>tdsservice</code>	The Data Source Properties service provides published data source metadata to client services like Ask Data.	Data Source Properties is added by default on the initial node. A Tableau Server installation must include at least one instance. For performance reasons, we recommend installing the Data Source Properties service on any node that has Application Server (VizPortal) installed.	No
Elastic Server	<code>elasticserver</code>	Elastic	Elastic Server pro-	No

		<p>Server is used by Ask Data to index data and by Content Exploration service to index searchable content.</p>	<p>cesses can be running on more than one node in a cluster. Optionally, they can be moved to any node. It is recommended to have an odd number of Elastic Server processes running.</p> <p>The Elastic Server heap size can be configured by using the <code>elasticsearchserver.vmopts</code> TSM configuration option. For more information, see <code>tsm configuration set Options</code>.</p>	
File Store	<code>filestore</code>	<p>File Store can be configured to run locally on Tableau Server or externally using SAN or NAS storage.</p>	<p>When File Store is installed, Data Engine is also installed, unless the node already has an instance of Data Engine or if File Store is configured externally.</p>	<p>Local File Store: No External File Store: Requires Advanced Management license.</p>

		When configured locally: The File Store automatically replicates extracts across Data Engine nodes.		
Gateway	<code>gateway</code>	The Gateway is a web server that handles all requests to Tableau Server from browsers, Tableau Desktop, and other clients.	Required on any node with an instance of VizQL Server or Vizportal.	No
Internal Data Source Properties Added in version 2020.1	<code>tdsnat- iveservice</code> Cannot be configured manually.	The Internal Data Source Properties service is an internal service that only communicates with the Data Source Properties ser-	One instance of Internal Data Source Properties is automatically configured on each node that has an instance of Data Source Properties on it.	No

Tableau Server on Linux Administrator Guide

		vice.		
<p>Messaging Service</p> <p>Added in version 2019.4.</p>	<code>activemqserver</code>	<p>The Messaging Service is used to support communication between microservices in Tableau Server.</p>	<p>Automatically installed on initial node when you install Tableau Server. One instance of the service is required.</p> <p>On multi-node installations of Tableau Server, you can move the Messaging Service to a different node. In version 2020.1 you can add a second instance of the Messaging Service on an additional node to provide some redundancy (in 2019.4 you cannot configure more than one instance in a cluster). For more information, see Tableau Server Messaging Service.</p>	No
<p>Metrics Service</p> <p>Added in version</p>	<code>metrics</code>	<p>The Metrics Service is responsible for reading and writing</p>	<p>Automatically installed on initial node with a single instance when you install Tableau Server. One instance</p>	No

2020.2.		Metric data in Tableau Server.	<p>of the service is required.</p> <p>You can add additional instances as necessary .</p> <p>We recommend at least one instance of the Metrics service on each node in a multi-node installation of Tableau Server. For more information, see Tableau Server Metrics Service.</p>	
Repository	pgsql	<p>The PostgreSQL repository is the main database for Tableau Server. It stores workbook and user metadata. When Tableau Catalog (or Tableau Metadata API) is</p>	<p>You are limited to a maximum of two instances of the repository in a cluster, and must have at least three nodes in the cluster to add a second repository instance.</p>	No

Tableau Server on Linux Administrator Guide

		enabled, the repository stores Tableau content and external assets metadata.		
SAML Service	Cannot be configured manually.	The SAML Service acts as a proxy between Tableau Server and SAML Identity Providers (IdPs).	Automatically installed on each node where you install Tableau Server. Shows a status of <code>stopped</code> in output of <code>tsm status -v</code> unless site SAML is enabled. You cannot configure the SAML Service manually.	No
Search And Browse	<code>searchserver</code>	The Search Service handles fast search, filter, retrieval, and display of content metadata on the server.		No
Tableau Prep Con-	<code>flowprocessor</code>	The Tableau	By default, it is auto-	Yes

ductor		Prep Conductor runs flows and processes flows for ingestion by Data Catalog. It leverages the scheduling and tracking functionality of Tableau Server so you can automate running flows to update the flow output.	atically enabled on a node where backgrounder is enabled. If the node role is set to exclude flows, then Tableau Prep Conductor is not installed on that node. For more information, see Workload Management through Node Roles. Starting in 2020.4 the Data Management Add-on is not needed to enable this process on Tableau Server.	
Tableau Prep Flow Authoring Added in version 2020.4.	<code>floweditor</code>	Provides the interactive Prep Flow experience in the browser.	When Tableau Prep Flow Authoring (<code>floweditor</code>) is installed, Data Engine and Tableau Prep Flow Service (<code>flowqueryservice</code>) are also installed, unless the node already has an instance of each of those.	Yes
Tableau Prep Min-	<code>flowminerva</code> Note: previously	Used by Tableau	By default, it is automatically enabled on a	Yes

Tableau Server on Linux Administrator Guide

<p>erva Service</p> <p>Added as Tableau Prep Flow Service in version 2020.4 and renamed in version 2021.2.</p>	<p><code>flowqueryservice</code></p>	<p>Prep Flow Authoring (<code>floweditor</code>) for querying datasources.</p>	<p>node where Tableau Prep Flow Authoring (<code>floweditor</code>) is enabled.</p>	
<p>VizQL Server</p>	<p><code>vizqlserver</code></p>	<p>The VizQL Server loads and renders views, computes and executes queries.</p>	<p>When VizQL Server is installed, Data Engine is also installed, unless the node already has an instance of Data Engine.</p> <p>In most situations you can change the number of VizQL instances on an existing node of a running server without causing a stop and restart the server. An exception is if you are adding VizQL to an existing node that did not previously have</p>	<p>Yes</p>

			VizQL or any other process that also installs Data Engine. For more information, see Tableau Server Dynamic Topology Changes.	
<p>Tableau Microservice Container Processes These processes are automatically added when the first instance of Backgrounder or Application Server is added to a node. If all instances of Backgrounder or Application Server are removed from a node, the microservice container process is also removed.</p> <p>Container status depends on the status of the microservices within the container. If all microservices are running, the container process has a status of <code>running</code>. If all microservices are stopped, the container process status is <code>error</code>. If one or more microservices is running while others are not, the container service has a status of <code>degraded</code>. For more information, see Tableau Server Microservice Containers.</p>				
Interactive Microservice Container		Container process for internal Tableau Server microservices that are bundled together for ease of deployment and scalability purposes.	These containers and the microservices they contain cannot be manually configured. The microservices may change over time.	No

Tableau Server on Linux Administrator Guide

Non-Interactive Microservice Container	<code>noninteractive</code>	Container process for internal Tableau Server microservices that are bundled together for ease of deployment and scalability purposes.	These containers and the microservices they contain cannot be manually configured. The microservices may change over time.	No
<p>Tableau Services Manager (TSM) Processes These processes have a status of <code>running</code> once TSM has been initialized, and remain running even when Tableau Server is stopped.</p>				
Activation Service Added in version 2021.1	Cannot be configured manually.	The Activation Service, also known as the authorization-to-run service (ATR), enables you to activate Tableau Server without running out of licenses. It provides	Automatically installed on the initial node when you are using Server ATR to activate Tableau Server.	Yes

		short-term leases of configurable duration until the product key expires.		
Administration Agent	Cannot be configured manually.	The TSM Agent monitors the Coordination Service for changes to configuration or topology and delivers new configurations to each service (configuration) or deploys new services and removes old ones (topology)	Automatically installed on each node where you install Tableau Server. You cannot configure the Administration Agent manually. For more details, see Tableau Server Administration Agent.	No
Administration Controller	Cannot be configured manually, except to move it to another node. For more information, see Recover from an Initial Node Failure.	The TSM Controller handles requests to TSM and orchestrates configuration and topology changes and	Automatically installed when you install TSM on the initial node. You cannot configure the Administration Controller manually except to move it to	No

		<p>workflow across service processes. The Controller also serves as the REST API endpoint (HTTPS).</p>	<p>another node. For more information, see Recover from an Initial Node Failure.</p> <p>For more details, see Tableau Server Administration Controller</p>	
Client File Service	<code>cli-entfileservice</code>	<p>The Client File Service (CFS) manages most shared files in a multi-node cluster. For example, authentication related certificates, keys, and files (OpenID, mutual SSL, SAML, and Kerberos), and customization files are managed by CFS.</p>	<p>Automatically installed on the initial node. No other instances are installed unless you explicitly configure them. See Configure Client File Service.</p> <p>In multi-node deployments, we recommend you configure an instance of CFS on each of the nodes where you deploy the Coordination Service. Redeploying the Coordination Service does not have any impact on CFS.</p> <p>CFS does not display in the Status page or the Configuration</p>	No

			<p>page but is visible in the output of the <code>tsm status -v</code> command.</p> <p>To view or set instances of CFS, use the <code>tsm topology</code> command.</p>	
Coordination Service	Cannot be set with <code>tsm topology set-process</code> .	The Coordination Service serves as the single source of truth.	Automatically installed on the initial node. No other instances are installed unless you explicitly deploy a new Coordination Service ensemble. For details, see Deploy a Coordination Service Ensemble .	No
Service Manager	Cannot be configured manually.	The Service Manager	<p>Automatically installed on all nodes.</p> <p>You cannot configure the Service Manager manually.</p>	No
License Manager	Cannot be configured manually.	The License Manager handles licensing.	<p>Automatically installed on the initial node when you install TSM.</p> <p>A single instance of this is installed on a Tableau Server</p>	No

			<p>cluster. The License Manager process should only be manually configured if the initial node fails. For more information, see Recover from an Initial Node Failure</p>	
<p>Tableau Server Maintenance Processes These processes have a status of <code>stopped</code> unless they are actively running to complete a job.</p>				
Database Maintenance	Cannot be configured manually.	The Database Maintenance service is responsible for performing maintenance operations on the Tableau Server repository.	<p>Automatically installed on each node where you install Tableau Server.</p> <p>Shows a status of <code>stopped</code> in output of <code>tsm status -v</code> unless it is actively performing database maintenance. Maintenance can include updates related to enabling remote access to the repository and changing passwords used to access the repository.</p> <p>You cannot configure the Database Main-</p>	No

			tenance service manually.	
Backup/Restore	Cannot be configured manually.	The Backup and Restore service is responsible for performing backup and restore operations on the data stored in the Tableau Server repository and file store.	<p>Automatically installed on each node where you install Tableau Server.</p> <p>Shows a status of <code>stopped</code> in output of <code>tsm status -v</code> unless it is performing a backup or restore operation.</p> <p>You cannot configure the Backup and Restore service manually.</p>	No
Site Import/Export	Cannot be configured manually.	The Site Import and Export service is responsible for migrating Tableau Server sites between server clusters.	<p>Automatically installed on each node where you install Tableau Server.</p> <p>Shows a status of <code>stopped</code> in output of <code>tsm status -v</code> unless it is performing an import or export.</p> <p>You cannot configure the Site Import and Export service manually.</p>	No

			ally.	
--	--	--	-------	--

Process workflow

The Tableau Server processes and how they interact depend on what action or activity is taking place. For example, the processes that are used, and how they interact, differ when you publish a workbook from those used when you sign in using SAML. For some interactive views on process workflow, see the below workbook. This allows you to select a particular workflow and follow it step by step from start to finish.

Disclaimer: This workbook is published on Tableau Public and is not maintained by Tableau documentation. We cannot guarantee that it is up-to-date with the latest version of Tableau Server.

Flow: Authenticate with AD

Step 1: Choose a Workflow Step 2: Choose a Data Source Step 3: Drag Slider to Observe Workflow

Authenticate with AD None [Slider]

Description of each Stage

1	A request to access Tableau Server is sent through the browser or Tableau D..
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	

+ a b l e a u [Navigation icons]

Tableau Server Administration Agent

The Administration Agent monitors the Coordination Service for changes to configuration or topology and delivers new configurations to each service (configuration) or deploys new services and removes old ones (topology). The Administration Agent also checks each of the services for status and reports this back to the Coordination Service. This process will be automatically configured on each node of the cluster during installation—no explicit configuration is required or possible.

The Administration Agent may also be referred to as the *TSM Administration Agent*.

Process	Administration Agent
Status	Status of the Administration Agent process is not visible on the Status Page.

	Use the TSM CLI to view status. For more information, see View Server Process Status
Logging	Logs generated by the the Administration Agent process are located in <code>/var/opt/tableau/tableau_server/data/tabsvc/logs/tabadminagent</code> . For more information, see Tableau Server Logs and Log File Locations

What happens when an Administration Agent process fails? All other Tableau Server processes running on the same node will display as “unavailable” on the TSM status page. Tableau Server will continue to work as expected, however you will not be able to make configuration/topology changes to the cluster. Failed Administration Agent processes automatically restart as long as the computer itself is otherwise healthy. If the Administration Agent doesn’t start up on the node, you can try to start the service manually by running the following command:

```
sudo su -l tableau
systemctl --user start tabadminagent_0
```

Tableau Server Administration Controller

The Administration Controller process hosts the TSM REST API for configuring and managing your Tableau Server deployment. There can only be a single instance of the Administration Controller in the entire cluster.

This process will be automatically configured on the initial node of the cluster during installation—no explicit configuration is required.

The Administration Controller is also referred to as the *TSM Controller* and the *TSM Administration Controller*.

Process	Administration Controller
Status	Status of the Administration Controller process is visible on the Status Page, displayed as TSM Controller . For more information, see View Server Pro-

	cess Status
Logging	Logs generated by the Administration Controller process are located in <code>/var/opt/tableau/tableau_server/data/tabsvc/logs/tabadmincontroller</code> . For more information, see Tableau Server Logs and Log File Locations

What happens when the Administration Controller process fails?

All other Tableau Server processes running on the same node will display as “unavailable” on the TSM status page. Tableau Server will continue to work as expected, however you will not be able to make configuration/topology changes to the cluster. Failed Administration Controller processes automatically restart as long as the computer itself is otherwise healthy. If the Administration Controller doesn’t start up on the node, you can try to start the service manually by running the following command:

```
sudo su -l tableau
```

```
systemctl --user start tabadmincontroller_0
```

Moving the Administration Controller

If the initial node fails, you need to move the Administration Controller and the Licensing Service to a different node so that Tableau Server can continue to function. For details on how to do this, see [Recover from an Initial Node Failure](#).

Restarting the Administration Controller

Restart the TSM Administration Controller (as *tableau* system account):

```
sudo su -l tableau -c "systemctl --user restart tabadmincontroller_0.service"
```

Note: It may take a few minutes for `tabadmincontroller` to restart. If you attempt to apply pending changes in the next step before the controller has fully restarted,

TSM will not be able to connect to the controller. You can verify that the controller is running by using the `tsm status -v` command. Tableau Server Administration Controller should be listed as "is running".

Tableau Server Application Server

The Application Server (VizPortal) handles the web application and REST API calls. Application Server also supports browsing and searching. To ensure high availability of Application Server, configure instances on each node in the Tableau Server cluster.

Process	Application Server
Status	Status of the Application Server process is visible on the Status Page. For more information, see View Server Process Status
Logging	Logs generated by the Application Server process are located in <code>/var/opt/tableau/tableau_server/data/tabsvc/logs/vizportal</code> . For more information, see Tableau Server Logs and Log File Locations

What happens when an Application Server process fails? Requests being handled by that instance will fail, but subsequent requests will be routed to other running Application Server processes. Assuming the node containing the failed Application Server is still running, the failed process should automatically restart within seconds.

Tableau Server Backgrounder Process

The Backgrounder process runs server jobs, including extract refreshes, subscriptions, flow runs, and data driven alerts. Jobs are initiated both from scheduled tasks and when started manually using 'Run Now', REST API, or `tabcmd` commands.

Process	Backgrounder
Status	Status of the Backgrounder process is visible on the Status Page. For more information, see View Server Process Status

Logging	Logs generated by the <code>Backgrounder</code> process are located in <code>/var/opt/tableau/tableau_server/data/tabsvc/logs/backgrounder</code> . For more information, see Tableau Server Logs and Log File Locations
----------------	--

What happens if a `Backgrounder` process goes down? Jobs on the failed `Backgrounder` process are retried once the `Backgrounder` process recovers from failure. Most background jobs are scheduled to run periodically, and the same background task will be picked up and performed normally at the next scheduled time by a functioning `Backgrounder` process.

Failed `Backgrounder` processes automatically restart as long as the computer itself is otherwise healthy, and the failed jobs will be retried.

To make the `Backgrounder` process highly available, you should configure one or more instances to run on multiple nodes in the cluster.

Managing Backgrounder Resources

Backgrounders as mentioned earlier in this topic, run server tasks, and can be resource intensive. There are several ways in which you can manage the resources that `Backgrounder` needs to run the server tasks:

- Increase the number of instances on a specific node: `Backgrounder` is single-threaded. It can only launch a single job at a time. Adding more `Backgrounder` instances to a node can increase the number of jobs that can be run in parallel on that node, but keep in mind that each job launched can itself use multiple threads. You can add `Backgrounder` instances up to one half the number of cores. When deciding where and how many Backgrounders to run, consider that each `Backgrounder` process launched for a job can use multiple threads, so adding `Backgrounder` instances may limit the effectiveness of each process. Also keep in mind how other server processes will affect each machine's available capacity.
- Isolate `Backgrounder` process: If you are running Tableau Server on a multi-node cluster, you can dedicate one or more nodes for running `Backgrounder`. For more

information, see Recommended Baseline Configurations.

- **Node Roles:** You can also separate the type of jobs or workload that the Backgrounder on a node does. For example, you can have one node dedicated to running extract refreshes only. For more information, see Workload Management through Node Roles.
- **Restricting users from manually running jobs:** Currently users can run extract refreshes, flow run, and subscriptions using the web interface, REST API, and `tabcmd` commands, and can run them at any time of the day. This can take up server resources during times that your server is busy doing other activities. Starting in Tableau Server 2020.1, a new Run Now settings allows the server administrator to choose whether or not to allow users to run jobs manually. By disabling the Run Now option, you have better control over how backgrounders are utilized as well as being able to better predict the load. This does not apply or affect jobs that are generated for scheduled tasks. For more information on configuring this setting, see Server Settings (General and Customization).

Related content

- Improving group synchronization performance

Tableau Server Cache Server

The Cache Server provides a shared external query cache. It's a cache of key/value pairs which hold information from previous queries to speed up future requests. To make Cache Server highly available, configure one or more Cache Server processes on multiple nodes of the cluster.

Process	Cache Server
Status	Status of the Cache Server process is visible on the Status Page. For more information, see View Server Process Status
Logging	Logs generated by the Cache Server process are located in <code>/var/opt/tableau/tableau_</code>

<code>server/data/tabsvc/logs/cacheserver</code> . For more information, see Tableau Server Logs and Log File Locations

What happens when Cache Server process goes down? The consequences are relatively mild. Tableau Server will still work but actions may take longer as they do not have cached results available. As the queries rerun, the restarted Cache Server gets repopulated, eventually speeding things up for end users. In effect, Cache Server does not have an availability impact; however, it does have an impact on various end user performance scenarios. To reduce impact on user performance, run multiple processes of this type across the cluster.

A failed Cache Server process is automatically restarted; as long as the computer itself is otherwise healthy, the Cache Server process will relaunch.

Tableau Server Client File Service

The Client File Service (CFS) stores and distributes files needed by TSM (e.g. certificates, customization files, etc.). Files that are managed by the Client File Service are renamed and obfuscated before they are distributed across the deployment. This process also parametrizes the file attributes that are required by Tableau services. As a result, files are not mapped to a single file location on the file system. Be sure you have an off-box backup of all files managed by CFS.

The following files are managed by CFS:

- SAML certificate file
- SAML key file
- SAML IdP metadata file
- The custom certificate installed by `tsm security custom-cert add`
- `OpenID.static.file`
- `Kerberos.keytab` file
- LDAP Kerberos keytab file
- LDAP Kerberos conf file
- Mutual SSL certificate file
- Mutual SSL revocation file
- Customization header logo file

Tableau Server on Linux Administrator Guide

- Customization sign-in logo file
- Customization compact logo file

The following files are not managed or distributed by CFS:

- External SSL files. The certificate and key files for external SSL are stored and managed by the Coordination Service. You do not need to manually distribute these files.
- SSL files for LDAP external identity store. You must distribute the SSL certificate file manually to each node in the cluster. See [Configure Encrypted Channel to LDAP External Identity Store](#).

The Client File Service functions much like the File Store does for files needed by business services. By default, CFS is only installed on the initial node of your Tableau Server installation. To configure CFS for high availability, we recommend that you configure an instance of CFS on each of the nodes where you deploy the Coordination Service.

In a cluster, if a node that is running your only instance of CFS fails, any files being managed by CFS will be lost, and you will need to repopulate CFS those files by reimporting certs and custom images, and making any related configuration changes.

Process	Client File Service
Status	Status of the Client File Service process is not visible on the Status Page. Use the TSM CLI to view status. For more information, see View Server Process Status
Logging	Logs generated by the Client File Service process are located in <code>/var/opt/tableau/tableau_server/data/tabsvc/logs/clientfileservice</code> . For more information, see Tableau Server Logs and Log File Locations

What happens when a CFS process fails? Nothing, as long as there is still at least one functioning CFS process in the cluster. The controller will redirect file transfer requests to the other working CFS process.

Failed Client File Service processes automatically restart as long as the computer itself is otherwise healthy.

Tableau Server Collections Service

The Tableau Server Collections service was added in Tableau Server version 2021.2. The Collections service powers the Collections feature. It provides information about collections, and connects with the Content Exploration service to get meta data about collections and items within collections. The Collections service also supports the Favorites feature.

Server Configuration

The Collections Service is automatically installed on the first node where Application Server (vizportal) is installed.

Multi-Node Configuration

For high availability and better performance, we recommend you install one instance of the Collections Service on every node that is running the Application Server (vizportal).

Process	Collections Service
Status	Status of the Collections Service is visible on the Status Page and from the command line using the <code>tsm status -v</code> command. For more information, see View Server Process Status
Logging	Logs generated by the the Collections Service process are located in <code>/var/opt/tableau/tableau_server/data/tabsvc/logs/collections</code> . For more information, see Tableau Server Logs and Log File Locations

Impact if the Collections Service is not running properly

If the Collections Service stops, it should automatically restart as long as the computer itself is otherwise healthy. If it does not restart you can try to restart it manually from the command line:

```
sudo su -l tableau
```

Tableau Server on Linux Administrator Guide

```
systemctl --user start collections_0
```

One instance of Collections service

If you have a single instance of the Collections service configured and that instance fails, collections and favorites will not be available.

Multiple instances of Collections service

If you have a multi-node installation of Tableau Server and have configured instances of the Collections service on multiple nodes, when one instance fails, collections and favorites may not be available to some users.

Log Files

The Collections Service creates two sets of log files:

- `control_collections*.log`: These logs will contain information about the service starting and being enabled.
- `collections_*.log`: Any errors or problems are logged here.

For more information, see [Tableau Server Logs and Log File Locations](#).

Tableau Server Content Exploration Service

The Tableau Server Content Exploration Service extends the capabilities of the Search and Browse process and is responsible for indexing all content in Tableau Server. This service is available in Tableau Server 2021.1 and later and required to search and browse external assets such as databases and tables. Databases and tables are only available if you have [Tableau Catalog](#) enabled.

The Content Exploration Service is applicable to both quick and filtered search. This service is not based on SOLR but depends on the Elastic Search to perform its functions.

Server Configuration

The Content Exploration Service is automatically installed on the initial node.

Multi-Node Configuration

For high availability and better performance, we recommend you install at least one instance of the Content Exploration Service on every node that is running the Application Server.

Process	Content Exploration Service
Status	Status of the Content Exploration Service is visible on the Status Page and can be accessed using the TSM CLI to view. For more information, see View Server Process Status
Logging	Logs generated by the Content Exploration Service are located in <code>/var/opt/tableau/tableau_server/data/tabsvc/logs/contentexploration</code> . For more information, see Tableau Server Logs and Log File Locations

What happens when the Content Exploration Service fails?

If the Content Exploration Service stops working, any search requests for the content type it provides would fail to appear on the search results. If multiple instances of the service are installed, subsequent requests are routed to a healthy instance.

A failed Content Exploration service is automatically restarted; as long as the computer itself is otherwise healthy, the service will relaunch.

Performance Tuning

The Content Exploration Service has the ability to scale up as needed but based on the available memory on the node. However, there is a memory allocation set by default that determines the maximum amount of memory that can be used by the service. While it is not typically recommended that you change this setting if you are seeing performance issues due to insufficient memory allocation, you can run the following commands to change the maximum memory allocation setting.

First, retrieve the current maximum memory allocation by running the following command:

Tableau Server on Linux Administrator Guide

```
tsm configuration get -k content-exploration.vmopts
```

Run the following commands to change the maximum memory allocation by updating the -Xmx value only. Do not change any other options:

```
tsm configuration set -k content-exploration.vmopts -Xmx1024m -  
XX:+ExitOnOutOfMemoryError -Dspring.profiles.active=monolith  
  
tsm pending-changes apply
```

Log Files

The Content Exploration Service creates two sets of log files:

- `control_contentexploration*.log`: These logs will contain information about the service starting and being enabled.
- `contentexploration_*.log`: Any errors or problems are logged here.

For more information, see [Tableau Server Logs and Log File Locations](#).

Search Accuracy

There are couple of ways the Content Exploration Service is used to make sure that searchable content is kept up to date:

- Whenever content managed by Tableau Server or Tableau Online is modified, including permissions, the Content Exploration Service re-indexes the relevant documents to keep its search indexes up to date.
- A system generated Backgrounder task that periodically validates that all content is properly synchronized and re-indexes any discrepancies that are found.

Re-indexing

During restore, the restore process will initiate a full re-indexing of the content and external assets managed by Tableau Server. The re-indexing process consumes CPU resources which may be noticeable during backup and restore.

Tableau Server Coordination Service

The Coordination Service is built on [Apache ZooKeeper](#), an open-source project, and coordinates activities on the server, guaranteeing a quorum in the event of a failure, and serving as the source of "truth" regarding the server topology, configuration, and state. The service is installed automatically on the initial Tableau Server node, but no additional instances are installed as you add additional nodes. Because the successful functioning of Tableau Server depends on a properly functioning Coordination Service, we recommend that for server installations of three or more nodes, you add additional instances of the Coordination Service by deploying a new Coordination Service ensemble. This provides redundancy and improved availability in the event that one instance of the Coordination Service has problems.

The hardware for your cluster can have some effect on how well the Coordination Service runs. In particular:

- **Memory.** The Coordination Service maintains state information in memory. By design, the memory footprint is small, and is typically not a factor in overall server performance.
- **Disk speed.** Because the service stores state information on disk, it benefits from fast disk speed on the individual node computers.
- **Connection speed between nodes.** The service communicates continuously between cluster nodes; a fast connection speeds between nodes helps with efficient synchronization.

Process	Coordination Service
Status	Status of the Coordination Service process is not visible on the Status Page. Use the TSM CLI to view status. For more information, see View Server Process Status
Logging	Logs generated by the the Coordination Service process are located in <code>/var/opt/tableau/tableau_server/data/tabsvc/logs/app-zookeeper</code> . For more information, see Tableau Server Logs and Log File

	Locations
--	-----------

Configuration for the Coordination Service

The Coordination Service is installed automatically on the initial node of Tableau Server. If you are running a single-node installation, you do not need to do anything to deploy or configure the Coordination Service. If your installation includes three or more nodes, you'll be prompted to configure a Coordination Service ensemble when you add your third node. This is not required, but is highly recommended as the Coordination Service serves a key function for high availability, acting as the source of "truth" about server topology, configuration, and state.

To configure a Coordination Service ensemble, use the TSM CLI and add the Coordination Service to the nodes you want running it. For details on how to deploy a Coordination Service ensemble, see [Deploy a Coordination Service Ensemble](#).

The Coordination Service Quorum

To ensure that the Coordination Service can work properly, the service requires a *quorum*—a minimum number of instances of the service. This means that the number of nodes in your installation impacts how many instances of the Coordination Service you want to configure in your ensemble.

Number of Coordination Service instances to use

The maximum number of Coordination Service instances you can have in an ensemble on Tableau Server depends on how many Tableau Server nodes you have in your deployment. Configure a Coordination Service ensemble based on these guidelines:

Total number of server nodes	Recommended number of Coordination Service nodes in ensemble (must be 1, 3, or 5)	Notes
1-2	1 node	This is the default and requires no changes unless

Total number of server nodes	Recommended number of Coordination Service nodes in ensemble (must be 1, 3, or 5)	Notes
nodes		you want to move the Coordination Service off your initial node and onto your additional node.
3-4 nodes	3 nodes	
5 or more nodes	3 nodes or 5 nodes	<p>Five is the maximum number of Coordination Service instances you can install. A 3-node Coordination Service ensemble allows for one of the ensemble nodes to fail without causing Tableau Server to fail. A 5-node ensemble allows for two of the ensemble nodes to fail without causing Tableau Server to fail.</p> <p>For most installations, three Coordination Service nodes are adequate, and because of the I/O-intensive nature of the Coordination Service, this is the most performant configuration.</p> <p>If high availability is your absolute priority, you may want to consider deploying a 5-node Coordination Service ensemble. This provides the most redundancy in the event that one or more nodes fail but will require more system resources. A maximum of two of the ensemble nodes can fail without impacting Tableau Server (as long as any other services on the node also exist on still-functioning nodes).</p> <p>To reduce performance impact, locate the Coordination Service on nodes that are running fewer other services or consider using Coordination Service-only</p>

Total number of server nodes	Recommended number of Coordination Service nodes in ensemble (must be 1, 3, or 5)	Notes
		nodes. For details, see Configure Tableau Server for High Availability with Coordination Service-Only Nodes .

If you reduce the number of nodes

If you reduce the nodes in your cluster from three (or more) to two nodes, a warning tells you Tableau Server can no longer support high availability:

A minimum of three Tableau Server nodes are required for high availability. You can add a third node now, or continue with only two nodes. Continuing with only two nodes means Tableau Server will not be highly available. You can always add a third node later. Click OK to continue with 2 nodes, or Cancel to go back and add a node.

If you continue, Tableau Server will run, but you will not have any automatic failover of the repository.

Viewing Coordination Service Status

The Coordination Service is not included in the listing when you View Server Process Status. To see the state of the service, you can use the `tsm status` command:

```
tsm status -v
```

The output from the command shows you whether the service is running:

```
node1: TABLEAUSVR01
Status: RUNNING
```

```
'Tableau Server Gateway 0' is running.  
'Tableau Server Application Server 0' is running.  
'Tableau Server VizQL Server 0' is running.  
'Tableau Server VizQL Server 1' is running.  
'Tableau Server VizQL Server 2' is running.  
'Tableau Server VizQL Server 3' is running.  
'Tableau Server Cache Server 0' is running.  
'Tableau Server Cache Server 1' is running.  
'Tableau Server Coordination Service 0' is running.  
'Tableau Server Cluster Controller 0' is running.  
'Tableau Server Search And Browse 0' is running.  
'Tableau Server Backgrounder 0' is running.  
'Tableau Server Backgrounder 1' is running.  
'Tableau Server Data Server 0' is running.  
'Tableau Server Data Server 1' is running.  
'Tableau Server Data Engine 0' is running.  
'Tableau Server File Store 0' is running.  
'Tableau Server Repository 0' is running (Active Repository).  
'Tableau Server Administration Agent 0' is running.  
'Tableau Server Administration Controller 0' is running.  
'Tableau Server Service Manager 0' is running.  
'Tableau Server License Manager 0' is running.  
'Tableau Server Client File Service 0' is running.  
'Tableau Server Database Maintenance 0' is stopped.  
'Tableau Server Backup/Restore 0' is stopped.  
'Tableau Server Site Import/Export 0' is stopped.  
'Tableau Server SAML Service 0' is stopped.
```

Tableau Server Data Engine

Hyper is Tableau's in-memory Data Engine technology optimized for fast data ingests and analytical query processing on large or complex data sets. Starting in Tableau 10.5 release, Hyper powers the Data Engine in Tableau Server, Tableau Desktop, Tableau Online, and Tableau Public. The Data Engine is used when creating, refreshing or querying extracts. It

is also used for cross-database joins to support federated data sources with multiple connections.

Process	Data Engine
Status	Status of the Data Engine process is visible on the Status Page. For more information, see View Server Process Status
Logging	Logs generated by the Data Engine process are located in <code>/var/opt/tableau/tableau_server/data/tabsvc/logs/hyper</code> . For more information, see Tableau Server Logs and Log File Locations

Memory and CPU usage

The Data Engine is designed to leverage all available CPU and memory on the machine to provide the fastest response times.

CPU usage

Hyper technology leverages the new instruction sets in CPU and is capable of parallelizing and scaling to all the available cores. Hyper technology is designed to scale to many cores efficiently, and also to maximize the use of each single core as much as possible. This means that you can expect to see the CPU being fully used during query processing. Adding more CPU is expected to result in performance improvement.

Modern operating systems such as Microsoft Windows, Apple macOS, and Linux have mechanisms to make sure that even if a CPU is fully used, incoming and other active processes can run simultaneously. In addition, to manage overall resource consumption and to prevent overloading and completely starving other processes running on the machine, the Data Engine monitors itself to stay within the limits set in the Tableau Server Resource Manager (SRM). Tableau Server Resource Manager monitors the resource consumption and notifies Data Engine to reduce the usage when it exceeds the predefined limit.

Since the Data Engine is designed to utilize the available CPU, it is normal to see spikes in CPU usage at times. If however, you see high CPU usage (ex: 95%) for extended periods of time (an hour or more), this can mean a couple of things:

- There is a high load of queries. This can happen if a server is under stress due to overload of multiple client requests and the queries are queuing up. If this happens often, it is an indication that more hardware is required to serve the clients. Adding more CPU in this case should help to improve performance.
- There is one long running query. In this case, the Tableau Server resource Manager will stop long running queries based on the timeout settings. This was also true for the Tableau Server versions earlier than version 10.5

For more information on Tableau Server Resource Manager, see [General Performance Guidelines](#).

Memory usage

Memory usage of the Data Engine depends on the amount of data required to answer the query. The Data Engine will try to run this in-memory first. A working set memory is allocated to store an intermediate data structure during query processing. In most cases, systems have enough memory to do these types of processing, but if there isn't enough available memory, or if more than 80% of RAM is utilized, the Data Engine shifts to spooling by temporarily writing to disk. The temporary file get deleted after the query has been answered. Therefore, spooling is an indication that more memory may be needed. Memory usage should be monitored and upgraded appropriately to avoid performance issues caused by spooling.

To manage memory resources on the machine, the maximum memory limit for Data Engine is set by Tableau Server Resource Manager (SRM).

Server configuration, Scalability, and Performance

- A single instance of Data Engine is automatically installed per node where an instance of File Store, Application Server (VizPortal), VizQLServer, Data Server, or Backgrounder is installed on Tableau Server. The Data Engine can scale by itself

and uses as much CPU and memory as needed, thus removing the need for multiple instances of the Data Engine. For more information on the server processes, see [Tableau Server Processes](#).

- The instance of Data Engine installed on the node where File Store is installed is used for querying data for view requests. The instance of Data Engine installed on the node where backgrounder is installed is used for extract creation and refreshes. This is an important consideration when you are doing performance tuning. For more information, see [Performance Tuning Examples](#).
- Data Server, VizQL Server, and the Application Server (VizPortal) all use the local instance of Data Engine to do cross-database joins and create shadow extracts. Shadow extract files are only created when you work with workbooks that are based on non-legacy Excel or text, or statistical files. Tableau creates a shadow extract file in order to load the data more quickly.
- In Tableau Server 10.5 one instance of Data Engine is installed automatically when you install backgrounder. The backgrounder process uses the single instance of Data Engine (hyperd.exe) installed on the same node.

Important! There are exceptions to when the Data Engine is installed on the same node as File Store. When File Store is configured external to Tableau Server, Data Engine is no longer installed with File Store. In this configuration where Tableau Server is configured with an External File Store, Data Engine, will continue to be installed with the other process as noted above. In addition, you can also configure Data Engine on a node without other processes - but only when File Store is configured externally. For more information on External File Store, see [Tableau Server External File Store](#).

Scalability:

You can scale up with the new Data Engine: Since cores are fully utilized, adding more cores makes individual queries execute faster which in turn allows for more queries to execute in less time.

Memory usage should be monitored and upgraded appropriately to avoid the performance issues caused by spooling.

For more information on Scalability, see [Tableau Server Scalability](#).

Performance:

Performance benefits

Starting in 10.5, Hyper technology has been integrated with Tableau Data Engine to give you the following key benefits:

- **Faster extract creation:** With Hyper technology, extracts are generated almost as fast as the source system can deliver data, no sorting needed.
- **Support for larger extracts:** Prior to this release, you might have not been able to get all your data into a single extract. With Hyper technology, much larger amounts of data can be included in a single extract.
- **Faster analysis of extracts:** In many cases you will see faster querying of data for larger extracts, or workbooks with complex calculations.

Here are some reasons why the Data Engine powered by Hyper performs better on larger or complex extracts and is optimized for faster querying:

- **Hyper technology is designed to consume data faster.** Unlike in previous versions, the Data Engine does not do any post processing like sorting. With Hyper, post processing steps like sorting are not needed giving the Data Engine the ability to perform better with larger extracts.
- **Hyper technology is memory-optimized.** This means that when needed, all data lives in memory. This results in fast data access times.
- **Hyper technology is CPU optimized.** This means that Data Engine now fully parallelizes the query execution and utilizes available CPU in such a way that query execution time scales almost linearly with the number of cores in the machine.

- **Hyper is a compiling query engine.** Queries are either interpreted or compiled to the machine code for maximum performance and allowing the Data Engine to get most performance out of modern hardware (CPU, large main-memory capacities).
- **Hyper technology uses advanced query optimizations to make queries faster.** Along with many additional advanced techniques such as, materializing min and max values for each column, mini-indices to optimize search ranges, more granular data block-level dictionaries, advanced logic for join and sub-query performance optimizations, the new Data Engine offers many improvements over the previous Tableau Data Engine in terms of performance and scalability.

For more information on performance, start with [General Performance Guidelines](#), and [Performance Tuning](#)

Tableau Server Data Server

The Data Server manages connections to Tableau Server data sources. To make Data Server highly available, configure one or more Data Server processes to run on multiple nodes of the cluster.

Process	Data Server
Status	Status of the Data Server process is visible on the Status Page. For more information, see View Server Process Status
Logging	Logs generated by the Data Server process are located in <code>/var/opt/tableau/tableau_server/data/tabsvc/logs/dataserver</code> . For more information, see Tableau Server Logs and Log File Locations

What happens if a Data Server process fails? Queries running through the Data Server process will fail, resulting in a failed view rendering, extract refresh, or alert. Subsequent requests, including a retry of the failed operation, should succeed as long as a working Data Server is available to accept rerouted requests.

Tableau Server is not dependent on Data Server to function; however, without a running Data Server, workbooks on the server lose the ability to query or to connect to published

data sources. Any view that does not use Data Server for any of its data sources will still function correctly.

Tableau Server Data Source Properties Service

Introduced in version 2020.1.0, the Tableau Server Data Source Properties service provides metadata for published data source from the Application Server (VizPortal) to client services like Ask Data.

One instance of the Data Source Properties service is configured on the initial node of Tableau Server. You can add additional instances on the initial node or on other nodes in a multi-node installation. For performance reasons, we recommend you configure an instance of the Data Source Properties service on any node that has an instance of Application Server.

Process	Data Source Properties Service
Status	Status of the Data Source Properties Service is visible on the Status Page and from the command line using the <code>tsm status -v</code> command. For more information, see View Server Process Status
Logging	Logs generated by the Data Source Properties Service are located in <code>/var/opt/tableau/tableau_server/data/tabsvc/logs/tdsservice</code> . For more information, see Tableau Server Logs and Log File Locations

Log files for the Data Source Properties service

The Data Source Properties service creates two sets of log files:

- `control-tdsservice*.log`. These logs will contain information about the service starting and being enabled.
- `tdsservice_*.log`. Any errors or problems are logged here.

For more information, see [Log File Snapshots \(Archive Logs\)](#).

Data Source Properties service in a multi-node cluster

How you configure the Data Source Properties service in a multi-node cluster depends on how you configure your nodes. In any installation of Tableau Server you must have at least one instance of the service. In addition, for performance reasons, we recommend you configure one instance of Data Source Properties on any node that is configured with the Application Server (VizPortal).

Tableau Server File Store

This topic describes File Store process when configured to run locally on Tableau Server. However, File Store can be run locally as well as external to Tableau Server. For more information on Tableau Server External File Store, see Tableau Server External File Store.

The Tableau Server File Store process controls the storage of extracts. When File Store is installed, an instance of the Data Engine is also installed unless the node already has an instance of the data engine. In highly available (HA) environments, the File Store ensures that extracts are synchronized to other file store nodes so they are available if one file store node stops running.

Process	File Store
Status	Status of the File Store process is visible on the Status Page. For more information, see View Server Process Status
Logging	Logs generated by the File Store process are located in <code>/var/opt/tableau/tableau_server/data/tabsvc/logs/filestore</code> . For more information, see Tableau Server Logs and Log File Locations

The decommission Command

If you want or need to remove a File Store instance you need to decommission File Store first using the `tsm topology filestore decommission` command. If you don't decommission File Store before you attempt to remove it, you will be prompted to do so. Decommissioning puts the File Store instance into read-only mode and copies any unique data contained in the instance to the other File Store(s) in the cluster. While a File Store instance

is being decommissioned, this shows on the Status page, and once all unique content has been copied to other File Store nodes, the decommissioned node shows as ready to be removed.

Decommissioning File Store when co-located with the Administrative Controller

Tableau backup is fastest when an instance of File Store is located on the same node as the TSM Administrative Controller. If you are removing an instance of File Store that is co-located with the Controller, you will be warned about the performance impact for backups.

Tableau Server Gateway Process

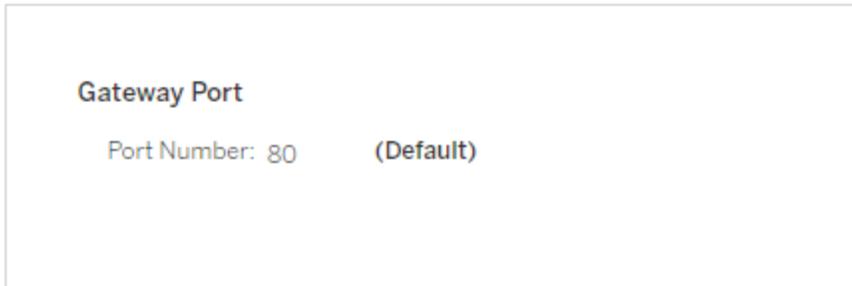
The Tableau Server gateway process is an Apache web server component (`httpd.exe`). Its role is to handle requests to the server from all clients—Tableau Desktop, mobile devices, a proxy, a load balancer, etc.

The server runs a single instance of the gateway process; you can't run more than one per machine. The gateway process is required on any node with an instance of VizQL Server or Vizportal.

Process	Gateway
Status	Status of the Gateway is visible on the Status Page. For more information, see View Server Process Status
Logging	Logs generated by the repository are located in <code>/var/opt/tableau/tableau_server/data/tabsvc/logs/httpd</code> . For more information, see Tableau Server Logs and Log File Locations

Port assignment

By default, the gateway process listens for requests on port 80 (for HTTP requests) and 443 (for SSL requests). When you install Tableau Server on a computer, part of the server configuration makes sure that this port is open in the computer's firewall. If the computer is running a different process that requires port 80 for HTTP, you can change the port assignment for the gateway process during installation. (You cannot change the SSL/HTTPS port.)



If you need to set the HTTP port after installation, then run the following TSM command, where *node* is the node number (for example, `node1`) and *nn* is the new port number:

```
tsm topology set-ports --node-name node --port-name gateway:primary --port-value nn
```

If you are running Tableau Server with a reverse proxy server, you will need to configure other port and host-related settings on Tableau Server. See [Configuring Proxies for Tableau Server](#).

Log files for the gateway process

The gateway process creates two sets of log files:

- **Activity logs.** The name for these log files has the format `access.yy_mm_dd_hh_mm_ss.log`.
- **Error logs.** All errors are logged in a single file named `error.log`.

For more information, see [Log File Snapshots \(Archive Logs\)](#).

Gateway processes in a cluster

If your server environment is distributed across multiple machines, you can run a single gateway process on each node of the cluster. The most common scenario for running a gateway process on multiple computers in the cluster is that you have a load balancer in front of the cluster. In this scenario, the load balancer distributes requests to any gateway in the cluster. If you need to take a node off line (for example, to perform maintenance on that node), you

can disable the load balancer's routing to that machine. When the maintenance is complete, you can re-enable the node on the load balancer.

You must have a gateway process running on at least one computer in the cluster. If you remove the gateway process from the primary server, you must make sure that another computer in the cluster is running the gateway process. You must also make sure that that computer is reachable by clients.

If the Tableau Server is configured to use SSL, you must make sure that the certificate for SSL support is in the same location on each computer in the cluster that has the gateway process running. For more information about using SSL, see [Configure SSL for External HTTP Traffic to and from Tableau Server](#).

Similarly, if the server installation uses a custom logo, the logo must be in the same location on every computer that is running the gateway process.

If you need to change the port number that the gateway process listens on, as explained earlier, you can use the configuration dialog box or run the following command for each worker computer that is running the gateway process:

```
tsm topology set-ports --node-name node --port-name gate-  
way:primary --port-value nn
```

Additional information

[Configuring Proxies for Tableau Server](#)

[Add a Load Balancer](#)

Tableau Server Internal Data Source Properties Service

The Tableau Server Internal Data Source Properties service was introduced in version 2020.1.0 and communicates with the Data Source Properties service. It is managed internally by Tableau Server, and cannot be configured by an administrator.

An instance of the Internal Data Source Properties service is automatically configured on any node that has an instance of the Data Source Properties service.

Process	Internal Data Source Properties Service
Status	Status of the Internal Data Source Properties Service is visible on the Status Page and from the command line using the <code>tsm status -v</code> command. For more information, see View Server Process Status
Logging	Logs generated by the Internal Data Source Properties Service are located in <code>/var/opt/tableau/tableau_server/data/tabsvc/logs/tdsnativeservice</code> . For more information, see Tableau Server Logs and Log File Locations

Log files for the Internal Data Source Properties service

The Internal Data Source Properties service creates two sets of log files:

- `control-tdsnativeservice_*.log`. These logs will contain information about the service starting and being enabled.
- `nativeapi_tdsnativeservice_*.log`. Any errors or problems are logged here.

For more information, see Log File Snapshots (Archive Logs).

Internal Data Source Properties service in a multi-node cluster

One instance of the Internal Data Source Properties services is added to any node that has an instance of Data Source Properties configured. Because the Internal Data Source Properties service is managed internally by Tableau, there is nothing an administrator can configure related to the service.

Tableau Server Messaging Service

The Tableau Server messaging service uses Apache ActiveMQ beginning with version 2019.4. This is a publish/subscribe platform that enables secure, scalable, performant, and highly available message-oriented communication for microservices. The Messaging Service is used to support communication between microservices in Tableau Server.

The server runs a single instance of the Messaging Service.

Process	Messaging Service
Status	Status of the Messaging Service is visible on the Status Page and from the command line using the <code>tsm status -v</code> command. For more information, see View Server Process Status
Logging	Logs generated by the repository are located in <code>/var/opt/tableau/tableau_server/data/tabsvc/logs/activemqserver</code> . For more information, see Tableau Server Logs and Log File Locations

Impact if Messaging Service is not running properly

If the Messaging Service stops or fails to start the consequences are relatively mild. The actual impact depends on whether you have one, or two instances of the Messaging Service configured.

Multiple instances of Messaging Service (multi-node, version 2020.1 and later)

You can have two instances of the Messaging Service configured if you have version 2020.1 or later, *and* you have a multi-node installation of Tableau Server, you can install two instances of the Messaging Service. In this case, if one of the instances fails, the other instance handles all traffic, and you can remove the failed instance and reinstall an instance.

One instance of Messaging Service

If you have a single-node installation of Tableau Server, or are running version 2019.4.x, you are limited to a single instance of the Messaging Service. In this case, if the configured instance fails, Tableau Server will still work but status will show as "Degraded", and users mentioned in comments will not receive email notification of the mention. There is no loss of data.

If the Messaging Service stops, it is automatically restarted; as long as the computer itself is otherwise healthy, the Messaging Service should restart. If you cannot restart the service, you can disable it so that messaging uses the older mechanism.

Disabling a failed Messaging Service instance

If you have a single instance of the Messaging Service that will not automatically restart:

1. Disable use of the service by using the TSM CLI to set these options to `false` and apply the pending changes:

```
tsm configuration set -k features.MessageBusEnabled -v false
```

```
tsm configuration set -k features.ActiveMQ -v false
```

```
tsm pending-changes apply
```

2. Restart Tableau Server (if you did not restart the server when you applied pending changes).

These steps configure Tableau Server to use the pre-2019.4 messaging mechanism.

Messaging Service in a multi-node cluster

How you configure the Messaging Service in a multi-node cluster depends on what version of Tableau Server you are running. In all cases, you must run one instance of the Messaging Service.

- 2020.1 and later—Starting with version 2020.1, you can run two instances of the service in a multi-node environment. We recommend you run two instances as this provides redundancy. You can add a second instance to any node that does not already have the Messaging Service configured. You cannot add more than a combined total of two instances in your multi-node cluster.
- 2019.4.x—In version 2019.4.x, you are limited to a single instance of the service. In a multi-node installation, you can run that instance on any one of the nodes.

Tableau Server Metrics Service

The Metrics service is responsible for reading and writing Metric data in Tableau Server. The service is required in order for Metrics to work properly. To make the Metrics service highly available, configure one or more instances of the service to multiple nodes of the cluster. We

recommend you configure at least one instance on every node that is running the Application Server (VizPortal).

Process	Metrics Service
Status	Status of the Metrics Service is visible on the TSM Status Page and from the command line using the <code>tsm status -v</code> command. For more information, see View Server Process Status
Logging	Logs generated by the Metrics Service process are located in <code>/var/opt/tableau/tableau_server/data/tabsvc/logs/metrics</code> . For more information, see Tableau Server Logs and Log File Locations

Impact if the Metrics Service fails

The Metrics Service is required for Metrics to function properly on Tableau Server. In order to minimize issues, install multiple instances of the service in your server installation. If you have a single-node instance of Tableau Server, configure at least two instances of the service on your node. If the Metrics Service stops working, any Metrics requests that are in-process will fail. Subsequent requests are routed to a healthy instance of the service.

Metrics Service in a multi-node cluster

The Metrics Service is required for Metrics to function properly on Tableau Server. In order to minimize issues, install multiple instances of the service in your server installation. In multi-node installations, we recommend you configure at least one instance of the Metrics Service on every node that is running the Application Server (VizPortal). This provides redundancy and maximum performance.

Log files for the Metrics service

The Metrics Service creates two sets of log files:

- `control-metrics*.log`. These logs will contain information about the service starting and being enabled.
- `metrics_*.log`. Any errors or problems are logged here.

For more information, see [Log File Snapshots \(Archive Logs\)](#).

Tableau Server Microservice Containers

By default, one instance of the Interactive Microservice Container is added to every node that has Application Server (Vizportal) installed, and one instance of the Non-Interactive Microservice Container is added to every node that has Backgrounder installed. Although you cannot add a Microservice container directly, you can use the TSM CLI to change the number of instances for both Microservice Containers, when necessary. If all instances of Backgrounder or Application Server are removed from a node, the container process is also removed.

Microservice Containers and the microservices:

- Interactive Microservice Container:
 - MessageBus Microservice
 - Relationship Query Microservice

- Non-Interactive Microservice Container:
 - Relationship Ingestor Microservice
 - External Content Provider Microservice
 - Flow Provider Microservice
 - Content Provider Microservice

Viewing Microservice Container Status

You can see the status of the Microservice Container processes from the TSM Status page, or from the command line, using the `tsm status -v` command. When you use the TSM Status page to View Server Process Status, the status of each container process is visible, but you cannot see the status of any of the microservices in the containers. When you use the command line, more detail is shown, including the status of each individual microservice.

Microservice Container Status

The status of a container process depends on the status of the microservices within the container. When all microservices within a container process are running as expected, the container status is *Active* (on the TSM Status page) or *running* (when viewed from the TSM command line). If all microservices within a container process are stopped, the status for the container is *Error* (on the TSM Status page) or *stopped* (from the TSM command line). If a microservice is stopped but at least one other microservice is running, the container status is *Degraded* (on the TSM Status page) or *degraded* (from the TSM command line).

When all microservices within a container process have a status of running, the container status is *Active*. If any microservice in a container is in an error state (has a status of *stopped*) the container process status is *degraded*. If all microservices in a container are in an error state, the container status is *error*.

Use the TSM web interface

To view the Microservice Container status from the TSM Status page:

1. Open TSM in a browser:

```
http://<tsm-computer-name>:8850
```

2. Click **Status**:

The page displays the status for the Interactive Microservice Container and Non-Interactive Microservice Container processes, as well as status for other processes running as part of TSM or Tableau Server.

You cannot see the status of any individual microservice within a container process, but if the container process has a status of *Active* (a green check), the microservices it contains are all running as expected. To see the status of individual microservices, use the TSM command line.

Note: The status of the container processes does not display on the older Tableau Server status page. For details about the two status pages and how they differ, see [View Server Process Status](#).

Use the TSM CLI

To view the Microservice Container status from the TSM command line:

1. Open a command prompt as administrator on the initial node (the node where TSM is installed).
2. Run the following command:

```
tsm status -v
```

The output from the command shows you the status of the container services and the microservices in them:

```
node1: TABLEAUSVR01
Status: RUNNING
'Tableau Server Gateway 0' is running.
'Tableau Server Application Server 0' is running.
'Tableau Server Interactive Microservice Container 0' is
running.
  'MessageBus Microservice 0' is running.
  'Relationship Query Microservice 0' is running.
'Tableau Server VizQL Server 0' is running.
'Tableau Server VizQL Server 1' is running.
'Tableau Server VizQL Server 2' is running.
'Tableau Server VizQL Server 3' is running.
'Tableau Server Cache Server 0' is running.
'Tableau Server Cache Server 1' is running.
'Tableau Server Coordination Service 0' is running.
```

Tableau Server on Linux Administrator Guide

```
'Tableau Server Cluster Controller 0' is running.
'Tableau Server Search And Browse 0' is running.
'Tableau Server Backgrounder 0' is running.
'Tableau Server Backgrounder 1' is running.
'Tableau Server Non-Interactive Microservice Container 0'
is running.
  'Relationship Ingestor Microservice 0' is running.
  'External Content Provider Microservice 0' is run-
ning.
  'Flow Provider Microservice 0' is running.
  'Content Provider Microservice 0' is running.
'Tableau Server Data Server 0' is running.
'Tableau Server Data Server 1' is running.
'Tableau Server Data Engine 0' is running.
'Tableau Server File Store 0' is running.
'Tableau Server Repository 0' is running (Active Repos-
itory).
'Tableau Server Tableau Prep Conductor 0' is running.
'Tableau Server Elastic Server 0' is running.
'Tableau Server Ask Data 0' is running.
'Tableau Server Administration Agent 0' is running.
'Tableau Server Administration Controller 0' is running.
'Tableau Server Service Manager 0' is running.
'Tableau Server License Manager 0' is running.
'Tableau Server Client File Service 0' is running.
'Tableau Server Database Maintenance 0' is stopped.
'Tableau Server Backup/Restore 0' is stopped.
'Tableau Server Site Import/Export 0' is stopped.
'Tableau Server SAML Service 0' is stopped.
c:\Program Files\Tableau\Tableau Server-
\packages\scripts.near.18.1216.1859>
```

Tableau Server Repository

Tableau Server Repository is a database that stores server data. This data includes information about Tableau Server users, groups and group assignments, permissions, projects, data sources, and extract metadata and refresh information.

The Repository is also referred to as *PostgreSQL*.

Process	Repository
Status	Status of the Repository is visible on the Status Page. For more information, see View Server Process Status
Logging	Logs generated by the repository are located in <code>/var/opt/tableau/tableau_server/data/tabsvc/logs/pgsql</code> . For more information, see Tableau Server Logs and Log File Locations

Preferred active repository

When you configure Tableau Server you have the option to specify a node as the preferred active repository. When Tableau Server is configured for repository failover, the preferred active repository node is the one used for the active repository. This is an optional step, and if you do not specify a preferred active repository node, Tableau Server will select the active repository node on startup.

To configure the preferred active repository, use the `tsm configuration set` command to configure the `pgsql.preferred_host` option:

```
tsm configuration set -k pgsql.preferred_host -v "<host_name>"
```

Note: The `host_name` is case-sensitive and must match the node name shown in the output of `tsm status -v`.

Configure a preferred active repository node if you want Tableau Server to select a specific node on startup. You might want to do this if you have a particular server you want to use for

your active repository (a computer with more disk space or memory for example), or if you are using custom administrative views. Custom administrative views have embedded connection information that refers to the repository for which you created the views. For more information on connecting to the Tableau Server repository, see [Collect Data with the Tableau Server Repository](#)

The failoverrepository Command

If failover occurs and your passive repository becomes the active repository, it remains the active repository until either Tableau Server restarts or you use the `tsm topology failover-repository` command to switch back. Specify the repository you want to be the active one, or specify that the preferred active repository (if configured) should be made active again. For more information, see `tsm topology failover-repository`.

Tableau Server SAML Service

For Tableau Server installations that have site-specific SAML enabled, there will also be a SAML Service instance running on each node that is configured with Application Server. This will be automatically configured when site-specific SAML has been enabled on the server. SAML Service on Tableau Server will show as stopped unless site SAML is enabled.

Process	SAML Service
Status	Status of the SAML Service process is not visible on the Status Page. Use the TSM CLI to view status. For more information, see View Server Process Status
Logging	Logs generated by the the SAML Service process are located in <code>/var/opt/tableau/tableau_server/data/tabsvc/logs/samlservice</code> . For more information, see Tableau Server Logs and Log File Locations

If this process goes into a failed state, then users will not be able to sign in to Tableau Server if the user request is routed to an Application Server on this node. Like other processes, when a node containing the SAML Service fails, the failed process should automatically restart within seconds.

Tableau Server Search and Browse

The Search & Browse process, based on Apache SOLR and also known as searchserver, handles fast search, filter, retrieval, and display of content metadata on the server. To configure high availability for the Search & Browse process, configure the process on multiple nodes. Starting in Tableau Server version 2021.1, the functionality of the Search and Browse process is enhanced by the Content Exploration Service. For more information, see [Tableau Server Content Exploration Service](#)

Process	Search & Browse
Status	Status of the Search & Browse process is visible on the Status Page. For more information, see View Server Process Status
Logging	Logs generated by the Search & Browse process are located in <code>/var/opt/tableau/tableau_server/data/tabsvc/logs/search-server</code> . For more information, see Tableau Server Logs and Log File Locations

What happens if a Search & Browse process fails? Users can still sign in to Tableau Server, but workbook content will appear to be missing. The content is not actually missing. Rather, the content is not being returned in the search results; it will be displayed again after the Search and Browse process restarts.

If more than one Search & Browse process is configured and running on multiple nodes when the failure occurs, requests made to a failed Search & Browse process will also fail, but subsequent requests will be routed to working Search & Browse processes. Each Search & Browse process indexes across all nodes in the cluster, therefore if all but one Search & Browse process fails, results will still be returned across all nodes.

Tuning the Search & Browse Process

There are several ways you can change how the Search & Browse process works in Tableau Server. In most cases you should not need to change anything, but here are some configuration changes you can make.

Search & Browse Max Heap Memory

You can configure Search & Browse options using the `searchserver.javamemopts` configuration key. The most important setting you can configure is the one controlling how large the maximum heap memory should be and specified by the `-Xmx` option. By default this is set by Tableau, based on the available physical memory for the server node running the Search & Browse service. The value should be set as high as possible, based on available physical memory and memory usage, but no higher than 24 GB.

When changing the amount of max heap memory, you need to take into account any other services on the system. If you're running low on memory on the computer running Tableau Server, you should consider adding more memory, or scaling down the topology on the machine. If you are running Search & Browse on a node with few other processes, or with a large amount of physical memory, you can consider increasing the amount used by the search service. If you are running Search & Browse on a node with more than the default number of processes, you should consider decreasing the maximum heap memory allocation to avoid having the service attempt to use more memory than is available.

Default maximum heap memory allocations

The table below shows the *default* amount of heap memory Tableau allocates to Search & Browse on a server node, based on available memory:

System memory	SOLR heap memory
<= 16 GB	1 GB
<= 32 GB	2 GB

System memory	SOLR heap memory
<= 64 GB	4 GB
<= 128 GB	8 GB
> 128 GB	16 GB

To change the value of max heap memory, set the `-Xmx` value of `searchserver.javamemopts` using the `tsm configuration set` command.

First get the current values:

```
C:\WINDOWS\system32>tsm configuration get -k search-
server.javamemopts
-Xmx512m -Xms512m -XX:+ExitOnOutOfMemoryError -XX:-UsePerfData
```

Next, change the value of the `-Xmx` option. Include but do not change all other options:

```
tsm configuration set -k searchserver.javamemopts -v "-Xmx256m -
Xms512m -XX:+ExitOnOutOfMemoryError -XX:-UsePerfData"
```

For more details about the `searchserver.javamemopts` configuration key, see `searchserver.javamemopts`.

Client session timeouts

You can configure how long Search & Browse clients will wait to establish a connection to the Search & Browse server, and to Coordination Service (Zookeeper). Both timeout values are set to a relatively high value, but if you experience issues browsing to server content, and `vizportal` and `backgrounder` logs show timeouts connecting to SOLR, try increasing these settings. If this has no impact, you may be running into limited resources on the Tableau Server computer.

```
searchserver.connection_timeout_milliseconds
```

```
searchserver.zookeeper_session_timeout_milliseconds
```

To change the values use the `tsm configuration set` command:

```
tsm configuration set -k searchserver.connection_timeout_mil-
liseconds -v
```

For more details about the `searchserver.connection_timeout_milliseconds` configuration key, see `searchserver.connection_timeout_milliseconds`.

Zookeeper connection health check timeout at startup

When Tableau Server is starting, resource usage is high, especially related to CPU usage. If Search & Browse does not make a connection to Coordination Service (zookeeper), it will fail to start. To account for this, Tableau Server performs a health check on the Coordination Service before starting Search & Browse.

If your Tableau Server computer is especially busy, or if Search & Browse fails to start, increase this timeout value.

An error is written to the `control-searchserver.log` files when this health check fails: `Failed zookeeper health check. Refusing to start SOLR.`

To increase the amount of time Tableau Server waits for a successful health check, set the value of `searchserver.startup.zookeeper_healthcheck_timeout_ms` using the `tsm configuration set` command.

```
tsm configuration set -k searchserver.startup.zookeeper_
healthcheck_timeout_ms -v <nnnnnn>
```

For more details about the `searchserver.startup.zookeeper_healthcheck_timeout_ms` configuration key, see `searchserver.startup.zookeeper_healthcheck_timeout_ms`.

Tableau Server TSM Maintenance Services

There are three TSM Maintenance Services that are installed on every node of the cluster: Database Maintenance, Backup/Restore, and Site Import/Export.

Processes	Database Maintenance, Backup/Restore, and Site Import/Export.
Status	Status of the TSM Maintenance services are not visible on the Status Page. Use the TSM CLI to view status. For more information, see View Server Process Status
Logging	Logs generated by the the Service Manager process are located in <code>/var/opt/tableau/tableau_server/data/tabsvc/logs/</code> , in the <code>database</code> , <code>maintenancemaintenance</code> , <code>backuprestore</code> , and <code>siteimportexport</code> directories. For more information, see Tableau Server Logs and Log File Locations

These services remain stopped unless a specific maintenance task that requires them is initiated by the administrator. Additional high-availability configuration is not required for these services. These services are used only for maintenance tasks such as backup and restore and should not impact the functioning of Tableau Server for the end users.

Tableau Server VizQL Server

The VizQL Server loads and renders views, and computes and executes queries. To achieve high availability for the VizQL Server process, configure one or more instances to run on multiple nodes.

Process	VizQL Server
Status	Status of the VizQL Server process is visible on the Status Page. For more information, see View Server Process Status
Logging	Logs generated by the VizQL Server process are located in <code>/var/opt/tableau/tableau_server/data/tabsvc/logs/vizqlserver</code> . For more information, see Tableau Server Logs and Log File Locations

What happens if a VizQL Server process fails? If there is only one VizQL Server process and it fails, then Tableau Server will no longer be able to render any views. High availability

requires configuring redundant VizQL processes. A fairly typical configuration consists of two to four VizQL Server processes on each node.

This simultaneously serves the need for high availability and scalability. If multiple VizQL Server processes are running, then the failure of a single process will result in the failure of any requests and the loss of session data at the time of its failure. Any future requests will be routed to the other working VizQL Server processes on the Tableau Server cluster.

Tableau Prep Conductor

The Tableau Prep Conductor process runs flows and processes flows for ingestion by Data Catalog. It leverages the scheduling and tracking functionality of Tableau Server so you can automate running flows to update the flow output. Starting in 2020.4 the Data Management Add-on is only needed to schedule flows to run on Tableau Server. For more information, see [Tableau Prep Conductor](#).

Process	Tableau Prep Conductor
Status	Status of the Tableau Prep Conductor process is visible on the Status Page. For more information, see View Server Process Status
Logging	Logs generated by the Tableau Prep Conductor process are located in <code>/var/opt/tableau/tableau_server/data/tabsvc/logs/flow-processor</code> . For more information, see Tableau Server Logs and Log File Locations

Tableau Prep Conductor uses the following components to run flows:

- **Backgrounder:** Tableau Prep Conductor uses the Backgrounder process to run flows. Backgrounder is single threaded, so each instance of the Backgrounder process on a node can run one flow at a time. By adding more Backgrounders to a node, you can increase the number of flows that can be run in parallel on that node. The Backgrounder processes can be up to half the number of the physical cores of that node.
- **Connectors:** Prep Conductor uses the supported Tableau Data connectors to

connect to data. For a list of supported Connectors, see [Supported Connectors](#).

- **Data Engine:** Any changes to data or transformation steps in your flow that cannot be pushed to the underlying data source are processed using the Data Engine process. For example, SQL Server does not natively support regular expressions. When connecting to SQL Server, Tableau Prep lets you write regular expression calculations. Tableau Prep Conductor uses Data Engine to temporarily load the data and then perform the regular expression.

Performance and Scale Recommendations

- **Isolate flows to a separate node:** Running Tableau Prep Conductor on a separate node will isolate flow workflows from other Tableau workloads. This is highly recommended since Prep flows are CPU and RAM intensive.
- **Manage flow schedules:** You can control flow execution by creating flow schedules. These schedules let you determine when flows run, how frequently they run, the priority of that schedule, and whether to run items in that schedule serially or in parallel.
- **Add resources:** When scaling your Tableau Prep Conductor environment, we recommend scaling up to an 8 physical cores box per node running as many as 4 backgrounders on each. As you need more resources, we recommend adding more nodes to your server environment.

You can monitor user activity and performance of flows using Administrative views. For more information, see [Monitor Flow Health and Performance](#).

Topology and Configuration

By default, Tableau Prep Conductor is automatically enabled on a node where backgrounder is enabled. If the node role is set to exclude flows, then Tableau Prep Conductor is not installed on that node. For more information, see [Workload Management through Node Roles](#).

It is recommended that you enable Tableau Prep Conductor on a dedicated node to run flows. For more information, see the following topics:

- For new Tableau Server installations, see [Installing Tableau Server with Prep Conductor](#).
- To enable Tableau Prep Conductor on an existing installation of Tableau Server, see [Enable Tableau Prep Conductor to your Tableau Server Installation](#) topic.

Tableau Prep Flow Authoring

The Tableau Prep Flow Authoring process provides the interactive Prep Flow experience in the browser, which lets users create and interact with flows on Tableau Server to clean and prepare data. For more information, see [Create and Interact with Flows on the Web](#). It was added in version 2020.4.

Process	Tableau Prep Flow Authoring
Status	Status of the Tableau Prep Flow Authoring process is visible on the Status Page. For more information, see View Server Process Status
Logging	Logs generated by the Tableau Prep Flow Authoring process are located in <code>/var/opt/tableau/tableau_server/data/tabsvc/logs/floweditor</code> . For more information, see Tableau Server Logs and Log File Locations

Tableau Prep Flow Authoring is enabled by default. It uses these components:

- **Tableau Prep Minerva Service:** Used by Tableau Prep Flow Authoring for querying datasources. By default, it is automatically enabled on a node where Tableau Prep Flow Authoring is enabled.
- **Data Engine:** The Data Engine creates data extracts and processes queries. By default, it is automatically enabled on a node where Tableau Prep Flow Authoring is enabled.

Performance and Scale Recommendations

Tableau Prep Flow Authoring can be CPU and RAM intensive. It is recommended to monitor server and adjust your deployment as necessary by isolating flow authoring to a separate node or adding server resources.

Tableau Server on Linux Administrator Guide

Isolate flow authoring to a separate node

Running Tableau Prep Flow Authoring on a separate node will isolate flow authoring work-flows from other Tableau workloads. If you have a busy server with many extract refreshes, and a lot of viz editing and viewing and you don't want to cause disruption to this workload, then it is recommended to isolate Prep Flow Authoring to its own server node. This means that all flow editing will be directed to a dedicated node. You can also choose to isolate all background workloads such as backgrounders and Tableau Prep Conductor from all your interactive workloads such as Prep Flow Authoring and VizQL Server.

Add resources

As you need more resources, you can add more nodes to your server environment. When planning your nodes, several factors can influence your decision on how much additional hardware you need to allocate.

The main things to consider when planning your nodes are:

- The number of concurrent users or concurrent sessions you expect during peak hours. The number of concurrent sessions per user can be set using the TSM option `maestro.sessionmanagement.maxConcurrentSessionPerUser`. For more information, see [tsm configuration set Options](#).
- The number of data input nodes your flows have on average and the amount of data they have. The maximum row sampling limit can be set using the TSM option `maestro.app_settings.sampling_max_row_limit`. For more information, see [tsm configuration set Options](#).
- The complexity of the flows being authored and the number of nodes. Node types like joins, unions, aggregates, and pivots will, in general, require more resources.

Topology and Configuration

To make Tableau Prep Flow Authoring highly available, configure two instances of Tableau Prep Flow Authoring on nodes when enabling it.

Here is an example of a two node configuration:

Topology

- Security
- User Identity & Access
- Notifications
- Licensing

Topology

Configure and improve Tableau Server performance by adding or removing nodes and changing process configurations and other settings. [Learn more](#)

node1 ip-10-176-60-76	node2 ip-10-176-61-177
Gateway <input checked="" type="checkbox"/>	Gateway <input checked="" type="checkbox"/>
Application Server 1 ▾	Application Server 1 ▾
Interactive Microservic... 1 ▾	Interactive Microservic... 1 ▾
VizQL Server 2 ▾	VizQL Server 2 ▾
Cache Server 2 ▾	Cache Server 2 ▾
Cluster Controller <input checked="" type="checkbox"/>	Cluster Controller <input checked="" type="checkbox"/>
Search & Browse <input checked="" type="checkbox"/>	Search & Browse <input checked="" type="checkbox"/>
Backgrounder 2 ▾	Backgrounder 2 ▾
Non-Interactive Micros... 1 ▾	Non-Interactive Micros... 1 ▾
Data Server 2 ▾	Data Server 2 ▾
Data Engine <input checked="" type="checkbox"/>	Data Engine <input checked="" type="checkbox"/>
File Store <input checked="" type="checkbox"/>	File Store <input checked="" type="checkbox"/>
Repository <input checked="" type="checkbox"/>	Repository <input type="checkbox"/>
Tableau Prep Conductor <input checked="" type="checkbox"/>	Tableau Prep Conductor <input checked="" type="checkbox"/>
Tableau Prep Flow Auth... 0 ▾	Tableau Prep Flow Auth... 2 ▾
Tableau Prep Flow Serv... <input type="checkbox"/>	Tableau Prep Flow Serv... <input checked="" type="checkbox"/>
Ask Data <input checked="" type="checkbox"/>	Ask Data <input checked="" type="checkbox"/>
Elastic Server <input checked="" type="checkbox"/>	Elastic Server <input type="checkbox"/>
Metrics Service 1 ▾	Metrics Service 0 ▾
Messaging Service <input checked="" type="checkbox"/>	Messaging Service <input type="checkbox"/>
Data Source Properties... 1 ▾	Data Source Properties... 0 ▾
Internal Data Source Pr... <input checked="" type="checkbox"/>	Internal Data Source Pr... <input type="checkbox"/>
TSM Controller <input checked="" type="checkbox"/>	TSM Controller <input type="checkbox"/>
License Server <input checked="" type="checkbox"/>	License Server <input type="checkbox"/>
Activation Service <input type="checkbox"/>	Activation Service <input type="checkbox"/>
Content Exploration Se... 1 ▾	Content Exploration Se... 0 ▾
Collections Service 1 ▾	Collections Service 0 ▾
<input type="button" value="Remove Node"/>	<input type="button" value="Remove Node"/>

Tableau Server on Linux Administrator Guide

Here is an example of status page for a two node configuration:



Process	node1 ip-10-176-60-76	node2 ip-10-176-61-177
Gateway	✓	✓
Application Server	✓	✓
Interactive Microservice Container	✓	✓
VizQL Server	✓ ✓	✓ ✓
Cache Server	✓ ✓	✓ ✓
Cluster Controller	✓	✓
Search & Browse	✓	✓
Backgrounder	✓ ✓	✓ ✓
Non-Interactive Microservice Container	✓	✓
Data Server	✓ ✓	✓ ✓
Data Engine	✓	✓
File Store	✓	✓
Repository	✓	
Tableau Prep Conductor	✓	✓
Tableau Prep Flow Authoring		✓ ✓
Tableau Prep Flow Service		✓
Ask Data	✓	✓
Elastic Server	✓	
Metrics Service	✓	
Messaging Service	✓	
Data Source Properties Service	✓	
Internal Data Source Properties Service	✓	
2242 TSM Controller	Version: 2021.2 ✓	Tableau Software
License Server	✓	

Tableau Server on Linux Administrator Guide

Here is an example of a four node configuration:

Topology
 Configure and improve Tableau Server performance by adding or removing nodes and changing process configurations and other settings. [Learn more](#)

Service	node1	node2	node3	node4
Gateway	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Application Server	1	1	1	0
Interactive Microserv...	1	1	1	0
VizQL Server	2	2	2	0
Cache Server	2	2	2	0
Cluster Controller	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Search & Browse	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Backgrounder	2	2	2	0
Non-Interactive Micros...	1	1	1	0
Data Server	2	2	2	0
Data Engine	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
File Store	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Repository	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Tableau Prep Conductor	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Tableau Prep Flow Auth...	0	1	0	2
Tableau Prep Flow Serv...	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Ask Data	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Elastic Server	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Metrics Service	1	0	0	0
Messaging Service	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Data Source Properties...	1	0	0	0
Internal Data Source Pr...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
TSM Controller	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
License Server	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Activation Service	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Content Exploration Se...	1	0	0	0
Collections Service	1	0	0	0

Here is an example of status page for a four node configuration:

+ a b l e a u					STATUS	MAINTENANCE	CONFIGURATION	Tableau Server is running ? sign out	
Process	node1	node2	node3	node4					
Gateway	✓	✓	✓	✓					
Application Server	✓	✓	✓						
Interactive Microservice Container	✓	✓	✓						
VizQL Server	✓✓	✓✓	✓✓						
Cache Server	✓✓	✓✓	✓✓						
Cluster Controller	✓	✓	✓	✓					
Search & Browse	✓	✓	✓						
Backgrounder	✓✓	✓✓	✓✓						
Non-Interactive Microservice Container	✓	✓	✓						
Data Server	✓✓	✓✓	✓✓						
Data Engine	✓	✓	✓	✓					
File Store	✓	✓	✓						
Repository	✓	✓							
Tableau Prep Conductor	✓		✓						
Tableau Prep Flow Authoring		✓		✓✓					
Tableau Prep Flow Service		✓		✓					
Ask Data	✓	✓	✓						
Elastic Server	✓								
Metrics Service	✓								
Messaging Service	✓								
Data Source Properties Service	✓								
Internal Data Source Properties Service	✓								
TSM Controller	✓								
License Server	✓								
Activation Service									
Content Exploration Service	✓								
Collections Service	✓								

✓ Active
⚙ Busy
✓ Passive
⚠ Unlicensed
⚠ Degraded
❌ Error
🌐 External
⏸ Stopped
❓ Status Unavailable

Tableau Server Dynamic Topology Changes

With the introduction of TSM, Tableau Server also introduced the ability to make certain topology changes or updates without restarting the server. These are known as dynamic topology changes, and are possible with the Backgrounder and VizQL Server processes.

You can increase or decrease the number of backgrounder or VizQL Server instances on a node that already has at least one of these processes running, without any Tableau Server downtime. If you have a node already running Backgrounder, you add additional instances

of backgrounder or VizQL Server to the node without restarting Tableau. You can do the same with VizQL Server node running VizQL Server, adding or removing instances of either Backgrounder or VizQL Server. You must be only changing the number of instances of Backgrounder or VizQL Server. If you also add or remove another process, you need to restart Tableau Server. For example, if you have a node running four instances of VizQL Server, you can remove all those instances and add one or more instances of backgrounder instead, without restarting Tableau Server, taking advantage of dynamic topology changes. When adding the first instance of backgrounder or removing the only instance of Backgrounder from a node, the Non-Interactive Microservice Container is added or removed with the Backgrounder instance. When adding the first instance of Application Server (Vizportal) or removing the only instance of Application Server from a node, the Interactive Container Service is added or removed along with Application Server.

With version 2020.2.0 of Tableau Server, certain configuration changes can also be made dynamically. You can make dynamic topology changes at the same time you make dynamic configuration changes, without needing to restart Tableau server. For more information about dynamic configuration changes, see [What's New and Changed in Tableau Server in What's New and Changed for 2020.2 in Tableau Server](#).

Example Scenarios

To better understand why this might be useful, consider these examples:

- **Backgrounder**—At the end of a sales quarter your sales team is using Tableau Server to keep track of their numbers. Dashboards that depend on extracts are showing sales people how they are doing. Any delay in extract refreshes means your team is not seeing the most up-to-date numbers. You can add additional backgrounders to any node that already has at least one backgrounder or VizQL Server, and increase the throughput of extract refreshes, helping to guarantee the numbers are up-to-date as your team finished up their quarter. Later, after the quarterly push, you can reduce the backgrounder instances again to return Tableau Server to its original configuration.
- **VizQL Server**—Similarly, if Tableau Server is unable to keep up with view refreshes, you can quickly add additional VizQL Server instances to any node that already has at least one instance of either VizQL Server or backgrounder configured. In the above backgrounder example, you might want to remove VizQL Servers temporarily, to

accommodate additional backgrounders, and then re-add them back before your users arrive in the morning.

Making dynamic topology changes

You can make dynamic topology changes using the TSM Web UI, or on the command line. To use the Web UI, sign in to TSM using a browser, and on the Configuration tab, update the number of backgrounder or VizQL Server instances for the node you are updating. For details, see [Configure Nodes](#). To make your changes using the TSM CLI, at a command prompt, run the `tsm topology set-process` command. For details, see [Changing the number of processes on a node](#).

Impact of dynamic topology changes

When making dynamic topology changes that remove existing instances of VizQL Server or backgrounder, the instances are removed immediately. Be aware of the following potential impacts to users and currently running jobs:

- **Backgrounder**—Any currently running jobs are terminated. The normal Tableau Server retry logic will restart these jobs, using another backgrounder instance.
- **VizQL Server**—Any currently active sessions are terminated. Users may see an error message. Refreshing the browser should clear the error.

Best practices

Tableau recommends you test any dynamic topology changes you plan on using, before implementing them in your production environment. This will help you fully understand potential impacts to your users and scheduled refreshes and subscriptions, and allow you to most efficiently take advantage of the flexibility offered by dynamic topology.

Automating dynamic topology changes

You can automate dynamic topology changes. For example, if you have most of your extract refreshes scheduled overnight, and know your server has extra capacity because users are not signed in, you can use a script or other automated deployment tool to add

Tableau Server on Linux Administrator Guide

backgrounder instances when they can be most efficiently used, and then remove them before the start of the work day.

To automatically get the status of processes, use the `tsm status -v` command and parse the output in your script. Alternately, you can use the TSM REST API get server status. The API is currently in alpha. For more details, see [Get server status](#).

A sample script to set four instances of backgrounder on node2 might look like this:

```
echo Adding/Removing Processes
tsm topology set-process -pr backgrounder -n node2 -c 4
tsm pending-changes apply
echo Done!
```

Additional information

Tableau Server Processes

Server Process Limits

When you reconfigure processes for Tableau Server, there is a limit to the amount that you can increase the number of process instances. By default, the limit is set to eight. If your machine has enough RAM and CPU cores, and you want to go above this limit, you can change the limit using the `service.max_procs` configuration option. For each process instance, Tableau recommends that the machine running the process have at least 1 GB of RAM and 1 logical CPU core.

To change the maximum number of processes allowed:

1. Type the following command, where `number` is the maximum number of process instances you want to allow:

```
tsm configuration set -k service.max_procs -v <number>
```

For example:

```
tsm configuration set -k service.max_procs -v 10
```

2. Next type:

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

Tableau Services Manager Ports

The processes and services that make up the components of Tableau Services Manager (TSM) and Tableau Server on Linux use various ports to communicate. By default, most these ports are assigned (mapped) dynamically from a predefined range of ports. The port assignments are made for each service or process when it is installed. You can control what ports are used in a variety of ways.

A small subset of processes do not use dynamic port mapping and behave uniquely. For more information, see [Ports that are not dynamically mapped](#), below.

Note: This operation includes steps that you may need to perform using the TSM command line.

Firewall requirements

If you are running a firewall on the computer where you will be installing Tableau Server, then you will need to open the following default ports for Tableau Server traffic. All port numbers, except 443 can be changed.

Port	TCP/UDP	Used by ...	TYPE OF INSTALLATION	
			All	Distributed / High Availability
80	TCP	Gateway	X	
443	TCP	SSL. When Tableau Server is configured for SSL, the application server redirects requests to this port. Do not change this port.	X	
8850	TCP	Tableau Services Manager.	X	
8060	TCP	PostgreSQL database.	X	
8061	TCP	PostgreSQL backup verification port	X	
8000-9000	TCP	Range of ports reserved by default for dynamic mapping of Tableau processes		X
27000-27009	TCP	Range of ports used by Tableau Server for License service. This range must be open on the node running the License service and accessible from other nodes. By default, the initial node runs the License service.	X	

See Configure Local Firewall.

Port assignment

There are two approaches you can use for port assignment or mapping in TSM:

- **Dynamic port assignment.** This is the default and requires the least intervention by the administrator.

- **Manual port assignment.** This option requires an administrator to individually assign each port.

If you change port assignments and you are running a local firewall, see [Local firewall configuration](#).

Dynamic port assignment

You can control which ports are used by changing the range of ports available to the dynamic assignment process. Doing this leaves dynamic port assignment in place but restricts the ports that can be chosen. With dynamic port assignment in place, you can still choose to assign ports for certain processes manually. This approach enables you to assign specific ports to specific processes, while leaving the others to be mapped dynamically. Using dynamic mapping, with or without some individually assigned ports is the easiest approach, and should satisfy the requirements of most customers. Dynamically assigned ports are preserved if you export your Tableau Server configuration.

By default, ports are assigned for each service or process from available ports between 8000 to 9000. This assignment takes place when services are installed for the first time on a node. After Tableau Server is initialized, you can see which ports are being used by which services or processes by running this command:

```
tsm topology list-ports
```

If you have a multi-node cluster, ports on all nodes are listed.

Changing the port range

For organizations that have specific requirements for ports being used, the easiest way to control this is to change the range from which ports are dynamically selected. You need to do this at installation, by specifying a minimum and maximum port for the range in your configuration file.

Note: The minimum allowable size of your port range will depend on your server installation and how many services or processes you are running. As a general best practice you should not restrict the range too tightly because port assignment is done by selecting random ports within the range, and if you do not allow a large enough range, selection may fail to find an available port.

To limit the range from which available ports are chosen to those between 8300 and 8600, your configuration file would include an entry similar to this:

```
"configKeys": {
  "ports.range.min": "8300",
  "ports.range.max": "8600"
}
```

Blocking specific ports within the range

Beginning with version 2021.1.0, you can specify certain ports that are within the dynamically assigned range but should not be used by Tableau. This is useful if you have other software on your Tableau Server computers that rely on ports within the range Tableau is using.

To specify ports within the range that should not be used by TSM and Tableau Server, use the `ports.blocklist` configuration key:

```
tsm configuration set -k ports.blocklist -v <port>[,<port>,<-port>]
```

For example:

```
tsm configuration set -k ports.blocklist -v 8000,8088, 8090
```

Disabling dynamic port assignment

If you need more control of port assignment than you can get through a combination of restricting port range and individually assigning ports, you can disable dynamic port mapping at initial server configuration. Disabling dynamic port mapping requires you to manually

assign every port for every process, so we don't recommend this unless you need to control every single port assignment.

If you disable dynamic port mapping, you must configure the port for each process on each node of your installation.

To disable dynamic mapping, your configuration file would include an entry similar to this:

```
"configKeys": {  
  "service.port_remapping.enabled": false  
}
```

Important: When you disable dynamic port assignments, the License service port range is not included. This range (27000-27009) must be open on the node running the License service and accessible from other nodes. By default, the initial node runs the License service.

Manual port assignment

You can disable automatic port assignment entirely and assign a port for each process individually. If you do this, you must assign a port for every process on every node. You can assign ports either in a configuration file, when the processes are first installed, or after installation, using a TSM command. Only ports assigned at process installation are preserved if you export your Tableau Server configuration.

You can specify individual ports for specific processes, whether or not dynamic mapping is enabled. You might do this if you want a process to use a particular port, or if you've disabled dynamic mapping. There are two ways to specify ports for processes: during installation or after installation.

Configuring ports during installation

We recommend configuring port assignment during the installation process as described here. Changing ports after installation is a much more labor-intensive process.

To configure ports during installation, create a json file that specifies your port configuration. This process is similar to defining a non-default port range, but instead you specify a particular port for a specific service or process. If you are going to assign specific ports, this approach is the most robust way to do so because the port mapping is preserved if you export the server configuration and topology settings using the `tsm settings export` command.

To define ports at installation, add information to your configuration file to specify the node (`workerN`), process (`servicename`) and instance ID (`instanceid`), port type (`porttype`), and the port to be used. The format looks like this:

```
workerN.{servicename}_{instanceid}.{porttype}.port:X
```

Where:

- `workerN` is an optional parameter and identifies the node for which the remapping applies. Node numbers start with zero (0).
We recommend you do not include this parameter unless you need to map different ports for the same service on different nodes. If you leave this parameter off, you can map a service port on the initial node, or map the same service port on multiple nodes.
- `servicename` is the name of the process or service that will use the port.
- `instanceid` is the instance of the process. If you are going to be configuring multiple instances of a process on one node, you would need to increment this value for each instance. Start the `instanceid` at zero (0) and increment it by one (1) for each instance of the process. For services that only install a single instance on any given node, this must be left off.
- `porttype` If setting the primary port, do not include this option.
- `port` is the port the process or service should use.

For example, to set the port for the first instance of the file store process on the initial node to 8500, you would include a configuration file entry similar to this:

```
"configKeys": {  
  "filestore_0.port": "8500"  
}
```

The example above does not include the optional `workerN` parameter, so sets the port on all nodes in the cluster. It also leaves off the `porttype` option because it is setting the primary filestore port.

Important: When specifying port changes with a configuration file, you must include the `--force-keys` parameter with the `tsm settings import` command.

Configuring ports after installation

If you need to change ports after you have installed Tableau Server, use the `tsm topology set-ports` command. This approach allows you to specify a port for a specific process after that process has been installed. You are restricted by these limits:

- You must set ports individually, on each node.
- After you set an individual port, you must run `tsm restart`.
- The port assignments are not preserved if you need to import a Tableau Server configuration using `tsm settings import`.
- Port names use a different syntax for `tsm` commands than the syntax that is required for `configKeys`. The table at the end of this topic provides a syntax reference.

For example, to set second instance of the file store on the initial node to use port 8500:

```
tsm topology set-ports --node-name node1 --port-name filestore
--port-value 8500 --instance 2
```

The following example shows how to use shorthand commands to set the JMX ports:

```
tsm topology set-ports -n node1 --port-name vizqlserver:jmx.rmi
-pv 9403 -i 1
```

```
tsm topology set-ports -n node1 --port-name vizqlserver:jmx -pv
9404 -i 2
```

Note: Port entries are not validated when you enter them. Therefore, if you use a port that is already assigned, or if you mistype the syntax for a command, Tableau will not give an error until you restart. After restarting, you may see a generic error, *The reconfigure async job failed*.

If you add an incorrect portname:type with a valid port, you cannot delete the incorrect entry. To update the port, you must reassign an unused port to that value to free up the port again.

Ports that are not dynamically mapped

The Tableau Server repository uses two ports that are not dynamically mapped. These each have a default port that you can override using the `tsm configuration set` command and a process-specific parameter.

Port names	Port (default)	Description
pgsql.port	8060	Port for the Tableau Repository (PostgreSQL database). To override this port: <code>tsm configuration set -k pgsql.port <port></code>
pgsql.verify_restore.port	8061	Port for verifying the integrity of a repository backup. to override this port: <code>tsm configuration set -k pgsql.verify_restore.port <port></code>

Because these ports do not use the dynamic port mapping system, they do not show up in the output of the `tsm topology list-ports` command. To see the value of these you need to use the `tsm configuration get -k <config.value>` command. For example:

```
tsm configuration get -k pgsql.port
```

Controlling port remapping with initialize-tsm

Port assignments are made when services are installed. This means that in order to manually map ports for the TSM-specific processes, you need to assign the ports when you run the `initialize-tsm` script. The script includes options to specify ports for individual TSM services, as well as options for defining the minimum and maximum of the port range used with dynamic mapping, and you can disable dynamic mapping.

The table below lists the options for ports when running the `initialize-tsm` script.

Table: initialize-tsm script port options

Script option	Parameter	Description
-i	<port>	Sets the Coordination Service client port.
-e	<port>	Sets the Coordination Service peer port.
-m	<port>	Sets the Coordination Service leader port.
-n	<port>	Sets the TSM agent file transfer port.
-o	<port>	Sets the TSM Controller port.
-l	<min-port>	Sets the bottom of the port range used

Script option	Parameter	Description
		for dynamically mapping ports.
-r	<max-port>	Sets the top of the port range used for dynamically mapping ports.
--disable-port-remapping		Disables dynamic port mapping. If you do this you must assign ports for every service or process used by TSM and Tableau Server. For more information, see Manual port assignment above.

Dynamically mapped ports

This table lists the processes or services that use dynamically mapped ports.

Port names: syntax for json file (configKeys)	Port names: syntax for tsm CLI	Description
appzookeeper_0.client.port	appzookeeper:client	Coordination Service client port.
appzookeeper_0.peer.port	appzookeeper:peer	Coordination Service peer port.
appzookeeper_0.leader.port	appzookeeper:leader	Coordination Service leader port.
backgrounder_0.port	backgrounder	Backgrounder primary port.
backgrounder_0.debug.port	backgrounder:debug	Backgrounder debug port.

Port names: syntax for json file (configKeys)	Port names: syntax for tsm CLI	Description
backgrounder_0.jmx.port	backgrounder:jmx	Backgrounder jmx port.
backgrounder_0.jmx.rmi.port	backgrounder:jmx.rmi	Backgrounder jmx rmi port.
backgrounder_0.recommendations.trainer.port	backgrounder:recommendations.trainer	Backgrounder recommendations port.
backuprestore.port	backuprestore	Backup/Restore service port.
cacheserver_0.port	cacheserver	Cache server port.
clustercontroller.status.port	clustercontroller:status	Cluster Controller status port.
clustercontroller.storage.port	clustercontroller:storage	Cluster Controller storage port.
databasemaintenance.port	databasemaintenance	Database Maintenance port.
dataserver_0.port	dataserver	Data server primary port.
dataserver_0.debug.port	dataserver:debug	Data server debug port.
dataserver_0.jmx.port	dataserver:jmx	Data server jmx port.

Tableau Server on Linux Administrator Guide

Port names: syntax for json file (configKeys)	Port names: syntax for tsm CLI	Description
dataserver_0.jmx.rmi.port	dataserver:jmx.rmi	Data server jmx rmi port.
filestore.port	filestore	File store primary port.
filestore.status.port	filestore:status	File Store status port.
gateway.port	gateway	Gateway port. This defaults to 80, and if that is not available, to 8080. If that is not available, it tries 8000. That sequence is followed whether or not dynamic port assignment is enabled or not. If none of those ports are available and dynamic mapping is enabled, it takes an available port within the defined range. The gateway port must be the same on all nodes in a multi-

Port names: syntax for json file (configKeys)	Port names: syntax for tsm CLI	Description
		node cluster, so if port 80 is selected on the initial node this is the port that will be used on all nodes and if it is unavailable on one of the other nodes, gateway port selection will fail.
hyper.port	hyper	Data engine primary port.
hyper.connection.port	hyper:connection	Data engine connection port.
licenseservice.vendor_daemon.port	licenseservice:vendor_daemon	License service vendor daemon port. Used for licensing-related communications between nodes in a multi-node installation.
samlservice.port	samlservice	SAML service port.
searchserver.port	searchserver	Search server primary port.
searchserver.debug.port	searchserver:debug	Search server

Tableau Server on Linux Administrator Guide

Port names: syntax for json file (configKeys)	Port names: syntax for tsm CLI	Description
		debug port.
searchserver.jmx.port	searchserver:jmx	Search server jmx port.
searchserver.jmx.rmi.port	searchserver:jmx:rmi	Search server jmx rmi port.
searchserver.startup.port	searchserver:startup	Search server startup port.
siteimportexport.port	siteimportexport	Site Import/Export port.
tabadmincontroller.port	tabadmincontroller	TSM Controller port.
tabadminagent.columbo.port	tabadminagent:columbo	Administration Agent service discovery port
tabadminagent.filetransfer.port	tabadminagent:filetransfer	TSM Agent file transfer port.
vizportal_0.authentication.port	vizportal:authentication	Application server authentication port.
vizportal_0.authorization.port	vizportal:authorization	Application server authorization port.
vizportal_0.maintenance.port	vizportal:.maintenance	Application server maintenance port.

Port names: syntax for json file (configKeys)	Port names: syntax for tsm CLI	Description
vizportal_0.microservice.extensions.port	vizportal:.microservice:extensions	Application server extensions port.
vizportal_0.monolith_grpc.port	vizportal:monolith_grpc	Application server GRPC port.
vizportal_0.publishing.port	vizportal:publishing	Application server publishing port.
vizportal_0.recommendations.port	vizportal:recommendations	Application server recommendations port.
vizportal_0.port	vizportal	Application server primary port.
vizportal_0.debug.port	vizportal:debug	Application server debug port.
vizportal_0.jmx.port	vizportal:jmx	Application server jmx port.
vizportal_0.jmx.rmi.port	vizportal:jmx.rmi	Application server jmx rmi port.
vizqlserver_0.port	vizqlserver	VizQL server primary port.
vizqlserver_0.debug.port	vizqlserver:debug	VizQL server debug port.

Port names: syntax for json file (configKeys)	Port names: syntax for tsm CLI	Description
vizqlserver_0.jmx.port	vizqlserver:jmx	VizQL server jmx port.
vizqlserver_0.jmx.rmi.port	vizqlserver:jmx.rmi	VizQL server jmx rmi port.

Enable the JMX Ports

To help you work through a problem with Tableau Server, Tableau Support may ask you to enable the server's JMX ports. These ports can be useful for monitoring and troubleshooting, usually with a tool like JConsole.

To enable the JMX ports on Tableau Server:

1. **Stop the server.**
2. Enter the following command:

```
tsm configuration set -k service.jmx_enabled -v true
```

3. Apply pending changes:

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see `tsm pending-changes apply`.

4. Restart the Coordination Service and TSM Administration Controller (as *tableau*

system account):

```
sudo su -l tableau -c "systemctl --user restart app-
zookeeper_0.service"
```

```
sudo su -l tableau -c "systemctl --user restart tabad-
mincontroller_0.service"
```

It may take a few minutes for tabadmincontroller to restart. If you attempt to apply pending changes in the next step before the controller has fully restarted, TSM will not be able to connect to the controller. You can verify that the controller is running by using the `tsm status -v` command. Tableau Server Administration Controller should be listed as "is running".

5. Start the server.

Important Enabling JMX ports can introduce some security risk. To mitigate this risk, it is important to limit access to the JMX ports to the fewest number of clients that's practical for your scenario. You typically limit access using the host's firewall rules, an external security device, or routing rules.

How the JMX Ports Are Determined

By default, the JMX ports assigned dynamically, from within a range of available ports. For details on how port assignment is done, and how to override dynamic mapping, see [Tableau Services Manager Ports](#).

ATRDdiag.exe Command Line Reference

You can use the ATRDiag command line utility to manage licensing issues related to the login-based license management and authorization-to-run (ATR) features.

Synopsis

```
ATRDiag -dumpATR -showAll -showVerbose -product ["Tableau
Desktop"]["Tableau Prep"]["Tableau Server"]

-subProduct {subProduct} -version {version} -setVersion {ver-
sion}

-container -serverDataPath {path}

-log [Trace][Debug][Info][Warn][Error][Fatal][Off] -status -
deleteAllATRs

-deleteInvalidATRs -deleteATR {atrIdToDelete}

-enableATRFeature -disableATRFeature -enableLBLMFeature -dis-
ableLBLMFeature

-requireLBLMFeature -setDuration {seconds}
```

Note: All ATRDiag options are case-insensitive.

Options

-dumpATR

Display a valid authorization-to-run (ATR) on the command prompt (if present). Use -version/-product/-subProduct/-version to control which ATR is dumped if more than one is valid.

-showAll

Enumerate the contents of the registry to see available ATRs.(some of which might not be valid).

`-showVerbose`

Enumerate the contents of the registry and use `-version/-product/-sub-Product/-version` to provide per-product.

`-product ["Tableau Desktop"]["Tableau Prep"]["Tableau Server"]`

Defaults to "Tableau Desktop". Must specify "Tableau Server" to display Server ATR..

`-subProduct {subProduct}`

Defaults to "Professional".

`-container`

Container mode, only for Tableau Server. Must specify `-product "Tableau Server"`.

`-serverDataPath`

The location of Server data under Container mode. Defaults to `"/var/opt/tableau/tableau_server/"`.

`-version {version}`

No default; a valid value for this field is "Tableau 2021.1".

`-setVersion {version}`

Persist a default value for `-version`.

`-log [Trace][Debug][Info][Warn][Error][Fatal][Off]`

Display ATR log information.

Tableau Server on Linux Administrator Guide

`-status`

Provide ATR feature status (enabled or disabled), the license server, and dump the ATR.

`-deleteAllATRs`

Remove all ATRs present on the machine.

`-deleteInvalidATRs`

Remove all invalid ATRs.

`-deleteATR {atrIdToDelete}`

Remove an ATR by ID.

`-enableATRFeature`

Turn on the ATR feature. Must run as an administrator.

`-disableATRFeature`

Turn off the ATR feature. Must run as an administrator.

`-enableLBLMFeature`

Turn on login-based license management (LBLM). Must run as an administrator.

`-disableLBLMFeature`

Turn off login-based license management (LBLM). Must run as an administrator.

`-requireLBLMFeature`

Set the login-based license management (LBLM) feature to required. Must run as an administrator.

```
-setDuration {seconds}
```

Set `ATRRequestedDurationSeconds` to seconds. Must run as an administrator.

Global Options

```
-h, --help
```

Optional.

Show the command help.

Help Output for initialize-tsm Script

The following help content is the output when you run the following command:

```
sudo ./initialize-tsm -h
```

The `initialize-tsm` script is installed to `/opt/tableau/tableau_server-
/packages/scripts.<version_code>/`.

Output

REQUIRED

```
--accepteula          Indicate that you have accepted the End  
User License Agreement (EULA).
```

```
                    You can find the EULA in /opt/tableau/tableau_server-  
/packages/docs.<version_code>
```

OPTIONAL

```
-c config-name        Set the service configuration name.  
                    If not set, the default is "tabsvc".
```

```
-d data-directory     Set a custom location for the data
```

Tableau Server on Linux Administrator Guide

directory
if it's not already set. If not set, the default is
"/var/opt/tableau/tableau_server".

-b bootstrap-file Optional. Location of the bootstrap file
downloaded from the Tableau Services Manager
on existing node. Must be provided to join existing Tableau
Server cluster.

-u username Name of the user with admin privileges on
existing Tableau Services Manager.
Required if -b option specified.

-p password Password for the Tableau Services Manager
admin user.

-f Bypass warning messages.

-g Do NOT add the current user to the "tsmad-
min" administrative
group, used for default access to Tableau Services Manager
or
to the "tableau" group, used for easier access to log in.

-a username The provided username will be used as the
user to be added
to the appropriate groups, instead of the user running the
script. Providing both -a and -g is not allowed.

-q Quiet, suppress output except for errors and
warnings.

-i coordinationservice-client-port Client port for the
coordination service

<code>-e coordinationservice-peer-port</code> coordination service	Peer port for the
<code>-m coordinationservice-leader-port</code> coordination service	Leader port for the
<code>-t licenseservice-vendord daemon-port</code> the licensing service	Vendor daemon port for
<code>-n agent-filetransfer-port</code> the agent service	Filetransfer port for
<code>-o controller-port</code> controller service	Https port for the con-
<code>-l port-range-min</code> automatic selection	Lower port bound for
<code>-r port-range-max</code> automatic selection	Upper port bound for
<code>--disable-port-remapping</code> selection	Disable automatic port
<code>--unprivileged-user=<value></code> account to run Tableau Server. Default: "tableau".	Name of the unprivileged
<code>--tsm-authorized-group=<value></code> allows authorization to access Tableau Services Manager. Default: "tsmadmin".	Name of the group(s) that
<code>--disable-account-creation</code>	Do not create groups or

Tableau Server on Linux Administrator Guide

user accounts for Server and TSM authorization.

However, the values in: unprivileged-us

ized-group will still be used

in TSM configuration.

--debug
debugging purposes. Produces extensive

Print each command as it is run for
output.

--http_proxy=<value>
Tableau Server. Its value should be http://<proxy_address>:<-
proxy_port>/

Http forward proxy for

For example, --http_proxy=http://1.2.3

proxy=http://example.com:3128/

--https_proxy=<value>
Tableau Server. Its value should be http://<proxy_address>:<-
proxy_port>/

Https forward proxy for

For example, --

https_proxy=http://example.com:3128/

Take care to us

https_proxy environmental variable.

Do not specify

https_proxy environmental variable.

--no_proxy=<value>
URLs to bypass the forward proxy. For example,

--no_proxy-

y=localhost,127.0.0.1,localaddress,.localdomain.com

--activation-service
authorization-to-run (ATR) service to activate Tableau Server.

This option is ideal for cloud-based or

onments, but is available to anyone who

ing ATR for product activation is a permanent choice that cannot be undone later.

Related topics

- Controlling port remapping with initialize-tsm
- Install and Initialize TSM
- System user and groups
- Data directory

Help Output for upgrade-tsm Script

The following help content is the output when you run the following command:

```
sudo ./upgrade-tsm -h
```

The upgrade-tsm script is installed to /opt/tableau/tableau_server-
/packages/scripts.<version_code>/.

Output

```
Usage: upgrade-tsm --accepteula [optional arguments]
```

```
Upgrade Tableau Server cluster to version <version number>.
This script should be run from any Tableau Server cluster node
after Tableau Server <version number> package
has been installed on all nodes.
```

REQUIRED

```
--accepteula          Indicate that you have
accepted the End User License Agreement (EULA).
```

```
You can find the EULA in docs directory
```

Tableau Server on Linux Administrator Guide

`-u <value>, --username=<value>` TSM administrator user name. Required if it is run using a non-TSM administrator account on the initial node, or if upgrading prior to 2019.2.

`-p <value>, --password=<value>` TSM administrator password. Required if the `--username` option is specified. If a password is required but not provided, you will be prompted for the password.

OPTIONAL

`--debug` Print each command as it is run for debugging purposes. Produces extensive output.

`--trust-admin-controller-cert` Do not validate the server certificate.

`--no-prompt` Suppress script prompts. You will only be prompted for missing required parameters, for example, if you specify a password. If the script needs to stop or restart Tableau Server so without warning or prompting. Use this if you automate the process, for example, with a script.

`--external-repository-config-file=filename` Required if upgrading from a Tableau Server instance that uses a later major version of PostgreSQL and is configured to use an external repository. The filename is a file describing a new instance of the external repository. The filename should use the same type of external service as the current instance.

repository, but with the

supported version of PostgreSQL.

`--external-repository-cert-file=filename`

Required if upgrading from a Tableau Server

of Tableau Server

that uses a later major version of PostgreSQL.

figured to use an

external repository. The filename is a

certificate file for the new

external repository. The new repository

same type

of external service as the current external

but with the

supported version of PostgreSQL.

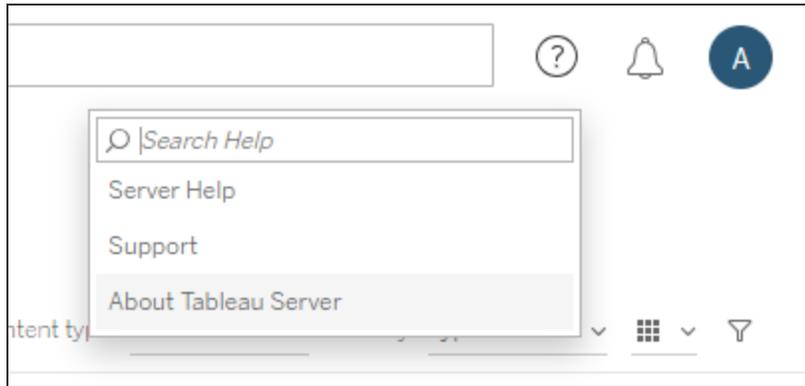
View Server Version

The version of Tableau Server you are running is important. It determines the functionality and capabilities you have access to. Version is also important when you are upgrading, because in some cases how you upgrade depends on which version you are upgrading from, and which version you are upgrading to. Knowing your version is easy, once you understand how to find it.

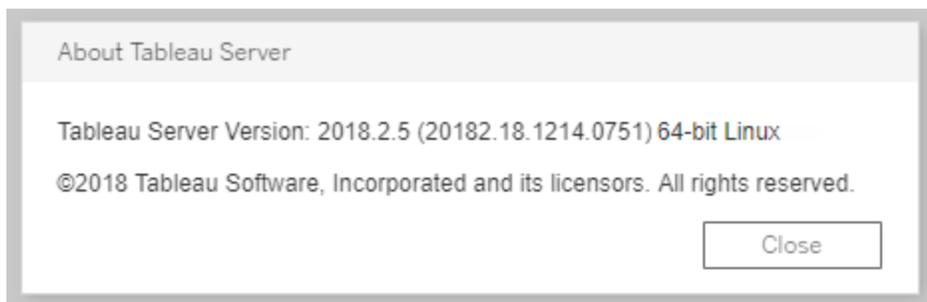
- Viewing version in Tableau Server—All server users can view the version of Tableau Server from the Help menu in the server web UI.
- Viewing version in Tableau Services Manager (TSM)—TSM administrators can view the versions of TSM and Tableau Server from the TSM command line (CLI).

Viewing the server version from the Tableau Server web UI

- While logged into Tableau Server, click the information icon (?) and **About Tableau Server**.



The version of Tableau Server is listed in the About Tableau Server dialog box:



Viewing the server version and TSM version from the TSM command line

1. Open a command prompt as administrator on the initial node (the node where TSM is installed).
2. Run the following command:

```
tsm version
```

The output displays the versions of Tableau Services Manager (TSM) and Tableau Server.

For example:

```
C:\>tsm version
Tableau Services Manager command line version
20182.18.1214.0751.
Tableau Server version 20182.18.1214.0751.
```

Short version, long version, and version_code

In most cases, when you need to know your version number, you need to know the "short" version. This version number displays in the About Tableau Server dialog box and is made up of three parts: major, minor, and maintenance versions. The short version number has this format: `nnnn.n.n`. For example: `2018.2.5`.

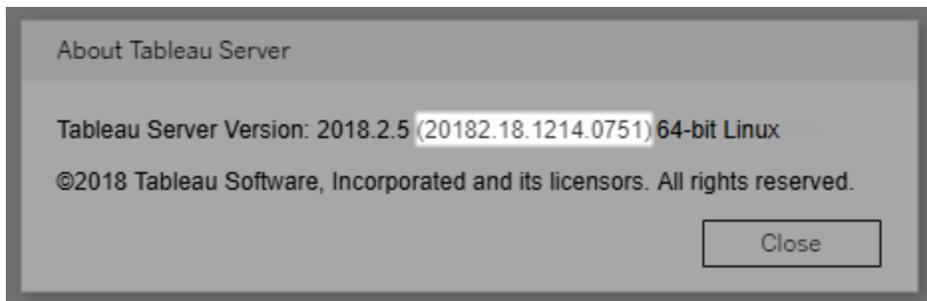
A longer version includes the major and minor version numbers, as well as other build-specific information, and has this format: `nnnnn.nn.nnnn.nnnn`, for example: `20182.18.1214.0751`. When you see a reference to `<version_code>` in this documentation, it means this longer version. The most common reference to this `version_code` or long version in the documentation is when we are discussing install locations or paths to the scripts or bin directories.

Tableau Server is installed in a `/packages` directory, with subdirectories that include the long version in the path name. This means that when you are navigating to a location within the `/packages` directory, you need to include the long version. For example, the `/scripts` directory: `/opt/tableau/tableau_server-
/packages/scripts.<version_code>/`.

For example: `/opt/tableau/tableau_server-
/packages/scripts.20182.18.1214.0751`

Finding the long version number

This longer version also displays in the About Tableau Server dialog box, in parentheses:



Configure Einstein Discovery Integration

Beginning with version 2021.1.0, Tableau Server supports integration with Einstein Discovery, making Einstein Discovery predictions available to authors and viewers of workbooks and dashboards. Starting in version 2021.2.0, Einstein Discovery predictions is also now available when authoring flows on the web.

Einstein Discovery in Tableau is powered by [salesforce.com](https://www.salesforce.com). Consult your agreement with [salesforce.com](https://www.salesforce.com) for applicable terms.

To integrate Einstein Discovery with Tableau Server, there are several necessary configuration steps, including some in Tableau Server, and some in the Salesforce org running Einstein Discovery. This overview outlines these steps for Dashboard extensions, Analytics extensions, and Tableau Prep extensions, and provides links to specific topics with steps for completing the server configuration.

For details on how to use Einstein Discovery predictions in Tableau, including licensing and permission requirements, see [Integrate Einstein Discovery Predictions in Tableau](#) in the Tableau Desktop and Web Authoring Help. For information about adding predictions in flows, see [Add Einstein Discovery Predictions to your flow](#).

Einstein Discovery dashboard extensions

The Einstein Discovery dashboard extension allow workbook authors to surface real-time predictions in Tableau. The dashboard extension delivers predictions interactively, on-

demand, using source data in a Tableau workbook and an Einstein Discovery-powered model deployed in Salesforce.

To configure Tableau Server for the Einstein Discovery dashboard extension you need to do the following:

1. In Tableau Server:
 - a. Enable saved OAuth tokens for data connections and extensions in Tableau Server. [Allow Saved Access Tokens](#)
 - b. Enable Dashboard extensions for the server. See: [Manage Dashboard Extensions in Tableau Server](#)
2. In Salesforce, in the organization running Einstein Discovery:
 - a. Configure CORS in Salesforce.com for Einstein Discover Integration in Tableau Server.
 - b. In Salesforce, in the organization running Tableau CRM, create a connected app. See [Create a Connected Salesforce App](#).
3. In Tableau Server, configure server for saved SF OAuth credentials using information from the connected app. [Configure Tableau Server for Salesforce.com OAuth](#)

Einstein Discovery analytics extensions

The Einstein Discovery analytics extension gives your users the ability to embed predictions directly in Tableau calculated fields. A table calc script requests predictions from a model deployed in Salesforce by passing its associated prediction ID and input data that the model requires. Use Model Manager in Salesforce to auto-generate a Tableau table calculation script, and then paste that script into a calculated field for use in a Tableau workbook.

To configure Tableau Server for either the Einstein Discovery analytics extension you need to do the following:

1. In Tableau Server:
 - a. Enable saved OAuth tokens for data connections and extensions in Tableau Server. [Allow Saved Access Tokens](#)
 - b. Enable analytics extensions for the server and configure a connection type. See: [Configure Connections with Analytics Extensions](#)

2. In Salesforce, in the organization running Einstein Discovery, create a connected app. See [Create a Connected Salesforce App](#).
3. In Tableau Server, configure server for saved SF OAuth credentials using information from the connected app. Configure Tableau Server for Salesforce.com OAuth

Einstein Discovery Tableau Prep extensions

Supported in Tableau Server and Tableau Online starting in version 2021.2.0

The Einstein Discovery Tableau Prep extension enables users to embed Einstein predictions directly in their flows when authoring flows on the web.

To configure Tableau Server or Tableau Online for the Einstein Discovery Tableau Prep extension you need to do the following:

1. In Tableau Server:
 - a. Enable saved OAuth tokens for data connections and extensions in Tableau Server. See [Allow Saved Access Tokens](#)
 - b. Enable Tableau Prep Extensions for the server. See [Enable Tableau Prep Extensions](#).
2. In Salesforce, in the organization running Einstein Discovery, create a connected app. See [Create a Connected Salesforce App](#).
3. In Tableau Server, configure server for saved SF OAuth credentials using information from the connected app. Configure Tableau Server for Salesforce.com OAuth

Configure CORS in Salesforce.com for Einstein Discover Integration in Tableau Server

In version 2021.1.0 the ability to integrate Einstein Discovery predictions into Tableau Dashboards was added. You can do this using the Einstein Discovery dashboard extension. A prerequisite for this is configuring Cross-Origin Resource Sharing (CORS) in the Salesforce org that hosts Tableau CRM and includes the model and predictions that are going to be used.

This procedure explains how an administrator in a Salesforce.com organization would do this configuration. You can find more information about CORS in the Salesforce documentation, [Set Up Cross-Origin Resource Sharing \(CORS\)](#).

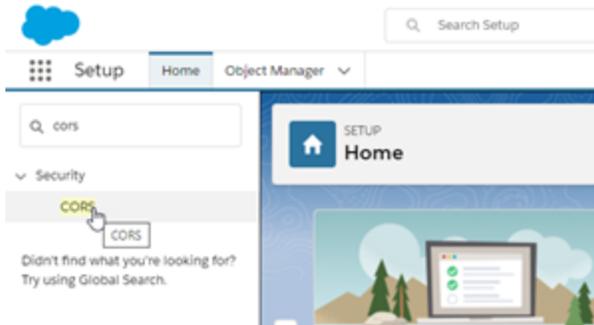
Configure CORS for Einstein Discovery.

Note: This procedure documents the process in Salesforce Lightning. If you are using the traditional interface, the navigation may be different but the configuration is the same.

1. Sign in to your Salesforce.com developer account, click your user name in the upper-right, and then select **Setup**.



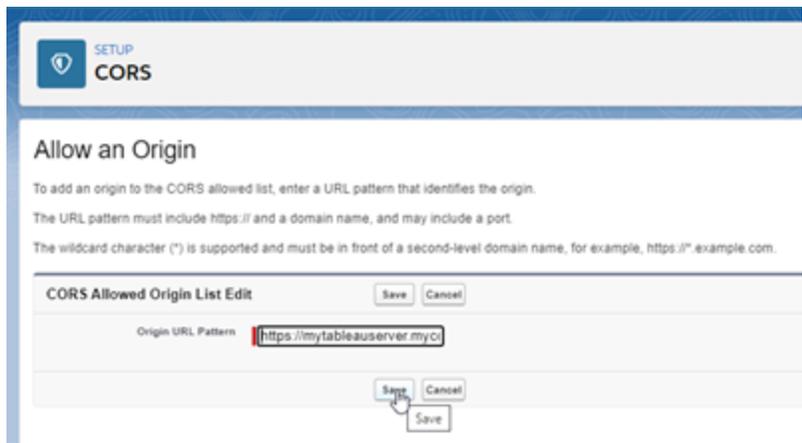
2. In the left navigation column, search for "cors" and select **CORS**.



3. In **CORS**, in the **Allowed Origins List** section, click **New**.



4. In **CORS Allowed Origin List Edit**, enter the URL of your Tableau server, beginning with "https://".



For more information about the URL pattern, see the Salesforce developer documentation: https://developer.salesforce.com/docs/atlas.en-us.chat-terapi.meta/chat-terapi/extend_code_cors.htm

5. Click **Save**.

Configure Connections with Analytics Extensions

Tableau supports a set of functions that your users can use to pass expressions to analytics extensions for integration with R, Python, and Einstein Discovery.

This topic describes how to configure sites on Tableau Server with analytics extensions.

Because Tableau Server provides an authentication mechanism, it can be more secure to expose analytics extensions functionality to users through Tableau Server than in Tableau Desktop.

For more information about user scenarios and configuring Tableau Desktop, see [Pass Expressions Analytics Extensions](#), in the *Tableau Desktop and Web Authoring Help*.

Feature change history:

- 2021.2 — You can configure multiple analytics extension connections for each site. (You are limited to a single Einstein Discovery connection per site.)

For information about how to determine analytics extension usage in workbooks, see [Determining analytics extensions usage](#).

- 2021.1 — Einstein Discover is included as an analytics extension option. Einstein Discovery in Tableau is powered by [salesforce.com](#). Consult your agreement with [salesforce.com](#) for applicable terms.
- 2020.2 — You can configure a different analytics extension connection for each site on your server. Prior to this change a single analytics extension configuration applied globally to all sites on the server.
- 2020.1 — This functionality is now called *analytics extensions*. Previously the feature was called "external services."

Server SSL

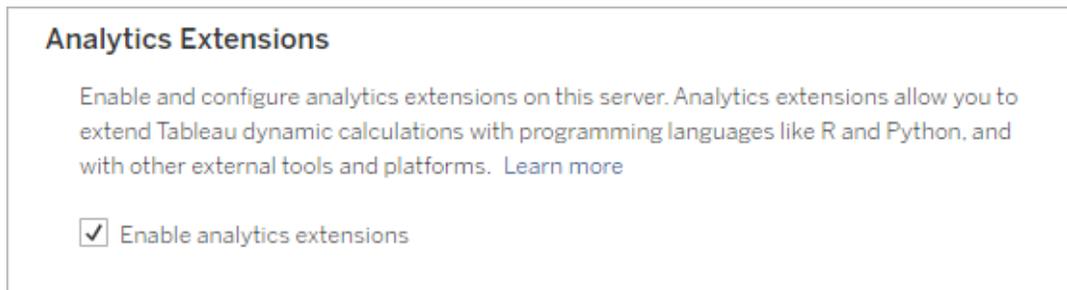
To configure SSL for analytics extensions, you must install a valid certificate on the computer running Tableau Server. The certificate must be trusted by the computer running Tableau Server. The certificate Subject field or one of the SAN entries on must exactly match the URI of the analytics extensions service configuration.

Enable analytics extensions

Before you configure extensions, you must enable analytics extensions server-wide.

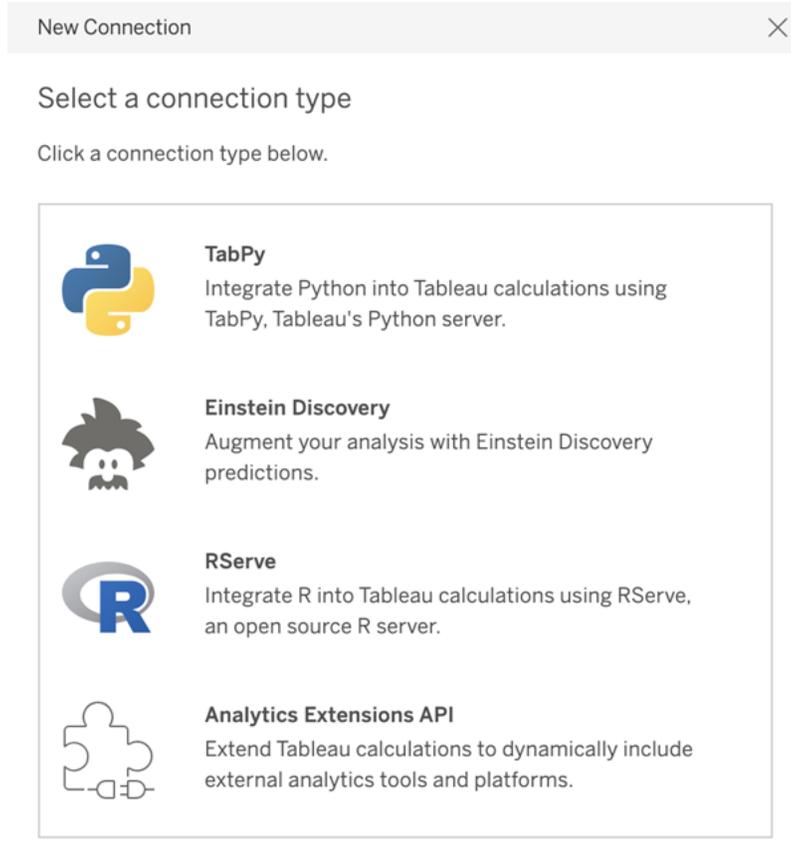
Tableau Server on Linux Administrator Guide

1. Sign in to the Tableau Server Admin Area.
 - If you only have a single site (default) on your server, click **Settings**, and then go to Step 2.
 - If you have multiple sites on your server:
 - a. Under **All Sites**, click **Manage all sites**.
 - b. Click the **Extensions** tab.
2. Scroll to **Analytics Extensions**, select **Enable analytics extensions**, and then click **Save**.



Configure analytics extensions settings

1. Sign in to the Tableau Server Admin Area.
2. On the Settings page, click the **Extensions** tab and then scroll to **Analytics Extensions**. (On multi-site deployments of Tableau Server, navigate to the site where you want to configure analytics extensions, and then click **Settings>Extensions**.)
3. **Multi-site deployments only**: you must enable Analytics Extensions on each site. Under Analytics Extensions, select **Enable analytics extensions for site**.
4. Under Analytics Extensions, click **Create new connection**.
5. In the **New Connection** dialog, click the connection type you want to add, then enter the configuration settings for your analytics service:



6. The options you need to configure depend on the connection type you choose:
- For an Einstein Discovery connection, click **Enable**.
 - For TabPy, RServer and Analytics Extensions API connections, enter the following information:
 - **Connection Name:** (Required) Specify the server type you are connecting to. RSERVE supports connections to R using the RServe package. TABPY supports connections to Python using TabPy, or to other analytics extensions.
 - **Require SSL** (Recommended): Select this option to encrypt the connection to the analytics service. If you specify a `HTTPS` URL in the **Hostname** field, then you must select this option.
 - **Hostname:** (Required) Specify the computer name or URL where the analytics service is running. This field is case sensitive.

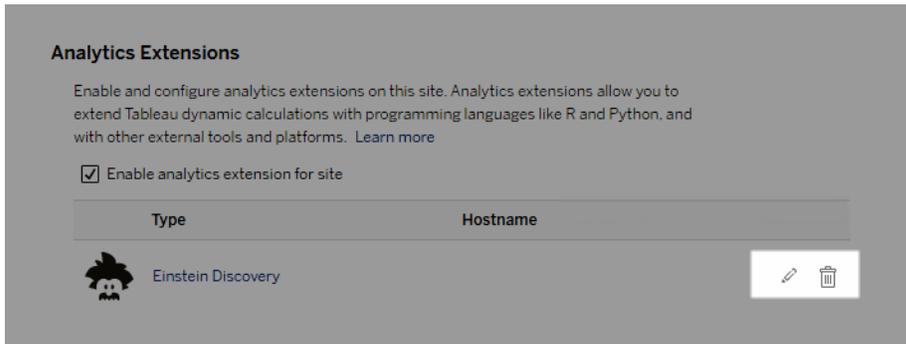
Tableau Server on Linux Administrator Guide

- **Port:** (Required) Specify the port for the service.
- **Sign in with a username and password**(Recommended): Select this option to specify user name and password that is used to authenticate to the analytics service.

7. Click **Create**.

Edit or delete an analytics extension connection

To edit or delete a configuration, navigate to **Analytics Extensions** on the **Extensions** tab of your site.



Click the **Edit** or **Delete** icon and follow the prompts to change the configuration.

Client requirement: Intermediate certificate chain for Rserve external service

As of Tableau Server version 2020.1, you must install a full certificate chain on Tableau Desktop computers (Windows and Mac) that are connecting to a Rserve external connection through Tableau Server. This requirement is due to how Rserve manages the handshake on secure connections.

Importing a root certificate on the Tableau Desktop is not sufficient, the entire certificate chain must be imported onto the client computer.

Script errors

Tableau cannot verify that workbooks that use an analytics extension will render properly on Tableau Server. There might be scenarios where a required statistical library is available on a user's computer but not on the analytics extension instance that Tableau Server is using.

A warning will be displayed when you publish a workbook if it contains views that use an analytics extension.

This worksheet contains external service scripts, which cannot be viewed on the target platform until the administrator configures an external service connection.

Determining analytics extensions usage

Beginning with version 2021.2, analytics extensions configurations are mapped at the workbook level. This allows administrators to use custom views to query the Tableau Repository and determine which workbooks are using which extensions, and how often they are used.

To do this you need to join the workbook connections table to tables showing workbook usage. For details about creating and using custom administrative views, see [Collect Data with the Tableau Server Repository and Create Custom Administrative Views](#).

Changing the Identity Store

Infrastructure or business changes may require you to change the identity store on Tableau Server. There are two kinds of identity stores: local and external. When you installed Tableau Server you configured either a local identity store or an external identity store.

When you configure Tableau Server with a local identity store, all user and group information is stored and managed in the Tableau Server repository. In the local identity store scenario, there is no external source for users and groups.

When you configure Tableau Server with an external store, all user and group information is stored and managed by an external directory service. Tableau Server must synchronize

with the external identity store so that local copies of the users and groups exist in the Tableau Server repository, but the external identity store is the authoritative source for all user and group data. Examples of external identity stores are OpenLDAP and Active Directory.

For more information about the Tableau identity store, see [Identity Store](#).

You can change from local store to an external store, or you can change from an external store to a local store. In either case, to change the identity store type, you complete these steps:

1. Uninstall (including manual deletion of directories) and then reinstall of Tableau Server. The procedure for full uninstall and clean install are at the end of this topic.
2. Restore content and permissions.

In these steps the term "restore" does not refer to using the `TSM maintenance restore` command to restore the backup you are making. You cannot restore a backup (`.tsbak`) created on a Tableau Server instance that uses a different identity store than the target Tableau Server. The backup is a best practice safeguard, in case you need to go back to your original Tableau Server configuration.



Warning

Changing the installation type on Tableau Server can be a complicated and time-consuming process. To avoid data loss or orphaning of content or users, you'll need to plan this process carefully. In all cases, user filters applied to workbooks and data sources will need to be updated manually after the change.

Most importantly, determine how you will transition content and permissions to the new identity store after you reinstall Tableau Server.

Methods for restoring content and permissions

The following list describes two methods for restoring content and permissions after you reinstall Tableau Server. Select the method that best fits with your environmental requirements.

- **Method 1: Use site export and import**—In this method, you start by exporting each site in your existing deployment. Then, you install the new server and configure it for the new identity store type. You then create new users in the default site on the new server. Finally, you import all the original sites. During the import stage, you can map the original identities to the new users that you created in the default site.

You must use the same version of Tableau Server for the export and import operations. Importing a site to a version of Tableau Server that is different from the exported site version is not supported.

Because this method exports all content and permissions at each site, it is the best method for organizations that require a high fidelity replica of the content and permissions after the identity store change is complete. Some organizations require an identity store change as the result of an authentication change. In these cases, a different user name syntax is often a requirement in the new model. This method, which includes a process of mapping original user names to new names, provides flexibility for such scenarios.

- **Method 2: Fresh installation; users republish content**—In this method, you install a new version of Tableau Server and select the new identity store type during setup. You also create new sites. You then create users and give them access, and they republish their workbooks and data sources. Unlike the other method, in this one, you do not reuse any of your existing Tableau Server infrastructure.

This method is most appropriate for smaller deployments with fairly autonomous and data savvy users. From an administrative perspective, this method is the simplest, since you're not actively porting over content. However, because you rely entirely on

users to republish content, this method may not be successful for large organizations or for those where centralized oversight of content is required.

User filters

User filters are domain-specific. Therefore, when the domain of Tableau Server changes or authentication type changes, filters no longer function as expected. Although the user filters are generated by Tableau Server, after they are set by the user, the filters are stored in the workbooks and data sources. Neither of these methods for changing the identity store modifies the contents of the workbooks or data sources.

As you plan the identity store change, you must also include a final task to correct user filtering in all workbooks and data sources with Tableau Desktop.

User names and the Tableau Identity store

If you are using Method 1, it's helpful to understand how Tableau Server stores user names in the Tableau identity store. Tableau stores all user identities in the repository, which coordinates content permissions and site membership with various services in Tableau Server. Generally, an identity store configured for Active Directory store user names in the format, `domain\username`. Some organizations use a UPN (`jsmith@domain.lan`).

On the other hand, organizations that configure Tableau Server with local identity store usually create standard, truncated user names, such as `jsmith`.

In all cases, these user names are literal strings that must be unique in the Tableau identity store. If you are changing from one identity store type to another, then your target authentication, SSO, or user provisioning solution may require a specific user name format.

Therefore, to maintain all permissions, content, and user viability, one of the following must be true after you change the identity store type:

- The new user names must match the original user names, or
- The original user names must be updated to match a new format.

If an authentication change is driving the identity store change, then the target authentication scheme will likely impose a user name syntax that is different than your original user names. Method 1 includes a process where you can map original user names to new user names.

It's possible that the original user name format will work with the new authentication type. For example, if you used UPN names in a local identity store deployment, you might be able to use the same user names in an Active Directory deployment. You could also use the `domain\username` format for local identity store, as long as users continue to use that format to sign in to Tableau Server.

If you are changing from local identity store to an external Active Directory store, review the topic, [User Management in Deployments with External Identity Stores](#), as part of your planning process.

Method 1: Use site export and import

You must use the same version of Tableau Server for the export and import operations.

1. Export all sites on your server. See [Export or Import a Site](#).
2. Back up, remove, and then reinstall .
3. Create new users on Tableau Server. You should have a new user that corresponds to each user on the original server.
4. Import the sites that you exported in Step 1. See [Export or Import a Site](#). During import, you will be prompted to map the new users to the original users.

Method 2: Fresh installation—users republish content

Even if you do not plan to port content as part of your identity store change, we recommend that you back up the server.

1. Back up, remove, and then reinstall .
2. Create users, sites, and groups.
3. Inform your users of the new Tableau Server, provide them with credentials, and allow them to republish their content.

Back up, remove, and then reinstall

Both methods include the following steps:

1. Back up Tableau Server
2. Remove Tableau Server.
3. Reinstall Tableau Server with the new identity store type.

Step 1: Back up Tableau Server

As a best practice, you should back up the server before proceeding.

Follow the procedure, Create a backup using the TSM command line interface (CLI). Run the `backup` command with the `-d` option. The `-d` option adds the datestamp.

When you are finished, copy the backup file (`.tsbak`) to a safe location that is not a part of your Tableau Server installation.

Step 2: Remove Tableau Server

You must completely remove Tableau Server from the computer. See [Remove Tableau Server from Your Computer](#).

Step 3: Reinstall Tableau Server with new authentication type

1. Go to the Tableau Customer Portal, sign in with your Tableau user name and password, and then download Tableau Server.
2. Install Tableau Server. See [Install and Configure Tableau Server](#) more information. During installation, you will select the new identity store type. See [Configure Initial Node Settings](#).

External Identity Store Configuration Reference

Tableau Server supports connecting to an external directory using LDAP. In this scenario, Tableau Server imports users from the external LDAP directory into the Tableau Server repository as system users.

This topic provides a description of all LDAP-related configuration options Tableau Server supports. If you are connecting to Active Directory, we strongly recommend that you automatically configure the LDAP connection with Tableau Server as part of Setup, rather than configuring the connection manually. See [Configure Initial Node Settings](#).

The options listed in this reference can be used for any LDAP-compliant directory. If you do not have experience configuring LDAP, then work with your directory administrator, or with an LDAP expert.

This is a reference topic. For more information about how Tableau Server stores and manages users, start with [Identity Store](#).

Configuration methods

Configuration parameters that enable Tableau Server to connect to your LDAP directory are stored in .yml files. These files are managed and synchronized by various services in Tableau Server. Updating the .yml files must be done using a Tableau Services Manager (TSM) interface.

Do not attempt to update .yml files directly with a text editor. TSM must manage all updates for proper operation.

The .yml configuration files are composed of key-value pairs. For example, the key, `wgserver.domain.username`, takes a username as a value. This key defines the username that will be used to authenticate to the LDAP directory during the bind operation.

There are four different TSM methods that can set yml key values. The four methods are described here, using the `wgserver.domain.username` key as an example to illustrate the different methods:

- **configKey key-value pairs**—You can update a .yml configuration file key by updating the `wgserver.domain.username` key running `tsm configuration set Options`, or by including the key in a JSON configuration file under a `configKey` entity. See [Configuration File Example](#).

The `configKey` key-value pairs in a JSON configuration file are the same as those used for `tsm configuration set` but they are set differently. This topic refers to both of these methods as *configKey*.

Unlike when using `configEntities` and native `tsm` commands that are described below, `configKey` input is not validated. When you set an option with a `configKey`, the value that you enter is copied as a literal string to the underlying `.yaml` configuration files. For example, for a key where `true` or `false` are the valid inputs, when you configure the key using a `configKey` key-value pair, you can enter an arbitrary string value and it will be saved for the key. In such cases, invalid values will undoubtedly lead to LDAP configuration errors.

We recommend using `configKeys` only when no option exists to set the configuration with the other three options listed below (`configEntities`, a native `tsm` command, or the TSM Web UI). When using `configKeys` be sure to double-check your values and be sure to mind case-sensitivity.

- `configEntities` JSON—You can update a `.yaml` configuration file by passing the `username` option in a *configEntities* JSON.

When you configure a value using `configEntities` options in a JSON file, the values are validated before they are saved. Values are case-sensitive. For details on how to configure a value using `configEntities`, see the `identityStore` Entity example. The JSON file is imported with the `tsm settings import` command. The options available for `configEntities` are a subset of all the `.yaml` key-value pairs.

Validation means that the import command will only succeed if all the values in the JSON file are valid data types. For example, if you enter `no` for a value that only accepts `true` or `false`, then you will receive an error and the configuration is not imported.

You can only import JSON configuration files only as part of the initial configuration. If you need to make LDAP changes after you have imported the JSON configuration file

and initialized Tableau Server, do not attempt to re-import the JSON file. Instead, make individual key changes with native tsm commands if available, or using `configKeys` and `tsm configuration set`.

- Native tsm commands—You can update a .yml configuration file by passing the `ldapuser` option with the *native tsm command* `tsm user-identity-store`. As with `configEntities`, values that you enter with the native tsm command are validated before they are saved.

Not all key-value pairs in a .yml file can be set using native tsm commands.

- TSM GUI—You can set configuration values during Setup, using the TSM GUI. If you are connecting to Active Directory, and configure the Tableau identity store during Setup, with the GUI, then you are prompted for an account with AD read access. The `wgserver.domain.username` key is set when you enter credentials.

This scenario only works if you are connecting to Active Directory. Tableau Server does not support arbitrary LDAP configuration as part of the GUI Setup process.

Consider using the [Tableau Identity Store Configuration Tool](#) to generate your LDAP json configuration file. The Tableau Identity Store Configuration Tool will also generate a list of key/value pairs that you can set by running `tsm configuration set Options`. The tool itself is not supported by Tableau. However, using a JSON file created by the tool instead of creating a file manually does not change the supported status of your server.

Configuring Active Directory

If you are configuring Tableau Server to use Active Directory, we recommend using the TSM Web UI during installation. The TSM Web UI is optimized to configure Tableau Server for Active Directory with the minimum necessary input. See [Configure Initial Node Settings](#).

Configuration reference table

con- figEntities	Nat- ive	configKey (Used with tsm con-	Sc- en-	Notes

option (Options are case sensitive)	tsm command	figuration set command or in the configKeys section of a JSON file)	ario	
type	N/A	wgserv-er.authenticate	A-D, L-D-A-P, L-oc-al	Where you want to store user identity information. Values: <code>local</code> or <code>activedirectory</code> . If you want to connect to any LDAP server, enter <code>activedirectory</code> .
sslPort	N/A	wgserv-er.domain.ssl_port	A-D, L-D-A-P	Use this option to specify the secure port of the LDAP server. We recommend secure LDAP for simple bind. LDAPS is usually port 636.
N/A	N/A	wgserv-er.-domain.ldap.starttls-enabled	A-D, L-D-A-P	Values: <code>true</code> or <code>false</code> . Beginning with version 2021.2, this key is set to <code>true</code> by default when Tableau Server is configured to connect to Active Directory. As a result, simple bind to LDAP directory is encrypted when a valid SSL/TLS certificate is present in the Tableau key store. For more information, see Configure Encrypted Channel to LDAP External Identity Store .

				<p>This key is set to <code>false</code> by default when Tableau Server is configured to connect to a an arbitrary (but not Active Directory) LDAP server.</p> <p>This key was introduced (but not set) in version 2021.1.</p>
port	N/A	wgserv- er.domain.port	A- D, L- D- A- P	Use this option to specify the non-secure port of the LDAP server. Plain-text is usually 389.
domain	dom- ain	wgserv- er.domain.default	A- D	<p>In Active Directory environments, specify the domain where Tableau Server is installed, for example, "example.lan".</p> <p>For non-AD LDAP: the string you enter for this value is displayed in the "Domain" column of user management tools. You can enter an arbitrary string, but the key cannot be blank.</p> <p>This key is redundant with <code>wgserv-er.domain.fqdn</code>. The values for both keys must be the same.</p> <p>Native tsm command: Uses <code>tsm user-identity-store set-connection [options]</code> command.</p>
username	ldap-	wgserv-	A-	The user name that you want to use to

Tableau Server on Linux Administrator Guide

	user-name	er.domain.username	D, L-D-A-P	<p>connect to the directory service.</p> <p>The account that you specify must have permission to query the directory service.</p> <p>For Active Directory, enter the username, for example, <code>jsmith</code>.</p> <p>For LDAP servers, enter the distinguished name (DN) of the user that you want to use to connect. For example, <code>"cn==jsmith,dc=example,dc=lan"</code>.</p> <p>Native tsm command: Uses <code>tsm user-identity-store set-connection [options]</code> command.</p>
password	ldap-password	wgserv-er.domain.password	A-D-L-D-A-P	<p>The password of the user account that you will use to connect to the LDAP server.</p> <p>Native tsm command: Uses <code>tsm user-identity-store set-connection [options]</code> command.</p>
directoryServiceType	N/A	wgserv-er.-domain.-directoryservice.type	A-D-L-D-A-P	<p>The type of LDAP directory service that you want to connect to. Values: <code>activedirectory</code> or <code>openldap</code>.</p>
kerber-	kerbpri-	wgserv-er.-	A-	<p>The service principal name for Tableau</p>

osPrincipal	ncipal	domain.ldap.-principal	D, L-D-A-P	<p>Server on the host machine. The keytab must have permission for this principal. Do not use the existing system keytab at <code>/etc/krb5.keytab</code>. Rather, we recommend that you register a new service principal name. To see principals in a given keytab, run the <code>klist -k</code> command. See Understanding Keytab Requirements.</p> <p>Native tsm command: Uses <code>tsm user-identity-store set-connection [options]</code> command.</p>
hostname	host-name	wgserv-er.-domain.ldap.host-name	A-D, L-D-A-P	<p>The hostname of the LDAP server. You can enter a hostname or an IP address for this value. The host that you specify here will be used for user/group queries on the primary domain. In the case where user/group queries are in other domains, Tableau Server will query DNS to identify the appropriate domain controller.</p> <p>Native tsm command: Uses <code>tsm user-identity-store set-connection [options]</code> command.</p>
member-sRetrievalPageSize	N/A	wgserv-er.-domain.ldap.member-s.retrieval.page.size	A-D, L-D-A-P	<p>This option determines the maximum number of results returned by an LDAP query.</p> <p>For example, consider a scenario where Tableau Server is importing an LDAP group that contains 50,000</p>

				<p>users. Attempting to import such a large number of users in a single operation is not a best practice. When this option is set to 1500, Tableau Server imports the first 1500 users in the first response. After those users are processed, Tableau Server requests the next 1500 users from the LDAP server, and so forth.</p> <p>We recommend that you modify this option only to accommodate the requirements of your LDAP server.</p>
N/A	N/A	wgserv-er.-domain.ldap.-connectionpool.enabled	A-D, L-D, A-P	<p>When this options is set to <code>true</code>, Tableau Server will attempt to reuse the same connection when sending queries to the LDAP server. This behavior decreases the overhead of having to re-authenticate with the LDAP server on each new request. Connection pooling only works with simple bind and TSL/SSL bind connections. Connection pooling is not supported for GSSAPI bind connections.</p>
N/A	N/A	wgserv-er.domain.accept_list	A-D	<p>Allows connection from Tableau Server to secondary Active Directory domains. A secondary domain is one that Tableau Server connects to for user synchronization, but is a domain where Tableau Server is not installed. To ensure that Tableau Server can connect to other Active Directory</p>

				domains, you must specify the trusted domains by setting the <code>wgserver.domain.accept_list</code> option with TSM. For more information, see <code>wgserver.domain.accept_list</code> .
N/A	N/A	<code>wgserver.domain.whitelist</code>	A-D	<p>Important: Deprecated as of version 2020.4.0. Use <code>wgserver.domain.accept_list</code> instead.</p> <p>Allows connection from Tableau Server to secondary Active Directory domains. A secondary domain is one that Tableau Server connects to for user synchronization, but is a domain where Tableau Server is not installed. To ensure that Tableau Server can connect to other Active Directory domains, you must specify the trusted domains by setting the <code>wgserver.domain.whitelist</code> option with TSM. For more information, see <code>wgserver.domain.whitelist</code>.</p>
<code>kerberosConfig</code>	<code>kerbconfig</code>	No direct mapping	A-D, L-D-A-P	<p>The path to the Kerberos configuration file on the local computer. If you are installing into Active Directory, we don't recommend using the existing Kerberos configuration file or keytab file that may already be on the domain-joined computer. See Identity Store</p> <p>Native tsm command: Uses <code>tsm user-</code></p>

				identity-store set-connection [options] command.
kerberosKeytab	kerb-keytab	No direct mapping	A-D, L-D-A-P	<p>The path to the Kerberos keytab file on the local computer. It is recommended that you create a keytab file with keys specifically for Tableau Server service and that you do not share the keytab file with other applications on the computer. For example, on Linux, you might place the keytab file in the <code>/var/opt/tableau/keytab</code> directory.</p> <p>Native tsm command: Uses <code>tsm user-identity-store set-connection [options]</code> command.</p>
nickname	N/A	wgserv-er.domain.nickname	A-D, L-D-A-P	<p>The nickname of the domain. This is also referred to as the NetBIOS name in Windows/Active Directory environments. The <code>nickname</code> option is required for all LDAP entities. The value cannot be null. If your organization does not require a nickname/NetBIOS, then pass a blank key, for example: <code>" "</code>.</p>
root	N/A	wgserv-er.domain.ldap.root	L-D-A-P	<p>If you do not use a dc component in the LDAP root or you want to specify a more complex root you need to set the LDAP root. Use the <code>"o=my,u=root"</code> format. For example, for the domain,</p>

				example.lan, the root would be "o=example,u=lan".
server-SideSorting	N/A	wgserv-er.-domain.ldap.server_side_sorting	L-D-A-P	Whether the LDAP server is configured for server-side sorting of query results. If your LDAP server supports server-side sorting, set this option to <code>true</code> . If you are unsure whether your LDAP server supports this, enter <code>false</code> , as misconfiguration may cause errors.
rangeRetrieval	N/A	wgserv-er.-domain.ldap.range_retrieval	L-D-A-P	Whether the LDAP server is configured to return a range of query results for a request. This means that groups with many users will be requested in small sets instead of all at once. LDAP servers that support range retrieval will perform better for large queries. If your LDAP server supports range retrieval, set this option to <code>true</code> . If you are unsure whether your LDAP server supports range retrieval, enter <code>false</code> , as misconfiguration may cause errors.
bind	N/A	wgserv-er.-domain.ldap.bind	L-D-A-P	The way that you want to secure communication to the directory service. Enter <code>simple</code> for LDAP unless you are connecting to an LDAP server with Kerberos. For Kerberos, enter <code>gssapi</code> .
N/A	N/A	wgserv-er.-domain.ldap.-	L-D-A-	Note: This key is only supported

		domain_custom_ports	P	<p>for Tableau Server on Linux.</p> <p>Allows you to map child domains and their LDAP ports. Domain and port are separated by a colon (:) and each domain:port pair is separated by a comma (,) using this format: <code>FQDN1:port,FQDN2:port</code></p> <p>Example: <code>tsm configuration set -k wgserv-er.domain.ldap.domain_custom_ports -v child-domain1.lan:3269,child-domain2.lan:3269,child-domain3.lan:389</code></p>
dis-tin-guishedNameAttribute	N/A	wgserv-er.-domain.ldap.dnAttribute	L-D-A-P	<p>The attribute that stores the distinguished names of users. This attribute is optional, but it greatly improves the performance of LDAP queries.</p> <p>Important: Do not set this option as part of the initial configuration. Only set this after you have validated overall LDAP functionality. You must have a dnAttribute set in your organization before setting</p>

				<div style="background-color: #f0f0f0; padding: 5px; border: 1px solid #ccc;"> <p>this key.</p> </div>
groupBaseDn	N/A	wgserv-er.-domain.ldap.-group.baseDn	L-D-A-P	<p>Use this option to specify an alternative root for groups. For example, if all of your group are stored in the base organization called "groups," then enter "o=groups".</p>
N/A	classnames	wgserv-er.-domain.ldap.-group.classnames	L-D-A-P	<p>By default Tableau Server looks for LDAP group object classes containing the string "group". If your LDAP group objects do not fit the default class name, override the default by setting this value. You can provide multiple classnames separated by commas.</p> <p>If your group names include commas, you must escape them with a backslash (\). For example, if you have a group name, groupOfNames, top, then enter "groupOfNames\, top".</p> <p>Tableau LDAP implementation interprets LDAP objects as either user or group. Therefore, be sure that you are entering the most specific class name. Overlapping class names between users and groups may cause conflicts.</p> <p>Native tsm command: Uses tsm user-</p>

				identity-store set-group-mappings [options] command.
groupBaseFilter	base-filter	wgserv-er.-domain.ldap.-group.baseFilter	L-D-A-P	<p>The filter that you want to use for groups of users of Tableau Server. You might specify an object class attribute and an organization unit attribute. For example:</p> <pre>" (&(objectClass=s=groupofNames) (ou=Group)) "</pre> <p>If " (&(objectClass=s=inetOrgPerson) (ou=People)) " doesn't work in your LDAP implementation, then specify the base filter that works for your Tableau user base.</p> <p>This is a required key. It cannot be blank.</p> <p>Native tsm command: Uses tsm user-identity-store set-group-mappings [options] command.</p>
groupName	groupname	wgserv-er.-domain.ldap.-group.name	L-D-A-P	<p>The attribute that corresponds to group names on your LDAP server.</p> <p>Native tsm command: Uses tsm user-identity-store set-group-mappings [options] command.</p>
groupE-	group-	wgserv-	L-	The attribute that corresponds to

mail	pem- ail	er.- domain.ldap.- group.email	D- A- P	group email addresses on your LDAP server. Native tsm command: Uses tsm user-identity-store set-group-mappings [options] command.
groupDe- scription	desc- rip- tion	wgserv- er.- domain.ldap.- group.description	L- D- A- P	The attribute that corresponds to group descriptions on your LDAP server. Native tsm command: Uses tsm user-identity-store set-group-mappings [options] command.
member	me- mbe- r	wgserv- er.- domain.ldap.- group.member	L- D- A- P	Specify the LDAP attribute that contains a list of distinguished names of users that are part of that group. Native tsm command: Uses tsm user-identity-store set-group-mappings [options] command.
N/A	N/A	wgserv- er.- domain.ldap.- group.memberURL	L- D- A- P	Specify the name of the LDAP attribute that stores the LDAP query for dynamic groups.
user- BaseDn	N/A	wgserv- er.- domain.ldap.user- .baseDn	L- D- A- P	Use this option to specify an alternative root for users. For example, if all of your users are stored in the base organization called "users," then enter "o=users".
N/A	clas-	wgserv-	L-	By default Tableau Server looks for

	snames	er.- domain.ldap.user- .classnames	D- A- P	<p>LDAP user object classes containing the string “user” and “inetOrgPerson”.</p> <p>If your LDAP user objects do not use these default class names, override the default by setting this value. You can provide multiple classnames separated by commas. For example: "userclass1, userclass2".</p> <p>If your names include commas, you must escape them with a backslash (\). For example, if you have a name, Names, top, then enter "Names\, top".</p> <p>Native tsm command: Uses tsm user-identity-store set-user-mappings [options] command.</p>
user-BaseFilter	base-filter	wgserv-er.- domain.ldap.user- .baseFilter	L- D- A- P	<p>The filter that you want to use for users of Tableau Server. You might specify an object class attribute and an organization unit attribute.</p> <p>For example: " (&(objectClass=s=inetOrgPerson) (ou=People)) "</p> <p>Native tsm command: Uses tsm user-identity-store set-user-mappings [options] command.</p>

user-Username	ldap-user-name	wgserv-er.-domain.ldap.user-.username	L-D-A-P	<p>The attribute that corresponds to user names on your LDAP server.</p> <p>Native tsm command: Uses tsm user-identity-store set-user-mappings [options] command.</p>
user-DisplayName	display-name	wgserv-er.-domain.ldap.user-.displayname	L-D-A-P	<p>The attribute that corresponds to user display names on your LDAP server.</p> <p>Native tsm command: Uses tsm user-identity-store set-user-mappings [options] command.</p>
userEmail	email	wgserv-er.-domain.ldap.user-.email	L-D-A-P	<p>The attribute that corresponds to user email addresses on your LDAP server.</p> <p>Native tsm command: Uses tsm user-identity-store set-user-mappings [options] command.</p>
user-Certificate	certificate	wgserv-er.-domain.ldap.user-.usercertificate	L-D-A-P	<p>The attribute that corresponds to user certificates on your LDAP server.</p> <p>Native tsm command: Uses tsm user-identity-store set-user-mappings [options] command.</p>
N/A	thumbnail	wgserv-er.-domain.ldap.user-.thumbnail	L-D-A-P	<p>The attribute that corresponds to user thumbnail images on your LDAP server.</p> <p>Native tsm command: Uses tsm user-identity-store set-user-mappings [options] command.</p>

Tableau Server on Linux Administrator Guide

user-JpegPhoto	jpeg-photo	wgserv-er.-domain.ldap.user-.jpegphoto	L-D-A-P	<p>The attribute that corresponds to user profile images on your LDAP server.</p> <p>Native tsm command: Uses tsm user-identity-store set-user-mappings [options] command.</p>
memberOf	memberof	wgserv-er.-domain.ldap.user-.memberof	L-D-A-P	<p>Group that the user is a member of.</p> <p>Native tsm command: Uses tsm user-identity-store set-user-mappings [options] command.</p>
groupClassNames	N/A	wgserv-er.-domain.ldap.-group.classnames	L-D-A-P	<p>By default Tableau Server looks for LDAP group object classes containing the string "group". If your LDAP group objects do not fit the default class name, override the default by setting this value.</p> <p>For configEntity: This option takes a list of strings, which requires passing each class in quotes, separated by a comma (no space) and within brackets. For example: ["basegroup", "othergroup"].</p> <p>For configKey: Enter each class, separated by a comma (no space) and within double quotes. For example: "basegroup, othergroup".</p>
user-ClassName	N/A	wgserv-er.-	L-D	<p>By default Tableau Server looks for LDAP user object classes containing</p>

es		domain.ldap.user-classnames	A-P	<p>the string “user” and “inetOrgPerson”. If your LDAP user objects do not use these default class names, override the default by setting this value.</p> <p>For configEntity: This option takes a list of strings, which requires passing each class in quotes, separated by a comma (no space) and within brackets. For example: ["user-class1", userclass2"].</p> <p>For configKey: Enter each class, separated by a comma (no space) and within double quotes. For example: "userclass1, userclass2".</p>
----	--	-----------------------------	-----	---

Calculated configKeys

The following Kerberos-related configKeys are calculated and set according to multiple environmental inputs. As such, they must be set by the native tsm command or configEntities. Do not attempt to set these configKeys manually.

Calculated configKey	To use the native TSM command:	To use configEntity json:
wgserver.domain.ldap.kerberos.conf, cfs.ldap.kerberos.conf	Set the Kerberos configuration file location with the <code>kerbconfig</code> option of <code>tsm user-identity-store set-connection [options]</code> command.	Set the Kerberos configuration file location with the <code>kerberosConfig configEntity</code> option.

<p>wgserver.domain.ldap.kerberos.keytab, cfs.ldap.kerberos.keytab</p>	<p>Set the Kerberos keytab file location with the <code>kerberosKeytab</code> option of <code>tsm user-identity-store set-connection [options]</code> command.</p>	<p>Set the Kerberos keytab file location with the <code>kerberosKeytab configEntity</code> option.</p>
---	--	--

Unsupported configKeys

Some unsupported configKeys are present in underlying .yml configuration files. The following keys are not intended for standard deployments. Do not configure these keys:

- wgserver.domain.ldap.kerberos.login
- wgserver.domain.ldap.guid
- wgserver.domain.fqdn: this key is redundant with wgserver.domain.default. The values for both keys must be the same. Only update wgserver.domain.fqdn if the value does not match wgserver.domain.default.

Basic Product Data

By default Tableau products send usage data to Tableau so that we can understand how customers use our software, and gain insights into where they are successful and where they might run into problems that we can address. For example, this data can help us learn where upgrades are commonly failing and allow us to make product changes to address those issues, or identify what portion of our user base needs informed about a security issue that applies to a specific version of Tableau Server. You can disable the sending of this data at installation time, or later. For details on how to do this, see the instructions for [Tableau Desktop](#) or [Tableau Server](#).

Even when you disable the sending of product usage data, certain basic product data is sent to Tableau. This Basic Product Data includes information about products and their processes, including which product or process is running, when they start up, what operating system they are running on, licensing information, which machine or cluster of machines has

sent the data (using unique pseudonymized identifiers), and whether the product is configured to send product usage data.

You can disable the sending of Basic Product Data at the machine level, or at the enterprise level, by blocking traffic sent to **prod.telemetry.tableausoftware.com**.

Disabling sharing of Basic Product Data on individual computers

Important: This procedure involves modifying your local `hosts` file. If you do not know what this is, you should not change it. You should only make this change if you understand the implications of making changes to the file, know how to change the file, and have made a backup of the file for safety.

Modifying `hosts` files changes network behavior for computers. Detailed instructions for modifying `hosts` files are provided by operating system providers such as Microsoft, Apple or Linux Distributions.

1. Make a copy of your existing `hosts` file and save it to a computer that is not your Tableau computer. This is your backup, in case you need to reverse your changes. Do not start modifying the file until you have made a backup copy of it.
2. Modify your computer's `hosts` file to include these lines:

```
# Stops sending Product Usage to Tableau (prod.telemetry.tableausoftware.com) .
# Learn more here: http:\\tableau.com\derived-data
127.0.0.1    prod.telemetry.tableausoftware.com
```

The first and second lines are comments, explaining the third line.

The third line prevents all traffic to `prod.telemetry.tableausoftware.com` (`http://prod.telemetry.tableausoftware.com/`) from leaving your local machine by sending it to the Internal host loopback address. The data does not get sent outside the computer.

Disabling the sharing of Basic Product Data at the enterprise level

To disable sending of Basic Product Data on an enterprise level, modify your Network Firewall to prevent outbound traffic to `prod.telemetry.tableausoftware.com`.

This domain is used by Tableau to receive the Basic Product Data about process launch and shutdown. It is also used for the more general Product Usage Data. Blocking traffic to this domain it you will prevent both kinds of data from being sent.

Traffic to this domain will occur on Ports 80 (for initial registration of our Product Data clients) and on Port 443 (for all subsequent traffic). To completely prevent product data from being sent, block all traffic to this domain.

For details on how to configure your network firewall, refer to your vendor or your internal IT department. Tableau cannot provide these instructions.

About Tableau Advanced Management

Tableau Advanced Management is a separately licensed Add-on to Tableau Server which will provide enhanced security, manageability, and scalability capabilities for Tableau Server.

Note: Tableau Advanced Management includes several Tableau Server features and two separately installed tools: Tableau Content Migration Tool and the Tableau Resource Monitoring Tool. For more information, see the [feature table](#) later in this topic.

Advanced Management Licensing Requirements

Advanced Management is licensed on a per Deployment basis, which may be User-Based or Core-Based. A Deployment includes a licensed production Tableau Server installation and two licensed non-production Tableau Server installations that support the production installation. For more information on Deployment, see the [EULA Documentation](#).

- The Advanced Management can only be activated on a licensed Tableau Server Deployment. This means that your Tableau Server must be first activated with a valid

key that is either User-Based or Core-Based, before applying the Advanced Management product key. For more information on how to purchase Advanced Management and get the product key, contact your account manager.

- When the product key is active and enabled, you can use all the features that are included in the Add-on.
- When the Server Management Add-on product key is removed or deactivated, you will no longer be able to use the features that require a valid Advanced Management license. Any associated data will not be deleted. Each feature might have slight differences on what happens when the license expires. For more information on the individual features use the links in the table below.

The following table lists the features that are included and require a valid Advanced Management license:

Feature	Description	Requirements to use the feature	Version
About Tableau Resource Monitoring Tool	Provides a comprehensive look at the health of Tableau Server. With the Resource Monitoring Tool you can identify issues that cause slow load times, extract failures, and other critical issues and can help you proactively address the issues that impact end user experience.	The Tableau Server that you are monitoring must have a Advanced Management license.	Available in 2019.3 and later. Linux support was introduced in version 2020.4.
About Tableau Content Migration Tool	The Content Migration Tool provides an easy way to copy or migrate content between Tableau Server projects, sites and deployments. You can do this between projects on separate Tableau Server installations (for instance, between	<ul style="list-style-type: none"> • Both the source Tableau Server (Server that you are moving the con- 	Available in version 2019.3 and later.

Feature	Description	Requirements to use the feature	Version
	<p>a development instance of Tableau Server and a production instance of Tableau Server), or between projects on a single Tableau Server installation.</p>	<p>tent from) and the target Tableau Server (Server that you are moving the content to) must have a valid Advanced Management license.</p> <ul style="list-style-type: none"> • The Content Migration Tool must be installed on a version of Microsoft Windows that supports .NET 4.6.1 (Windows 7 or later, Windows Server 2008R2 or later). 	
<p>Tableau Server External Repository</p>	<p>Allows you to deploy Tableau Server Repository external to Tableau Server. The Tableau Server Repository is a PostgreSQL database that stores data about all user interactions, extract refreshes,</p>	<p>The Tableau Server that is using an external repository must have a Advanced Management license.</p>	<p>Available in version 2019.3 and later. - AWS</p>

Feature	Description	Requirements to use the feature	Version
	and more.		supported on Tableau Server version 2019.3 and later. - Azure supported on Tableau Server version 2020.4 and later.
Workload Management through Node Roles	Using node roles, you can configure where certain types of workloads are processed on your Tableau Server installation. The node roles features allows you to dedicate and scale resources to specific workloads (ex: extract refreshes, subscriptions).	The Tableau Server must have a valid Advanced Management license.	Available in version 2019.3 and later.
Tableau Server Key Management System	Gives you additional functionality to configure Tableau Server to use AWS as the KMS for extract encryption.	Tableau Server must have a valid Advanced Management license.	Available in version 2019.3 and later.

Feature	Description	Requirements to use the feature	Version
			- AWS supported on Tableau Server version 2019.3 and later.
			- Azure supported on Tableau Server version 2021.1 and later.
Tableau Server External File Store	Allows you to use network attached storage(NAS) as your File Store. This removes the need to run File Store locally on your Tableau Server	Tableau Server must have a valid Advanced Management license.	Available in version 2020.1 and later.

Activating the Advanced Management license

The Advanced Management license is applied to a Tableau Server Installation and can be used for both User-Based and Core-Based installations.

Here is a quick overview of how you can activate the Advanced Management license on your Tableau Server Installation.

Use the TSM web interface

1. Open TSM in a browser:

`https://<tsm-computer-name>:8850`

2. Click **Licensing** on the **Configuration** tab. Click **Activate License**.
3. Enter or paste your Advanced Management product key and click **Activate**.
4. On the **Register** page, enter your registration information and click **Register**.
5. Follow the prompts and restart Tableau Server after registration is complete.

Note: If this is a new Tableau Server installation, and you apply the Server Management Add-on Key before you apply the Tableau Server product key, you will see an error. You may however, continue the installation and apply the Tableau Server product key using the same steps described above.

Use the TSM CLI

1. Open a command prompt as administrator on a node in the Tableau Server cluster.
2. Run the following command with your Advanced Management product key to activate the license:

```
tsm licenses activate -k <server-management-add-on-product key>
```

Note: If this is a new Tableau Server installation, run the command twice, first with the Tableau Server product key and then with the Server Management Add-on product key.

3. Apply the changes and restart the Server:

```
tsm pending-changes apply
```

Who can do this

Only Server Administrators can activate Advanced Management license keys.

About Tableau Resource Monitoring Tool

This document gives you an overview of the Tableau Resource Monitoring Tool.

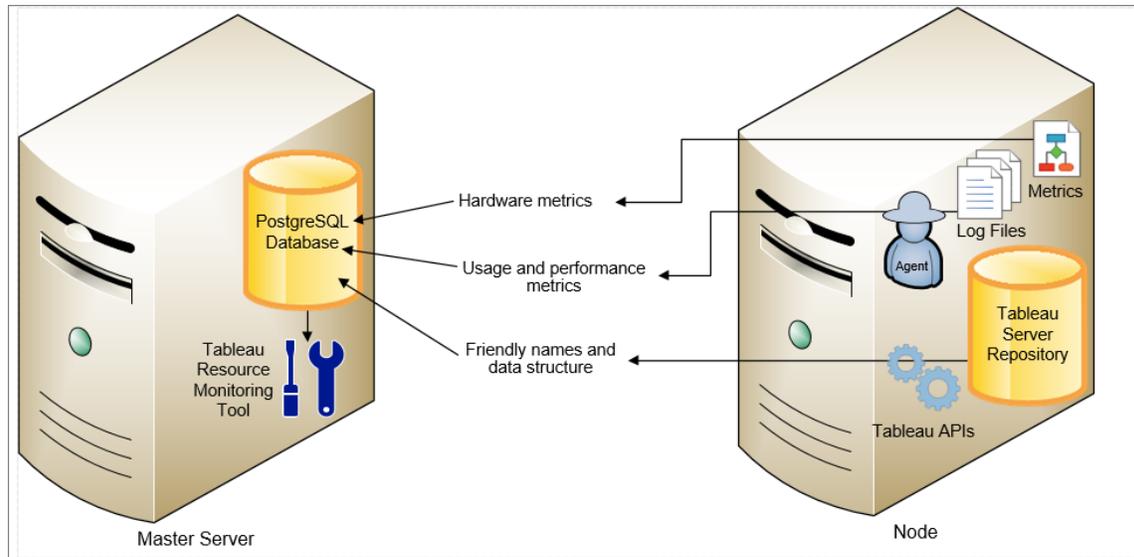
What is Resource Monitoring Tool?

The Resource Monitoring Tool is a tool that you can use to monitor the health and performance of your Tableau Server. It gathers data from your Tableau Server and provides a comprehensive look at the health of Tableau Server. Using this tool, you can identify what is causing slow load times, extract failures, and other critical issues. To use the Resource Monitoring Tool, you must have the Advanced Management enabled on your server. For more information on the Advanced Management, see [About Tableau Advanced Management](#)

The Resource Monitoring Tool has two main components: Master Server and Agent.

The Master Server where the data from Tableau Server is collated and served up through a web interface, where you can configure, monitor, and analyze the health and performance of Tableau Server.

An Agent runs on each of the nodes in your Tableau cluster to monitor their performance and activity. The following diagram illustrates the interaction between a Tableau Server node and the Resource Monitoring Tool Master Server.



Get Started with Tableau Resource Monitoring Tool

This article will help you get started with Tableau Resource Monitoring Tool. It contains links to other articles about information you need to prepare before installing Resource Monitoring Tool. It also has links to upgrade and other useful resources.

Pre-Installation

Product Compatibility with Tableau Server

For the latest compatibility information, see [Resource Monitoring Tool Compatibility with Tableau Server](#).

Resource Monitoring Tool Master Server and Agent Compatibility

Generally, we recommend that you install the same version of Master Server and Agent to be sure that they are compatible. If you have Agents using versions not compatible with the Master Server version, a critical incident is logged. For more information, see [Agent Incidents](#).

Here is the

Tableau Server on Linux Administrator Guide

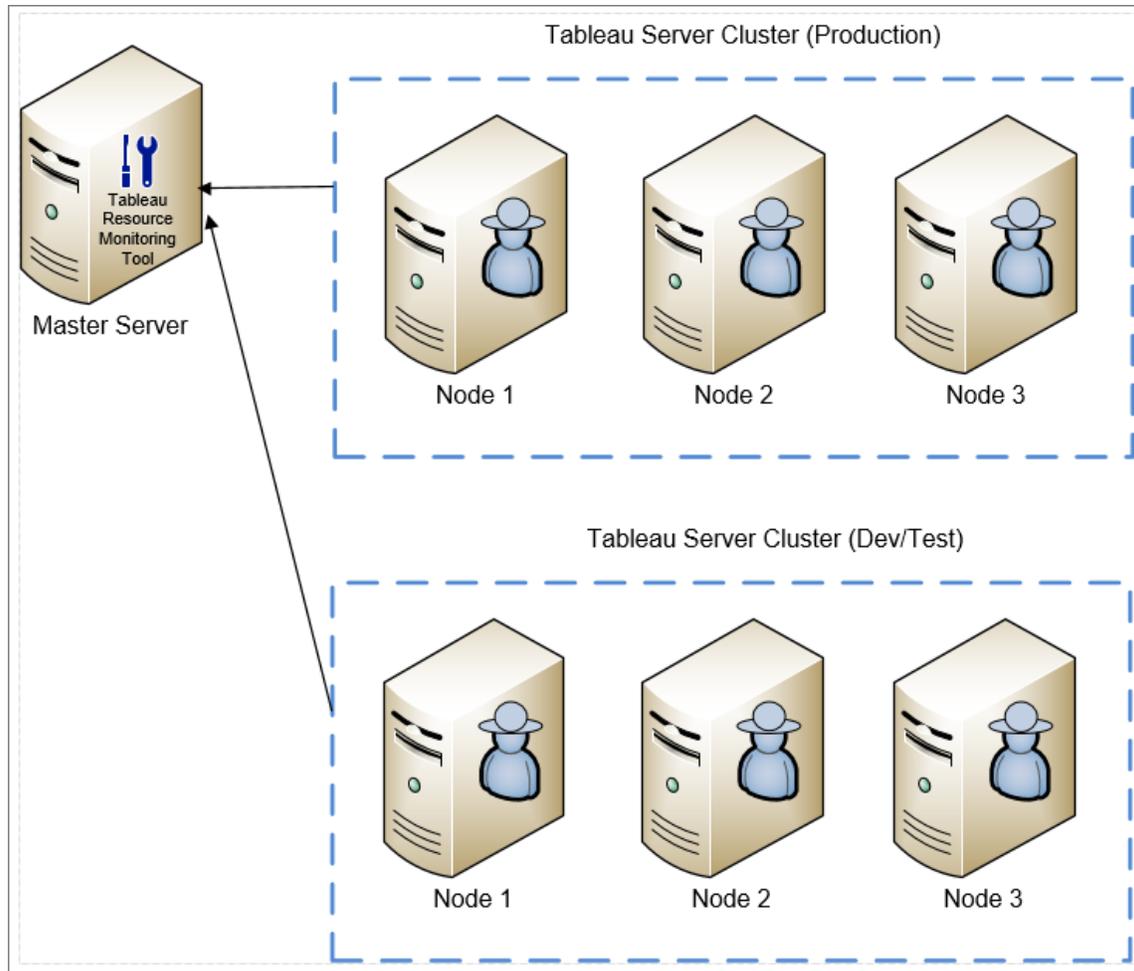
Concepts and Terms

Get familiar with the concepts and terms used in the Resource Monitoring Tool topics. For more information, see [Concepts](#).

Setup Architecture

Get an understanding of the setup architecture for Resource Monitoring Tool. The Resource Monitoring Tool has two main components: Master Server and Agent. The Master Server should be installed on its own machine, and the Agent is installed on each node of your Tableau Server Cluster. For a more details about the setup architecture, see [Install the Tableau Resource Monitoring Tool](#).

A high level view of the Resource Monitoring Tool setup. This diagram shows a three node Tableau Server Cluster.



Minimum Hardware Requirements and Recommendations for Resource Monitoring Tool

Resource Monitoring Tool handles all of the processing, aggregation, storage, and web interface to the monitoring data collected from your Tableau Server installation. For production use, the machine you install Resource Monitoring Tool on should meet or exceed the minimum hardware recommendations. For more details, see [Minimum Hardware Requirements and Recommendations for Tableau Resource Monitoring Tool](#).

Tableau Server on Linux Administrator Guide

Pre-Installation Checklist for Resource Monitoring Tool

Before you install Resource Monitoring Tool, review these action items and complete any necessary steps: [Pre-Installation Checklist - Tableau Resource Monitoring Tool](#)

Troubleshoot

Be sure to check our [Troubleshoot Tableau Resource Monitoring Tool Issues](#) for answers to common questions before contacting support.

Concepts

This document briefly explains some of Tableau Resource Monitoring Tool core concepts and defines some terminology you will see often.

Agent

Resource Monitoring Tool Agent runs as a background service on each of your Tableau servers. It must be installed on all the nodes in a Tableau Server installation.

The agent is a lightweight process that monitors the Tableau activity and general performance of the server it is installed on. It reports those details back to the Resource Monitoring Tool Master Server.

Master Server

The Resource Monitoring Tool Master Server hosts the web application that users interact with. It also does much of the background processing to collate and monitor the data from the agents.

Environment

Environment is the term used to refer to a single node Tableau Server installation, or a Tableau Server cluster. In the Master Server web interface, each node of the Tableau Server is called a **Server**.

Resource Monitoring Tool can monitor multiple Tableau Server installations. For example, you may have both production and staging Tableau Servers. In Resource Monitoring Tool,

you can create two environments (*Production* and *Staging*) and monitor both of those Tableau Server clusters.

Environment Status

The *environment status* indicates the state of the current Tableau environment. The status is determined by the number and type of incidents and the status of Tableau's various processes. The following sections will describe what each status means in these terms.

OK



No warning or critical incidents have occurred today and there are not currently any failed processes.

Warning



The Environment is in a **Warning** state when any of the following are true:

- At least one warning incident has occurred today.
- A process is **Unlicensed**.
- A process is **Offline**.

Critical



The Environment is in a **Critical** state when any of the following are true:

- At least one critical incident has occurred today.
- The Primary Gateway is **Offline**.
- All instances of a process are **Unlicensed**.
- All instances of a process are **Offline**.

Tableau Server on Linux Administrator Guide

Server

Server is the term used to refer to each of nodes in the Tableau Server installation. Every machine that you install the Resource Monitoring Tool Agent on is considered a Server.

Pre-Installation Checklist - Tableau Resource Monitoring Tool

Machine, Network, and Account Requirements

Complete the following prerequisites before installing the Tableau Resource Monitoring Tool.

1. Make sure you have administrative privileges for all computers that will run the Resource Monitoring Tool. This includes the computer running the Resource Monitoring Tool Master, and all the Tableau Server nodes in the installation you will be monitoring with Resource Monitoring Tool.
2. Open port 5672 to inbound TCP traffic on the computer running the Resource Monitoring Tool Master server

The Resource Monitoring Tool Agent application sends the data that it collects through port 5672, so the server where the Master program is installed will need to allow inbound TCP traffic on that port. Windows Firewall is a common blocker of this traffic.

3. Open the following ports:
 - Open port 80 to inbound TCP traffic on the computer running the Resource Monitoring Tool Master server

This is the port on which the Master application will host the web interface. Note, we can change the port number during installation if there is a 3rd party service already using it. To make the web interface accessible from outside the Master server, this port will need to be opened.

 - Open port 443 to inbound TCP traffic on the computer running the Resource

Monitoring Tool Master server.

This is the port on which connections come from the Agent when testing the connection to the Master.

- Open port 5672 to outbound TCP traffic on all Agent servers

The Agent application sends the data that it collects to the Master via this port.

- Open port 80 to outbound TCP traffic on all Agent computers.

Periodically, the Agent will make requests to the Master web interface.

- Open port 443 to outbound TCP traffic on all Agent computers.

This is the port on which the Agent tests its connection to the Master application.

For more information on ports and communications, see [Tableau Resource Monitoring Tool Communication Ports](#).

4. Verify that all Agent computers can contact the Master Server.

The web interface hosted by the Master application is bound to a specific host name or IP address during the setup process. It will only answer requests sent to that specific host name. For example, if the Master is configured to be hosted at `http://Master-Server/` but the Agent servers can only contact the Master server through `http://mymaster.myserver.com/` then the requests will not go through.

5. Gather credentials for a Tableau Server Administrator account.

Resource Monitoring Tool will use the Tableau Server web API to collect information about the content, so it will need credentials for a user that is a Tableau server administrator and can access all sites, projects, etc. We recommend creating one dedicated to Resource Monitoring Tool, but it can be any user that is a Tableau Server administrator.

6. (Optional) Enable access to the Tableau Server repository database:

Resource Monitoring Tool accesses the repository database directly for performance reasons. For this to work, access to the repository must be enabled, with a password set for the "readonly" database user. For details, see [Enable access to the Tableau Server repository](#).

If you want to use SSL connections between Resource Monitoring Tool and Tableau Server Repository, make sure Tableau Server is configured to use SSL for internal Postgres connections. For more information, see [Configure SSL for Internal Postgres Communication](#). The Resource Monitoring Tool allows you to use either the certificate file or thumbprint for the SSL connections. If you plan to use the certificate file, copy the certificate file generated by Tableau Server for internal Postgres SSL connections, to the machine where you plan to install the Resource Monitoring Tool Master Server. Make sure that the Operating System to trust the certificate.

7. Confirm that the REST API is enabled on Tableau Server (this is the default). Use the `tsm configuration get -k api.server.enabled` command to do this. A return value of `true` means the REST API is enabled. To enable the REST API, use the `tsm configuration set` command. For more information, see [api.server.enabled](#).

8. Review the size of Tableau Server logs. Once Agent Server is installed and configured, the Resource Monitoring Tool processes relevant historical data from Tableau Server logs before data is displayed. If there is a large amount of historical log data, it may take a while to process the information which in turn might result in a delay of processing newer events on the Server.

If you are concerned about the delay, and not having historical information does not concern you, you can do the following to clean up the existing files:

- Remove Unneeded Files, and consider Log File Snapshots (Archive Logs) before you remove log files.

Note: Performance data like CPU usage and memory usage are not gathered using historical log data and are collected after Agent is installed and configured so cleaning up historical data does not affect performance data.

9. (optional) Gather connections information for a SMTP server that the Master server can access.

If you want email notifications, you need to provide the Master with the location, username and password (if any), and port number for a SMTP server that it can use to send the email.

Who can do this

To do all the steps described above you need be an admin on the machine that you are installing Resource Monitoring Tool, and be a Tableau Server Administrator.

Minimum Hardware Requirements and Recommendations for Tableau Resource Monitoring Tool

Tableau Resource Monitoring Tool handles all of the processing, aggregation, storage, and web interface to the monitoring data collected from your Tableau Server installation.

For production use, the machine you install Resource Monitoring Tool on should meet or exceed the hardware recommendations below:

- 8 physical CPU cores (16 vCPUs)
- CPUs must support SSE4.2 and POPCNT instruction sets
- 32GB RAM
- 500GB Disk Space (SSD is recommended)

The Resource Monitoring Tool can provide real-time performance reporting for a Tableau Server deployment serving up to 5,500 views per hour. Above 5,500 views per hour, customers may experience delays in analyzing recent Server activity. Additional factors such as

Tableau Server on Linux Administrator Guide

Tableau Server Backgrounder load and Master server drive write performance may also contribute to delays. You can use the Traffic to Views Admin View to understand the peak view load in your deployment, using the following steps:

- Navigate to Server Status -> Traffic to views and make sure the filters are set to look at all sites, workbook, and views.
- Update the time range to capture all available data.
- Drill into the time axis on **What is the Total View Count by Day?** to get the view count per hour.

Resource Monitoring Tool Agent - Resource Utilization

The Resource Monitoring Tool Agent monitors Windows and Tableau Server processes and log files and sends performance metrics and Tableau log data to the Resource Monitoring Tool for near real time processing. The Agent needs to be installed on every machine that is running Tableau Server processes.

On Tableau Server installations that match Tableau's minimum requirements, we typically see the Agent using 0-5% CPU on average with infrequent spikes above that. Our performance target is an average of 10% CPU usage or less.

The agent typically uses 200 MB RAM or less with infrequent spikes during periods of heavier Tableau Server activity.

A minimum of 10 GB free disk space is recommended for the agent.

The Resource Monitoring Tool can provide performance reporting for a Tableau Server deployment serving up to 4,000 views per hour. Above 4,000 views per hour, customers may experience delays in analyzing recent Server activity. Additional factors such as Tableau Server Backgrounder load, and the type of drive on the Master Server may also contribute to delays. We recommend solid state drive (SSD) for your Master Server. You can use the Traffic to Views Admin View to understand the peak view load in your deployment, using the following steps:

1. Navigate to **Server Status -> Traffic to views** and make sure the filters are set to look at all sites, workbook, and views.
2. Update the time range to capture all available data.
3. Drill into the time axis on **What is the Total View Count by Day?** to get the view count per hour.

Installing in a Cloud Environment

Resource Monitoring Tool can be installed on a virtual machine in any cloud environment that you may be using for your Tableau Servers. The virtual machine will need to meet the same minimum hardware requirements described above.

Who can do this

To install Resource Monitoring Tool, you must be all the following:

- Administrator on the machine you are installing Resource Monitoring Tool.
- Tableau Server Administrator.
- Resource Monitoring Tool Administrator.

Install the Tableau Resource Monitoring Tool

The Tableau Resource Monitoring Tool consists of a Master Server that is installed on a dedicated machine, and Agents that are installed on every node in the Tableau Server installation you will be monitoring with Resource Monitoring Tool.

The Resource Monitoring Tool prerequisites and the Master Server must be installed on dedicated hardware to guarantee they have the necessary resources, and that there is no contention for machine resources with other programs or software. These should not be installed on your Tableau Server computer, except in exceptional cases (for example, for limited demonstration purposes in non-production environments).

The installer for installing Resource Monitoring Tool can be found on the [Server Management Product and Release Notes site](#). Starting in 2020.2, a single installer will install the prerequisites and the Master Server. Download all the files that start with **Tabrmt**. Starting in 2020.2, you will need only two installers: One for Master Server that includes prerequisites, and one for installing the Agent.

Tableau Server on Linux Administrator Guide

Before Installing the Resource Monitoring Tool

- Review the following requirements and instructions:
 - Minimum Hardware Requirements and Recommendations for Tableau Resource Monitoring Tool
 - Pre-Installation Checklist - Tableau Resource Monitoring Tool
 - The following OS is supported:
 - Red Hat Enterprise Linux (RHEL) 7.3 and higher (not 8.x) and Amazon Linux 2
 - CentOS 7.3 and higher (not 8.x)
 - The latest versions of Ubuntu 16.04 LTS and 18.04 LTS (not 17.04)

Additional notes on Linux distributions:

- Red Hat Enterprise Linux (RHEL), CentOS, Oracle Linux, and Amazon Linux distributions are collectively referred to in this documentation as RHEL-like.
- RHEL 8 is not supported.
- Non-LTS releases of Ubuntu are not supported.
- Ubuntu version 17.04 is not supported.
- Previous versions of CentOS and Ubuntu are not supported because Tableau Server requires systemd for process management.
- The version of the installer with the file suffix, .deb, installs on Ubuntu distributions.

As of July 2022, Debian distributions are no longer supported. For more information, see [this Tableau Community post](#).
- Custom kernels are not supported.
- Make sure the Tableau Server installation you are going to be monitoring is licensed with the Advanced Management license. You must have a valid Advanced Management license to use the Resource Monitoring Tool.
- Review the size of Tableau Server logs. Once Agent is installed and configured, the

Resource Monitoring Tool processes relevant historical data from Tableau Server logs before data is displayed. If there is a large amount of historical log data, it may take a while to process the information which in turn might result in a delay of processing newer events on the Server.

If you are concerned about the delay, and not having historical information does not concern you, you can do the following to clean up the existing files:

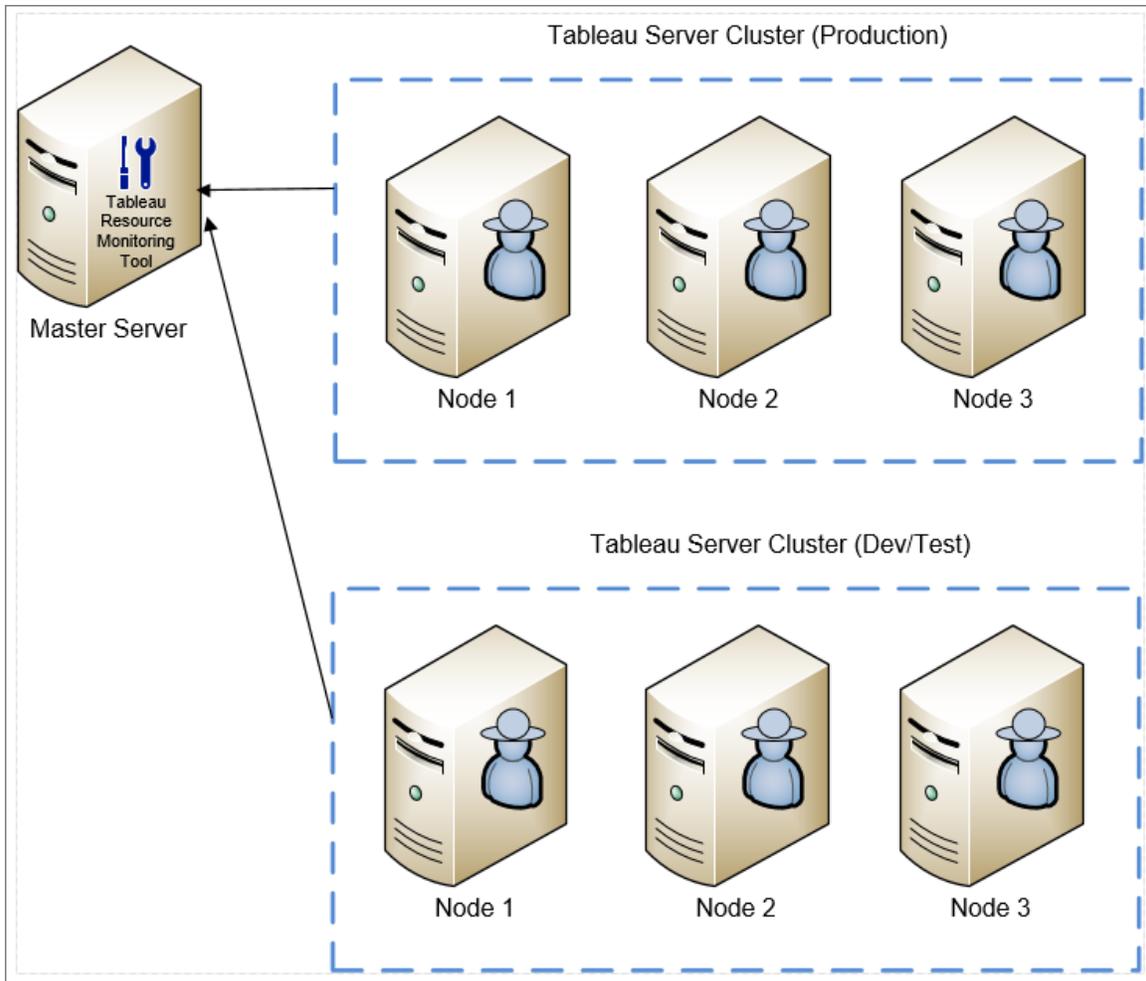
- Remove Unneeded Files, and consider Log File Snapshots (Archive Logs) before you remove log files.

Note: Performance data like CPU usage and memory usage are not gathered using historical log data and are collected after Agent is installed and configured so cleaning up historical data does not affect performance data.

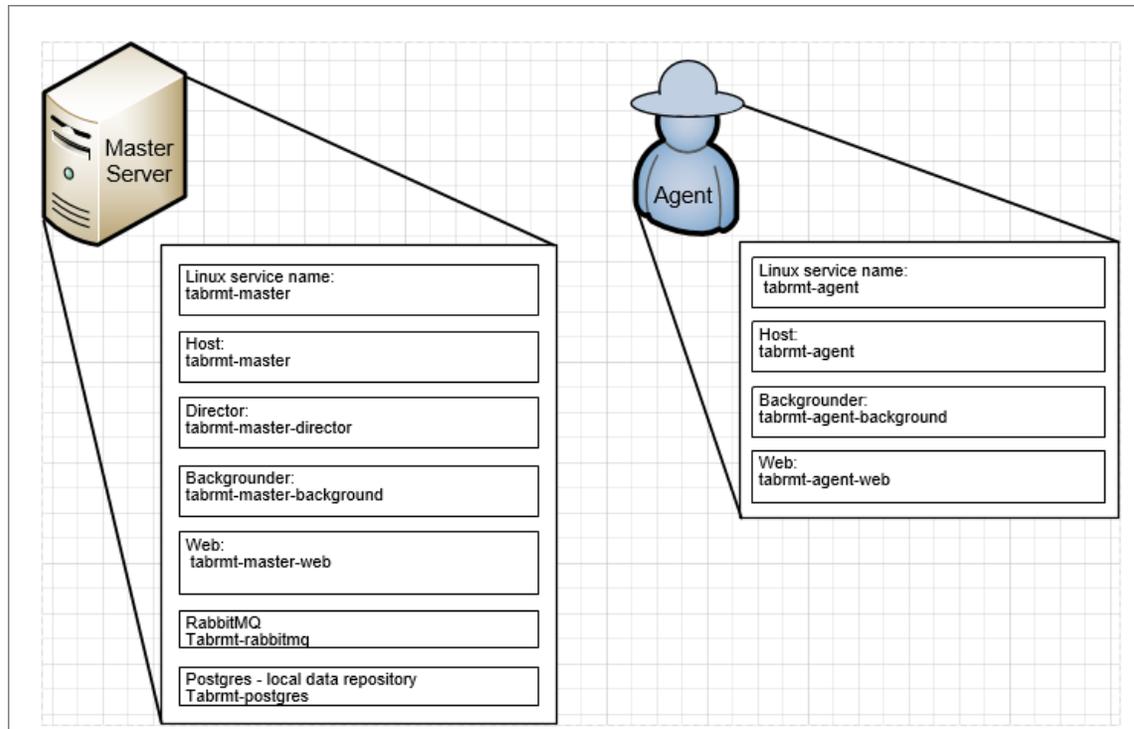
- Optional: Enable the readonly built-in user for Tableau Server Repository. This user is used when installing the Resource Monitoring Tool Agent. For more information, see [Collect Data with the Tableau Server Repository](#).

Setup Architecture

A high level view of the Resource Monitoring Tool setup. This diagram shows two Tableau Server clusters monitored by the Master Server.



A detailed view of the Resource Monitoring Tool components installed on the Master Server and Agent.



Installation Overview

Starting in version 2020.4, the Resource Monitoring Tool can be installed on Windows or Linux machines. The Master Server can be installed on either operating system and does not depend on the operating system that Agent is installed on. This means you can install Master and Agent on different operating systems.

However, Agent is installed on the Tableau Server Cluster nodes, hence it follows that the operating system choice for the Agent depends on the Tableau Server installation.

You can monitor multiple Tableau Server clusters using the same Resource Monitoring Tool. Each Tableau cluster should be setup as a separate environment.

Installation on Linux

You can only install Resource Monitoring Tool Master Server and Agent using the command line on Linux operating systems.

Here are the steps to installing Resource Monitoring Tool:

1. Install the Master Server [using command line](#).

When you start the Resource Monitoring Tool installer, the setup program will install certain programs required to run Resource Monitoring Tool first. The programs installed are RabbitMQ, Erlang, and a dedicated PostgreSQL database. It will then proceed to install the Master Server.

2. Install the Agent [using command line](#).

Install the Agent on every one of your Tableau Server nodes. The Agent sends information about Tableau Server usage and performance to the Master Server for reporting.

HTTPS

As a best practice, you should use HTTPS to protect sensitive information and user credentials.

The **Require HTTPS** setting in the Server configuration is used for communications between the users and the Master Server. It is also used when you register an Agent. Regular agent communications between Agent and Master to collect data is done through Rabbit MQ.

Initially, the Resource Monitoring Tool Master Server is installed with a self-signed certificate and will use that certificate for HTTPS communication which includes communication during Agent registration. You can use your own certificate to replace the self-signed certificate. This can be done during Master Server install in the Server Configuration page or after the installation is complete.

SSL Certificate Mode and Requirements

The resource monitoring tool supports the following modes of using SSL Certificates:

- **Default:** This mode uses the default self-signed certificate supplied by the installer.
- **Local:** This mode allows you to specify a file-based certificate in the `/var/opt/tableau/tabrmt/master/config` folder.

Follow these guidelines and requirements for your certificate:

- You must have a HTTPS certificate (like X.509) for the appropriate domains. This depends on your local security policies and certificate requirements. For example, if the Resource Monitoring Tool is using a CName or SSL passthrough proxy then you might need to use a SAN certificate. For multiple sub-domains, wildcard certificates are supported.
- The Resource Monitoring Tool supports only PKCS #12 and PEM formats.
- The Resource Monitoring Tool web server requires a certificate and a private key, and optionally chain-of-trust.

The private key can be either RSA or DSA.

These can be provided in a single file or grouped files.

- Single file examples:
 - PKCS #12: A single file with the .pfx or .p12 file extensions.
 - PEM: PEM-encoded certificate + private key (plus optionally intermediate CAs chaining up to root CA), in a single file with the .pem extension. The items in the file does not have to be in any specific order.
- Grouped file examples:
 - PEM-encoded certificate in a .crt or .cer file PLUS
 - PEM-encoded private key in a .key file PLUS (optionally)
 - PEM-encoded certificate authority in one or more .ca files

Default File and Directory locations:

Master:

- Installation directory: `/opt/tableau/tabrmt/master`
- Configuration file: `/var/opt/tableau/tabrmt/master/config.json`
- Logs: `/var/opt/tableau/tabrmt/master/logs`

Agent:

- Installation directory: `/opt/tableau/tabrmt/agent`
- Bootstrap: `/var/opt/tableau/tabrmt/agent/bootstrap`
- Logs: `/var/opt/tableau/tabrmt/agent/logs`

Tableau Server on Linux Administrator Guide

Who can do this

To install Resource Monitoring Tool, you must have all the following:

- User account with full sudo access.
- Tableau Server Administrator site role.
- Resource Monitoring Tool Administrator account.

Next Step

Install the Master Server:

- [Using web interface](#)
- [Using command line](#)

Install the Master Server Using Web Interface

The Resource Monitoring Tool Master Server hosts the web application that users interact with. It also does much of the background processing to collate and monitor the data from the Agents. The Master Server must be installed on dedicated hardware.

Installation through web interface is currently only supported for Windows Installations. If you want to install on Linux, see [Install the Master Server Using Command Line](#).

To install the Master Server:

1. Run the Master Server setup program.
2. After reading the EULA, select **I agree to the license terms and conditions**, and click **Install**.

Note: If you are planning to install to a non-default location, use the guidelines provided in the [Installing to a Non-Default Location](#). The default location is **C:\Program Files\Tableau\Tableau Resource Monitoring Tool**.

3. If the User Account Control dialog opens, click **Yes** to allow the installer to make changes.
4. The Master Server installer will first install certain prerequisites as the first step. The prerequisites include RabbitMQ, Erlang, and a PostgreSQL database. The PostgreSQL database is used to store usage data gathered from Tableau Server. It will then proceed to install the Master Server.
5. At the end of the installation, a **Server Configuration** web page opens.
6. The page should already be filled in. Make any updates to the values if needed.

Note: Make a note of the **Host Name**. The web interface used to access the Master Server uses this format: `https://<hostname>`.

7. By default, Resource Monitoring Tool uses a self-signed certificate to use for the HTTPS communications with the Master Server. To leverage your own certificate, replace the thumbprint value with the one for the certificate you want to use. Check the **Require HTTPS** option if you want to mandate secure HTTPS communications.

Checking the **Require HTTPS** option will also require you to choose a certificate mode and provide additional details if necessary. Mainly there are three options to choose from:

1. **Default:** This mode uses the default self-signed certificate supplied by the installer.
2. **Local:** This mode allows you to specify a file-based certificate in the Resource Monitoring Tool **/config** folder. When **Local** mode is selected, the **Certificate Name** field becomes available. The options listed will correspond with the certificate file groups located in the **/config** folder.
3. **Store:** This mode allows you to enter the thumbprint of a certificate in the Windows certificate store.

The **Password** field will only be used if the selected certificate requires a password.

8. Test the Server Configuration by clicking **Test Server Configuration**.
9. Test the connection to the dedicated PostgreSQL database by clicking **Test Database Connection**.

A **Success** message lets you know all the information is correctly entered.

Resource Monitoring Tool accesses the repository database directly for performance reasons. For this to work, access to the repository must be enabled, with a password set for the **readonly** database user. For details, see [Enable access to the Tableau Server repository](#).

10. Test the connection to the message queue by clicking **Test Message Queue Connection**.

A **Success** message lets you know all the information is correctly entered.

Note: The Enable TLS setting under the Message Queue section allows you to enable encryption when data is transmitted between the Master Server and the Agents. It requires additional RabbitMQ setup. For more information, see [Encrypted Data Collection](#).

There is a section at the bottom of this page that allows you to configure the password requirements that will apply to the user accounts and will be applied to the Administrative user account you will create a little later.

11. Click **Save and Restart Server**.
12. When the server has restarted, you are prompted to create an administrative user and password.

13. After signing in using the administrative credentials you just created in the previous step, the Resource Monitoring Tool you will see a message that no Environment has been created. Click **Add an environment** to setup a new environment.
14. Create a new environment. Fill in the details of the environment, like the name an identifier. The Tableau Server REST API and the Tableau Server Repository configurations are used to communicate with Tableau Server. The Tableau Server Repository configuration is optional, but is a preferred method to access Tableau Server.

You have the option to configure secure encrypted connection when RMT connects to Tableau Server Repository. In order to use SSL connections between RMT and Tableau Server Repository database, Tableau Server must be configured to use SSL. For more information, see [Configure SSL for Internal Postgres Communication](#).

In the **Tableau Repository Configuration** section:

1. In the **SSL Mode** drop down box, select **Prefer SSL** or **Require SSL** to configure SSL connections to Tableau Repository. Choosing **Disable** means SSL will never be used to make Tableau Server Repository connections.

In the **Prefer SSL** mode, the Resource Monitoring Tool will use SSL in the first attempt, and if that fails the subsequently attempts a non-encrypted connection.

In the **Require SSL** mode, if the SSL connection fails, the connections to Tableau Server Repository will fail entirely. In this case, Tableau Server REST API connections will be used to communicate with Tableau Server.

2. You can choose to either supply the thumbprint that was generated by Tableau Server, or copy the **server.crt** file to the Resource Monitoring Tool Master Server machine. If you choose to copy the certificate file, you don't have to supply the thumbprint. For more information, see [Configure Postgres SSL to Allow Direct Connections from Clients](#).

Tableau Server on Linux Administrator Guide

The screenshot shows the 'Environment Details' and 'Tableau Repository Configuration' sections of the Tableau Server configuration interface. The 'Environment Details' section includes fields for Name (Test Environment), Identifier (Test-Environment), Gateway URL (http://10.00000.111/), Tableau Version (v2021.1), Tableau API Username (tuser), and Tableau API Password (masked). The 'Tableau Repository Configuration' section includes fields for Server (Host or IP Address of Repository Node), Port (8060), Database (workgroup), SSL Mode (Prefer SSL), SSL Certificate Thumbprint (Optional SHA1 certificate hash to validate), Username (readonly), and Password. A 'Test Connection' button is present at the bottom of both sections. 'Cancel' and 'Save' buttons are at the bottom right.

Click **Save**. You will see a new section added to the page - **Agent Configuration**. This section allows you to download the bootstrap file needed to install and configure Agents.

Note: If you make any updates to the Environment configurations, you must click **Save** before downloading the Bootstrap file.

This screenshot shows the 'Agent Configuration' section added to the configuration page. It displays 'Agents Connected: 0/6' and provides instructions: 'To install and configure an agent: 1. Download the agent bootstrap configuration file below and locate your agent installer. The same installer and configuration file can be used to install multiple agents. 2. Run the installer for the new agent(s) and when prompted, provide the configuration file. For more information, view the agent setup documentation.' A 'Download Bootstrap' button is available. The 'Environment Details' and 'Tableau Repository Configuration' sections remain visible in the background.

- 15.
- 16. Download the bootstrap file and save to it a location that can be accessed from the Tableau Server. You should now be ready to install the Agent on Tableau Server nodes.

Installing to a Non-Default Location

Master Server

Tableau recommends using `\Tableau\Tableau Resource Monitoring Tool` as the location for the installing the Master Server . Example non-default location: **D:\Tableau\Tableau Resource Monitoring Tool**.

To choose a non-default location during installation, use the following steps:

1. Run the Master Server installer.
2. On the EULA page, choose **Customize**.
3. Under **Setup Options**, in the **Install location** field, enter the location.
4. Continue with Step 2 of the installation as described [Install the Master Server Using Web Interface](#).

Who can do this

To install Resource Monitoring Tool, you must have all the following:

- Administrator permissions on the machine you are installing Resource Monitoring Tool.
- Tableau Server Administrator site role.
- Resource Monitoring Tool Administrator account.

Next Step

Install the Agent Using the Web Interface

Install the Agent Using the Web Interface

The Agent is a lightweight process that consumes minimal server resources and sends data to the Master Server. Install the Resource Monitoring Tool Agent on each of your Tableau Server nodes. To install and register an Agent, download the Agent bootstrap configuration file and save it to a location that is accessible from the Resource Monitoring Tool Agent nodes.

Installation through web interface is currently only supported for Windows Installations. If you want to install on Linux, see [Install the Agent Using Command Line](#).

Steps to download the Agent bootstrap file

Use the following steps if you have not yet downloaded the bootstrap file from the Master Server.

1. Using the web interface (<https://<hostname>>) on the Master Server, from the **Admin** menu, select **Environments**.
2. The bootstrap file can be downloaded directly from the environment overview tab on the home page.

Note: If you haven't created an environment as part of the Master Server setup, follow steps 12- 15 described in the [Install the Master Server Using Web Interface](#) topic.

3. Save the bootstrap configuration file to a location that is accessible from the Tableau Server nodes where you will be installing Resource Monitoring Tool Agent.

Steps to install Agent

To install the Agent on each of your Tableau Server nodes:

1. Run the Agent setup program.
2. After reading the EULA, select **I agree to the license terms and conditions**, and click **Install**.

Note: If you are planning to install to a non-default location, use the guidelines provided in the [Installing to a Non-Default Location](#). The default location is **C:\Program Files\Tableau\Tableau Resource Monitoring Tool\agent**.

3. If the User Account Control dialog opens, click **Yes** to allow the installer to make changes.
4. When the installer is finished an **Agent Registration** web page opens.

Tip! Sometimes, it may take a while for the web page to open. If the web page fails to open for some reason, use the following URL on the machine that you are installing:

`http://localhost:9002/setup/register`

5. Review the information on this page about the Tableau Server log file size, historical data, and the implications. Once you proceed to the next step, you will no longer see this information.

Once Agent is installed and registered, the Resource Monitoring Tool processes relevant historical data from Tableau Server logs before data is displayed. If there is a large amount of historical log data, it may take a while to process the information which in turn might result in a delay of processing newer events on the Server.

If you are concerned about the delay, and not having historical information does not concern you, you can do the following to clean up the existing files:

Remove Unneeded Files, and consider Log File Snapshots (Archive Logs) before you remove log files.

Note: Performance data like CPU usage and memory usage are not gathered using historical log data and are collected after Agent is installed and configured so cleaning up historical data does not affect performance data.

6. Browse to the location of the bootstrap file you downloaded from the Master Server. Click **Import Bootstrap File**.
7. Once the import is successfully completed, a web page to enter the Server information is displayed. In the **Tableau Server gateway URL** field, enter the URL you use

to access Tableau Server.

Here are some examples on what the URL might look like:

- <https://MarketingServer/> (if you know the server's name)
- <https://10.0.0.2/> (if you know the server's IP address)
- <http://10.0.0.4/> (If your Tableau Server is not enabled to use SSL)

8. Click **Test Tableau Server Connection** to verify the Agent is able to reach Tableau Server.

A **Success** message displays to confirm the Agent can connect to Tableau.

9. The **Master Server URL** field should already be populated from the information in the bootstrap file. This is the URL of the web page used to do administration tasks, monitor performance and other tasks.

Here is an example of what the URL might look like:

- <https://MasterServer/>

10. Click **Test Master Server Connection** to verify the Agent is able to reach the Resource Monitoring Tool Master Server.

A **Success** message displays to confirm the connection to the Master Server works.

11. The **Master Server certificate thumbprint** field should be already be populated using the information in bootstrap file. The bootstrap file you saved should have this and the Master Server URL information. Enter the Master Server user name and password. This user must have the **Server/Environment Management** role.

Click **Get Registration Options**.

This takes you to the **Agent Registration - Message Queue** page.

12. Click **Test Message Queue Connection** to verify the message queue connection is

working.

A **Success** message displays if the connection works.

Note: The Enable TLS setting under the Message Queue section allows you to enable encryption when data is transmitted between the Master Server and the Agents. It requires additional RabbitMQ setup. For more information, see Encrypted Data Collection.

13. Click **Connect to Message Queue**.
14. On the final Agent Registration page the environment section should already be filled out. Verify the information and add any Tableau Server node details.
15. Click **Register Agent** to complete the agent installation and configuration.

You will see an option to disable the web interface. After registration the web interface is no longer necessary for the agent and can optionally be disabled. If you need to re-enable the web interface, use the following command:

```
rmtadmin set server.web.run true
```

For more information on Resource Monitoring Tool commands, see [rmtadmin Command Line Utility](#).

16. You can verify that the Agent is connected by navigating to the Master Server web interface. From the **Admin** menu, select **Environments**, and under environment details you can see the number of Agents that are currently connected.

Installing Agent on Multi-Node Tableau Server

1. Follow the steps described above to install the Agent on the Tableau Server initial node to connect to the Resource Monitoring Tool.
2. In the **Server** section on the final registration page, check the **Primary Server** option.

Tableau Server on Linux Administrator Guide

3. On the Tableau Server additional nodes, install the Agent using the same bootstrap file that you used to install Agent on the initial node and follow the steps described above.
4. On the final registration page, in the **Environment** section, select the environment you created when installing the Agent on the initial node.
5. In the **Server** section, select the **New Environment Server** option. The Primary Server option should remain unselected.
6. The agent registration process automatically adds this node to your existing environment.

Installing to a Non-Default Location

Tableau recommends using `\Tableau\Tableau Resource Monitoring Tool\agent` as the location for the prerequisites. Example non-default location: **D:\Tableau\Tableau Resource Monitoring Tool\agent**.

To choose a non-default location during installation, use the following steps:

1. Run the Agent installer.
2. On the EULA page, choose **Customize**.
3. Under **Setup Options**, in the **Install location** field, enter the location.
4. Continue with Step 2 of the installation as described [Install the Agent Using the Web Interface](#).

Who can do this

To install Resource Monitoring Tool, you must have all the following:

- Administrator permissions on the machine you are installing Resource Monitoring Tool.
- Tableau Server Administrator site role.

Next Steps

Resource Monitoring Tool Master Server Configuration

Install the Master Server Using Command Line

The Resource Monitoring Tool Master Server hosts the web application that users interact with. It also does much of the background processing to collate and monitor the data from the Agents. The Master Server must be installed on dedicated hardware.

This topic describes the steps you can use to install the Resource Monitoring Tool Master Server using command line. Command line installation is supported on both Windows and Linux operating systems.

Install on Linux

To install the Master Server:

1. Install the Master Server:
 1. Download the rpm or .deb Master Server Installer and save it to a location that you can access from the machine where you plan to install the Master Server.
 2. Login to the machine where you want the Master Server installed as a user that has sudo access.
 3. Run the following command to install the Master Server, where <version> is formatted as major-minor-maintenance:

For RHEL like distribution including CentOS:

```
sudo yum install <pathtomasterserverinstaller>/tabrmt-  
master-setup-<version>-x86_64.rpm
```

For Ubuntu distributions:

```
sudo apt install <pathtomasterserverinstaller>/tabrmt-  
master-setup-<version>-amd_64.deb
```

This installs the package and the prerequisites including RabbitMQ, Erlang, and a PostgreSQL database. The PostgreSQL database is used to store usage data gathered from Tableau Server. It will then proceed to install the Master Server.

2. Initialize

1. To accept EULA and initialize, run the following command :

```
sudo /opt/tableau/tabrmt/master/install-scripts/initialize-rmt-master --accepteula
```

The end user agreement (EULA) can be found in the `/opt/tableau/tabrmt/master/docs` folder.

3. Configure the Master Server

1. Run the following command as the `tabrmt-master` user:

```
sudo su --login tabrmt-master  
  
rmtadmin master-setup [options]
```

The configuration options can be supplied either through the command prompt, a configuration file. If you do not supply the options, the default values will be applied except for the administrator password. The administrator user name will be set to `admin` and you will be prompted to provide the password.

Example command including the required password parameter:

```
rmtadmin master-setup --admin-username=<name of the  
administrator user> --admin-password=<administrator  
user password>
```

The following table lists the required and some commonly used options to configure the Master Server. For a full list of the configurations options, see `rmtadmin Command Line Utility`.

Note: Require HTTPS option ensures secure communications between the Master Server and users. When you require HTTPS for communications, you must also select a mode for the certificate that should be

used for these communications. The table below includes the various options. To learn more about these modes and certificates, see [SSL Certificate Mode and Requirements](#)

Option	Required?	Default	Description
admin-pass-word	Yes Password can be supplied in the command line or provide a file with the password to use. If neither is provided, you will be prompted for the password.	n/a	The password for the administrator user.
admin-pass-word-file	No Password can be supplied in the command line or provide a file with the password to use. If neither is provided, you will be	n/a	The file where the password for the administrator user is stored. Note: <code>tabrmt-master</code> user must have access to this file.

Tableau Server on Linux Administrator Guide

Option	Required?	Default	Description
	prompted for the password.		
admin-user-name	No	admin	The username for the administrator user.
http-port	No	80	
require-https	No	False	Redirect http traffic to HTTPS.
https-certificate-mode	No	<p>"Default"</p> <p>Available options:</p> <p>Default</p> <p>Local</p>	<p>The type of certificate search to perform for the HTTPS certificate.</p> <p>Default: This mode uses the default self-signed certificate supplied by the installer.</p> <p>Local: Allows you to specify a file-based certificate in the <code>/var/opt/tableau/tabrmt/master/config</code> folder.</p>
https-certificate-local-name	No	Null	The name of the HTTPS certificate file without the file extension.
	Note: If not specified, the Resource Monitoring Tool is installed with a self-signed cer-		

Option	Required?	Default	Description
	tificate and will use that certificate for HTTPS communications.		
https-certificate-local-password	No	Null	The password to use for the HTTPS certificate.
https-certificate-local-password-file	No	Null	The path to the file containing the password to use for the HTTPS certificate.

4. Create an environment

1. Run the following command to create an environment:

```
rmtadmin create-env --env=<myenvironment> --api-username=<TableauServer API user name> --api-password=<password for the Tableau Server API user account>
```

Configure the environment using the options available for this command. Here are some key configuration options to consider:

- The Tableau Server REST API and the Tableau Server Repository configurations are used to communicate with Tableau Server. The Tableau Server Repository configuration is optional, but is a preferred method to access Tableau Server.

Tableau Server on Linux Administrator Guide

- You have the option to configure secure encrypted connection when RMT connects to Tableau Server Repository. In order to use SSL connections between RMT and Tableau Server Repository database, Tableau Server must be configured to use SSL. For more information, see [Configure SSL for Internal Postgres Communication](#).

The following table lists the some of the common options. To see a full list of options, see [rmtadmin Command Line Utility](#) .

Option	Required?	Default	Description
--name	Yes	n/a	The name of the environment.
--gateway-url	Yes	n/a	URL used to access the Tableau Server gateway.
--version	Yes	n/a	Tableau Server version that this environment will be monitoring.
--api-username	No	Null	User name of the account used to connect to Tableau Server APIs. The user account should be a Tableau Server administrator with access to all Tableau Server sites.
--api-password	No (If you specify the Tableau API user name, you will either provide	Null	Password of the Tableau Server API user account used to connect to Tableau Server APIs.

Option	Required?	Default	Description
	the password, or specify the file path and file that has the password)		
--api-password-file	No	Null	The path to the file and the name of the file containing the password of the Tableau Server API user account.
--repository-server	No	Null	This is the server name for the PostgreSQL database that is installed with Tableau Server
--repository-port	No	Null	The port number of the Tableau Server Repository database.
--repository-username	No	Null	Username used to connect to PostgreSQL database installed with the Tableau Server Repository. Resource Monitoring Tool accesses the Tableau Server Repository database directly for performance reasons. For this to work, access to the repository must be enabled, with a password set for the readonly database user.

Option	Required?	Default	Description
			For details, see Enable access to the Tableau Server repository.
--repository-password	No	Null	<p>Password for the user account used to connect to the PostgreSQL database that is installed with the Tableau Server .</p> <p>Resource Monitoring Tool accesses the Tableau Server Repository database directly for performance reasons. For this to work, access to the repository must be enabled, with a password set for the readonly database user. For details, see Enable access to the Tableau Server repository.</p>
--repository-password-file	No	Null	The path including the file name where the password for the user account used to connect to the PostgreSQL database that is installed with Tableau Server.
--repository-ssl-mode	No	Prefer	<p>Tableau Server Repository SSL Mode:</p> <p>Prefer SSL or Require SSL to configure SSL connections to Tableau Repository.</p> <p>Disable to never use SSL to make</p>

Option	Required?	Default	Description
			Tableau Server Repository connections.
--repository-ssl-thumbprint	No	Null	You can choose to either supply the thumbprint that was generated by Tableau Server, or copy the server.crt file to the Resource Monitoring Tool Master Server machine. If you choose to copy the certificate file, you don't have to supply the thumbprint. For more information, see Configure Postgres SSL to Allow Direct Connections from Clients .

2. Download the bootstrap file to a location that can be accessed from the Tableau Server nodes.

```
rmtadmin bootstrap-file --env=<myenvironment> --filename<The absolute or relative path including the filename>
```

5. **Optional step - only if not using SSD:** The Resource Monitoring Tool is optimized for SSD by default. If you are not using SSD hardware, run the command:

```
sudo /opt/tableau/tabrmt/master/tabrmt-master optimize --no-ssd
```

Install on Windows

To install the Master Server:

1. Run the Master Server setup program

1. Download the Master Server installer and save it to a location that you can access from the machine where you plan to install the Master Server.
2. Open the command prompt as an administrator.
3. Navigate to the location of the Master Server installer and run the exe file using the following command

```
tabrmt-master-<version>-64bit.exe /silent ACCEPTTEULA=1
```

Note: Use the /silent or /quiet switch to run the setup unattended and without displaying any UI or prompts.

The default install folder is usually like this: *C:\Program Files\Tableau\Tableau Resource Monitoring Tool\master*. To provide a different install location run the following command:

```
tabrmt-master-<version>-64bit.exe /silent ACCEPTTEULA=1  
InstallFolder="D:\Tableau Resource Monitoring Tool\  
master"
```

Full list of all the install command properties and switches can be found in [this section](#) below.

This installs the package and the prerequisites including RabbitMQ, Erlang, and a PostgreSQL database. The PostgreSQL database is used to store usage data gathered from Tableau Server. It will then proceed to install the Master Server.

2. Configure the Master Server.

1. Run the following command and provide the options:

```
rmtadmin master-setup [options]
```

The configuration options can be supplied either through the command prompt, a configuration file. If you do not supply the options, the default values will be

applied except for the administrator password. The administrator user name will be set to admin and you will be prompted to provide the password.

Example command including the required password parameter:

```
rmtadmin master-setup --admin-username=<name of the
administrator user> --admin-password=<administrator
user password>
```

The following table lists the required and some commonly used options used to configure the Master Server. For a full list of the configurations options, see [rmtadmin Command Line Utility](#).

Note: Require HTTPS option ensures secure communications between the Master Server and users. When you require HTTPS for communications, you must also select a mode for the certificate that should be used for these communications. The table below includes the various options. To learn more about these modes and certificates, see [Install the Tableau Resource Monitoring Tool](#)

Option	Required?	Default	Description
admin-password	Yes Password can be supplied in the command line or provide a file with the password	n/a	The password for the administrator user.

Tableau Server on Linux Administrator Guide

Option	Required?	Default	Description
	to use. If neither is provided, you will be prompted for the password.		
admin-pass-word-file	No Password can be supplied in the command line or provide a file with the password to use. If neither is provided, you will be prompted for the password.	n/a	The file where the password for the administrator user is stored.
admin-user-name	No	admin	The username for the administrator user.
http-port	No	80	
require-https	No	False	Redirect http traffic to

Option	Required?	Default	Description
			HTTPS.
https-certificate-mode	No	<p>Default</p> <p>Available options:</p> <ul style="list-style-type: none"> • Default • Store • Local 	<p>The type of certificate search to perform for the HTTPS certificate.</p> <p>Default: This mode uses the default self-signed certificate supplied by the installer.</p> <p>Store: This allows you to enter the thumbprint of a certificate in the Windows certificate store.</p> <p>Local: Allows you to specify a file-based certificate <i><installation directory>\config</i> folder. By default this is <i>C:\Program Files\Tableau\Tableau Resource Monitoring Tool\master\config</i>.</p>
https-certificate-store-thumbprint	No	Null	The HTTPS certificate hash/thumbprint to search for in 'store' certificate mode.

Option	Required?	Default	Description
https-certificate-local-name	No	Null Note: If not specified, the Resource Monitoring Tool is installed with a self-signed certificate and will use that certificate for HTTPS communications.	The name of the HTTPS certificate file without the file extension.
https-certificate-local-password	No	Null	The password to use for the HTTPS certificate.
https-certificate-local-password-file	No	Null	The path to the file containing the password to use for the HTTPS certificate.

3. Create an environment

1. Run the following command to create an environment:

```
rmtadmin create-env --env=<myenvironment> --api-user-name=<TableauServer API user name> --api-password=<password for the Tableau Server API user account>
```

Configure the environment using the options available for this command. Here are some key configuration options to consider:

- The Tableau Server REST API and the Tableau Server Repository configurations are used to communicate with Tableau Server. The Tableau Server Repository configuration is optional, but is a preferred method to access Tableau Server.
- You have the option to configure secure encrypted connection when RMT connects to Tableau Server Repository. In order to use SSL connections between RMT and Tableau Server Repository database, Tableau Server must be configured to use SSL. For more information, see [Configure SSL for Internal Postgres Communication](#).

The following table lists the some of the common options. To see a full list of options, see [rmtadmin Command Line Utility](#) .

Option	Required?	Default	Description
--name	Yes	n/a	The name of the environment.
--gateway-url	Yes	n/a	URL used to access the Tableau Server gateway.
--version	Yes	n/a	Tableau Server version that this environment will be monitoring.
--api-username	No	Null	User name of the account used to connect to Tableau Server APIs. The user account should be a Tableau Server administrator with access to all Tableau Server sites.
--api-password	No (If you specify the Tableau	Null	Password of the Tableau Server API user account used to connect to Tableau Server APIs.

Tableau Server on Linux Administrator Guide

Option	Required?	Default	Description
	API user name, you will either provide the password, or specify the file path and file that has the password)		
--api-password-file	No	Null	The path to the file and the name of the file containing the password of the Tableau Server API user account.
--repository-server	No	Null	This is the server name for the PostgreSQL database that is installed with Tableau Server
--repository-port	No	Null	The port number of the Tableau Server Repository database.
--repository-username	No	Null	Username used to connect to PostgreSQL database installed with the Tableau Server Repository. Resource Monitoring Tool accesses the Tableau Server

Option	Required?	Default	Description
			Repository database directly for performance reasons. For this to work, access to the repository must be enabled, with a password set for the readonly database user. For details, see Enable access to the Tableau Server repository.
--repository-password	No	Null	<p>Password for the user account used to connect to the PostgreSQL database that is installed with the Tableau Server .</p> <p>Resource Monitoring Tool accesses the Tableau Server Repository database directly for performance reasons. For this to work, access to the repository must be enabled, with a password set for the readonly database user. For details, see Enable access to the Tableau Server repository.</p>
--repository-password-file	No	Null	The path including the file name where the password for the user account used to connect to the PostgreSQL database that is installed with Tableau Server.

Option	Required?	Default	Description
--repository-ssl-mode	No	Prefer	Tableau Server Repository SSL Mode: Prefer SSL or Require SSL to configure SSL connections to Tableau Repository. Disable to never use SSL to make Tableau Server Repository connections.
--repository-ssl-thumbprint	No	Null	You can choose to either supply the thumbprint that was generated by Tableau Server, or copy the server.crt file to the Resource Monitoring Tool Master Server machine. If you choose to copy the certificate file, you don't have to supply the thumbprint. For more information, see Configure Postgres SSL to Allow Direct Connections from Clients.

2. Download the bootstrap file to a location that can be accessed from the Tableau Server nodes.

```
rmtadmin bootstrap-file --env=<myenvironment> --file-name<The absolute or relative path including the file name>
```

Windows install properties and switches

Switches:

Switch	Description	Comments
<code>/install /uninstall</code>	Run Setup to either install or uninstall Resource Monitoring Tool.	Default is to install, displaying UI and all prompts. If no directory is specified using the <code>InstallFolder</code> property on a fresh install, <code>C:\Program Files\Tableau\Tableau Resource Monitoring Tool\master</code> is assumed. If Resource Monitoring Tool is already installed, Setup will assume the same location as the current installation.
<code>/passive</code>	Run Setup with minimal UI and no prompts.	
<code>/quiet /silent</code>	Run Setup in unattended, fully silent mode. No web interface or prompts are displayed.	Use either <code>/quiet</code> or <code>/silent</code> , not both.
<code>/norestart</code>	Run Setup without restarting Windows, even if a restart is necessary.	In certain rare cases, a restart cannot be suppressed, even when this option is used. This is most likely when an earlier system restart was skipped. For example, if restart was skipped during installation of other software.
<code>/log <log-file></code>	Log information to the specified file and path. By default log files are created in %TEMP% with a naming convention of <code>Tableau_Resource_Monitoring_Tool_<version_code></code> .	If no file location is specified, the log file is written to the TEMP folder - <code>C:\Users\<username>\AppData\Local\Temp</code> . Check this log file for errors after installation. For example: <code><Setup file> /quiet /log="C:\Tableau\Logs\RmtInstall</code>

Properties:

Property	Description	Comments
<pre>InstallFolder- =<path\to\i- installation\directory></pre>	<p>Install to the specified non-default install location.</p>	<p>Specifies the location to install RMT. If not used, RMT is installed to <i>C:\Program Files\Tableau\Tableau Resource Monitoring Tool\master</i>.</p> <p>Example: <Setup file> /silent InstallFolder-r="D:\Tableau\Tableau Resource Monitoring Tool\master"</p>
<pre>ACCEPTTEULA=1 0</pre>	<p>Accept the End User License Agreement (EULA). Required for quiet, silent, and passive install on both initial</p>	<p>If not included when using /passive, /silent or /quiet, Setup fails silently. If included but set to 0, Setup fails.</p>

Property	Description	Comments
	<p>and additional nodes.</p> <p>1 = true, accept the EULA,</p> <p>0 = false, do not accept the EULA.</p>	

Who can do this

To install Resource Monitoring Tool, you must have all the following:

Windows

- Administrator permissions on the machine you are installing Resource Monitoring Tool.
- Tableau Server Administrator site role.
- Resource Monitoring Tool Administrator account.

Linux

- Full sudo access for the user account that is used to install the Agent.
- Resource Monitoring Tool Administrator account

Next Step

Install the Agent Using Command Line

Install the Agent Using Command Line

The Agent is a lightweight process that consumes minimal server resources and sends data to the Master Server. Install the Resource Monitoring Tool Agent on each of your Tableau Server nodes. To install and register an Agent, download the Agent bootstrap configuration file and save it to a location that is accessible from the Resource Monitoring Tool Agent nodes.

This topic describes the steps you can use to install the Resource Monitoring Tool Agent using command line. Command line installation is supported on both Windows and Linux operating systems.

Install on Linux

To install and configure the Agent:

1. Install:

1. Download the .rpm or .deb Agent Installer and save it to a location that you can access from the machine where you plan to install the Agent.
2. Run the following command to install the Agent where <version> is formatted as major-minor-maintenance:

For RHEL like distributions including CentOS:

```
sudo yum install tabrmt-agent-setup-<version>-x86_64.rpm
```

For Ubuntu:

```
sudo apt install tabrmt-agent-setup<version>-amd64.deb
```

2. Initialize:

1. To accept EULA and initialize, run the following command :

```
sudo /opt/tableau/tabrmt/agent/install-scripts/initialize-rmt-agent --accepteula
```

The end user agreement (EULA) can be found in the `/opt/tableau/tabrmt/agent/docs` folder.

3. Download the bootstrap file

Download the bootstrap file to a location that can be accessed from the Tableau Server nodes.

```
rmtadmin bootstrap-file --env=<myenvironment> --filename<The absolute or relative path including the filename>
```

4. Register:

1. Log off and log on as the `tabrmt-agent` user so you can run `rmtadmin` commands which always require that you run as the `tabrmt-agent` user. Also, when you log on again, you create a new session in which group membership changes have taken effect.

```
sudo su --login tabrmt-agent
```

2. Run the following command and provide the path where the bootstrap file is located. Provide a description of the node where the Agent is being installed.

```
rmtadmin register <bootstrap file path\file> --server-name=<Friendly name of machine> --server-description=<server description>
```

Note: The `tabrmt-agent` user defaults to run commands from the base working directory: `/var/opt/tableau/tabrmt/agent`, so you must specify the file path accordingly. For example, if you placed the bootstrap file in the `/var/opt/tableau/tabrmt/agent/bootstrap/` folder as recommended, the file path would be `/var/opt/tableau/tabrmt/agent/bootstrap/<bootstrap_file_name>`.

The following table lists the configuration options used to register the Agent:

Option	Required?	Default	Description
bootstrap file	Yes	<none>	The location of the bootstrap file.
server-name	No	Host name of machine	Name of the computer that has the Agent Installed. If no option is provided, this field will default to the host name of the machine.
server-description	No	<none>	Description of the computer that has the Agent installed. If no option is provided, this field will remain blank.

Install on Windows

To install and configure the Agent:

1. Download the bootstrap file to a location that can be accessed from the Tableau Server nodes.

```
rmtadmin bootstrap-file --env=<myenvironment> --file-name<The absolute or relative path including the file name>
```

2. Run the Agent setup program:

1. Download the Agent installer and save it to a location that you can access from Tableau Server machines.
2. Open the command prompt as an administrator.
3. Navigate to the location of the Agent installer and run the **exe** file using the following command

```
tabrmt-agent-<version>-64bit.exe /silent ACCEPTTEULA=1
```

Note: Use the /silent or /quiet switch to run the setup unattended and without displaying any UI or prompts.

The default install folder is usually like this: *C:\Program Files\Tableau\Tableau Resource Monitoring Tool\agent*. To provide a different install location run the following command:

```
tabrmt-agent-<version>-64bit.exe /silent ACCEPTTEULA=1
InstallFolder="D:\Tableau Resource Monitoring Tool\agent"
```

3. Register the Agent:

1. Run the following command to register the Agent:

```
rmtadmin register <bootstrap file path\file> --server-
name=<Friendly name of machine> --server-descrip-
tion=<server description>
```

The following table lists the configuration options used to register the Agent:

Option	Required?	Default	Description
bootstrap file	Yes	<none>	The location of the bootstrap file.
server-	No	Host	Name of the computer that has the

Option	Required?	Default	Description
name		name of machine	Agent Installed. If no option is provided, this field will default to the host name of the machine.
server-description	No	<none>	Description of the computer that has the Agent installed. If no option is provided, this field will remain blank.

Windows install properties and switches

Switches:

Switch	Description	Comments
/passive	Run Setup with minimal UI and no prompts.	
/quiet /silent	Run Setup in unattended, fully silent mode. No web interface or prompts are displayed.	Use either /quiet or /silent, not both.

Properties:

Property	Description	Comments
InstallFolder=<path\to\installation\directory>	Install to the specified non-default install location	Specifies the location to install RMT. If not used, RMT is installed to <i>C:\Program Files\Tableau\Tableau Resource Monitoring Tool\master</i> . Example: <Setup

Property	Description	Comments
	ation.	file> /silent InstallFolder= ="D:\- \Tableau\Tableau Resource Mon- itoring Tool\a- gent"
ACCEPTTEULA=1 0	Accept the End User License Agreement (EULA). Required for quiet, silent, and passive install on both initial and	If not included when using /passive, /silent or /quiet, Setup fails silently. If included but set to 0, Setup fails.

Property	Description	Comments
	additional nodes. 1 = true, accept the EULA, 0 = false, do not accept the EULA.	

Installing Agent on Multi-Node Tableau Server

Run the steps described above on each of the nodes of Tableau Server. On the web interface of the Master Server you should be able to see all the nodes where the Agent is installed.

Who can do this

To install Resource Monitoring Tool, you must have all the following:

Windows:

- Administrator permissions on the machine you are installing Resource Monitoring Tool.
- Tableau Server Administrator site role.

Linux:

- Full sudo access for the user account that is used to install the Agent.

Next Step

Configure Tableau Resource Monitoring Tool

Tableau Resource Monitoring Tool Prerequisites - Licenses

Tableau Resource Monitoring Tool contains the following open source applications:

- **Erlang:** version 23.2.6, copyright 2016 Industrial Erlang User Group, Apache 2.0. For more information, see the [Erlang](#) and [Apache 2.0](#) sites.
- **RabbitMQ:** version 3.8.14, copyright , MPL 2.0, Copyright (c) 2007-2021 VMware, Inc. or its affiliates. For more information, see [Mozilla Public License](#) site.

Resource Monitoring Tool contains a single modified RabbitMQ source file, which is available upon request.

- **PostgreSQL:** version 12.5, copyright Portions Copyright © 1996-2021, The PostgreSQL Global Development Group, Portions Copyright © 1994, The Regents of the University of California, PostgreSQL license. For more information, see [PostgreSQL](#) site.
- **OpenSSL:** version 1.1.1k, copyright Copyright (c) 1998-2019 The OpenSSL Project, Copyright (c) 1995-1998 Eric A. Young, Tim J. Hudson, OpenSSL license, double license under both the OpenSSL License and the original SSLeay license. For more information, see [Open SSL License](#).

Upgrading Resource Monitoring Tool

These instructions are for upgrading an existing installation of Tableau Resource Monitoring Tool.

Upgrade Notes

Consider the following before you start the upgrade process:

Tableau Server on Linux Administrator Guide

1. You may need to reboot the machine during an upgrade. You might need to consider doing your upgrades during non-work hours.
2. Sometimes, the Resource Monitoring Tool will modify the database and in such cases, the upgrade process will include a database migration. In case of a database migration, you will see a message banner at the top of the Master Server web page with a time estimate for completing the process. An example screen shot is shown below:

> As part of the upgrade process, the Tableau Resource Monitoring Tool database is being migrated to a new format. Current progress: **42% complete**. [Refresh this page](#) for an updated estimate.

On expanding the banner you can review the details of the impact of the upgrade. Changes are unique to each release and the details shown are specific to the release.

∨ As part of the upgrade process, the Tableau Resource Monitoring Tool database is being migrated to a new format. Current progress: **42% complete**. [Refresh this page](#) for an updated estimate.

During migration:

- Averages for events and content pages that existed before the upgrade will show a null value: "--", until the data aggregation is complete.
- Averages for new events and content pages that occurred after the upgrade will initially show as 0 until the data aggregation is complete.

[Learn More](#)

3. The Resource Monitoring Tool is backward compatible with Tableau Server versions but not forward compatible. This means that Resource Monitoring Tool version should be equal to or greater than the Tableau Server version that you are monitoring.

Tableau recommends that you upgrade the Resource Monitoring Tool first and then follow with Tableau Server upgrade for compatibility reasons. For more information, see [Product Compatibility with Tableau Server](#).

How to Upgrade the Resource Monitoring Tool

1. Log in to the Resource Monitoring Tool in the browser and confirm it is working before starting the upgrade.
2. Copy the new version of the Master Server package on the Master Server.
3. Copy the Agent package to the machines where you have Agents installed. Agents are installed on Tableau Server nodes that you are monitoring.

4. Stop the Resource Monitoring Tool Agent service on all Agents using the following command:

```
rmtadmin stop --agent
```

5. Stop the Resource Monitoring Tool on the Master server using the following command:

```
rmtadmin stop --master
```

6. Once the services have been stopped, it is best practice to check for any Resource Monitoring Tool processes that are running after the services have been stopped: Any with `tabrmt-agent` or `tabrmt-master`. This does not include PostgreSQL or RabbitMQ. You can check the status using the following command:

```
rmtadmin status
```

7. It can take a little time to fully stop all of the Master processes, make sure to wait until there are not any running processes with **tabrmt** in the name before the next step.
8. Run the upgrade commands on the Master Server. This will upgrade the existing version to the new version:

For RHEL like distributions including CentOS:

```
sudo yum install <pathtomasterserverinstaller>/<tabrmt-master-setup-<version>-x86_64.rpm>
```

```
sudo /opt/tableau/tabrmt/master/install-scripts/upgrade-rmt-master --accepteula
```

For Ubuntu distributions:

If you are upgrading from version 2020.4 to 2020.4.1 or later:

Tableau Server on Linux Administrator Guide

```
touch /tmp/tabrmt-master-upgrading.txt && sudo apt install  
<tabrmt-master-setup-<version>-amd_64.deb>
```

```
sudo /opt/tableau/tabrmt/master/install-scripts/upgrade-  
rmt-master --accepteula
```

Note: The touch command is only required when upgrading from 2020.4. If you do not run the touch command before installing the package, your existing version of the Resource Monitoring Tool will be uninstalled before upgrading.

If you are upgrading from version 2020.4.1 to 2020.4.2 or later:

```
sudo apt install <pathtomasterserverinstaller>/<tabrmt-mas-  
ter-setup-<version>-amd_64.deb>
```

```
sudo /opt/tableau/tabrmt/master/install-scripts/upgrade-  
rmt-master --accepteula
```

9. Once the Master Server has been upgraded, upgrade all the Agents by running the following command:

For RHEL like distributions including CentOS:

```
sudo yum install <pathtoagentinstaller>/<tabrmt-agent-  
setup-<version>-x86_64.rpm>
```

```
sudo /opt/tableau/tabrmt/agent/install-scripts/upgrade-rmt-  
agent --accepteula
```

For Ubuntu distributions:

If you are upgrading from version 2020.4 to 2020.4.1 or later:

```
touch /tmp/tabrmt-agent-upgrading.txt && sudo apt install  
<tabrmt-agent-setup-<version>-amd_64.deb>
```

```
sudo /opt/tableau/tabrmt/agent/install-scripts/upgrade-
rmt-agent --accepteula
```

Note: The touch command is only required when upgrading from 2020.4. If you do not run the touch command before installing the package, your existing version of the Resource Monitoring Tool will be uninstalled before upgrading.

If you are upgrading from version 2020.4.1 to 2020.4.2 or later:

```
sudo apt install <pathtoagentinstaller>/<tabrmt-agent-
setup-<version>-amd_64.deb>
```

```
sudo /opt/tableau/tabrmt/agent/install-scripts/upgrade-
rmt-agent --accepteula
```

10. Confirm the Agent and Master services are running. Start the Agent and Master services if they do not automatically restart after the upgrade is complete.
11. You can verify the Agents that have been upgraded. From the **Admin** menu, select Environment and click edit to see the environment details. In the **Server** tab, you can see the version of the Agent. This can be useful to determine which Agents have been upgraded when you a multi-node Tableau Server cluster.

Who can do this

To upgrade Resource Monitoring Tool, you will need to have the following permissions:

- User account with full sudo access.
- Tableau Server Administrator.
- Resource Monitoring Tool Administrator.

Uninstalling Resource Monitoring Tool

The Resource Monitoring Tool installers include uninstallers.

To uninstall the Resource Monitoring Tool, do the following:

Using remove:

- For RHEL like distribution including CentOS:

```
sudo yum remove tabrmt-master
```

- For Ubuntu distributions:

```
sudo apt remove tabrmt-agent
```

The remove script stops and removes all the Tableau Resource Monitoring Tool service from the computer from which you are running the command. It also removes data, Resource Monitoring Tool user accounts and groups. It however, preserves configuration files, logs, and backup files, by moving them to a temporary directory under the `opt/tableau/tabrmt/data/` folder.

You can force remove all the files including the logs and backups using the parameters:

- `-y`: Required.

Removes Resource Monitoring Tool from this computer. Must be specified three times (`-y-y-y`) to confirm.

- `-a`: Required to uninstall agent.

Removes Resource Monitoring Tool Agent if installed.

- `-m`: Required to uninstall master.

Removes Resource Monitoring Tool Master Server.

- `-k`: Optional

Does not copy backups to the `logs-temp` directory

- `-g`: Optional

Does not copy logs to the logs to the `logs-temp` directory.

Using Obliterate to remove all installation files:

If you run into issues when installing using the remove commands, you can use the obliterate script to remove all the installation files from your computer. The information to run the script is described below:

The obliterate script is located at:

```
/opt/tableau/tabrmt/master/install-scripts/tableau-rmt-obliterate
```

During uninstall, the obliterate script is copied to: `/var/tmp/tableau-rmt-obliterate`

- As the root user, run the following command to completely uninstall Resource Monitoring Tool Master Server:

```
/var/tmp/tableau-rmt-obliterate -m -y -y -y
```

- As the root user, run the following command to completely uninstall Resource Monitoring Tool Agent:

```
/var/tmp/tableau-rmt-obliterate -a -y -y -y
```

This should be run on every Tableau Server node where the Agent is installed.

Who can do this

To uninstall Resource Monitoring Tool, you must have all the following:

- Administrator permissions on the machine you are installing Resource Monitoring Tool.
- Tableau Server Administrator site role.
- Resource Monitoring Tool Administrator account.

Configure Tableau Resource Monitoring Tool

This section includes topics that provide information on how to configure Tableau Resource Monitoring Tool to suit your requirements.

Resource Monitoring Tool Master Server Configuration

This topic describes the Master Server configuration options you can set using the web interface.

Post Install Setup Configurations

You can update the configurations you specified during the setup.

Following are the two recommended ways to make configuration changes:

- **To do this using the web interface:** On the machine where Master Server is installed, go to: `http://<hostname>/setup/server`.
- **To do this using command line,** use `rmtadmin master-setup` command with the `--skip-admin-creation` option to make sure you are not prompted to create the admin user post installation. For more information, see `rmtadmin master-setup`.

Note: Configuration values are saved in the **configuration file**. Changes can be made directly to this file, but it is advised to leverage the configuration options in the UI and through the `rmtadmin` command line utility. Changes to the config file will require a restart to be applied.

Here are some examples of the type of updates you might want to do after the initial setup.

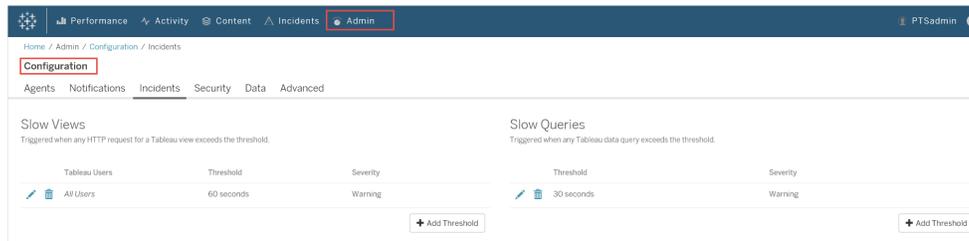
- **Changing SSL Configuration:** The default configuration is set to false. You can change this configuration to require HTTPS. Use the `rmtadmin master-setup` command to update this setting.
- **SSL Certificate Updates:** You might have completed your initial Master Server setup using the default certification mode for secure communications, and want to update the certification with your own certificate.

- **Changes Ports:** Sometimes you will need to use different port numbers than the default based on your network requirements. To learn more about ports and communications between the various components of Resource Monitoring Tool, see [Tableau Resource Monitoring Tool Communication Ports](#).

Note: Adding or removing a Tableau Server node: If you add a new node to the Tableau Server Cluster that you are monitoring using Resource Monitoring Tool, you will need to install and register this node. For more information, see [Tableau Server Topology Changes](#).

To do this through command line, use `rmtadmin-master-setup` command. For more information, see [rmtadmin Command Line Utility](#).

You can configure Resource Monitoring Tool by clicking on “Configuration” under the Admin menu. Configuration sections are divided by tabs. To do this using a configuration file, see [Master Server Configuration File](#).



Notifications

Notifications can be set at a global level by choosing Configuration from the Admin menu and selecting the Notifications tab.

The email configuration can be overridden at the environment level. To specify the notifications for each environment, from the **Admin** menu, select the environment that you want to modify and choose **Edit Environment**. From the **Notifications** tab, choose **Custom** for configuration type to override the global settings. Set the severity level, sender and recipient email information.

Configure the incident notification and SMTP Settings.

Incident Notifications settings:

- **Minimum Severity:** This determines the level at which a notification will be sent. Three options are available: Info, Warning, and Critical.
- **Sender Email:** The email account used to send the email.
- **Recipient Email(s):** The email account(s) that should get this email.

SMTP Server settings:

- **Server:** Name of the SMTP server.
- **Certificate Thumbprint:** This is optional. The SHA1 certificate used if SSL is required.
- **Port:** Port settings for the email server.
- **Username:** The name of the account used to authenticate to the email server.
- **Password:** The password for the account used to authenticate to the email server.

Incident Thresholds

Configure the global incident notification settings. These global thresholds apply to all environments by default unless overridden by the individual environments.

See [Incidents](#) for information on what incident options are available and how to configure them.

Security

Configure the security settings for Resource Monitoring Tool user accounts.

Data

By default, Resource Monitoring Tool stores two weeks of detailed activity data and ten years of aggregated reporting data.

Data Retention: Detailed activity data from your Tableau Server powers the dashboards and incidents for diagnosis of recent performance incidents. Due to the associated storage and processing requirements, this data is only stored for two weeks by default.

Reporting Data: Aggregated activity data from your Tableau Server is stored for reporting purposes (e.g., the [Chargeback](#) report). This data is compact and can easily be stored for many years of historical reporting.

Advanced

Allows configuration of the Resource Monitoring Tool diagnostic logging levels.

See the [Log Files](#) for more information about logging.

Who can do this

Resource Monitoring Tool Administrator or a Resource Monitoring Tool user with **Server-Environment Management** role.

Master Server Configuration File

This topic describes the configuration options you can use using the configuration file. To do this using the web interface, see [Resource Monitoring Tool Master Server Configuration](#) .

The configuration file is located at `/var/opt/tableau/tabrmt/master/config.json`.

Note: Configuration values are saved in the configuration file. Changes can be made directly to this file, but it is advised to leverage the configuration options in the UI and through the `rmtadmin` command line utility. Changes to the config file will require a restart to be applied.

You will need to restart the Master Server after making changes to the configuration file.

Data Retention

By default the Resource Monitoring Tool stores two weeks of detail data and ten years of aggregated reporting data.

This is configurable. Here is an example snippet defining the data retention settings:

```
{
```

```

"db": {
  "cleanup": {
    "afterDays": 14,
    "aggregate": {
      "afterDays": 3650
    }
  }
}
}

```

SMTP Configuration

An example `config.json` snippet defining the SMTP information:

```

{
  "smtp": {
    "server": "localhost",
    "port": 25,
    "username": "my-username",
    "password": "my-password",
    "requireSSL": true
  }
}

```

The full list of configuration values available in the `smtp` configuration section:

Key	Data Type	Required?	Description
<code>server</code>	String	Optional	The SMTP server to use. Default value: <code>localhost</code>
<code>port</code>	Number	Optional	The SMTP server's port number. Default value: <code>25</code>
<code>username</code>	String	Optional	The username to use

Key	Data Type	Required?	Description
			if authentication is required by the server.
password	String	Optional	The password to use if authentication is required by the server.
requireSSL	Boolean	Optional	true if your SMTP server requires SSL, otherwise false. Default value: false

Notification Configuration

An example `config.json` snippet defining the notification information:

```
{
  "notifications": {
    "email": {
      "from": "sender@domain.com",
      "to": "recipient1@domain.com,recipient2@domain.com"
    }
  }
}
```

Global notification information (as shown above) can be overridden per environment in the `environments` section:

```
{
  "environments": {
    "Environment1": {
      "notifications": {
        "email": {
          "from": "sender@domain.com",
          "to": "recipient1@domain.com,recipient2@domain.com"
        }
      }
    }
  }
}
```

```

    }
  }
}
}

```

The full list of configuration values available in the `notifications.email` configuration section:

Key	Data Type	Required?	Description
<code>from</code>	String	Required	The sender's email address.
<code>to</code>	String	Required	The recipient email address (es). Multiple addresses must be separated by commas or semi-colons.
<code>minimumIncidentSeverity</code>	String	Optional	The minimum incident severity at which emails will be sent. Default value: <code>critical</code> . Also see Tableau Resource Monitoring Tool - Incidents.

Histogram Configuration

The boundaries used to generate histograms in the web interface are configurable, using an array of values that represent each boundary.

The full list of configurable histograms in the `monitoring.histograms` section:

Key	Data Type	Required?	Description
<code>viewLoadDuration</code>	Array of Numbers	Optional	The histogram boundaries for view loads. Values are in milliseconds. Default value: [1000, 3000, 6000, 10000]
<code>externalDataRequestDuration</code>	Array of Numbers	Optional	The histogram boundaries for external data requests. Values are in milliseconds. Default value: [1000, 3000, 6000, 10000]
<code>backgroundTaskDuration</code>	Array of Numbers	Optional	The histogram boundaries for back-

Key	Data Type	Required?	Description
			ground tasks. Values are in milliseconds. Default value: [60000, 300000, 600000, 1800000]

As an example, to use the following histogram buckets for everything at a global level:

- ≤ 1 second
- > 1 second and ≤ 10 seconds
- > 10 seconds and ≤ 30 seconds
- > 30 seconds

The configuration would look like:

```
{
  "monitoring": {
    "histograms": {
      "viewLoadDuration": [1000, 10000, 30000],
      "externalDataRequestDuration": [1000, 10000, 30000],
      "backgroundTaskDuration": [1000, 10000, 30000]
    }
  }
}
```

Histogram boundaries can also be set per environment. As an example, for an environment whose identifier is “staging-environment” to use the following view histogram buckets:

- ≤ 2.5 seconds
- > 2.5 seconds and ≤ 5 seconds

- > 5 seconds and ≤ 30 seconds
- > 30 seconds and ≤ 1 minute
- > 1 minute and ≤ 10 minutes
- > 10 minutes

The configuration would look like:

```
{
  "environments": {
    "staging-environment": {
      "monitoring": {
        "histograms": {
          "viewLoadDuration": [2500, 5000, 30000, 60000,
600000]
        }
      }
    }
  }
}
```

The “staging-environment” would fall back to the global histogram configuration for background tasks.

Incident Configuration

See [Incidents](#) for information on what incident options are available and how to configure them.

Master Logging

See [Tableau Resource Monitoring Tool Log Files](#).

Agent

The agent service’s configuration file is located at %PROGRAMFILES%\Tableau\Tableau Resource Monitoring Tool\agent\config\config.json. This path may vary depending on your installation.

You will need to restart the agent service after making changes to the config file.

Tableau Server Detection

In almost all situations agents will automatically detect the Tableau Server installation and no configuration is needed beyond the standard [agent setup process](#). If desired, you can however manually configure the Tableau Server information through the `config.json` file.

An example `config.json` snippet defining the Tableau Server information needed to run the agent:

```
{
  "agent": {
    "tableauServer": {
      "override": true,
      "productVersion": 9.3,
      "applicationDirectory": "C:\\Tableau\\Tableau
Server\\9.3"
    }
  }
}
```

Backslashes should be escaped as shown above for the JSON format.

The full list of configuration values available in the `agent.tableauServer` configuration section:

Key	Data Type	Required?	Description
<code>applicationDirectory</code>	String	Required	The Tableau Server application root directory. E.g., <code>C:\Program Files\Tableau\Tableau Server\9.3</code> contains subdirectories like <code>admin</code> , <code>apache</code> , and <code>bin</code> .

Key	Data Type	Required?	Description
<code>dataDirectory</code>	String	Optional	The directory to get Tableau Server data files. E.g., <code>C:\ProgramData\Tableau\Tableau Server\data\tabsvc</code> - contains subdirectories <code>backgrounder</code> , <code>cacheserver</code> , <code>config</code> , and <code>logs</code> .
<code>override</code>	Boolean	Optional	<code>true</code> to have the configuration values take precedent over any automatically detected values. <code>false</code> to have the configuration values act as a fallback to the automatically detected values.
<code>productVersion</code>	Number	Required	The version number of the Tableau Server. E.g., <code>9.3</code> .

Agent Logging

See Tableau Resource Monitoring Tool Log Files.

Common

Common configuration values are available in both master server and agent applications. See application-specific sections for guidance on locating the configuration file.

You will need to restart the application service after making changes to the config file.

Encrypted Messaging

To enable encrypted messaging, the RabbitMQ server must be first configured to allow TLS. See the [Encrypted Data Collection](#) administrator guide for more details.

When configuring the master server or agent(s) for encrypted messaging:

Tableau Server on Linux Administrator Guide

- Both the `enabled` flag and the `certificateHostName` must be configured for encryption to be enabled.
- The `certificateHostName` variable **MUST** match the canonical name (CN=) on the server certificate or the connection will fail.
- The `port` setting in the `mq` section will likely need to be changed based on the TLS port you configured RabbitMQ.

```
{
  "mq": {
    "port": 5671,
    "tls": {
      "enabled": true,
      "certificateHostName": "foo"
    }
  }
}
```

The full list of configuration values available in the `mq.tls` configuration section:

Key	Data Type	Required?	Description
<code>enabled</code>	Boolean	Optional	<code>true</code> enables TLS encryption for messaging connections. <code>false</code> uses unencrypted connections for messaging. Default value: <code>false</code>
<code>certificateHostName</code>	String	Optional	<code>certificateHostName</code> MUST match the canonical name (CN=) of the server certificate or the connection will fail. Default value: ""

Who can do this

Resource Monitoring Tool Administrator or a Resource Monitoring Tool user with **Server-Environment Management** role.

Tableau Resource Monitoring Tool Communication Ports

This article lists the communication ports the Tableau Resource Monitoring Tool uses, and how to change these ports when your network requirements needs them to be different than what is set by default.

Master Server

The lists describe the default port configuration. You can change or update the port information using:

- `rmtadmin master-setup` command
- Web interface: From the machine where Master Server is installed, go to: *<http://localhost/setup/server>*

Master Server Communications

Ports used: 9001, 443, and 80:

- Setup initially defaults to port 9001, but changes the default to 80 after initial configuration.
- The SSL self signed certificate on initial installation is for port 443. Master Server listens on port 80 by default and will attempt to forward traffic to 443 for SSL. Any user specified certificate will also be for port 443.
 - The certificates are used during user traffic to the Master Server and when the Agent is registered using the Web interface
 - We generally do not recommend changing port 443. If however, you need to change this due to your environment restrictions, you will need to explicitly list the port number when connecting to Master Server. If you are registering the Agent using the web interface, you must explicitly specify the port number to connect to the Master Server.
- Port 80 is used for non SSL traffic and can be changed either during initial setup or post installation.

Tableau Server on Linux Administrator Guide

- The Master Server will reach out directly to the Tableau Server gateway for REST API calls
- The Master Server will reach out directly to the Tableau Server repository to query information if PostgreSQL info is supplied (optional)

Note: Currently the communication between Master Server and Tableau Server repository cannot be configured to use SSL.

RabbitMQ

RabbitMQ is the message que the Agent uses to send information to the Master Server.

Ports used: 5672

- RabbitMQ is installed and listens to port 5672. This cannot be changed.
- TLS communication for RabbitMQ can be setup but not through the Resource Monitoring Tool. For more information see the [Rabbit MQ documentation](#).

PostgreSQL Database

Ports used: 5555

- PostgreSQL database is installed and listens to port 5555 and cannot be changed.
- PostgreSQL has an admin users that is used by the Resource Monitoring Tool. The readonly user is used to download the TDS files. For more information on downloading the TDS files, see [Explore Monitoring Data Using Tableau Data Source Files](#).
- PostgreSQL requires SCRAM-SHA-256 authentication. For more information, see [Explore Monitoring Data Using Tableau Data Source Files](#).

Agent

Ports used: 9002, 443, 5672

- During installation, the Agent installer opens a web browser that is only used for registration that listens on port 9002.
- If Agent registration is done using the web interface, the registration process uses port 443. If the registration is done using the command line, the registration process goes through RabbitMQ and uses port 5672.
- Agent sends log files and WMI information through RabbitMQ over port 5672.

Server Roles in Tableau Resource Monitoring Tool

The Tableau Resource Monitoring Tool has various server roles that you can assign to a user to give them permissions to do certain tasks.

When you install Resource Monitoring Tool, during configuration, you create an admin user that has permissions to do all tasks on the Master Server by default.

The following table lists the available roles and what each role allows the user to do:

Server Roles	Role Description
Server/Environment Management	Has permissions to update all configurations.
User Management	Create and edit users.
Download Log Bundles	Download log files.
Download TDS Files	Download .tds files.
Generate Chargeback Reports	Create chargeback reports.

Create a New User

To create a new user in Resource Monitoring Tool, from the **Admin** menu, select Users. Click New User. Provide a user name, password, and apply the server roles. Click Save.

Tableau Resource Monitoring Tool - Incidents

Incidents

There are two types of incidents for the Resource Monitoring Tool.

1. Incidents that are built-in and are set by default - these cannot be configured. When the event occurs, the incident is reported under the **Incidents** tab:

Incident	Incident Level	Description
Agent Down	warning	An incident is logged and reported when either one or more Agents are down, and the Master Server is unable to communicate with the Agent. By default, an email will be sent to the email specified in the Notifications tab.
Agent Unlicensed	critical	This can happen if Tableau Server is not properly licensed to use Tableau Resource Monitoring Tool, or due to connection issues. For more information on the possible causes, see Agent Incidents By default, an email will be sent to the email specified in the Notifications tab.
Environment Down	critical	An incident is logged and reported when Tableau Server is offline. By default, an email will be sent to the email specified in the Notifications tab.
Extract Failure	warning	An incident is logged and reported when there is an extract failure in Tableau Server. By default, an email will be sent to the email specified in the Notifications tab.

2. Incidents that can be configured by you:

Incidents can be configured and used to help detect when unusual situations occur. Incidents can be configured for a variety of events for both Tableau Server and the server it is running on. Following are the events that you can configure incident thresholds for:

- Hardware: CPU Usage, Memory Usage, Memory availability, Disk Space, and Disk queue length.
- Slow queries
- Slow views

Incidents can be configured either globally to apply to all environments, or can be customized for each environment.

Global incidents can be configured by clicking on **Configuration** under the **Admin** menu, and clicking on the **Incident Thresholds** tab. Global configurations are applied to existing environments or any new environments created, unless the environment is using custom thresholds.

Incidents may also be overridden per environment. Do the following to set a specific environment to use custom thresholds:

1. Choose **Environments** under **Admin** menu and click edit on the environment that you want to use custom thresholds.
2. At this time the environment has all the global thresholds.
3. Navigate to the **Incident Thresholds** tab, and click the **Use Custom Thresholds** option at the top of the page. This will allow you to create new thresholds or edit the existing thresholds. In addition, from this point forward new or updates to global thresholds will not be automatically applied to this environment.

Incident Severity Levels

Incidents are configured with different levels of severity.

Severity Level	Key	Description
Info	<code>information</code>	The lowest severity. By default, an email will not be sent when an incident at this level is discovered. Also, the environment status will not be effected.
Warning	<code>warning</code>	Incidents at this level will, by default, send a notification email when triggered. Also, the environment status will be effected.
Critical	<code>critical</code>	The highest severity. By default, a notification email will be sent when triggered. Also, the environment status will be effected.

To change when emails are sent for incidents using the **web interface**, see [Resource Monitoring Tool Master Server Configuration](#)

To change when emails are sent for incidents using the **configuration file**, see Master Server Configuration File.

Environment Down Incidents

Environment Down incidents will be logged as critical when we detect that your Tableau Server is offline. Following are couple of use cases where Tableau Server is considered offline by Tableau Resource Monitoring Tool:

- Resource Monitoring Tool is unable to get a response from Tableau Server. Resource Monitoring Tool will retry three times before the incident is logged.
- If Resource Monitoring Tool detects that a process is down across all the nodes, an environment down incident will be logged.

Resource Monitoring Tool polls Tableau Server's `http://{Tableau-ServerUrl}/admin/systeminfo.xml` page every 30 seconds (by default) to check the status. If the 30 second polling interval is not sufficient, increase the interval to 60

seconds. If the status is offline, or does not reply within the set value for three consecutive polling intervals then a critical incident is created.

Note: The polling interval cannot be set using the `rmtadmin set` command. This setting can only be changed by editing the `config.json` file."

An example `config.json` snippet is below:

```
{
  "background": {
    "run":true,
    "loglevel":"Information",
    "pollRates": {
      "processStatus": 30000
    }
  }
}
```

Key	Data Type	Required?	Description
<code>processStatus</code>	Number	Optional	The amount of time in milliseconds to pause between Tableau Server status polling.

Who can do this

Resource Monitoring Tool Administrator or a Resource Monitoring Tool user with **Server-Environment Management** role.

Agent Incidents

Agents Unlicensed (critical)

Incompatible Agent Version (critical)

Agent Down (warning)

Agents Unlicensed

Agents Unlicensed incidents will be logged as critical when we detect that your Tableau Server has not been properly licensed to use Tableau Resource Monitoring Tool.

To monitor a Tableau Server using the Resource Monitoring Tool your Tableau Server must be licensed with the Tableau Advanced Management.

Agents will no longer collect data in the unlicensed state and send to the Master Server. You can however, continue to see the data that was previously collected when the environment was properly licensed.

To resolve this issue, license Tableau Server with a valid Advanced Management product key. For more information, see [About Tableau Advanced Management](#).

Note: It may take up to an hour after adding a valid license to Tableau Server, for the incident to be cleared.

There are other possible causes that can cause an **Agents Unlicensed** incident:

- Tableau Server API credentials may not be correct. In the environment configuration page, verify that the username and password are correct and that the account used has the Server Administrator role on Tableau Server.
- Unable to connect to Tableau Server REST API. Make sure that REST API is enabled on Tableau Server and that Tableau Server is running.

To resolve the REST API connectivity issue, from the **Admin** menu, select **Environments**, click **Edit Environment** on the environment you want to modify, and in the **Environment Details** tab, click the **Test Connection** button in the **Tableau Server REST API** section on the right side of the page.

- The Resource Monitoring Tool Master Server was unable to connect to Tableau Server, likely due to a network connectivity or similar issue.

Incompatible Agent Version

Agent incompatible incidents will be logged as critical when one or more Agents in the environment are on a version that is not compatible with the Master Server version.

To resolve the issue, you need to make sure that the Agent is on a version that is compatible with the current Master Server version.

Use one of the following to resolve this issue:

We recommend that the Master Server and all Agents be on the same version.

- If the Agent is on a version that is earlier than the minimum compatible version, then you must upgrade the Agent to at least the minimum compatible version that is listed in the incident report.
- If the Agent is on a version that is later than the Master Server version, we recommend that you upgrade the Master Server to that same later version as well. You may need to upgrade other Agents depending on the version that they are currently on.

Upgrading Agents:

Use the following steps to upgrade the Agent:

1. Copy the Agent package to the machines where you have Agents installed. Agents are installed on Tableau Server nodes that you are monitoring. The Agent should be on the same version as the Master Server or use a version that is compatible with the Master Server version. The incident details provides the minimum compatible

version.

2. Stop the Resource Monitoring Tool Agent service on all Tableau Server nodes using the following command:

```
rmtadmin stop --agent
```

3. Upgrade all the Agents by running the following command:

For RHEL like distributions including CentOS:

```
sudo yum install <pathtoagentinstaller>/<tabrmt-agent-setup-<version>-x86_64.rpm>
```

```
sudo /opt/tableau/tabrmt/agent/install-scripts/upgrade-rmt-agent --accepteula
```

For Ubuntu and Debian distributions:

If you are upgrading from version 2020.4 to 2020.4.1 or later:

```
touch /tmp/tabrmt-agent-upgrading.txt && sudo apt install <tabrmt-agent-setup-<version>-amd_64.deb>
```

```
sudo /opt/tableau/tabrmt/agent/install-scripts/upgrade-rmt-agent --accepteula
```

Note: The touch command is only required when upgrading from 2020.4. If you do not run the touch command before installing the package, your existing version of the Resource Monitoring Tool will be uninstalled before upgrading.

If you are upgrading from version 2020.4.1 to 2020.4.2 or later:

```
sudo apt install <pathtoagentinstaller>/<tabrmt-agent-setup-<version>-amd_64.deb>
```

```
sudo /opt/tableau/tabrmt/agent/install-scripts/upgrade-  
rmt-agent --accepteula
```

4. Confirm that the Agent is running and has been upgraded. You can view Agent registration status by navigating to **Admin -> Environments -> Edit the Environment -> servers** tab to see a full list of Tableau Server nodes and the status of the Agent.

Upgrading Master

Upgrade the Master if one or more Agents are on a later version.

Use the following steps to upgrade Master:

1. Copy the new version of the Master Server package on the Master Server.
2. Stop the Resource Monitoring Tool on the Master server using the following command:

```
rmtadmin stop --master
```

3. Stop the Resource Monitoring Tool Agents on all Tableau Server Nodes using the following command:

```
rmtadmin stop --agent
```

4. Once the services have been stopped, it is best practice to check for any Resource Monitoring Tool processes that are running after the services have been stopped: Any with tabrmt-agent or tabrmt-master. This does not include PostgreSQL or RabbitMQ. You can check the status using the following command:

```
rmtadmin status
```

5. Run the upgrade commands on the Master Server. This will upgrade the existing version to the new version:

For RHEL like distributions including CentOS:

Tableau Server on Linux Administrator Guide

```
sudo yum install <pathtomasterserverinstaller>/<tabrmt-master-setup-<version>-x86_64.rpm>
```

```
sudo /opt/tableau/tabrmt/master/install-scripts/upgrade-rmt-master --accepteula
```

For Ubuntu and Debian distributions:

If you are upgrading from version 2020.4 to 2020.4.1 or later:

```
touch /tmp/tabrmt-master-upgrading.txt && sudo apt install <tabrmt-master-setup-<version>-amd_64.deb>
```

```
sudo /opt/tableau/tabrmt/master/install-scripts/upgrade-rmt-master --accepteula
```

Note: The touch command is only required when upgrading from 2020.4. If you do not run the touch command before installing the package, your existing version of the Resource Monitoring Tool will be uninstalled before upgrading.

If you are upgrading from version 2020.4.1 to 2020.4.2 or later:

```
sudo apt install <pathtomasterserverinstaller>/<tabrmt-master-setup-<version>-amd_64.deb>
```

```
sudo /opt/tableau/tabrmt/master/install-scripts/upgrade-rmt-master --accepteula
```

6. Confirm the Agent and Master services are running. Start the Agent and Master services if they do not automatically restart after the upgrade is complete.

Agent Down

Agent Down incidents will be logged as warning when the Tableau Resource Monitoring Tool is unable to communicate with the Agents on Tableau Server.

Use the following steps to troubleshoot this issue:

1. Make sure that the hardware that hosts the Agents is running and available to communicate with the Resource Monitoring Tool Master Server.
2. Make sure that the Agent service is successfully registered, and is running. You can view Agent registration status by navigating to **Admin** -> **Environments** -> **Edit the Environment** -> **servers** tab to see a full list of Tableau Server nodes and the status of the Agent.

Agent polling and incident creation times

- Agent sends a *heartbeat message* to the Resource Monitoring Tool server every 5 minutes.
- The Resource Monitoring Tool server checks to see if the most recent *heartbeat message* from Agent is less than 15 minutes old.
 - If the latest *heartbeat message* received from the Agent is less than 15 minutes old, then the Agent is considered to be online and no incident is created, and any existing **Agent Down** incidents are cleared.
 - If the latest *heartbeat message* received from Agent is more than 15 minutes old, then it is considered to be the first failed attempt. Every minute thereafter, a check is done, two more times. If after three attempts (first attempt and two retries) there is still no recent heartbeat message, then an **Agent Down** incident is created.

Note: Based on the polling interval described above, it takes about 17 (15+1+1) minutes for an **Agent down** incident to be created after the Agent is offline.

Who can do this

Resource Monitoring Tool Administrator or a Resource Monitoring Tool user with **Server-Environment Management** role.

Extract Failure Incidents

Extract Failure incidents will be logged as a warning when there is an extract failure in Tableau.

Who can do this

Resource Monitoring Tool Administrator or a Resource Monitoring Tool user with **Server-Environment Management** role.

Hardware Incidents

Hardware incidents monitor the server itself. These can be used to help identify server issues that may affect Tableau Server's performance.

You can set thresholds for the following issues:

- CPU Usage
- Available Memory
- Memory Usage
- Free Disk Space

The following incidents are configured by default when you install a new Tableau Resource Monitoring Tool:

- When the available disk space falls below 10 GB for 10 minutes or more a warning incident is logged, and when available disk falls below 5 GB for 10 minutes or more, a critical incident is logged.
- When available memory falls below 8 GB for over 10 minutes, a warning incident is logged.
- When the CPU usage for the entire server is 80% or more for 5 minutes, a warning incident is logged.

Note: Memory related incidents are configured in binary multiples of bytes.

You can configure thresholds using the Resource Monitoring Tool Master Server web interface or by updating the configuration file `config.json`.

Use the Master Server web interface

To set the thresholds for hardware incidents, under the **Admin** menu, select **Configuration**, and go to the **Incidents** tab.

For **CPU Usage**, set the following:

To set the thresholds for hardware incidents, under the **Admin** menu, select **Configuration**, and go to the **Incidents** tab.

Key	Required?	Description
Severity	Required	See Incident Severity Level .
Process	Required	The threshold applies to the entire Tableau Server or for a single process as specified.
Start Threshold	Required	The CPU usage must surpass the value specified before an incident is created and monitored. Set the percent and the duration for this threshold.
End Threshold	Optional	The CPU usage that must fall below the value specified before an incident is considered resolved.

For **Available Memory**, set the following:

Key	Required?	Description
Severity	Required	See Incident Severity Level .
Start Threshold	Required	The available memory must fall below the value specified before an incident is created and monitored. Set the percent and the duration for this threshold.
End Threshold	Optional	The available memory must be above the value specified before an incident is considered resolved.

For **Memory Usage**, set the following:

Key	Required?	Description
Severity	Required	See Incident Severity Level .
Process	Required	The threshold applies to the entire Tableau Server or for a single process as specified.
Start Threshold	Required	The memory usage must be equal to the value specified before an incident is created and monitored. Set the percent and the duration for this threshold.
End Threshold	Optional	The memory usage must be below the value specified before an incident is considered resolved.

For **Free Disk Space**, set the following:

Key	Required?	Description
Severity	Required	See Incident Severity Level .
Start Threshold	Required	The free disk space must fall below the value specified before an incident is created and monitored. Set the percent and the duration for this threshold.
End Threshold	Optional	The free disk space must be above the value specified before an incident is considered resolved.

For **Disk Queue Length**, set the following:

Key	Required?	Description
Severity	Required	See Incident Severity Level .
Start Threshold	Required	The disk queue length must be equal to the value specified before an incident is created and monitored. Set the percent and the duration for this threshold.
End Threshold	Optional	The disk queue length must be below the value specified before an incident is considered resolved.

Use the configuration file (config.json)

An example `config.json` snippet defining two hardware incidents:

```
{
  "monitoring": {
    "incidents": {
      "triggers": [
        {
          "counter": "DiskSpaceAvailableKB",
          "severity": "warning",
          "threshold": 1048576
        },
        {
          "counter": "ProcessorTimePercent",
          "severity": "warning",
          "threshold": 0.95,
          "thresholdDuration": 300000,
          "endThreshold": 0.90,
          "endThresholdDuration": 5000
        }
      ]
    }
  }
}
```

- The **DiskSpaceAvailableKB** incident will trigger a warning once the available disk space falls below 10 GB.
- The **ProcessorTimePercent** incident will trigger a warning once the CPU has had at least 95% utilization for over 5 minutes. The incident will be considered resolved once the CPU is below 90% utilization for 5 seconds.

The default settings may or may not meet your requirements, and can be changed based on your environment. As an example, for an environment whose identifier was “staging-envir-

onment” to trigger a warning when the available disk space falls below 2 GBs, the configuration would look like:

```
{
  "environments": {
    "staging-environment": {
      "monitoring": {
        "incidents": {
          "triggers": [
            {
              "counter": "DiskSpaceAvailableKB",
              "severity": "warning",
              "threshold": 2097152
            }
          ]
        }
      }
    }
  }
}
```

Key	Data Type	Required?	Description
counter	String	Required	The identifier for the hardware incident to monitor. available options are: <ul style="list-style-type: none"> • ProcessorTimePercent • DiskSpaceAvailableKB • DiskQueueTotalLength • MemoryAvailableKB • MemoryCommittedKB
severity	String	Optional	See Incident Severity Level . Default value: Warning
threshold	Number	Required	The threshold that must be surpassed

Key	Data Type	Required?	Description
			before an incident is monitored.
<code>thresholdDuration</code>	Number	Optional	The amount of time in milliseconds to monitor the situation before triggering an incident. If not specified, an incident will be triggered as soon as the <code>threshold</code> is reached.
<code>endThreshold</code>	Number	Optional	The threshold that must be surpassed before an incident is considered resolved.
<code>endThresholdDuration</code>	Number	Optional	The amount of time in milliseconds to monitor the situation before completing the incident. If not specified, an incident will be resolved as soon as the <code>endThreshold</code> is reached. If <code>endThreshold</code> is not defined, then <code>threshold</code> is used.

Who can do this

Resource Monitoring Tool Administrator or a Resource Monitoring Tool user with **Server-Environment Management** role.

Slow Query Incidents

Slow Query incidents are triggered when any data query takes too long to complete. By default, a Slow Query incident will trigger a warning if any data query takes at least 30 seconds to execute.

You can configure thresholds using the Master Server web interface or by updating the configuration file `config.json`.

Use the Master Server web interface

To set the thresholds for slow query incidents, under the **Admin** menu, select **Configuration**, and go the **Incidents** tab.

Key	Required?	Description
Severity	Required	See Incident Severity Level .
Duration	Required	The minimum duration for a query to be considered slow. Values are in seconds.

Use the configuration file (config.json)

An example `config.json` snippet defining a Slow Query incident:

```
{
  "monitoring": {
    "incidents": {
      "triggers": [
        {
          "counter": "DataQueryDuration",
          "severity": "critical",
          "threshold": 45000
        }
      ]
    }
  }
}
```

This Slow Query incident will trigger if a data query takes at least 45 seconds to execute.

Incidents can be configured per environment. As an example, for an environment whose identifier was “staging-environment” to trigger a warning when a data query takes longer than 30 seconds to execute, the configuration would look like:

```
{
  "environments": {
    "staging-environment": {
      "monitoring": {
        "incidents": {
          "triggers": [
            {
              "counter": "DataQueryDuration",
              "severity": "warning",
              "threshold": 30000
            }
          ]
        }
      }
    }
  }
}
```

Key	Data Type	Required?	Description
counter	String	Required	Slow Query incidents use the <code>DataQueryDuration</code> identifier.
severity	String	Optional	See Incident Severity Level . Default value: <code>Warning</code>
threshold	Number	Required	The minimum duration for a query to be considered slow. Values are

Key	Data Type	Required?	Description
			in milliseconds.

Who can do this

Resource Monitoring Tool Administrator or a Resource Monitoring Tool user with **Server-Environment Management** role.

Slow Views Incidents

Slow View incidents are triggered when any view request takes too long to load.

By default, a Slow View incident will trigger a warning if any view request takes at least 1 minute to load.

You can configure thresholds using the Master Server web interface or by updating the configuration file `config.json`.

Use the Master Server web interface

To set the thresholds for slow view incidents, under the **Admin** menu, select **Configuration**, and go the **Incidents** tab.

Key	Required?	Description
Severity	Required	See Incident Severity Level .
Duration	Required	The minimum duration for a query to be considered slow. Values are in seconds.
Tableau Users	Required	This threshold is can be either applied to specific users when they make view requests or for all users. Default is All Users .

Use the configuration file (config.json)

An example `config.json` snippet defining a Slow View incident:

```
{
  "monitoring": {
    "incidents": {
      "triggers": [
        {
          "counter": "ViewRequestDuration",
          "severity": "critical",
          "threshold": 45000
        }
      ]
    }
  }
}
```

This Slow View incident will trigger if a view request takes at least 45 seconds to load.

Incidents can be configured per environment. As an example, for an environment whose identifier was “staging-environment” to trigger a warning when a view request takes longer than 30 seconds to load, the configuration would look like:

```
{
  "environments": {
    "staging-environment": {
      "monitoring": {
        "incidents": {
          "triggers": [
            {
              "counter": "ViewRequestDuration",
              "severity": "warning",
              "threshold": 30000
            }
          ]
        }
      }
    }
  }
}
```

```

    }
  }
}
}
}

```

Key	Data Type	Required?	Description
counter	String	Required	Slow View incidents use the ViewRequestDuration identifier.
severity	String	Optional	See Incident Severity Level . Default value: Warning
threshold	Number	Required	The minimum duration for a view request to be considered slow. Values are in milliseconds.
usernames	Array of Strings	Optional	Filters the trigger to only specified users.

```

{
  "monitoring": {
    "incidents": {
      "triggers": [
        {
          "counter": "ViewRequestDuration",
          "severity": "critical",
          "threshold": 10000,
          "usernames": ["vip_jessie","vip_james"]
        },
        {
          "counter": "ViewRequestDuration",
          "severity": "critical",

```

```

        "threshold": 45000
      }
    ]
  }
}

```

Per-User Slow View Triggers

Use the `usernames` field to configure per-user Slow View incidents triggers. Use multiple `ViewRequestDuration` triggers to finely tune when an incident is created.

An example `config.json` snippet defining a Slow View incident using `usernames`:

This Slow View incident will trigger if a view request from either “vip_jessie” or “vip_james” takes at least 10 seconds to load, or if any other user makes a view request that takes at least 45 seconds to load. Only one incident will be created per view request. Triggers with matching usernames are prioritized over triggers without usernames. **Incident Severity Level** is prioritized over everything.

Who can do this

Resource Monitoring Tool Administrator or a Resource Monitoring Tool user with **Server-Environment Management** role.

Encrypted Data Collection

Encrypted communication between the agent(s) and master server is possible by performing the following:

- Configuring RabbitMQ with SSL/TLS certificates.
- Configuring the master server and agent(s) to enable encrypted messaging.
- Configuring encrypted connections to Tableau Repository.

Tableau Server on Linux Administrator Guide

RabbitMQ Setup

For details on RabbitMQ server setup please reference RabbitMQ's documentation for [TLS Setup](#).

Tableau Resource Monitoring Tool Setup

After RabbitMQ has been configured for TLS all client applications: the Tableau Resource Monitoring Tool Master Server and all Agents will need to be configured to enable encrypted messaging.

Tableau Resource Monitoring Tool Setup

After RabbitMQ has been configured for TLS all client applications: the Tableau Resource Monitoring Tool Master Server and all Agents will need to be configured to enable encrypted messaging. Do the following on the Master Server web interface:

1. Navigate to: `http://<hostname>/setup/server`.
2. In the Message Queue section, check the Enable TLS check box, and provide the Certificate Host Name.
3. Update the port information if needed.

When configuring the master server and agent(s) for encrypted messaging:

- Both the `enabled` flag and the `certificateHostName` must be configured for encryption to be enabled.
- The `certificateHostName` variable must match the canonical name (CN=) on the server certificate or the connection will fail.
- The `port` number will likely need to be changed based on the TLS port you configured on RabbitMQ.

Note: If the Agents were already registered before SSL was configured, then you must re-register the agent. To do this, download the new bootstrap file and re-register the Agent using the new bootstrap file. For more information on re-registering the Agent, see [Re-registering an Agent](#).

Tableau Repository SSL Configuration

1. Make sure Tableau Server is configured to use SSL connections for internal Postgres connections. For more information, see [Configure SSL for Internal Postgres Communication](#). The Resource Monitoring Tool allows you to use either the certificate file or thumbprint for the SSL connections. If you plan to use the certificate file, copy the certificate file generated by Tableau Server for internal Postgres SSL connections, to the machine where you plan to install the Resource Monitoring Tool Master Server. For more information, see [Configure Postgres SSL to Allow Direct Connections from Clients](#).
2. On the Master Server web interface, from the Admin menus, select Environments. Click on the edit environment icon.

In the **Tableau Repository Configuration** section:

1. In the **SSL Mode** drop down box, select **Prefer SSL** or **Require SSL** to configure SSL connections to Tableau Repository. Choosing **Disable** means SSL will never be used to make Tableau Server Repository connections.

In the **Prefer SSL** mode, the Resource Monitoring Tool will use SSL in the first attempt, and if that fails the subsequently attempts a non-encrypted connection.

In the **Require SSL** mode, if the SSL connection fails, the connections to Tableau Server Repository will fail entirely. In this case, Tableau Server REST API connections will be used to communicate with Tableau Server.

2. You can choose to either supply the thumbprint that was generated by Tableau Server, or copy the **server.crt** file to the Resource Monitoring Tool Master Server machine. If you choose to copy the certificate file, you don't have to supply the thumbprint. For more information, see [Configure Postgres SSL to Allow Direct Connections from Clients](#).

Who can do this

To configure encrypted data collection you must be both a Tableau Server Administrator and Resource Monitoring Tool.

Tableau Server on Linux Administrator Guide

Hardware Changes to Master Server - Tuning PostgreSQL Database

Sometimes you may need to upgrade or change hardware on the machine that you have the Master Server installed on. Whenever you make any hardware changes, specifically to memory and CPU, it would be beneficial to tune the underlying PostgreSQL database that is installed with the Master Server.

When you install Master Server, the setup program configures the PostgreSQL database to optimize the use of available hardware on the machine. Example optimizations include managing the buffer and cache size. When you change the hardware on your machine, it may affect the performance.

Use the following steps to make sure the configuration is updated to reflect the change in hardware:

1. Connect to the Master Server as a user with complete sudo access.
2. Navigate to the master directory and run the following command:

```
sudo /opt/tableau/tabrmt/master/tabrmt-master optimize
```

You should see a message that indicates that the configuration file has been updated. It will also state that a restart of the PostgreSQL service is required for the changes to take effect. The updates to the configuration file does not require downtime of the server.

3. Restart the PostgreSQL database. You can do this during off-work hours when the Resource Monitoring Tool is not in use.

rmtadmin Command Line Utility

Resource Monitoring Tool includes a command line interface, `rmtadmin.cmd`, located in the installation folder. The default installation folder is:

- Master Server `/var/opt/tableau/tabrmt/master/.`
- Agent: `/var/opt/tableau/tabrmt/agent/.`

Note: You must run these commands as the `tabrmt-master` user:

```
sudo su --login tabrmt-master
```

`rmtadmin` is included in both the **Master Server** and **Agent** installations.

Some commands may vary based on whether you are using the Master Server or Agent `rmtadmin` command line utility. E.g., the `users` command only works from the *Master*. The `ziplogs` command is available everywhere but only includes the log files from the application the command is run on.

Here are the commands that can be used with the `rmtadmin` command line:

Note: The `rmtadmin` commands use both positional parameters and options. The positional parameters should be specified using only the values. You don't need to specify the actual keyword. The option keyword and the value should be specified using an equal sign.

Example:

```
rmtadmin <command> <positional parameter value> --<option keyword>=<value>
```

- `rmtadmin agents`
- `rmtadmin bootstrap-file`
- `rmtadmin cleanup`
- `rmtadmin create-env`
- `rmtadmin delete-env`
- `rmtadmin delete-env-data`
- `rmtadmin delete-server`
- `rmtadmin delete-server-data`
- `rmtadmin deregister-agent`
- `rmtadmin environments`
- `rmtadmin get`
- `rmtadmin help`
- `rmtadmin master-setup`
- `rmtadmin passwd`
- `rmtadmin query`
- `rmtadmin register`
- `rmtadmin restart`

- **rmtadmin-servers**
- rmtadmin set
- rmtadmin start
- rmtadmin stop
- rmtadmin status
- rmtadmin test-env
- rmtadmin update-env
- rmtadmin users
- rmtadmin version
- rmtadmin ziplogs

rmtadmin agents

Note: Added in version 2021.2

Lists all the registered Agents on Tableau Server nodes for all environments.

This is useful to see where the Resource Monitoring Tool Agent is installed on Tableau Server.

This command only works on the Master Server.

Synopsis

```
rmtadmin agents [options][global option]
```

Options

--env

Optional: Use this option to see the list of Agents for the specific environment.

--keys

Optional: includes the agent key in the output.

Example: `rmtadmin agents --env=<myenvironment> --keys`

```
rmtadmin bootstrap-file
```

Note: Added in version 2021.2

Creates and saves the bootstrap file used to register Agents on Tableau Server nodes.

Bootstrap file will be created and saved to the specified absolute path or relative path. Relative paths are resolved to the current working directory.

This command only works on the Master Server.

Synopsis

```
rmtadmin bootstrap-file [options][global option]
```

Options

`--env`

Required. Name of the environment that will be monitoring this Tableau Server. The Agent on a Tableau Server node can only registered to one environment at a time.

`--filename`

Optional. The absolute or relative path including the file name. The relative paths are resolved to the current working directory. If not specified, the default output filename is used.

`--force`

Optional. Overwrites the existing file with the same name.

Example:`rmtadmin bootstrap-file --env=<myenvironment>`

```
rmtadmin cleanup
```

Note: Added in version 2021.2

Deletes the log files that are older than a certain number of days. By default it deletes log files older than 31 days.

This command on both Master Server and Agent.

Note: If the Resource Monitoring Tool processes are running, the active log files will not be cleaned up by the command. If you want to be sure that all files are deleted, you will need to stop RMT using the `rmtadmin_stop` command and then run cleanup using the following command, `rmtadmin cleanup --log-files-retention=0`, and then restart the processes using the `rmtadmin_start` command.

Synopsis

```
rmtadmin cleanup [option][global option]
```

Option

```
--log-files-retention
```

Optional. Deletes the log files older than the specified number of days.

Example: `rmtadmin cleanup --log-files-retention=<number of days>`

```
rmtadmin create-env
```

Note: Added in version 2021.2

Creates a new environment.

This command only works on the Master Server.

Synopsis

```
rmtadmin create-env [options][global option]
```

Options

Option	Required?	Default	Description
--name	Yes	n/a	The name of the environment.
--gateway-url	Yes	n/a	URL used to access the Tableau Server gateway.
--version	Yes	n/a	Tableau Server version that this environment will be monitoring.
--non-interactive	No	Interactive prompts are enabled by default.	Disables all interactive prompts.
--no-test	No	API and Repository connection testing is enabled by default.	Disables testing API and repository connections.
--id	No	System generated based on the name of the environment.	Identifier of the environment used in web interface URLs.
--api-username	No	Null	User name of the account used to connect to Tableau Server APIs. The user account should be a Tableau Server administrator with access to all Tableau Server sites.
--api-password	No	Null	Password of the Tableau Server API user account used to connect

Option	Required?	Default	Description
	(If you specify the Tableau API user name, you will either provide the password, or specify the file path and file that has the password)		to Tableau Server APIs.
--api-password-file	No	Null	The path to the file and the name of the file containing the password of the Tableau Server API user account.
--repository-server	No	Null	This is the server name for the PostgreSQL database that is installed with Tableau Server
--repository-port	No	Null	The port number of the Tableau Server Repository database.
--repository-username	No	Null	<p>Username used to connect to PostgreSQL database installed with the Tableau Server Repository.</p> <p>Resource Monitoring Tool accesses the Tableau Server Repository database directly for performance reasons. For this to work, access to the repository must be enabled, with a password set</p>

Option	Required?	Default	Description
			for the readonly database user. For details, see Enable access to the Tableau Server repository.
--repository-password	No	Null	<p>Password for the user account used to connect to the PostgreSQL database that is installed with the Tableau Server .</p> <p>Resource Monitoring Tool accesses the Tableau Server Repository database directly for performance reasons. For this to work, access to the repository must be enabled, with a password set for the readonly database user. For details, see Enable access to the Tableau Server repository.</p>
--repository-password-file	No	Null	The path including the file name where the password for the user account used to connect to the PostgreSQL database that is installed with Tableau Server.
--repository-ssl-mode	No	Prefer	<p>Tableau Server Repository SSL Mode:</p> <p>Prefer SSL or Require SSL to configure SSL connections to Tableau Repository.</p> <p>Disable to never use SSL to make</p>

Option	Required?	Default	Description
			Tableau Server Repository connections.
--repository-ssl-thumbprint	No	Null	You can choose to either supply the thumbprint that was generated by Tableau Server, or copy the server.crt file to the Resource Monitoring Tool Master Server machine. If you choose to copy the certificate file, you don't have to supply the thumbprint. For more information, see Configure Postgres SSL to Allow Direct Connections from Clients .

Example: `rmtadmin create-env --name=<myenvironment> --api-user-name=<TableauServer API user name> --api-password=<password for the Tableau Server API user account> --gateway-url <Tableau Server Gateway URL> --version <Tableau Server version>`

`rmtadmin delete-env`

Note: Added in version 2021.2

Deletes a specific environment and all data that has been collected for that environment. It also removes the connection and topology information about the Tableau Server that the environment is monitoring and deregisters all the Resource Monitoring Tool Agents.

This command only works on the Master Server.

Synopsis

```
rmtadmin delete-env [positional parameter][options] [global option]
```

Positional Parameter

`env`

Required. The name of the environment you want to delete.

Options

`--confirm`

Optional. Confirm that you want to delete the environment. If you do not provide this option, you will be prompted to confirm.

`--skip-agent-disconnect`

Optional. Skips disconnecting the Agent and continues with deregistering the Agent. Use this option if you think the Agent is inaccessible - For example, if the Tableau Server node has been removed or if the Agent has been uninstalled on that node.

Example: `rmtadmin delete-env <myenvironment>`

`rmtadmin delete-env-data`

Permanently deletes all Tableau Server related data collected for a specific environment. Environment configuration, Tableau Server information, and Agent registration will not be removed.

This is useful for clearing all the existing data in an environment without removing the environment itself. After deleting the existing data, new data sent by the Agents will continue to be processed.

This command only works on the Master Server.

Synopsis

```
rmtadmin delete-env-data [positional parameter][option][global option]
```

Positional Parameter

env

The name of the environment for which the data should be deleted.

Option

--confirm

Optional. Confirm that you want to delete the environment data. If you do not provide this option, you will be prompted to confirm.

Example: `rmtadmin delete-env-data <myenvironment>`

`rmtadmin delete-server`

Note: Added in version 2021.2

Deletes the configuration information of the Tableau Server node from the environment, deregisters the Agent on that node, and deletes all monitoring data collected specific to that node.

This command only works on the Master Server.

Synopsis

```
rmtadmin delete-server [options][global option]
```

Options

`--env`

Required. The name of the environment that the Tableau Server node is connected to.

`--hostname`

Required to delete information about one or more connected to the environment and if **--all** is not specified.

This is the machine name of the Tableau Server node. Use commas to separate multiple values if specifying more than one host name.

`--all`

Optional, but required if **--hostname** is not specified.

Deletes the information for all the Tableau Server nodes connected to the environment.

`--skip-agent-disconnect`

Optional. Skips disconnecting the Agent and continues with deregistering the Agent. Use this option if the Agent is inaccessible- For example, if the node has been removed from Tableau Server or if the Agent has been uninstalled on that node.

`--confirm`

Confirm that you want to delete all the information about the Tableau Server node. If you do not provide this option, you will be prompted to confirm.

Example:`rmtadmin delete-server --hostname=<machine name of the Tableau Server node>`

```
rmtadmin delete-server-data
```

Note: Added in version 2021.2

Deletes the data collected from one or more Tableau Server nodes. Data removed only pertains to the nodes that are specified. If you specify *all*, then all the data collected from all the nodes will be deleted. Environment configuration, Tableau Server information, and Agent registrations will not be deleted.

This command only works on the Master Server.

Synopsis

```
rmtadmin delete-server-data [options][global option]
```

Options

`--env`

Required. The name of the environment that the Tableau Server node is connected to.

`--hostname`

Required to delete information about on or more connected to the environment and if **-all** is not specified.

This is the machine name of the Tableau Server node. Use commas to separate multiple values if specifying more than one host name.

`--all`

Optional, but required if **--hostname** is not specified.

Deletes the monitoring data for all the Tableau Server nodes connected to the environment.

`--confirm`

Confirm that you want to delete all the monitoring data from the Tableau Server nodes. If you do not provide this option, you will be prompted to confirm.

Example:`rmtadmin delete-server-data --all`

`rmtadmin deregister-agent`

Note: Added in version 2021.2

Deregisters the Agent from the environment. Monitoring data from this node will no longer be collected. Existing data will remain and not be deleted.

This command only works on the Master Server.

Synopsis

```
rmtadmin deregister-agent [options][global option]
```

Options

`--env`

Required if `--key` is not specified.

This is name of the environment where the Agent is currently registered.

`--key`

Optional if `--env` is specified.

Tableau Server on Linux Administrator Guide

This is the Agent key. Use commas to separate multiple values. You can get the key values by running the `rmtadmin agents` command.

`--hostname`

Optional. Machine name of the Tableau Server node where the Agent is installed. Use commas to separate multiple values.

`--all`

Optional. Use this option to deregister Agents on all the nodes.

`--ignore-agent-errors`

Optional. Ignores any errors that occur if unable to connect with the Agents during this process.

`--skip-agent-disconnect`

Optional. Skips disconnecting the Agent and continues with deregistering the Agent. Use this option if you think the Agent is inaccessible - For example, if the node has been removed from Tableau Server or if the Agent has been uninstalled on that node.

`--confirm`

Optional. Confirm that you want to deregister the Agents. If you do not provide this option, you will be prompted to confirm.

Example: `rmtadmin deregister-agent --env=<myenvironment> --all`

`rmtadmin environments`

Note: Added in version 2021.2

Lists all the environments on the Resource Monitoring Tool Master Server.

This command only works on the Master Server.

Synopsis

```
rmtadmin environments [global option]
```

```
rmtadmin get
```

Gets the value of a specified configuration key.

This command can be used both on the Master Server and Agent.

Synopsis

```
rmtadmin get <config.key> [global option]
```

Positional Parameter

```
key
```

Name of the configuration key.

Example: `rmtadmin get db:database`

Supported configuration keys

Key	Data Type	Applies To	DEFAULT	Description
<code>db:database</code>	String	Master Server	<code>tabrmtdb</code>	Name of the PostgreSQL database that RMT uses.
<code>db:readOnlyUsername</code>	String	Master Server	<code>readonly</code>	Username of a PostgreSQL

Tableau Server on Linux Administrator Guide

Key	Data Type	Applies To	DEFAULT	Description
				user account which has only read-only access to the RMT data.
<code>db:readOnlyPassword</code>	String	Master Server	Generated by the installer program.	Password of the PostgreSQL read-only user.
<code>mq:tls:certificateHostName</code>	String	Master Server and Agent	n/a	Host name in the certificate to use when connecting to RabbitMQ via TLS.
<code>mq:tls:enabled</code>	Boolean	Master Server and Agent	FALSE	Enable/disable TLS connection to RabbitMQ.
<code>mq:port</code>	Integer	Master Server and Agent	5672	RabbitMQ host port number.
<code>mq:virtualHost</code>	String	Master Server and Agent	tabrmt	Connection to the Master Server or Agent. For Agent, you can modify this setting by

Key	Data Type	Applies To	DEFAULT	Description
<code>server.web.run</code>	Boolean	Agent	TRUE	registering the Agent with a bootstrap file. For the Master the value is managed by the installer. Enable or disable the Agent's web interface.

Use the `set` command to change the current configuration value.

```
rmtadmin help
```

Shows general help about the command line interface and the available commands.

This command can be used both on the Master Server and Agent.

Synopsis

```
rmtadmin help
```

Show help and usage information for a specific command:

This command can be used both on the Master Server and Agent.

```
rmtadmin help [command]
```

```
rmtadmin master-setup
```

Configures the Master Server with specified options. Used both during install and post installation configurations.

Synopsis

```
rmtadmin master-setup [options][global option]
```

The configuration options are listed below:

Option	Required?	Default	Description
<code>--admin-password</code>	Yes (Required for initial installation only) Password can be supplied in the command line or provide a file with the password to use. If neither is provided, you will be prompted for the password.	n/a	The password for the administrator user.
<code>--admin-password-file</code>	Yes (Required for initial installation only) Password can be supplied in the command line or provide a file with the	n/a	The password for the administrator user.

Option	Required?	Default	Description
	password to use. If neither is provided, you will be prompted for the password.		
--admin-user-name	No	admin	The username for the administrator user.
--skip-admin-creation	No This option must be and only used when you making configuration updates post installation.	False	Skip creating the administrator user and password. If this option is not used during post installation configuration updates, the command will fail and no updates will be made.
--http-port	No	80	
--require-https	No	False	Redirect http traffic to HTTPS.
--https-certificate-mode	No	'Default' Available options: <ul style="list-style-type: none"> • Default • Store (Windows only) • Local 	The type of certificate search to perform for the HTTPS certificate. Default: This mode uses the default self-signed certificate supplied by the installer. Store: This allows you to enter the thumbprint of a certificate in

Option	Required?	Default	Description
			<p>the Windows certificate store.</p> <p>Local: Allows you to specify a file-based certificate in the config folder.</p>
--https-certificate-store-thumbprint	No	Null	The HTTPS certificate hash/thumbprint to search for in 'store' certificate mode.
--https-certificate-local-name	No	Null Note: If not specified, the Resource Monitoring Tool is installed with a self-signed certificate and will use that certificate for HTTPS communications.	The name of the HTTPS certificate file.
--https-certificate-local-password	No	Null	The password to use for the HTTPS certificate.
--https-certificate-local-password-file	No	Null	The path to the file containing the password to use for the HTTPS certificate.
--confirm	No	Prompt for confirmation.	Confirms the restart of the master service.

Option	Required?	Default	Description
--host	No Note: We don't recommend changing this value.	Current configuration value or machine name.	The preferred URL for accessing the Resource Monitoring Tool server. The server will listen to all the IPs regardless of the host name.
--db-server	No	Current configuration value or installer default.	This is the server name for the PostgreSQL database that is installed with the Resource Monitoring Tool.
--db-port	No	Current configuration value or installer default.	Port number for the database server.
--db-username	No	Current configuration value or installer default.	Username used to connect to PostgreSQL database installed with the Resource Monitoring Tool. Resource Monitoring Tool accesses the repository database directly for performance reasons. For this to work, access to the repository must be enabled, with a password set for the readonly database user. For details, see Enable access to the

Tableau Server on Linux Administrator Guide

Option	Required?	Default	Description
			Tableau Server repository.
--db-password	No	Current configuration value or installer default.	<p>Password for the user account used to connect to the PostgreSQL database that is installed with the Resource Monitoring Tool.</p> <p>Resource Monitoring Tool accesses the repository database directly for performance reasons. For this to work, access to the repository must be enabled, with a password set for the readonly database user. For details, see Enable access to the Tableau Server repository.</p>
--mq-server	No	Current configuration value or installer default.	The name of the message queue server.
--mq-port	No	Current configuration value or installer default.	The message queue port.
--mq-vhost	No	Current configuration value or installer default.	The message queue virtual host.
--mq-user-	No	Current con-	The user name used to connect to

Option	Required?	Default	Description
name		figuration value or installer default.	the message queue.
--mq-password	No	Current configuration value or installer default.	The password for the user account used connect to the message queue.
--mq-enable-tls	No	Current configuration value or false .	Requires TLS connection to connect to the message queue.
--mq-tls-certificate-host	No	Current configuration value or null .	The canonical name of the message queue server. This must match the name on the certificate.
--password-salt	No	Random	A global salt to use to hash the password. This applies to the local user accounts created in the Resource Monitoring Tool.
--password-min-length	No	Current configuration value or 10 .	The minimum length for the password. This applies to the local user accounts created in the Resource Monitoring Tool.
--password-min-numeric	No	Current configuration value or 1 .	The minimum required numeric characters in the password. This applies to the local user

Option	Required?	Default	Description
			accounts created in the Resource Monitoring Tool.
--password-min-special	No	Current configuration value or 1 .	The minimum required special characters in the password. This applies to the local user accounts created in the Resource Monitoring Tool.
--password-min-latin	No	Current configuration value or 5 .	The minimum required latin characters in the password. This applies to the local user accounts created in the Resource Monitoring Tool.
--password-require-mixed-case	No	Current configuration value or true .	Requires mixed case characters in passwords. This applies to the local user accounts created in the Resource Monitoring Tool.

Examples:

To specify the admin password: `rmtadmin master-setup --admin-password=<password>`

To update the port post install: `rmtadmin master-setup --http-port=8000 --skip-admin-creation`

`rmtadmin passwd`

Resets the password for a specific Resource Monitoring Tool user account.

This command works only on the Master Server.

Synopsis

```
rmtadmin passwd [positional parameter][global option]
```

Positional Parameter

username

Name of the user that you want to change the password for.

Example: `rmtadmin passwd <username>`

rmtadmin query

Executes a raw SQL query against the Resource Monitoring Tool database and saves the results to an output file.

This command works only on the Master Server.

Synopsis

```
rmtadmin query [positional parameter][options] [global option]
```

Positional Parameter

sql

SQL command text to run. You can provide multiple SQL commands. To use a file with the SQL commands, use `@` as a prefix to the name of the file.

Options

`--outfile=VALUE`

Name of the zip file you want to output the query results. Default is *queryresults.zip*

`--force`

Overwrites the existing file.

`--timeout=VALUE`

timeout for the query. Specify this in seconds

`--commit`

Commits any changes made by the SQL command to the database. By default, the SQL command is run as a transaction, but is rolled back at completion.

Example: `rmtadmin query <SQLCommand> --outfile=<path and the output file name>`

`rmtadmin register`

Note: Added in version 2020.2

Registers the Agent using a bootstrap file. The bootstrap file can be downloaded using the [rmtadmin bootstrap-file](#) command.

This command is can only be used on the Agent and should be run on the machine where you want to install the Agent.

Synopsis

```
rmtadmin register [options][positional parameter] [global option]
```

Positional Parameter

bootstrap

The file path including the name of the bootstrap file.

Options

--server-name

Name of the node where Agent is being installed. It defaults to the machine name if no name is specified.

--server-description

Custom description for the Server.

Example: `rmtadmin register <bootstrap file name and path> --server-name=<server name>`

`rmtadmin restart`

Restarts the Resource Monitoring Tool applications. When run from the Master Server it restarts the Master Server application, and when run on a node where the Agent is installed, it restarts just the Agent application on that particular node.

This command can be run from both the Master Server and Agent.

Synopsis

```
rmtadmin restart [options] [global option]
```

Options

At least one option must be specified:

Master Server:

`--all`

Restarts all services.

`--master`

Restarts the Master service.

`--mq`

Restarts the message queue service.

`--db`

Restarts the database service.

Agent:

`--agent`

Restarts the Agent service running on the machine.

Example: `rmtadmin restart --db`

`rmtadmin servers`

Note: Added in version 2021.2

Lists all the Tableau Server nodes across all environments or a specific environment.

This command only works on the Master Server.

Synopsis

```
rmtadmin servers [positional parameter][global option]
```

Positional Parameter

env

Optional. Specify the name of the environment to get a list of Tableau Server nodes registered in that environment.

```
rmtadmin set
```

Sets the value of a specified configuration key.

This command works both on the Master Server and Agent.

Synopsis

```
rmtadmin set [positional parameter] [global option]
```

Positional Parameters

key

The configuration key you want to change the value for.

value

The new value you want to use.

Example: `rmtadmin set mq:port <port number>` where `mq:port` is the key and the `<port number>` is the value.

Supported configuration keys

Key	Data Type	Applies To	DEFAULT	Description
<code>mq:tls:certificateHostName</code>	String	Master Server and Agent	n/a	Host name in the certificate to use when connecting to RabbitMQ via TLS
<code>mq:tls:enabled</code>	Boolean	Master Server and Agent	FALSE	Enable/disable TLS connection to RabbitMQ
<code>mq:port</code>	Integer	Master Server and Agent	5672	RabbitMQ host port number.
<code>server.web.run</code>	Boolean	Agent	TRUE	Enable or disable the Agent's web interface.

Options

`--backup`

Creates a backup of the configuration file before making changes.

Use the `get` command to view the current configuration value.

```
rmtadmin start
```

Starts the Resource Monitoring Tool services. When this is run on the Master Server, it will start the Master Server. When run on an Agent, it will start the Agent on the machine you are running the command from.

This command works both on the Master Server and Agent

Synopsis

```
rmtadmin start [options] [global option]
```

Options

At least one option must be specified:

Master Server:

```
--all
```

Starts all services.

```
--master
```

Starts the Master service.

```
--mq
```

Starts the message queue service.

```
--db
```

Starts the database service.

Agent:

```
--agent
```

Starts the Agent service running on the machine.

Example: `rmtadmin start --all`

```
rmtadmin status
```

Checks the status of the application and running services on the machine you are running this command on. For Master Server, the status report confirms that the master is connected to the database and message queue. For Agents, the status report confirms that the agent is connected to the master.

This command can be run both on the Master Server and the Agent.

This command returns the following:

- Application status (running or not)
- License status
- MQ connection
- Queue details
- Memory used
- Disk space used
- Tableau server version on machine
- Process topology of Tableau Server that is stored in Agent config file.

Synopsis

```
rmtadmin status [global option]
```

```
rmtadmin stop
```

Stops the Resource Monitoring Tool services. When this command is run on the Master Server, it will only stop the Master Server. When run on a machine where Agent is installed,

it will only stop the Agent on that machine.

This command can be run both on the Master Server and the Agent.

Synopsis

```
rmtadmin stop [options] [global option]
```

Options

At least one option must be specified:

Master Server:

`--all`

Stops all services.

`--master`

Stops the Master service.

`--mq`

Stops the message queue service.

`--db`

Stops the database service.

Agent:

`--agent`

Stops the Agent service running on the machine.

Example: `rmtadmin stop --db`

`rmtadmin test-env`

Note: Added in version 2021.2

Tests the various connection points between Resource Monitoring Tool and Tableau Server. This includes testing Tableau Server Repository connection, Tableau Server API connections, and Server Management Add-on licensing.

This command only works on the Master Server.

Synopsis

```
rmtadmin test-env [positional parameter][global option]
```

Positional Parameter

`env`

Required. The name of the environment that you want to run the test on.

`rmtadmin update-env`

Note: Added in version 2021.2

Updates the environment settings.

This command only works on the Master Server.

Synopsis

```
rmtadmin update-env [options][global option]
```

Options

Option	Required?	Description
--non-interactive	No	Disables all interactive prompts.
--no-test	No	Disables testing API and repository connections.
--name	Yes	The name of the environment.
--id	No	Identifier of the environment used in web interface URLs.
--gateway-url	No	URL used to access the Tableau Server gateway.
--version	No	Tableau Server version that this environment will be monitoring.
--api-username	No	User name of the account used to connect to Tableau Server APIs. The user account should be a Tableau Server administrator with access to all Tableau Server sites.
--api-password	No	Password of the Tableau Server API user account used to connect to Tableau Server APIs.

Option	Required?	Description
--api-password-file	No	The path to the file containing the password of the Tableau Server API user account.
--repository-server	No	This is the server name for the PostgreSQL database that is installed with the Resource Monitoring Tool.
--repository-port	No	The port number of the Tableau Server Repository database.
--repository-username	No	<p>Username used to connect to PostgreSQL database installed with the Tableau Server Repository.</p> <p>Resource Monitoring Tool accesses the Tableau Server Repository database directly for performance reasons. For this to work, access to the repository must be enabled, with a password set for the readonly database user. For details, see Enable access to the Tableau Server repository.</p>
--repository-password	No	<p>Password for the user account used to connect to the PostgreSQL database that is installed with the Tableau Server .</p> <p>Resource Monitoring Tool accesses the Tableau Server Repository database directly for</p>

Option	Required?	Description
		performance reasons. For this to work, access to the repository must be enabled, with a password set for the readonly database user. For details, see Enable access to the Tableau Server repository.
--repository-password	No	The path including the file name where the password for the user account used to connect to the PostgreSQL database that is installed with Tableau Server.
--repository-ssl-mode	No	Tableau Server Repository SSL Mode: Prefer SSL or Require SSL to configure SSL connections to Tableau Repository. Disable to never use SSL to make Tableau Server Repository connections.
--repository-ssl-thumbprint	No	You can choose to either supply the thumbprint that was generated by Tableau Server, or copy the server.crt file to the Resource Monitoring Tool Master Server machine. If you choose to copy the certificate file, you don't have to supply the thumbprint. For more information, see Configure Postgres SSL to Allow Direct Con-

Option	Required?	Description
		nections from Clients.

Example: `rmtadmin update-env --name=<new name>`

`rmtadmin users`

Shows a list of the Resource Monitoring Tool user accounts.

This command works only on the Master Server.

Synopsis

`rmtadmin users`

`rmtadmin version`

Shows the current version information for the Master Server when run on the Master Server. It shows the current version information of the Agent installed on the machine that you are running the command from.

This command can be run both on the Master Server and the Agent.

Synopsis

`rmtadmin version`

`rmtadmin ziplogs`

Creates a ZIP archive file containing the Resource Monitoring Tool log files.

This command works both on the Master Server and Agent but only includes the log files from the application the command is run on.

Synopsis

`rmtadmin ziplogs [positional parameter] [option] [global option]`

Positional Parameter

filename

Name of the output zip file. Defaults to log.zip if no name is provided.

Option

--force

Overwrites the existing file.

Example: `rmtadmin ziplogs <zip file name> --force`

Global Option

--help

Shows the help for the command.

Tableau Server Topology Changes

This article will help you understand what you need to do when you make certain topology changes to Tableau Server. The changes include: adding or removing a node, adding or removing a process from a node, and changing the number of processes on a node.

Adding a Node

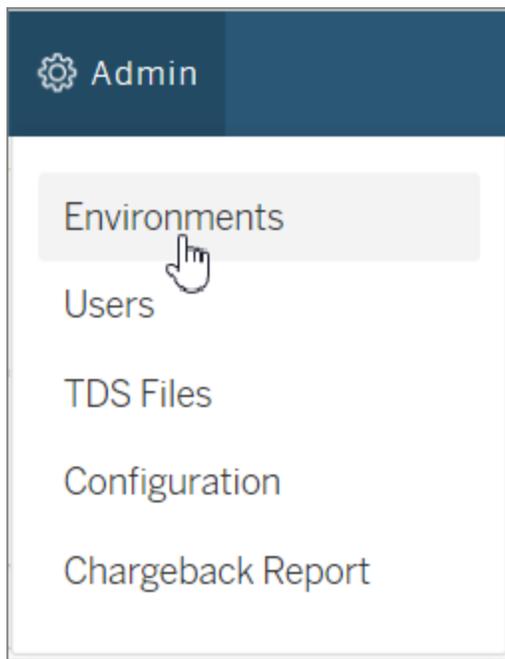
When you add a new node to Tableau Server, you must install the agent on this node. For more information on installing Agent on Tableau Server cluster, see [Install the Tableau Resource Monitoring Tool](#). Until you install the agent, information about this node will not be included in the reports.

Removing a Node

When you remove an existing node from Tableau Server cluster, you must update the environment on Resource Monitoring Tool Master Server.

Use the following steps to update the environment:

1. Log into the Master Server web interface.
2. From the **Admin** menu, choose **Environments**. Select the environment that should reflect this change.

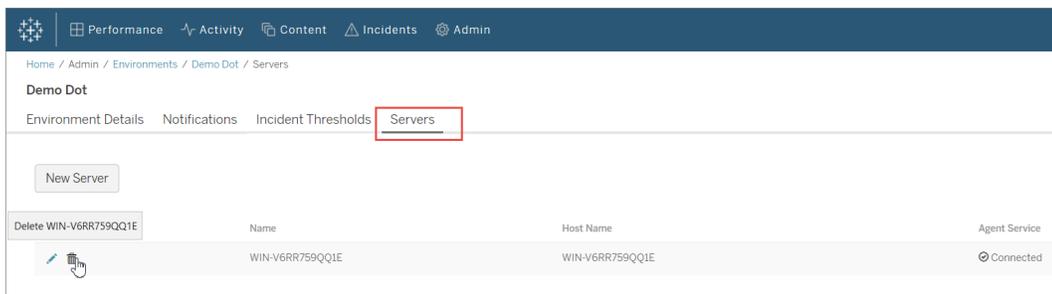


3. Choose **Edit Environment**. This opens more detailed information about the environment.



4. From the **Servers** tab, select the node that was removed from Tableau Server, and choose **Delete**.

Note: The host name should match the name or the IP address of the node.

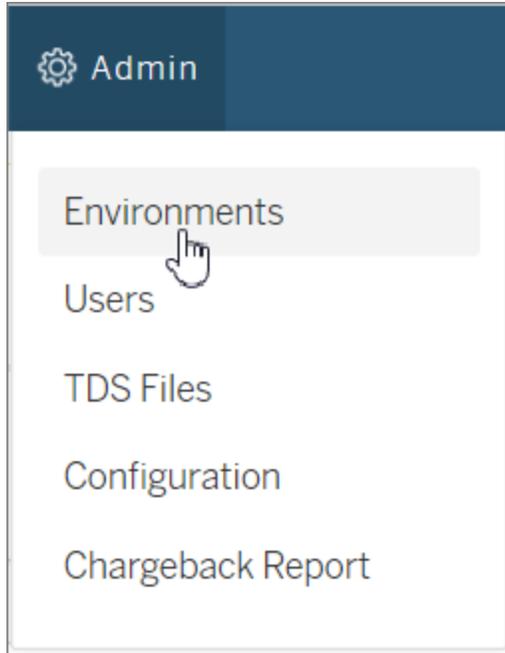


Adding or Removing a Tableau Server Process

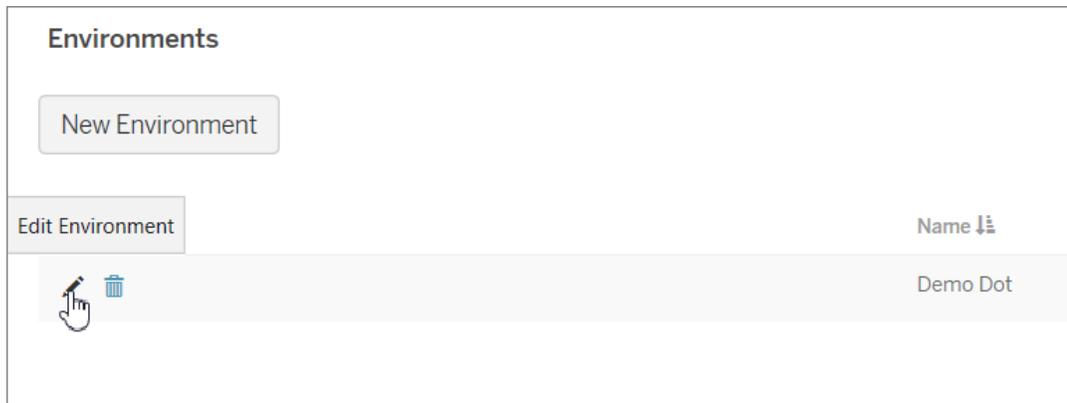
When you add or remove a process, or change the number of processes on a node, you must manually update the information on the Master Server. Until then the data reported may not be up to date or data may be missing.

Use the following steps to update the process information on Master Server:

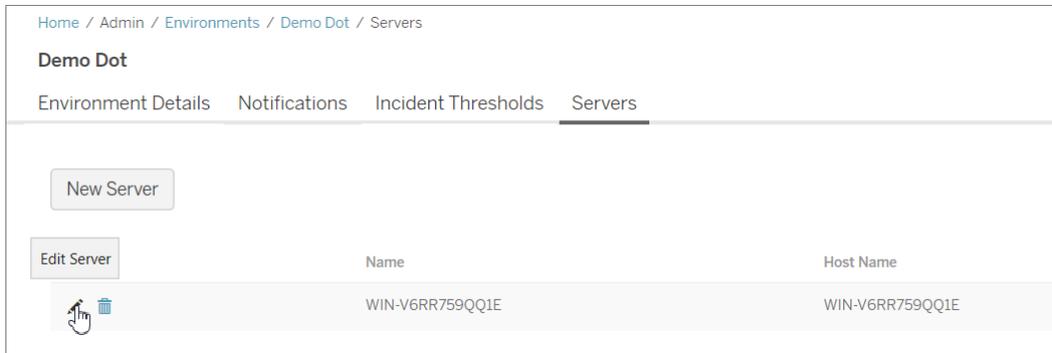
1. Log into the Master Server web interface.
2. From the **Admin** menu, choose **Environments**. Select the Environment.



3. Select **Edit Environment**. This opens more detailed information about the environment.



4. From the **Servers** tab, choose the node where the change was made, and select **Edit Server**.



5. Choose **Tableau Processes**. Add or remove processes to align with the changes that were made on Tableau Server.

WIN-V6RR759QQ1E

[Server Details](#) [Tableau Processes](#)

Server Processes

Edit the processes and ports being used on this server.

Add Process ▼

- Application Server
- Backgrounder Server
- Cache Server
- Cluster Controller
- Coordination Service
- Data Engine Server
- Data Server
- File Store
- Gateway Server
- Hyper
- Repository Server
- Search and Browse Server
- VizQL Server

	Port Number	
ver	8798	
ver	8189	
ver	8967	
ver	8633	
ver	8441	
ver	8214	
Cluster Controller	8156	

Delete

Re-registering an Agent

There can be situations when you might want to re-register an Agent. One such use case might be when you want to point one node or the entire Tableau Cluster to a different environment. Another example might be if you had to re-image a Tableau Server node and need to install and re-register the Agent. Use the following instructions to re-register the Agent.

1. On the Master Server web interface, navigate to **Admin -> Environments**. Select the environment. From the **Servers** tab, click delete to completely remove the Agent registration for this node on this environment. This also removes historical monitoring data that has been collected on this node. Use this for scenarios where you are ok with losing the history. For example, if you planning to point this node to a new environment, it might be acceptable to remove the historical data and start fresh.

Note: In the Master Server web interface, each node in a Tableau Server is referred to as a **Server**. The entire Tableau Server constitutes an environment. For more information, see Concepts

Who can do this

To make topology changes, you must be both a Tableau Server Administrator and a Resource Monitoring Tool Administrator.

Tableau Resource Monitoring Tool Log Files

Log Files

Tableau Resource Monitoring Tool includes many components. Each component maintains its own set of log files.

By default, 31 log files with a max file size of 1 GB each will be retained in each set. Excess log files beyond that will be deleted automatically.

Components

Component	Log File Location	Description
	/var/- opt/tableau/tabrmt/master/logs	
Backgrounder	background\YYYYMMDD-pts.log	Logs generated

Tableau Server on Linux Administrator Guide

Component	Log File Location	Description
	/var/ opt/tableau/tabrmt/master/logs	
		by the general background processor.
Director	director\YYYYMMDD-pts.log	Logs generated by the director background processor.
Host	host\YYYYMMDD.log	Logs generated by the Windows service host.
Web Server	web\YYYYMMDD-pts.log	Logs generated by the web server.

For advanced troubleshooting, verbose logging can also be enabled for more internal activities. These activity logs are stored in the associated component's log directory.

Activity	Log File Location	Description
Database	logs*\YYYYMMDD-ef.log	Logs of internal database queries. By default, only errors are written to this log. Enable by changing <code>db.logLevel</code> to <code>Debug</code> or higher.
Message Queue	logs*\YYYYMMDD-mq.log	Logs of internal message queue communication. By default,

Activity	Log File Location	Description
		this log will not be generated. Enable by changing <code>mq.logLevel</code> to <code>Debug</code> or higher.

Log Level Configuration

Resource Monitoring Tool log levels can be adjusted in the following ways:

- Resource Monitoring Tool web interface: From the Admin menu, navigate to Configuration and select the Advanced tab. You can set the log levels and also specify if you want to include database queries and message queue communications.

Logging levels you can set using the web interface:

Level
Default: Includes Information, Warning, Error, and Critical levels
Verbose: Includes Information, Warning, Error, Critical (everything in the default) and Debug levels

- Settings in the `config.json` files. Log levels are configured independently for each component. The default config file paths are below. This may vary depending on your installation folder.

Install	Default Location
Tableau Resource Monitoring Tool	<code>/var/opt/tableau/tabrmt/master/config.json</code>
Tableau Resource Monitoring Tool	<code>/var/opt/tableau/tabrmt/agent/config.json</code>

Tableau Server on Linux Administrator Guide

Install	Default Location
Agent	

Logging levels that can be configured using the configuration file:

Level
Trace
Debug
Information
Warning
Error
Critical
None

Below is a snippet demonstrating some default log level settings:

```
{
  "db": {
    "logLevel": "Error"
  },
  "mq": {
    "logLevel": "Warning"
  },
  "server": {
    "background": {
      "logLevel": "Information"
    },
    "director": {
      "logLevel": "Information"
    }
  }
}
```

```

    "web": {
      "logLevel": "Information"
    }
  }
}

```

Sending Log Files to Tableau Customer Support

If you are working with Tableau Support and they ask you to send log files, zip the files up before you send them:

1. Connect to Master Server and each server that has the Resource Monitoring Tool Agent installed.
2. Open a command prompt and run: `rmtadmin ziplogs <output file path>`, to create a ZIP archive of the log files.

For more information on sending log files to Tableau, see the [Tableau Knowledge Base](#).

Who can do this

Resource Monitoring Tool Administrator or a Resource Monitoring Tool user with **Server-Environment Management** role.

Tableau Log Files

Tableau Resource Monitoring Tool agents monitor the Tableau log files in near real time and send log messages to the master server for processing and reporting.

Below is a list of the specific Tableau log files that are monitored.

All directory paths are relative to the Tableau Server data directory. By default, this is located at: `/var/opt/tableau/tabrmt/data/tabsvc`

Directory	File Name
logs\backgrounder	backgrounder-*.log

Tableau Server on Linux Administrator Guide

Directory	File Name
logs\httpd	*.log
logs\vizportal	vizportal-*.log
vizqlserver\logs	dataserver_*.txt
vizqlserver\logs	vizqlserver_*.txt

Upgrading Tableau

When upgrading a Tableau Server installation that is monitored by Tableau Resource Monitoring Tool, there are a few additional steps that need to be followed.

Ensure Resource Monitoring Tool supports the new Tableau version

Before upgrading Tableau, you'll want to make sure that the version of Resource Monitoring Tool you have installed supports the version of Tableau that you are upgrading to.

The quickest way to check is to log in to the Resource Monitoring Tool web interface and navigate to the **Admin | Environments** screen. Edit the environment you are upgrading and check the **Tableau Version** drop down for the version you are upgrading to.

Stop Agents

Resource Monitoring Tool agents should be stopped while you upgrade Tableau Server. Follow these steps on each machine in your Tableau Server cluster:

Run the following command as the `tabrmt-master` user:

```
sudo su --login tabrmt-master
```

```
rmtadmin stop --agent
```

Upgrade Tableau

Follow the normal process for upgrading Tableau Server. This process is outlined in [the Tableau Server documentation](#).

Update Tableau Version in Resource Monitoring Tool

1. Log in to the Resource Monitoring Tool web interface.
2. Go to **Admin | Environments**
3. Edit the environment you upgraded.
4. Modify the **Tableau Version** to match

Restart Agents

After the upgrade is complete you are ready to restart the Resource Monitoring Tool agents. Follow these steps on each machine in your Tableau Server cluster:

Run the following command as the `tabrmt-master` user:

```
sudo su --login tabrmt-master
```

```
rmtadmin restart --agent
```

Who can do this

To upgrade Resource Monitoring Tool, you must have all the following:

- Full sudo access for the user on machine you are installing Resource Monitoring Tool.
- Tableau Server Administrator site role.
- Resource Monitoring Tool Administrator account.

Monitor Tableau Server Performance

Tableau Resource Monitoring Tool is used to monitor and analyze Tableau Server health and performance. Performance, usage and hardware metrics are collected through the Agents installed on the Tableau Server nodes and sent to the Master Server. The aggregated and analyzed data is then displayed in the form of charts and views on the web interface of the Master Server.

The web interface of the Master Server has built-in charts and views that you can use to identify what is causing slow load times, extract failures, and other critical issues. For more information, see [Charts](#).

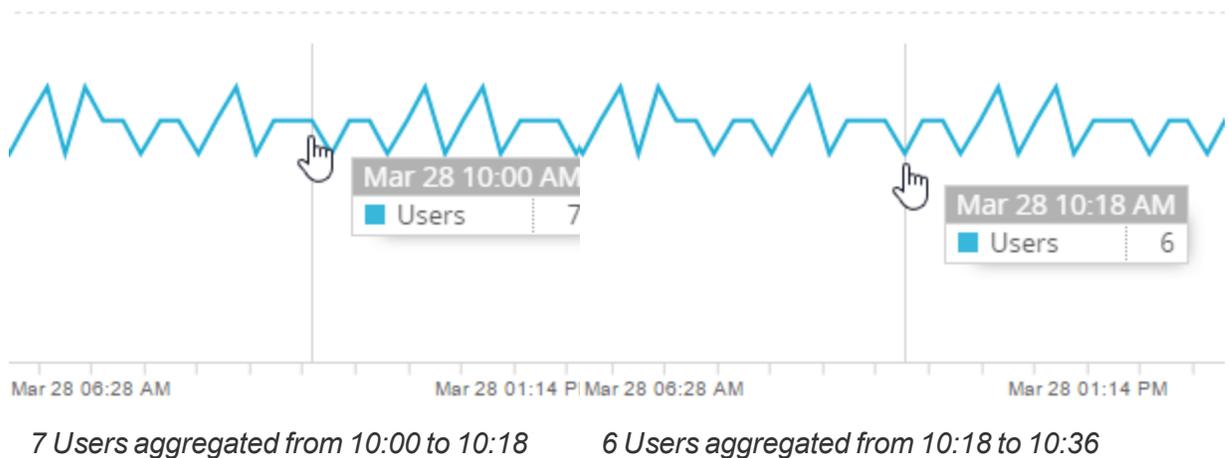
You can also download the data that is used to create the pre-built charts and explore it further. For more information, see [Explore Monitoring Data Using Tableau Data Source Files](#).

Charts

Line Chart Resolution

Line chart resolution simply means how many points of data are shown for a given date range. Tableau Resource Monitoring Tool shows up to 80 points on each line chart by default. This prevents lengthy date ranges from drawing too many data points on the screen, which would cause performance problems.

When looking at one hour, each point represents 45 seconds of data. When looking at one full day, each point represents 18 minutes of data. The data is aggregated from one point until the next point. That is to say that given a full day, a point at “10:00” represents an aggregate of all the data from “10:00” to “10:18”.

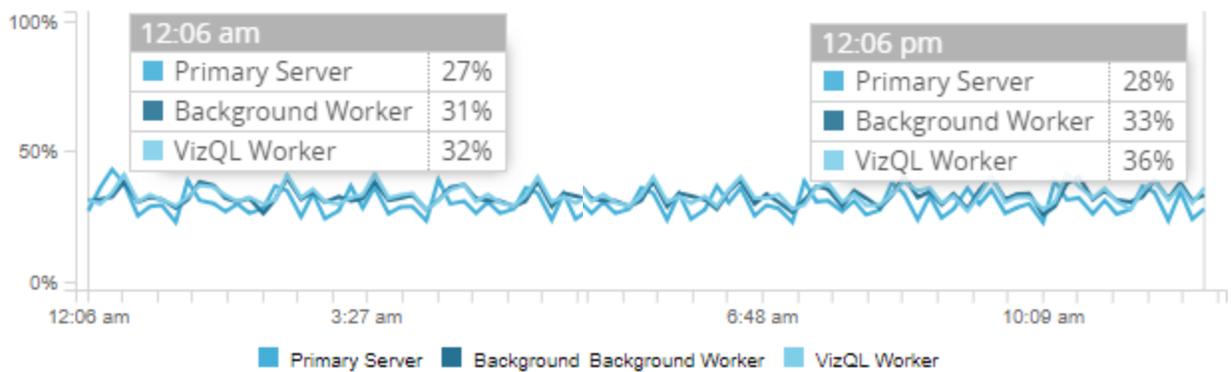


Line Chart Date Range

Each point on the line chart represents a span of time (see [Line Chart Resolution](#) for more info) and the chart is constructed so that the span is the same length for each point. In order to do this, the exact start date and end date of the chart may need to vary slightly from the user specified date range.

For example when looking at a twelve hour range from “12:10 am” to “12:10 pm”, with an 80 point resolution, each point represents 9 minutes of data. The first point will be adjusted to begin at “12:06 am” and the last point will be adjusted to begin at “12:06 pm”. These times are calculated by finding the first 9 minute interval from 1/1/1970 00:00:00 UTC that contains the specified start date and the first 9 minute interval that contains the specified end date.

Note: research “Unix time” if you are interested in learning more about why the interval calculation begins at 1/1/1970 00:00:00 UTC.



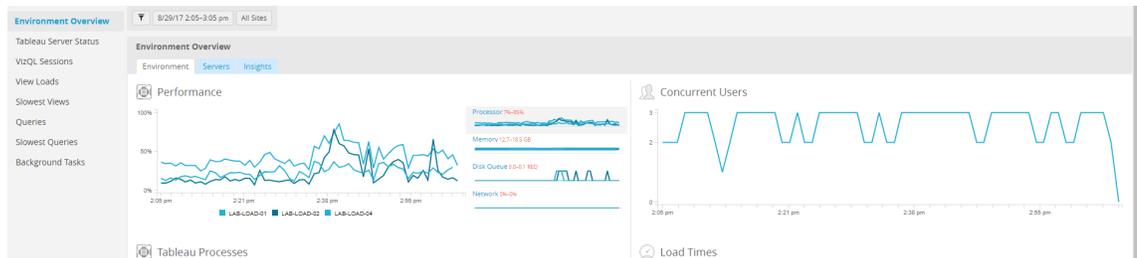
The chart begins with values aggregated from 12:06 to 12:15 am

the chart ends with values aggregated from 12:06 to 12:15 pm

Environment Overview

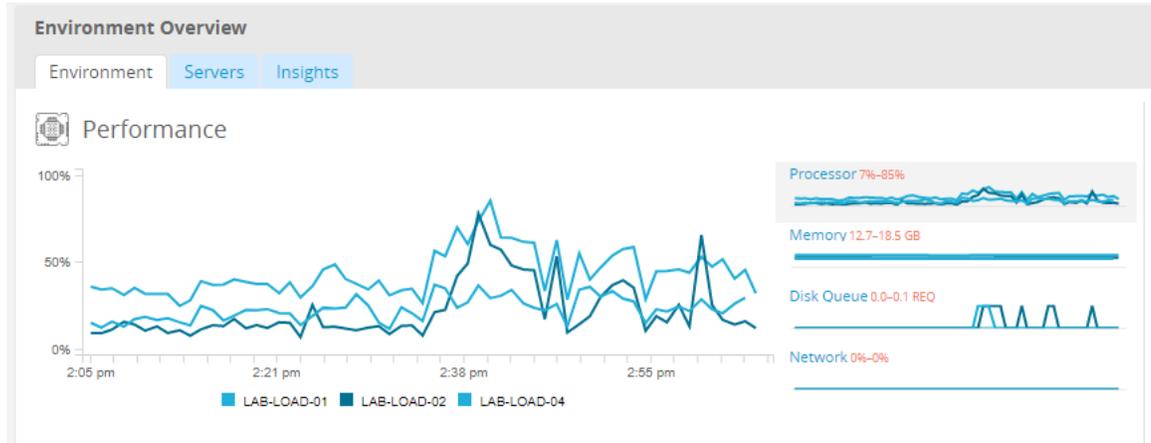
Dashboard that uses tabs to separate charts into a few categories that are outlined below.

Environment Tab



Environment Tab Overview

Performance



Performance Chart

Each point is an *average* of the data sampled.

Current performance values are sampled once every minute by default. In multicore environments, every CPU is aggregated for the Processor chart. To see each CPU individually, select the server by its name from the Performance dashboard menu.

The Processor, Memory, Disk Queue, and Network charts are all calculated this way.

Note: Network charts are not available for Tableau Server installed on Linux.

Tableau Processes

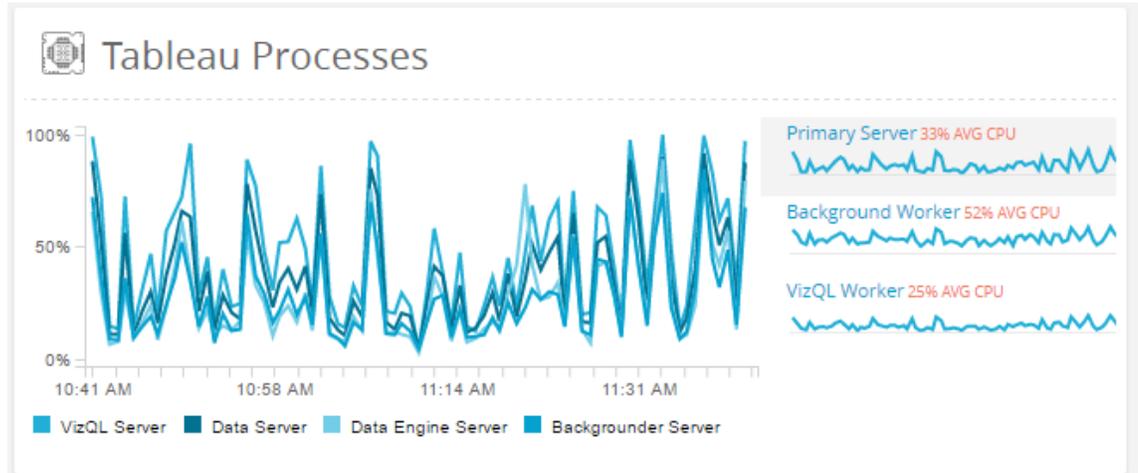


Tableau Processes

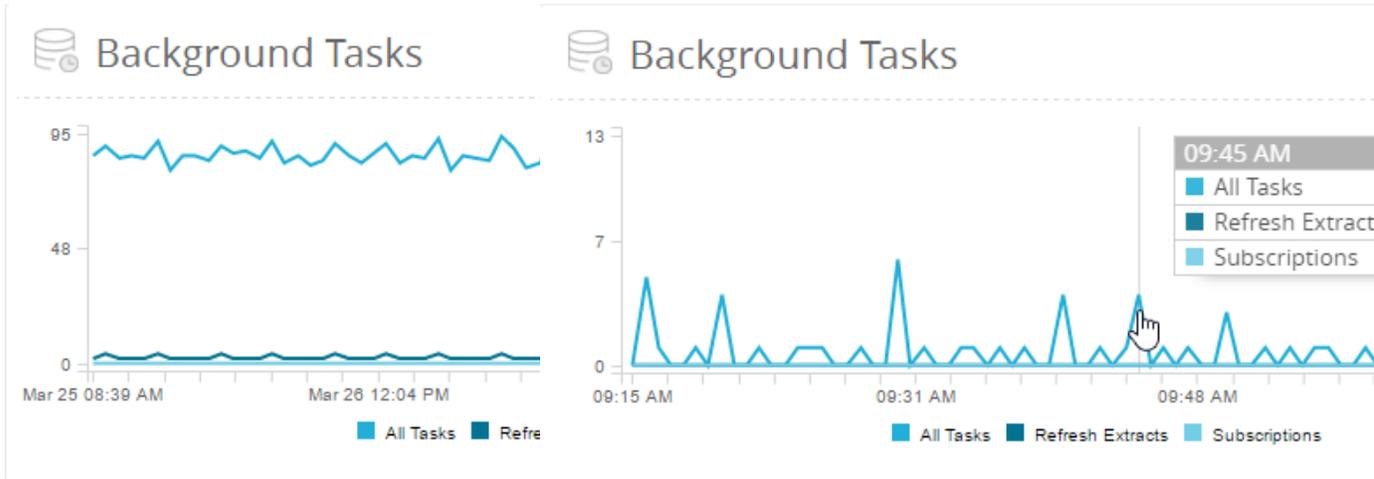
Each point is an *average* of the CPU data sampled for the given Tableau process.

Environments with multiple agents will have a chart for each agent. In this scenario, it's normal for some agents to have no information about a Tableau process, as some workers could have no processes for a given Tableau process assigned to it.

Background Tasks

Each point is a *summation* of background tasks that are running or just finished.

For example, a *Refresh Extract* task will be counted for as long as it's running, including the point where it finishes. This accounts for background tasks that are very quick. Lengthy date ranges will yield larger tasks counts because this chart uses a summation of activity. See [Line Chart Resolution](#) at the top of this article for more information.



84 total tasks reported at 9:45 when viewing 5 days of data

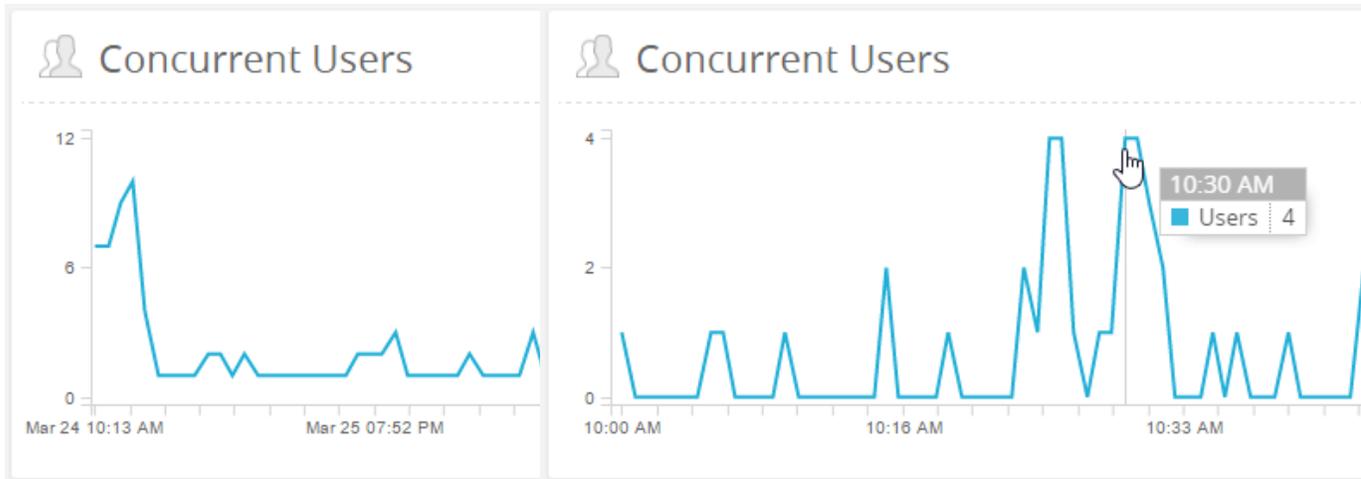
4 total tasks reported at 9:45 when viewing 1 hour of the same data

All Tasks is a summation of Refresh Extracts, Subscriptions, and all other background tasks. Navigate to the *Background Tasks* page to see more detail about each task accounted for.

Concurrent Users

Each point is a *summation* of the unique Tableau user activity.

Gateway requests like viewing a worksheet or dashboard are activities. If a unique Tableau user does two activities within one point, then it will only be counted once. Lengthy date ranges will yield larger Concurrent User counts because this chart uses a summation of activity. See [Line Chart Resolution](#) at the top of this document for more information.

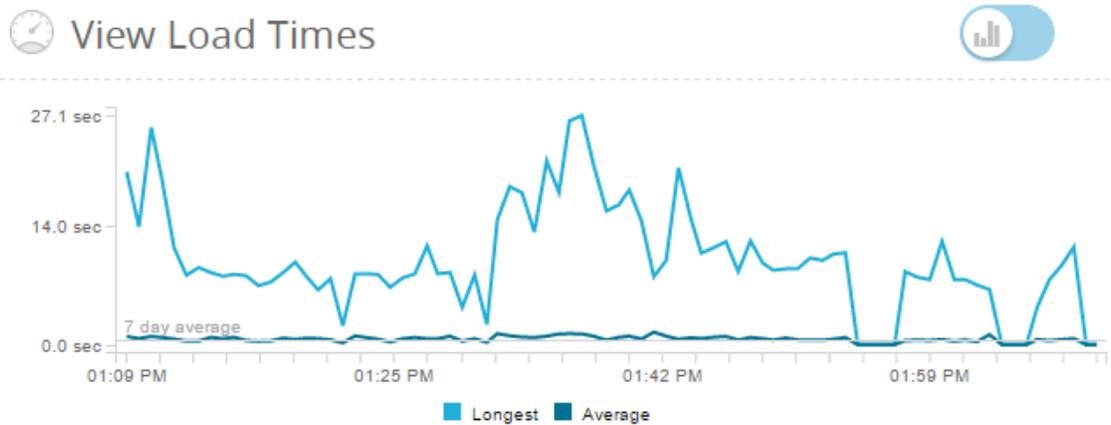


12 unique Tableau Users reported at 10:30 when viewing 5 days of data

4 unique Tableau Users reported at 10:30 when viewing 1 hour of the same data

Load Times

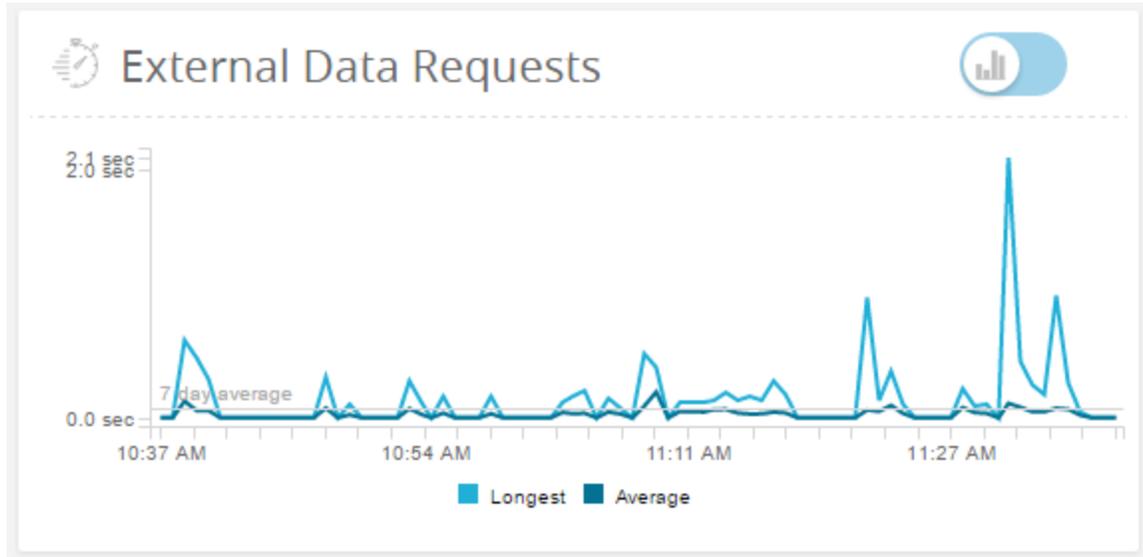
View Load Times



View Load Times

Each point of the *Longest* line represents the maximum duration of all view loads at that point. Each point of the *Average* line represents the average duration of all view loads at that point. Individual view load durations can be found on the *View Loads* page.

External Data Requests



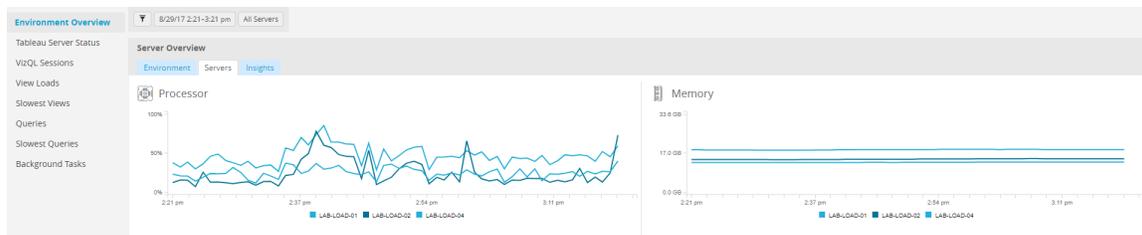
External Data Requests

External data requests are requests to the data server from an IP address that is not one of the machines in the Tableau cluster (not a worker, not the master). Each point of the *Longest* line represents the maximum duration of all external requests at that point. Each point of the *Average* line represents the average duration of all external requests at that point.

Background Tasks

Refer to the *Background Tasks* chart on the *Environment Overview* page.

Server Tab



The *Servers* tab has all of the charts that were previously on the *Server Dashboard* and is focused on monitoring the hardware performance of your servers.

The charts will show aggregated performance data for your entire environment.

Each chart will have a line series per server in the environment.

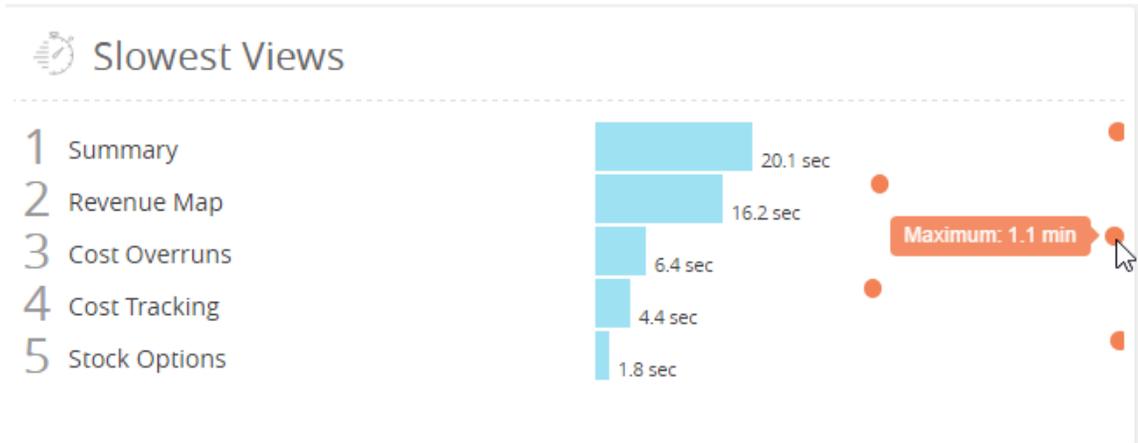
Refer to the *Performance* chart on the *Environment Tab* page.

Insights Tab



Insights Tab

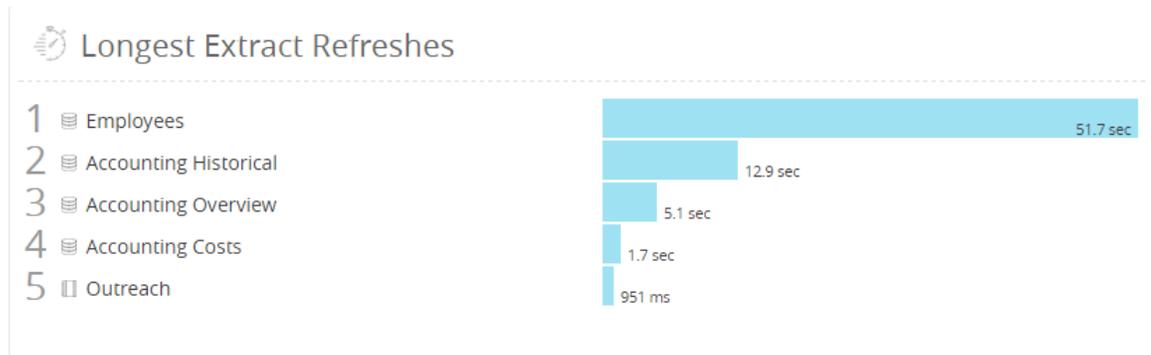
Slowest Views



Slowest Views

This chart shows the top five slowest views ordered by *average* view load duration. The orange dot indicator represents that view's *longest* load duration

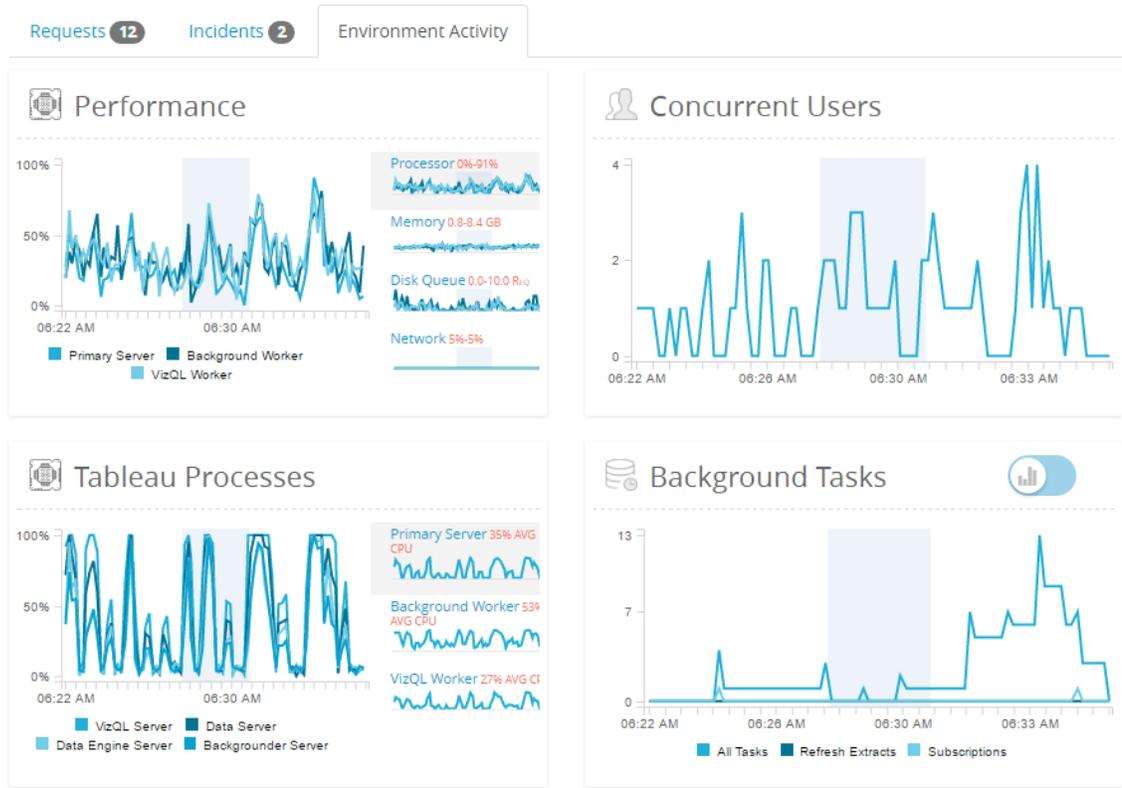
Longest Extract Refreshes



Longest Extract Refreshes

This chart shows the top five slowest *Refresh Extract* tasks ordered by duration. Both data source extracts and workbook extracts are considered. They are distinguishable by the icon next to their name.

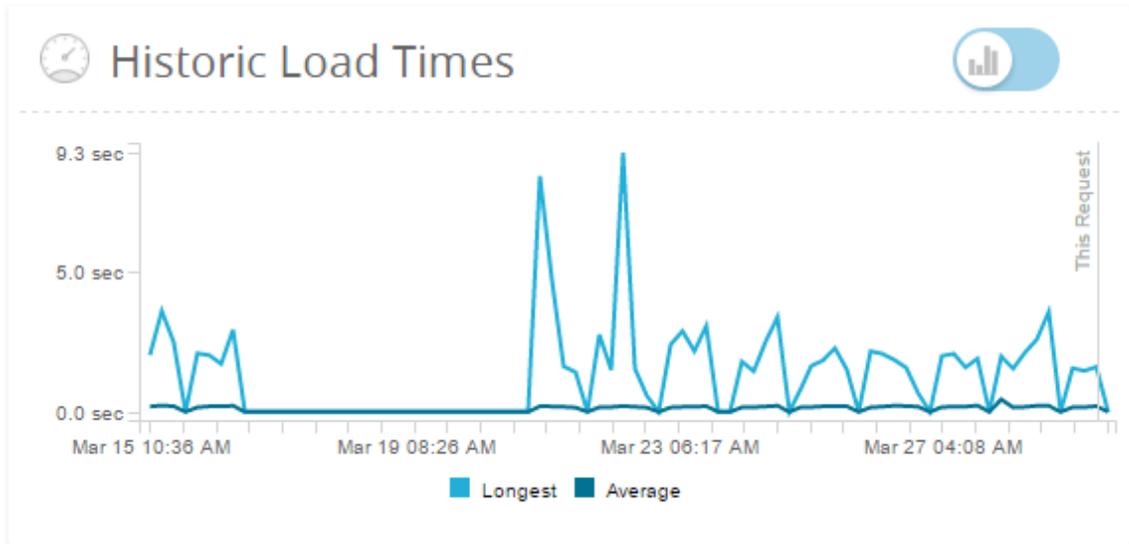
VizQL Session Environment Activity



VizQL Session Details - VizQL Processor and VizQL Memory spark lines

These charts show the same information as the [Environment Overview](#) charts. The VizQL Session's time frame is used as the date range.

Historic Load Times



View Load Request Details - Historic Load Times

This chart shows up to 14 days of load times for the selected view. Each point of the *Longest* line represents the maximum duration of view loads at that point. Each point of the *Average* line represents the average duration of view loads at that point.

View Load Environment Activity

These charts show the same information as the [Environment Overview](#) charts. Typically, 5 minutes before and after the time the view was loaded is used as the date range.

Who can do this

Any Resource Monitoring Tool user can view the charts.

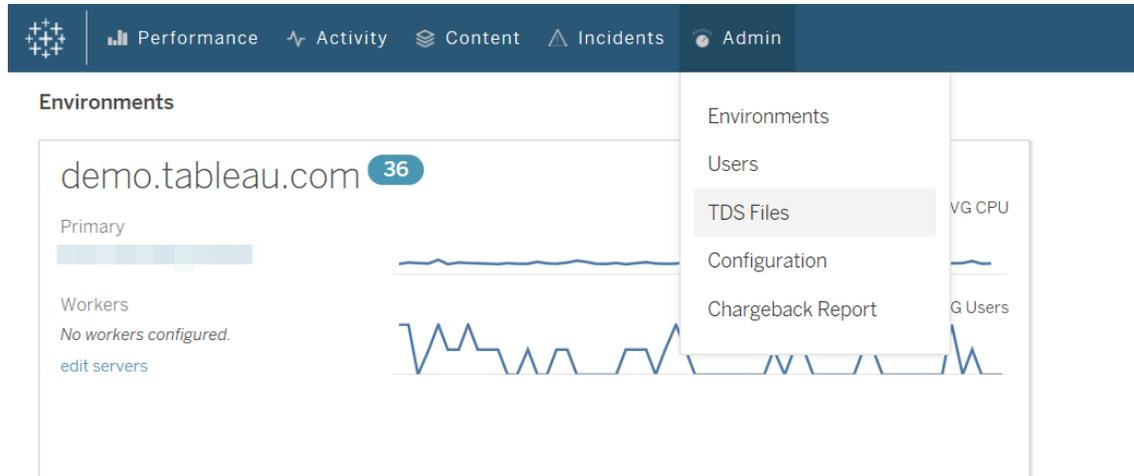
Tools used in Data Collection

Below is a list of the specific classes used to collect monitoring data :

Class	Command	Monitoring Category
LinuxSystemCpuCollector	top	CPU
LinuxProcessPerformanceCollector	top	Process
LinuxSystemMemoryCollector	free	Memory
LinuxDiskQueueLengthCollector	iostat	Disk
LinuxDiskUsageCollector	df	Disk

Explore Monitoring Data Using Tableau Data Source Files

The Tableau Resource Monitoring Tool includes built-in charts that you can use to monitor and analyze Tableau Server health and performance. The data that the monitoring tool leverages can be downloaded as .tds files for exploration within Tableau Desktop. You can download the **TDS files** from the **Admin** menu using the Resource Monitoring Tool web interface.



Following is the list of **TDS files** you can download:

- **Background Tasks:** Includes information about scheduled background tasks like extract refreshes, subscriptions, and flows.

Tableau Server on Linux Administrator Guide

- **Data Queries:** Information about all queries executed by Tableau Server.
- **Gateway Requests:** HTTP requests handled by Tableau Server including VizQL Server session details.
- **Incidents:** Incidents recorded by the Resource Monitoring Tool .
- **Server Performance:** Tableau Server hardware and process information that is gathered by the Resource Monitoring Tool.
- **Tableau Entities:** Information about the Tableau Server sites, projects, workbooks, and views gathered by the Resource Monitoring Tool.

Requirements

- The Tableau Server *tds* (data source) file contains a connection to the Resource Monitoring Tool PostgreSQL database. The Resource Monitoring Tool is running PostgreSQL version 12, which is only supported in Tableau Desktop 2020.4 and later. You must use Tableau Desktop 2020.4 or later to open the *tds* file.
- Encryption used is SCRAM-SHA-256 which is also supported by Tableau Desktop 2020.4.

Downloading TDS files from a remote computer

The **TDS files** are generated by connecting to the PostgreSQL database where the information is stored. When you connect locally on the Master Server, it works as expected. However, if you want to be able to download the **TDS files** from remote machine, you must make some configuration to allow connecting to the PostgreSQL database and downloading the files from a remote machine.

Use the following instructions to make the configuration changes that allow you to connect to the PostgreSQL database and download the **TDS files** from a remote machine:

1. Open `postgresql.conf` file: By default, the file is located at: `/var/opt/tableau/tabrmt/data/postgresql<version>`
2. Update `Listen_addresses = 'localhost'` to `Listen_addresses = '*'`.

Note: You must remove the '#' from this line to make it into an active command.

3. Open `pg_hba.conf` file: This file is also located in the same directory as the `postgresql.conf` file. By default the file is located at: `/var/opt/tableau/tabrmt/data/postgresql<version>`
4. Add the following to the end of the file:

```
host all all 0.0.0.0/0 scram-sha-256
```

```
host all all ::/0 scram-sha-256
```

Tip! Using the **scram-sha-256** hash function requires a user name and password to use the TDS files. You can find this information in the config file as described below:

The user name is stored under `db:readOnlyUserName`, and `db:readOnlyPassword` settings. To retrieve the user name, open the Resource Monitoring Tool master config.json file and locate the `db:readOnlyUserName` setting. To retrieve the password, run the `rmtadmin get db.readOnlyPassword` command on the Master Server.

5. Restart PostgreSQL Server. using the following command:

```
rmtadmin restart --db
```

If after completing these steps, if you still have problems downloading the **TDS files** from a remote machine, contact [Tableau Customer Support](#).

Who can do this

Resource Monitoring Tool Administrator or a Resource Monitoring Tool user with **Download TDS Files** server role.

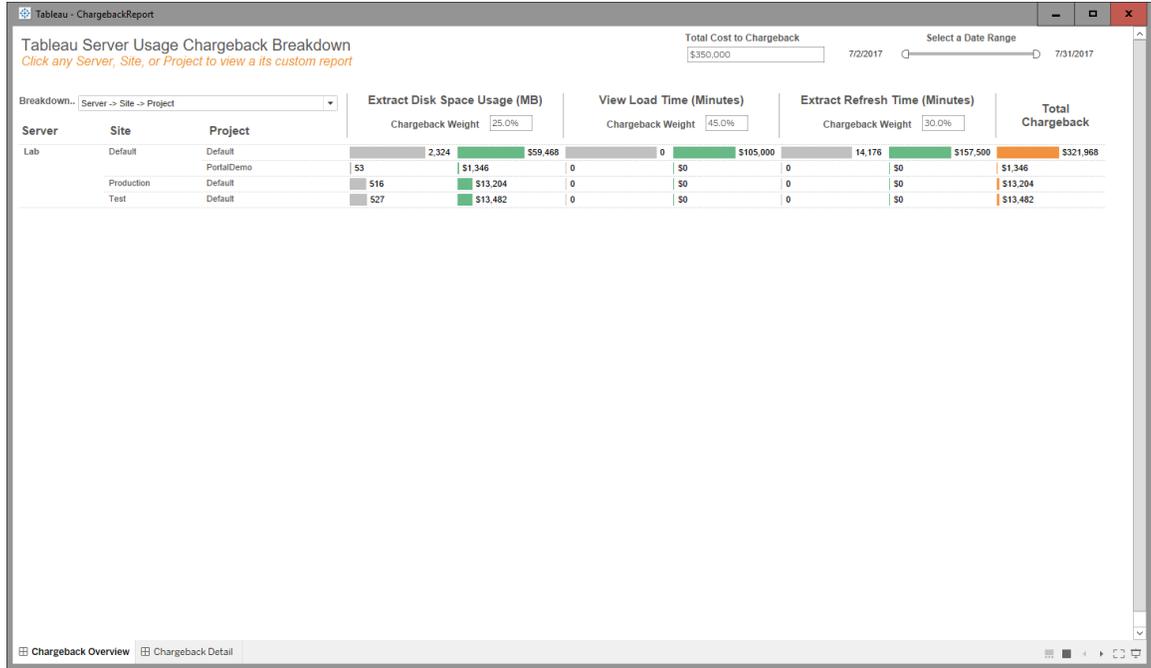
Chargeback Reports

Chargeback reports show usage on a per-project or per-site basis and are available for allowed users through the **Admin** menu. The report is generated as a Tableau workbook

Tableau Server on Linux Administrator Guide

using a generated extract, allowing you to modify the report or reuse the extract as needed.

The **Chargeback Overview** worksheet shows a breakdown of various metrics based on site and/or project:



Overview

Dollar amounts can be customized using the total cost field, and by adjusting the cost weights for each metric.

Clicking on a project or site navigates to a customized invoice-style detail report for that project/site:

Tableau Server Usage Report

Between July 2, 2017 and July 31, 2017
 Server: Lab
 Site: Default
 Project: Default

Total Chargeback
\$321,968
 92.0%
 of Total \$350,000

Metric	Usage	Total Server Usage	% of Total Usage	Allocated Cost
Extract Diak Space (MB)	2,324	3,420	68.0%	\$59,468
Extract Refresh Time (Minutes)	0	0	100.0%	\$105,000
View Request Time (Minutes)	14,176	14,176	100.0%	\$157,500

Chargeback Overview Chargeback Detail

Detail

Security

User access is controlled through the **Generate Chargeback Reports** server role.

Data Generation

Data for the chargeback report is aggregated at the daily level, with the following metrics included in the data set:

Metric	Description
Extract File Usage	The size, in Kilobytes, of extract files is collected once per day. For the default report this is simply totaled across the time period for ratio/comparison purposes between projects/sites.
Extract Query Duration	The length of time, in milliseconds, that queries running against extract files ran, totaled per day.

Metric	Description
Extract Refresh Task Duration	The length of time, in milliseconds, that extract refresh background tasks ran, totaled per day.
Query Duration	The length of time, in milliseconds, that queries ran, totaled per day. This includes both extract and non-extract queries.
View Request Duration	The length of time, in milliseconds, that requests to render views ran, totaled per day. This includes the duration of any data queries that blocked view rendering.

Who can do this

Resource Monitoring Tool Administrator or a Resource Monitoring Tool user with **Server-Environment Management** role.

Troubleshoot Tableau Resource Monitoring Tool Issues

This section includes articles that describe troubleshooting steps and tips. If you have any questions or encounter other issues not described here, contact [Tableau customer support](#).

Troubleshoot Missing Hardware Performance Data

Processor (CPU) usage, memory usage, disk queue, and network performance stats are considered hardware performance data in the Tableau Resource Monitoring Tool. The most common places to see this data are:

- On the **Environment Overview** dashboard in the Performance and Tableau Processes charts
- On the **Servers** dashboard

This data is reported in near-real-time by the Resource Monitoring Tool agent processes running on each of your Tableau Server machines.

If these charts show no data for an extended period of time, there may be a connectivity problem with the agent. Follow the recommendations below to isolate the issue:

Step 1: Check the Agent connection status

First, check that the Agents are currently connected to the Resource Monitoring Tool. The Agents send a regular heartbeat message to the Master service to indicate their connection status.

1. Log in to the Resource Monitoring Tool as an administrator.
2. Navigate to the **Admin > Environments** page.
3. Click the Edit link for the environment that is missing performance data.
4. Locate the **Servers** list and ensure that each server shows the Agent Service as **Connected**. You can hover over the **Connected** status for a timestamp of when the last heartbeat message was received.

Step 2: Ensure the Agent is running

If the Agent shows as **Disconnected** in the Resource Monitoring Tool, then the Agent's Windows service may not be running.

1. Connect to the machine the agent is running on.
2. Ensure that the Tableau Resource Monitoring Tool Agent Windows service is running.

Step 3: Ensure the Agent is configured correctly

If the agent shows as **Disconnected** in the Resource Monitoring Tool, but the service is running, then the agent may not be able to reach the Resource Monitoring Tool's message queue.

1. Connect to the machine the agent is running on.
2. Navigate to the agent's installation folder. For example: `/opt-tableau/tabrmt/agent`
3. Run the `rmtadmin status` command.

The `rmtadmin status` command will test the agent's connectivity to the message queue and the Resource Monitoring Tool's master server.

Tableau Server on Linux Administrator Guide

Step 4: Restart the Agent

In some cases, the Agent may be running and all status indicators show success but the agent continues to not send hardware performance data. This is a known issue that can result from transient connectivity errors between the Agent and the message queue. For example, when restarting the message queue server or during brief network interruptions.

To ensure this isn't the case, restart the Agent and wait a few minutes to confirm whether the performance data is working again.

1. Connect to the machine the Agent is running on.
2. Restart the Tableau Resource Monitoring Tool Agent Windows service.
3. Ensure the service starts successfully.
4. Wait at least 10 minutes and then log in to Resource Monitoring Tool and check Servers dashboard to see if any hardware performance data has been received.

Step 5: Contact Support

If, after following the above steps the issue is still not resolved, please contact support.

The support team will need a copy of the Resource Monitoring Tool log files from the master server and from each of the agents that are having connection issues. For more information on how to collect log files and sending them to Tableau customer support, see [Tableau Log Files](#).

Who can do this

Resource Monitoring Tool Administrators.

Troubleshoot Unknown Status of Tableau Server Processes

Under certain conditions you might see that the status of a Tableau Server process as reported as **Unknown**. This is often due to a change in the process configuration in Tableau Server which has not yet been updated in Tableau Resource Monitoring Tool.

Note: After updating any configuration, it may take a few minutes for the status reported by the Resource Monitoring Tool to refresh. By default, the Resource Monitoring Tool checks the Tableau Server status every 15 seconds but it may be longer if you configured it to check less frequently.

Use the following troubleshooting steps to resolve this issue:

Step 1: Check Tableau Server Environment Settings

The Resource Monitoring Tool connects to Tableau Server for monitoring and data collection. If the Tableau Server is upgraded, or the Tableau Server credentials expire then Resource Monitoring Tool will not be able to monitor it as expected.

To confirm that the Resource Monitoring Tool is able to connect to your Tableau Server:

1. Go to **Admin > Environments list** page.
2. Edit the environment that is having the issue.
3. Confirm that the selected Tableau Server version is correct. When you upgrade your Tableau Server you may need to update the version in the Resource Monitoring Tool.
4. Test the Tableau Server REST API connection using the “Test Connection” button.

Optionally, test the Tableau Repository connection using the “Test Connection” button.

Step 2: Update Tableau Server Machines and Processes

To correctly monitor a Tableau Server, the Resource Monitoring Tool needs to be configured with a complete record of your Tableau Server machine names and process ports. This information is automatically gathered when creating a new environment but may need manually updated if the Tableau Server configuration is changed.

1. Go to the **Admin > Environments** page.
2. Edit the environment that is having the issue.

Tableau Server on Linux Administrator Guide

3. Confirm that the list of **Servers** contains your initial node/gateway Tableau Server machine as well as all additional node machines.
4. Follow the steps below to confirm that each server is configured correctly.

Step 3: Update Machine Name

1. Open this Tableau Server URL: *http://<your TableauServer URL>/admin/systeminfo.xml*.
2. Compare and update the Resource Monitoring Tool configuration with this page to ensure it is up-to-date:
 - Update the server's Host Name to exactly match the `<machine name="{HOST NAME}" />` attribute value.

Step 4: Update Processes and Ports

1. Open this Tableau Server URL: *http://<your TableauServer URL>/admin/systeminfo.xml*.
2. Compare and update the Resource Monitoring Tool configuration with this page to ensure it is up-to-date:
 - Locate the `<machine>` element in the systeminfo.xml response that corresponds to the server you are editing in the Resource Monitoring Tool.
 - Each child element of `<machine>` corresponds to a Tableau Server process that is running on that machine. Update the list of Processes in the Resource Monitoring Tool to exactly match the information in systeminfo.xml for the server.

Step 5: Contact Support

If, after following the above steps the issue is still not resolved, contact [Tableau customer support](#).

VizQL Session details page says the VizQL process is unknown

HTTP requests for a Tableau view are linked to the VizQL process (the actual VizQL PID) that last locked the request's VizQL session ID.

In some cases we may not be able to find a matching VizQL process (PID) and in these cases you will see a message on the view session details page that the VizQL process is unknown.

This can happen in a few rare cases:

- The Tableau cluster has been modified with the addition of a new VizQL worker instance.
- Tableau adjusting the VizQL process's port number to avoid a conflict with another process.

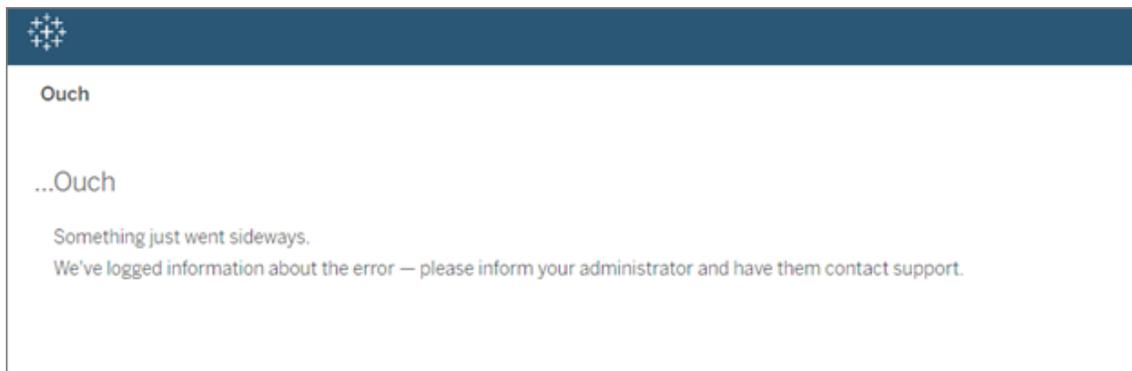
If this happens, sign in to the Resource Monitoring Tool and navigate to the environment admin screen. You'll want to check the process definitions for your servers. Ensure that VizQL processes are defined on the expected servers and with the correct port numbers.

Who can do this

To troubleshoot Tableau Server processes issues, you need to be both a Tableau Server Administrator and a Resource Monitoring Tool Administrator.

Troubleshoot Web Interface Timeouts

When you try to navigate to the Tableau Resource Monitoring Tool web interface, you see the following error:



This can happen when the load time takes longer than the default 30 seconds threshold.

The load time can be affected by the following:

- The size of the underlying PostgreSQL database.
- If RMT is connected to Tableau Server environments that push the current scale that RMT can support, this may also have an affect on the load times.

To resolve this issue you can either change the data retention time period or change the threshold to greater than 30 seconds.

To change the data retention time period, use the following steps:

1. Navigate to the Configuration UI as described in [this topic](#).
2. Navigate to the Data tab.

There are two types of Data Configurations:

- Data retention that is set to 2 weeks by default. It includes detailed activity data and the data displayed in the charts.
 - Reporting data that is set to 10 year time period by default. It includes aggregated historical data and data used for chargeback reports.
3. Tableau recommends changing the data retention time period to 1 week.

To change the timeout threshold, use the following steps:

1. Log into the machine where Resource Monitoring Tool Master Server is installed.
2. Find the master configuration file. The file when installed in the default location can be found at:

```
/var/opt/tableau/tabrmt/master/config.json
```

3. Add or change the key: **db:timeoutSeconds**

If you need to reduce the size of the PostgreSQL database immediately, Use the `rmtadmin` command line utility to delete the data. For more information, see [rmtadmin Command Line Utility](#).

Who can do this

You need to be a Administrator on the machine and be a Resource Monitoring Tool Administrator in order to make configuration and database changes.

Upgrade Power Tools for Server to Tableau Resource Monitoring Tool

Power Tools Server has been rebranded to Tableau Resource Monitoring Tool. The installer has been changed to reflect this rebranding and new ownership. This means the product will install side-by-side with the legacy InterWorks product instead of upgrading it in place.

The 2019.3 version of Power Tools Server (now named the Tableau Resource Monitoring Tool) requires a license key. For information about activating the license key, see [Tableau Resource Monitoring Tool Legacy License Key Activation](#).

Follow these steps to upgrade to Resource Monitoring Tool. The recommended steps to upgrade are:

1. Stop all installed InterWorks agents.
2. Wait for the data processing to complete.
3. Stop and uninstall InterWorks master server.
4. Run the Tableau master server installer to install Tableau master server.
5. Stop Tableau master server.
6. Copy over *config/config.json* from InterWorks Master Server directory to Tableau Master Server directory.
7. If you are using Postgres as your database, the following steps are necessary for the database to function with the security improvements in 2019.3. If using any other database, skip to step 8.

Postgres SQL Update Instructions:

- Locate the *postgres pg_hba.config* file. It will be at *<PTS installation directory>\data\postgresql\pg_hba.config*.
- Edit the *pg_hba.config* file, entries at the end will look like this: host all all 127.0.0.1/32 trust. Change the last column for all lines from **trust** to **md5**. So the new line should look like this: host all all 127.0.0.1/32 md5.

Tableau Server on Linux Administrator Guide

- Save the *pg_hba.config* file.
 - Restart the PTS postgresql service.
8. Restart Tableau Master Server.
 9. Run the Tableau Agent installers to install Tableau agents.
 10. Stop Tableau Agents.
 11. Copy over *config/config.json* from InterWorks agent directory to Tableau Agent directory.
 12. Once the Tableau Resource Monitoring Tool installation is complete, and you have confirmed it is working as expected, uninstall the InterWorks/Tableau Power Tools for Server.

Tableau Resource Monitoring Tool Legacy License Key Activation

The 2019.3 version of Power Tools Server (now named the Tableau Resource Monitoring Tool) requires a license key. The instructions below provide the steps to activate this key:

1. The license key will be passed over as a file in format *.dat*.
2. The *.dat* file needs to be added to the config directory of the Master Server.

The default location is: C:\Program Files\Tableau\Tableau Resource Monitoring Tool\master\config.

3. Restart the Master Server service:

Navigate to **Windows Server Manager** and restart the **Tableau Resource Monitoring Tool** service.

About Tableau Content Migration Tool

This set of articles guides you through setting up, using, and maintaining the Tableau Content Migration Tool.

What is Content Migration Tool?

The Content Migration Tool provides an easy way to copy or migrate content between Tableau Server projects. You can do this between projects on a single Tableau Server installation, or if you have user-based licensing, between projects on separate Tableau Server installations (for example, between a development instance of Tableau Server and a production installation). The Content Migration Tool User Interface walks you through the steps necessary to build a "migration plan" that you can use a single time, or as a template for multiple migrations.

Before migrating content, we recommend reviewing the [Content Governance](#) section in Tableau Blueprint.

Help and Support

If you have problems that you cannot solve with this documentation, contact [Tableau Technical Support](#).

Getting Started with Tableau Content Migration Tool

This article will help you get started with the Tableau Content Migration Tool. It contains links to other articles about information you need to prepare before installing the Content Migration Tool, as well as steps to design a migration plan and upgrade existing installations.

Tableau Server on Linux Administrator Guide

Pre-Installation

Installation requirements

The Content Migration Tool can only be installed on Windows operating systems. Before installing, you must be able to connect to the Tableau Server computer with the source site (the site you are migrating) and the computer with the destination site (the site you are migrating to). Both the source and destination sites must have a valid Server Management Add-on license. For more information, see [Install Tableau Content Migration Tool](#).

Compatibility with Tableau Server

The Content Migration Tool supports content migration for Tableau Server versions 2019.3 and later.

The table lists compatible versions of Tableau Server based on the installed version of Content Migration Tool.

CMT Version	Tableau Server Version
2021.2.x	2019.3.x - 2021.2.x
2021.1.x	2019.3.x - 2021.1.x
2020.4.x	2019.3.x - 2020.4.x
2020.3.x	2019.3.x - 2020.3.x
2020.2.x	2019.3.x - 2020.2.x
2020.1.x	2019.3.x - 2020.1.x
2019.4.x	2019.3.x - 2019.4.x
2019.3.x	2019.3.x

Compatibility with Tableau Online

Tableau Online migration using the Content Migration Tool is in pre-release and under active development. To learn about the terms and conditions of Tableau's Pre-Release program,

see Tableau Environment Migrations on the [Pre-Release website](#).

Compatibility with Tableau content

The Content Migration Tool supports migrating workbooks and published data sources saved in the eight most recent versions of Tableau. While you can migrate existing data sources, only data sources that use the connection types in the table below can be changed and modified during migration. For more information, see Data Source Transformations in Migration Plans: Workbooks and Migration Plans: Published Data Sources.

Action Matrix	HortonWorks Hadoop Hive	Pivotal Greenplum Database
Action Vectorwise	HP Vertica	PostgreSQL
Amazon Athena	IBM DB2	Progress OpenEdge
Amazon Aurora	IBM Netezza	Salesforce
Amazon Redshift	Map R Hadoop Hive	SAP HANA
Aster Database	Microsoft Access	SAP Sybase ASE
Box	Microsoft Analysis Services	SAP Sybase IQ
Cloudera Hadoop	Microsoft Excel	Snowflake
Delimited Text File	Microsoft Excel Direct	Spark SQL
EXASOL	Microsoft OneDrive	Statistical File
Firebird	Microsoft SQL Server	Tableau extracts
Google Analytics	MySQL	Teradata
Google BigQuery	OData	Text File
Google Cloud SQL	Oracle	Web Data Connector
Google Drive	Oracle Essbase	Other Databases (ODBC)

Tableau Server on Linux Administrator Guide

Post-Installation

Limitations when migrating content

Before you start, make sure you understand the limitations when migrating content using the Content Migration Tool. For more information, see [Migration Limitations](#).

Create a migration plan

The Content Migration Tool walks you through migrating content across projects on a single site, to a new site on the same Tableau Server instance, and to sites that exist on different Tableau Server instances. The plan you create can be saved and used again for future migrations. For more information, see [Migration Plan Overview](#).

Upgrading Content Migration Tool

For instructions on how to upgrade Content Migration Tool, see [Upgrade Tableau Content Migration Tool](#).

Install Tableau Content Migration Tool

Installing Tableau Content Migration Tool is straightforward and easy.

Installation requirements

The Content Migration Tool tool is run from a Windows computer and can connect to Tableau Server 19.3 and later with a valid Advanced Management license. For more information about compatible versions, see [Compatibility with Tableau Server](#).

The requirements for the computer you install Content Migration Tool are:

- Microsoft Windows 8 or newer (x64)
- Intel Core i3 or AMD Ryzen 3 (Dual Core)
- 4 GB memory or larger
- Can connect to the source and destination Tableau Server computers. The source

and destination sites must have a valid Advanced Management license to migrate content.

- 2 GB HDD or larger. The drive where the `\temp` folder resides must have enough disk space to hold a copy of all content being migrated in a single migration. All content is stored locally on the disk and deleted when the migration is complete.
- Have enough free disk space to hold the application and its logs.

In addition, confirm that the REST API is enabled on Tableau Server (this is the default).

Use the `tsm configuration get -k api.server.enabled` command to confirm this. A return value of `true` means the REST API is enabled. To enable the REST API, use the `tsm configuration set` command. For more information, see `api.server.enabled`.

Installing Content Migration Tool

To install the Content Migration Tool:

1. Download the Content Migration Tool installer (`Tabcmt-64bit-<version>.exe`) for your version of Tableau Server from the [Tableau Server Management Releases](#) page.
2. Run the Content Migration Tool Setup program.
3. After reading the EULA, select **I agree to the license terms and conditions**, and click **Install**.
4. If the User Account Control dialog opens, click **Yes** to allow the installer to make changes.

Who can do this

A user with Administrator access on the machine.

Upgrade Tableau Content Migration Tool

Upgrading Tableau Content Migration Tool is straightforward and easy.

Tableau Server on Linux Administrator Guide

Before Upgrading

- Running the Content Migration Tool Setup program will overwrite the previous version.
- Content Migration Tool does not support side-by-side installation of previous versions.

Upgrading Content Migration Tool

To upgrade Content Migration Tool:

1. Download the latest Content Migration Tool installer for your version of Tableau Server from the [Tableau Releases](#) page.
2. Run the Content Migration Tool Setup program.
3. After reading the EULA, select **I agree to the license terms and conditions**, and click **Install**.
4. If the User Account Control dialog opens, click **Yes** to allow the installer to make changes.

Who can do this

A user with Administrator access on the machine.

Using Tableau Content Migration Tool

The following steps are designed to guide you through using the Tableau Content Migration Tool:

- Migration Plan Overview
 - Migration Plans: Servers
 - Migration Plans: Source Projects
 - Migration Plans: Workbooks
 - Migration Plans: Published Data Sources
 - Migration Plans: User Permissions Mapping
 - Migration Plans: Migration Scripts
 - Migration Plans: Plan Options

- Migration Rollback
- Using the Tableau Content Migration Tool Console Runner

Tableau Content Migration Tool Use Cases

Tableau Content Migration Tool as the name suggests, is primarily used for moving Tableau Server content from one site to another. However there are many features in the tool that makes it ideal for accomplishing several tasks related to content migration and maintenance.

Note: In many of the use cases we use the term migration to describe moving content from one environment, site, or project to another. However, technically the Content Migration Tool copies content and does not automatically delete or archive the original or source content.

The uses cases describe below are some common use cases that the Content Migration Tool is useful for:

Content Promotion

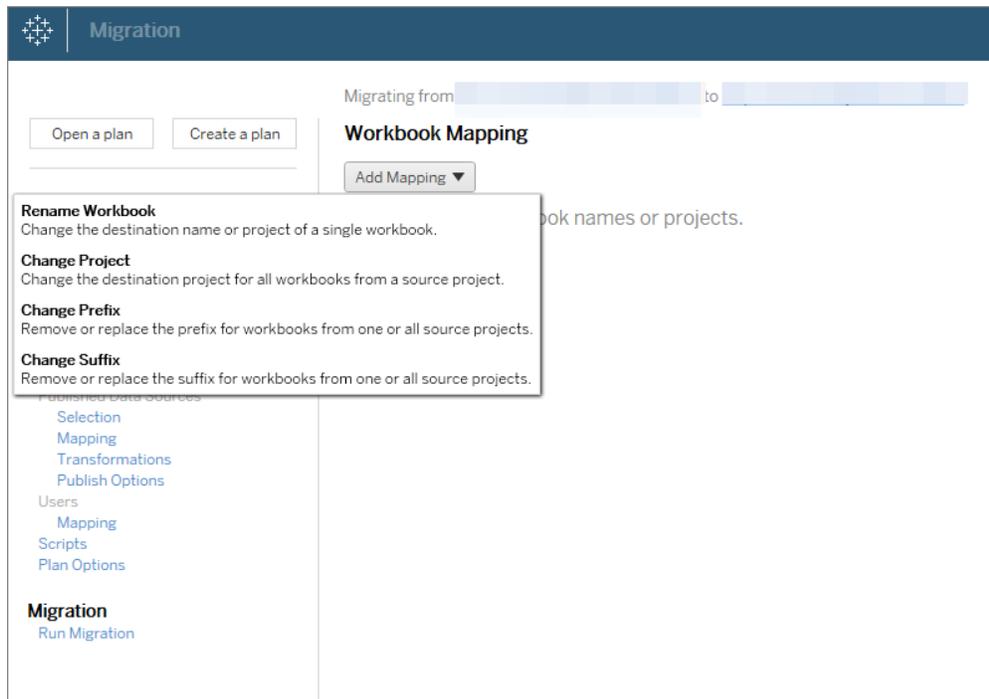
You can use the Content Migration Tool to create content in development sites and do routine migrations to staging or production environments.

Use the following steps to promote content to production environments:

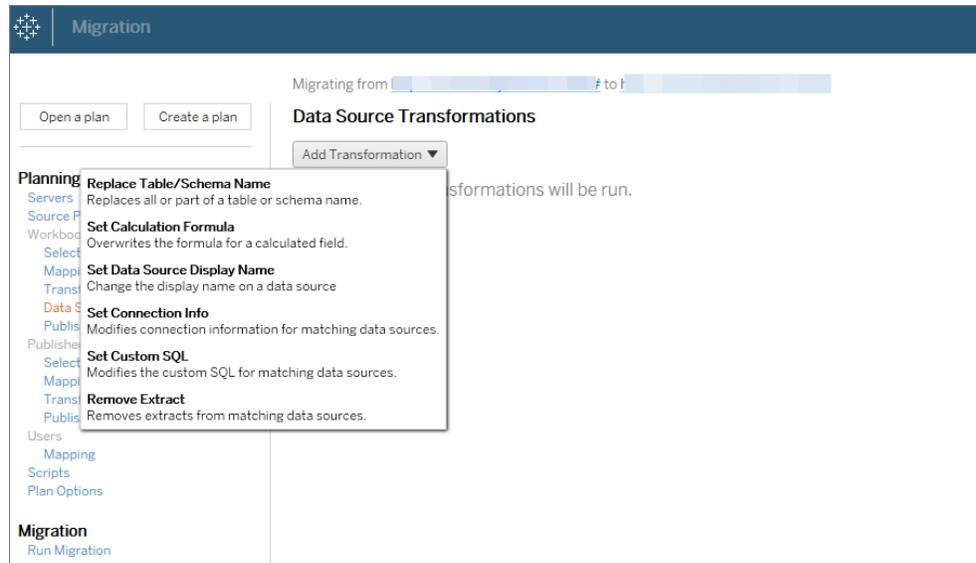
1. **Create a plan** and select the site used for development as the source and the staging or production site as the destination. For more information, see [Create a Plan in Migration Plan Overview](#) topic.
2. **Select the content** you want to migrate from your source site. You can select entire projects, specific workbooks and data sources, and user permissions. For more information, see [Planning in Migration Plan Overview](#) topic.
3. If you need to make any **changes or transformations** to the content during this

migration, you can configure that in the plan as well. This is referred to as **Mapping**. The types of mapping you can make include:

- **Changes to workbooks:** including renaming workbooks and changing destination project. For a full list of workbook transformations, see Migration Plans: Workbooks.



- **Changes to data sources:** Includes replacing table or schema names, setting calculation formulas, and setting connection information. For a full list of data source transformations, see Migration Plans: Workbooks (embedded data sources) and Migration Plans: Workbooks (published data sources).



- **Changes to Users:** Includes domain, user, and group name changes in the destination.
4. When you are ready, click on **Verify & Run** to end the Planning phase of your migration and prepare to run your plan.
 5. **To schedule** this to run on a regular basis, you can [script this as a job](#) using the Content Migration Tool Runner and schedule it. For more information on using the Content Migration Tool Runner, see, [Using the Tableau Content Migration Tool Console Runner](#).

Tailoring Content for Customers

When working in a consulting scenario, you can customize content for each of your customers using the Content Migration Tool. Each workbook functions as a template for your migration plan, allowing you to apply styling (text, images, etc.) and replace data sources for specific customers.

Use the following steps to customize content for your customers:

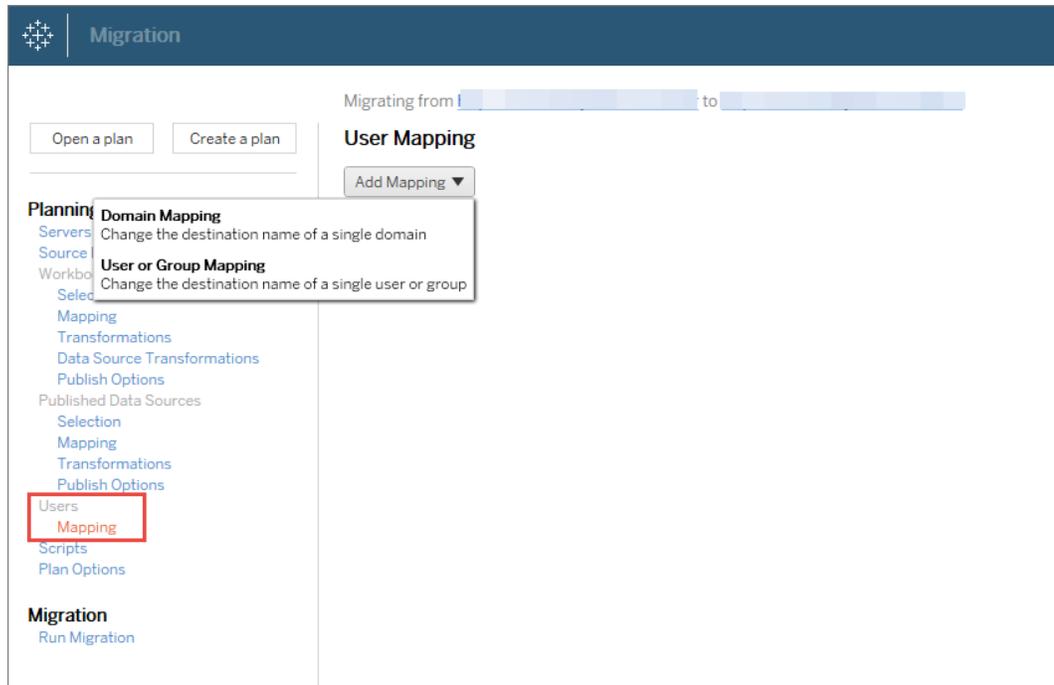
1. **Create a plan** and select the production site as the source and the customer site as the destination. For more information, see [Create a Plan in Migration Plan Overview](#) topic.
2. **In the Workbooks** step of the Migration Plan, use workbook mappings and transformations to customize your content. Below are two examples of frequently used transformations. For a full list of workbook transformations, see [Migration Plans: Workbooks](#).
 - To personalize content, you can use the **Replace Image** and **Replace Text** transformations to update the workbook with a customer's company name and logo.
 - When it comes to data sources, you can use the **Replace Table/Schema Name** or **Set Custom SQL** transformations to modify content for your customer.
3. **Verify and run** the plan. When you are ready, click on **Verify & Run** to end the Planning phase of your migration and prepare to run your plan.

Content Migration

You can use the Content Migration Tool to migrate content between Tableau Server deployments.

Use the following steps to migrate content between Tableau Server deployments:

1. **Create a plan** and **select the site** you want to migrate from as your source site. For more information, see [Create a Plan in Migration Plan Overview](#) topic.
2. **Select the content** you want to migrate from your source site. You can select entire projects, specific workbooks and data sources, and user permissions.
3. **Create user permissions mappings** to customize and secure content. For more information, see [Migration Plans: User Permissions Mapping](#).



4. **Verify and run** the plan. When you are ready, click on **Verify & Run** to end the Planning phase of your migration and prepare to run your plan.

Tips

- You can migrate your content in stages, test and validate content iteratively before final migration is complete. There is no server downtime when you use this method of migration. It can be a replacement for Site import/export.
- Content Migration Tool migration does not handle embedded credentials, subscriptions, and custom views. These will have to be migrated manually.

External Content Sharing

You can use the Content Migration Tool to share internal content with external collaborators, without allowing access to your site. This keeps your data secure and allows you to publish only select workbooks and data sources. Once content has been shared, collaborators sign in to their Tableau Server or Tableau Online site to view and make changes, without affecting content stored on your internal Server.

Before you continue, make sure the content you are sharing is compatible between the internal and external sites. The external site should typically be running the same release version (2019.X, 2020.X, etc) of Tableau Server or later. To learn more about compatibility, see [Make Workbooks Compatible Between Versions](#) in Tableau Desktop help.

Use the following steps to share content externally:

1. **Work with** the external Tableau Online or Tableau Server administrator to determine a user account that has publishing rights on the external site. You will use this user account to create the migration plan. For more information, see [Set Users' Site Roles and Permissions](#).
2. **Prepare internal content.** As a best practice, we recommend separating content on the internal server to a project, with locked permissions and strict governance rules. Workbooks and data sources should be clearly labeled to indicate the content is for external use. For more information, see [Use Projects to Manage Content Access](#).

Note: Content shared with external sites must use data extracts unless the data source is publicly accessible. For information about creating extracts and replacing data sources, see [Extract Your Data](#) and [Replace Data Sources](#) in Tableau Desktop help.

If you have implemented row level security, those data sources must be updated to reflect user filters and other details for the external site. For more information about row level security, see [Restrict Access at the Data Row Level](#) in Tableau Desktop help.

3. **Create a plan** and select the internal site as the source and the external Tableau Online or Tableau Server site as the destination. For more information, see [Create a Plan](#) in Migration Plan Overview topic.
4. **Select the content** you want to share with the external site. You can select entire

projects, specific workbooks and data sources, and user permissions.

5. **Verify and run** the plan. When you are ready, click on **Verify & Run** to end the Planning phase of your migration and prepare to run your plan.

Validating Database Migrations

This use case is when you intend to validate content after a migration of the underlying databases. One example of database migration is moving from SQL Server to Snowflake. CMT can help you validate the content built from both data sources is the same before you finalize your migration, but it cannot perform the actual database migration.

Use the following steps to validate database migrations:

1. **Create a plan** and select the Tableau Server site that you are going to use as your source. For more information, see Create a Plan in Migration Plan Overview topic.
2. **Select the content** you want to change the data source or database connections and select a new project as the destination. Let's call the **source** project as **Project A**, and the new or the **destination** project as **Project B**.
3. **Verify and run** the plan. When you are ready, click on **Verify & Run** to end the **Planning** phase of your migration and prepare to run your plan.
4. **Update the content** in **Project B** with the new database connections or replace the data sources. This needs to be done manually by authoring.
5. **Test each workbook** in **Project A** with the copy in **Project B** and review for any inconsistencies in the data due to the change in data source.
6. Once you have confirmed everything is working as expected, **overwrite the content** in Project A with the updated content in Project B.

Important! If the content already exists in the destination project and you do not select the overwrite option, the content will not be copied over to the destination project.

Geographical Content Migration

If you are maintaining a geographically distributed, multi-server environment, you will need some of this content to be shared and available across all the servers. This use case describes how to migrate content between Tableau Servers in different geographies. The Tableau Server can be in the same country or across continents.

Best Practices:

- We recommend migrating content in one direction only meaning Primary to Secondary. Here we use the term Primary to indicate the source Tableau Server and Secondary to indicate the destination. You can have one or more destination.
- We recommend prioritizing the content that you most need in multiple geographies.

Use the following steps to migrate between Tableau Server that are distributed in different geographies:

1. **Create a plan** on the Primary server. For more information, see [Create a Plan in Migration Plan Overview](#) topic.
2. **Select content** that you want to share between the Primary and Secondary.
3. **Verify and run** the plan. When you are ready, click on **Verify & Run** to end the Planning phase of your migration and prepare to run your plan.
4. **To schedule** this to run on a regular basis, you can [script this as a job](#) using the Content Migration Tool Runner and schedule it. For more information on using the Content Migration Tool Runner, see, [Using the Tableau Content Migration Tool Console Runner](#).
5. **Review the content** on the source **periodically** to determine if new items should be added to the migration plan.

Maintenance Tasks

You can use the Content Migration Tool to perform a variety of maintenance tasks.

Tagging stale content

Using the Tableau Content Migration Tool, you can manage archiving stale content. For example, you can build a plan that runs on a regular schedule that can automatically pick up content tagged as Stale Content and move it to an Archive project. After a certain amount of time, the content in this project can be purged from the system. For more information see, [Migration Plans: Workbooks](#).

Restoring content

You can use the Content Migration Tool to restore content removed (accidentally or purposefully) from the production Tableau Server with content from a backup Server. The restoration process is simple and does not require downtime, compared to restoring with a backup file.

Use the following steps to restore content from a backup Server:

1. **Create a plan** and select the backup Tableau Server as the source and the production Server as the destination. For more information, see [Create a Plan in Migration Plan Overview](#) topic.
2. **Select the content** you want to restore from the backup Tableau Server.
3. **Verify and run** the plan. When you are ready, click on **Verify & Run** to end the Planning phase of your migration and prepare to run your plan.
4. **Review the content** on the production Server.

Partial backup

If you follow a Disaster Recovery plan, the Content Migration Tool can be used to maintain a backup Tableau Server environment. You can use the Content Migration Tool to perform partial backups by transferring new content from production to the backup Tableau Server. Performing partial backups will keep your backup environment updated and ready for fail-over. If you have not configured a backup Tableau Server environment, see [Disaster Recovery for Tableau Server](#) whitepaper for more information

Note: Before you perform a partial backup, make sure you understand the Migration Limitations when using the Content Migration Tool. You may still need to periodically perform a full backup and restore to backup all Tableau Server content. For more information, see [Perform a Full Backup and Restore of Tableau Server](#)

Use the following steps to perform a partial backup of your content:

1. **Create a plan** and select the backup Tableau Server as the source and the production Server as the destination. For more information, see [Create a Plan in Migration Plan Overview](#) topic.
2. **Select the content** you want to backup. You can select entire projects, specific workbooks and data sources, and user permissions. To migrate only new content, make sure the publish options **Overwrite Newer Workbooks** and **Overwrite Newer Data Sources** are not selected. For more information, see [Migration Plans: Workbooks](#).
3. **Verify and run** the plan. When you are ready, click on **Verify & Run** to end the Planning phase of your migration and prepare to run your plan.
4. **To schedule** this to run on a regular basis, you can script this as a job using the Content Migration Tool Runner and schedule it. For more information on using the Content Migration Tool Runner, see, [Using the Tableau Content Migration Tool Console Runner](#).

Migration Plan Overview

Tableau Content Migration Tool creates a streamlined process for Tableau Server migrations. The easy-to-follow plan can be audited, is repeatable, and works via a batch process so any number of workbooks and data sources can be migrated in a simple and efficient process.

The Content Migration Tool will display contextual tips to walk you through creating or editing a migration plan, with an option to roll back a migration if you choose. Once you select the

source and destination sites, a summary of your migration will be displayed at the top of the screen as follows:

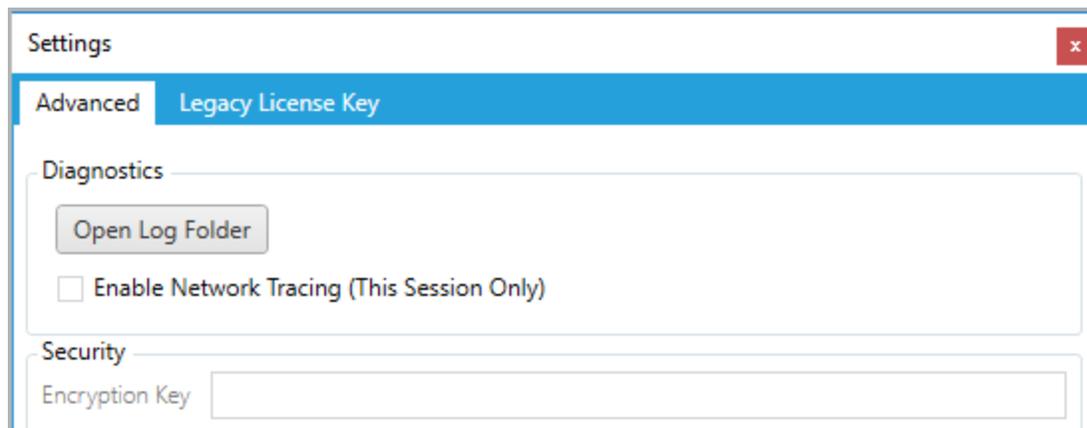
Migrating from <http://admin@win-vj23dhvudie:80/#/site/Accounting-sandbox> to <http://admin@win-vj23dhvudie:80/#/site/Accounting>

Limitations when migrating content

Before you start, make sure you understand the limitations when migrating content using the Content Migration Tool. For more information, see Migration Limitations.

Encryption keys

Each migration plan file is generated with an encryption key unique to the application that created the plan. Encryption keys can be shared if the migration plan needs to be run through an application that did not originally generate the file. When sharing encryption keys, you will need to overwrite the existing key in the application to run the migration plan. To view your encryption key, select **Help > Settings**.



If you will be using the Content Migration Tool Console Runner for migration plans, you must specify the encryption key using the `tabcmt-runner encryption` command before running the plan. For more information, see Using the Tableau Content Migration Tool Console Runner.

Tableau Server on Linux Administrator Guide

Migration process

Step 1: Start

The core of the migration process is creating a plan, which you can save and re-use for future migrations or modify and update as needed. The first step is choosing whether to create a new plan, or select a previously saved plan.

To create a new plan, click **Create New Plan**. If you already created a migration plan and want to use it, click **Browse for a Plan**.



By default, all of your saved migration plans will be stored in the `Tableau Content Migration Tool Plans` folder in your My Documents folder. All migration plans are saved with a `.tcmx` extension, with recently accessed plans listed separately to make them easy to select:

Recent Plans	Last Updated
DefaultToDefault2.tcmx C:\Users\jsmith\Documents\Tableau Content Migration Tool Plans\DefaultToDefault2.tcmx	7/12/2019 6:25 PM

You can select a recently accessed plan and duplicate it to modify the plan and save it as a new plan. Select the plan you want to copy and click **Duplicate**.



Step 2: Planning

The Content Migration Tool guides you through building or editing your migration plan in six steps.

Click on each step for detailed instructions:

- Migration Plans: Servers
- Migration Plans: Source Projects
- Migration Plans: Workbooks
- Migration Plans: Published Data Sources
- Migration Plans: User Permissions Mapping
- Migration Plans: Migration Scripts
- Migration Plans: Plan Options

Step 3: Migration

Once you have completed your plan, you are now ready to run the batch process for migration. When you reach the final step of the migration, a plan summary displays for your verification:

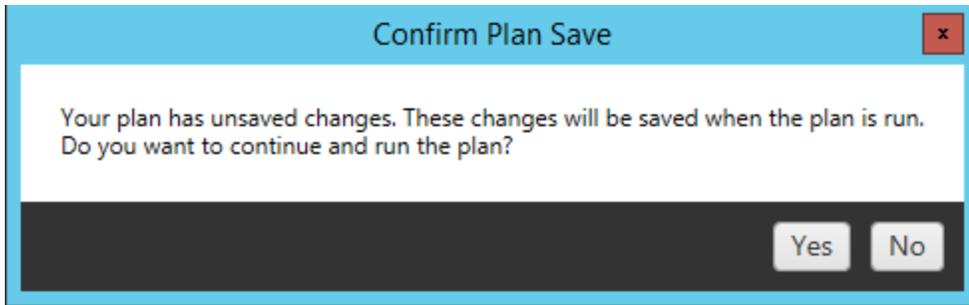
Review

Need help? 

Source:	http://admin@win-vj23dhvudie:80/#/site/Accounting-sandbox
Destination:	http://admin@win-vj23dhvudie:80/#/site/Accounting
Projects:	Default Mkt-Q3 Mkt-Q4
Workbooks:	Test Data - 2019 [Project: Mkt-Q4]
Published Data Sources:	All data sources
Auto Archive:	No

If you want to change any aspects of your plan, you can click on a section in the left sidebar to go directly to that phase. When you are ready, click **Run** to begin your migration.

When you click **Run**, the migration tool will prompt you about any unsaved elements of your plan. By default, any unsaved elements will be saved when you click **Yes**. Remember you can always keep your previous plan without making any changes by duplicating it during the Start phase of the migration process.

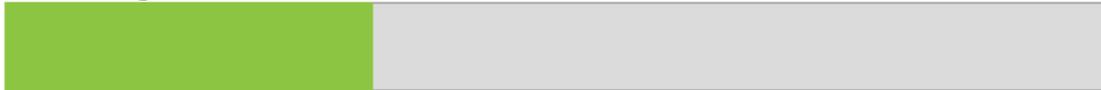


Your migration plan will run and a status bar displays for the overall plan progress and each workbook being sent to the destination server.

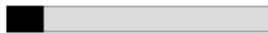
Running...

Need help? 

Downloading Source Workbooks



Test Data - 2019



When the plan finishes running, you can click the tabs at the bottom of the screen for more information about the migration.

Published workbooks

Published Workbooks details the newly published workbooks and the projects where they were migrated.

Published Workbooks			Published Data Sources	Output	Errors and Warnings
Workbook	Project				
Test Data - 2019	Mkt-Q4		View on Tableau Server		

Published data sources

Published Data Sources details the newly published data sources and the projects where they were migrated.

Output

The **Output** tab details the migration log of your plan.

Published Workbooks	Published Data Sources	Output	Errors and Warnings
<pre> ----- Tableau Content Migration Tool Version 2019.3.0 Build 20193.19.0712.1501+165d952 ----- Started : 7/15/2019 10:23:52 PM Plan : DefautToDefault2.tcmx File : C:\Users\jsmith\Documents\Tableau Content Migration Tool Plans\DefautToDe Migration ID : 1d60b6bb-9eaf-48a7-878d-53f1887009ee Source : http://admin@win-vj23dhvudie:80/#/site/Accounting-sandbox Destination : http://admin@win-vj23dhvudie:80/#/site/Accounting </pre>			

You can save this log by clicking **Save Log**.



Errors and warnings

The **Errors and Warnings** tab highlights any problems that occurred during the migration.

Published Workbooks	Published Data Sources	Output	Errors and Warnings
Message	Workbook or Data Source	Project	
⊗ Destination project [Mkt-Q4] does not exist. To avoid this error, enable automatic destination project creation in Options or create the project manually.			
⊗ Migration failed.			

You can correct these and rerun your plan. When you have completed your migration and saved your plan, click **Done** to finish.



Optional: Roll back

Content Migration Tool also offers you the ability to roll back to a previous migration. For more information, see Migration Rollback.

Who can do this

Tableau Server user with an Explorer role or higher. To migrate content, you must have View and Download Workbook/Save a Copy permissions on the source site, and Publishing rights for the destination site. For more information, see Permissions.

Migration Limitations

There are certain limitations to migrations using the Tableau Content Migration Tool.

Server-level configurations

The following server-level configurations from the source site will not be migrated to the destination site when using the Content Migration Tool:

- [Users](#)
- [Project permissions and locked state](#)
- [Site settings](#)

Workbooks with user configurations

The following user configurations from the source site will not be migrated to the destination site when using the Content Migration Tool:

- [Data-driven alerts](#)
- [Subscriptions](#)
- [Custom views](#)
- [Favorites](#)
- [Comments](#)

Compatibility with Tableau content

The Content Migration Tool supports migrating workbooks and published data sources saved in the eight most recent versions of Tableau. Workbooks and published data sources saved before version 2018.1.x are not supported by CMT. For more information, see [Getting Started with Tableau Content Migration Tool](#).

Tableau Online migrations

Tableau Online migration using the Content Migration Tool is in pre-release and under active development. Users can sign in to Tableau Online as the source or destination servers to create migration plans. To learn about the terms and conditions of Tableau's Pre-Release program, see [Tableau Environment Migrations on the Pre-Release website](#).

Embedded credentials

For security purposes, Tableau Server removes embedded credentials from data sources during the download process. To include embedded credentials when publishing to the

Tableau Server on Linux Administrator Guide

destination site, use the Set Connection Info data source transformation. For more information, see [Migration Plans: Published Data Sources](#).

Content metrics

Metrics created on the source site are not migrated to the destination site when using the Content Migration Tool. The historical values are removed from views, and users must recreate metrics on the destination site. For more information, see [Create and Troubleshoot Metrics](#).

Descriptions for projects, workbooks, and data sources

Descriptions for projects, workbooks, and data sources on the source site are not migrated to the destination site when using the Content Migration Tool. If you own the item or have the appropriate permissions, you can edit the item's description on the destination site. For more information, see [Add or edit descriptions](#) in Tableau Desktop help.

Thumbnails for workbooks and views

Workbooks and views that are migrated using the Content Migration Tool will retain their original thumbnails, even if the migration plan includes transformations that result in the views being rendered differently (for example, if data connections change). To update thumbnails, edit the workbook or view on the destination site and re-save it.

Tableau Prep flows

Tableau Prep flows published to the source site are not migrated to the destination site when using the Content Migration Tool. To run flows on a schedule using Tableau Prep Conductor, users must republish flows to the destination site with [Tableau Prep Builder](#).

Migration Plans: Servers

The first step when creating a migration plan in the Tableau Content Migration Tool is to sign in to the source and destination servers.

In the Servers section of the Planning phase, you'll sign in to the source and destination servers. The sites and projects you see when creating a migration plan are governed by the

permissions of the user credentials you use. You can only migrate content that the user has access to.

Note: Tableau Online migration using the Content Migration Tool is in pre-release and under active development. To learn about the terms and conditions of Tableau's Pre-Release program, see Tableau Environment Migrations on the [Pre-Release website](#).

The screenshot displays two side-by-side panels for selecting source and destination connections. Each panel contains a 'Sign in to Tableau' button.

Source	Destination
Select a connection for your source Sign in to Tableau	Select a connection for your destination Sign in to Tableau

Required permissions and licenses

The user account(s) used to sign in to the source and destination servers must have an Explorer role or higher, and the following permissions for the content you want to migrate:

- View
- Download Workbook/Save a Copy
- Optional: Administrator (to select workbooks, to access a user list)

Both the source and destination sites must have a valid Advanced Management license.

For more information, see [About Tableau Advanced Management](#).

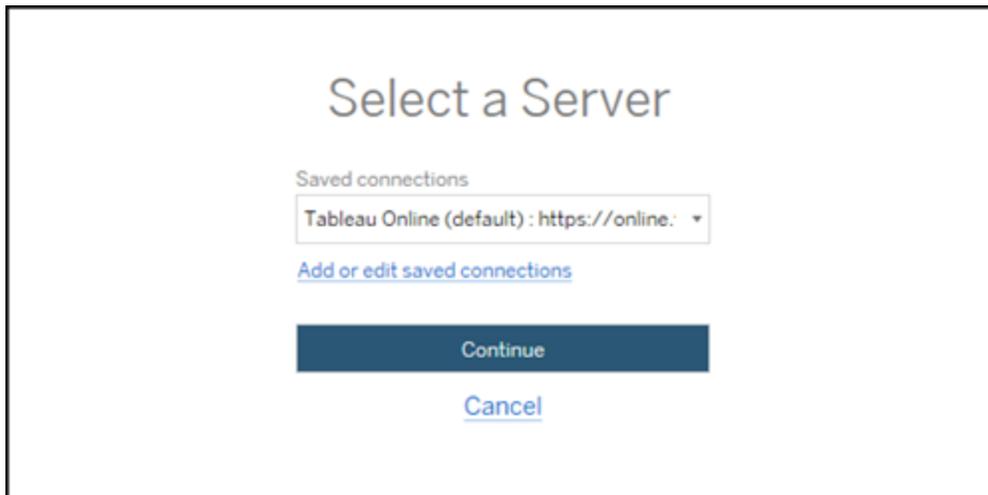
Step 1: Source

Here is the starting point of the migration. Sign in to your server, and then select which *site* to use as the source. Sites are independent silos of workbooks, data, and user lists created within Tableau to group related content for selected users. You can only migrate content from one site at a time.

Sign in to the source server

Use the steps below to sign in to Tableau Online or Tableau Server.

1. Click **Sign in to Tableau**.
2. In the **Select a Server** dialog window, select a saved connection and click **Continue**.



If no connections are available, click **Add or edit saved connections** to add a new connection. For more information, see [Saved connections](#).

3. Enter your username and password, and click **Sign In**.

If your server is configured for SAML or Single Sign-On, you are redirected to the Identity Provider sign-in page to complete the authentication process.

4. Select the site you want to use.



To change the source server or update your site selection, click **Select a different source**.

Step 2: Destination

Repeat the server sign-in process for the destination server (the server you are migrating content to).

If you are migrating your workbooks between two sites on the same Tableau Server, your sign-in credentials for the source and destination servers may be very similar or identical (including the Server URL). At a minimum, they will have a different site selected.

Saved connections

Using saved connections allow you to quickly sign in to the source and destination servers by creating a reusable server connection. When adding a saved connection, you must specify the preferred sign-in method for your server. As of 2021.2, the Content Migration Tool supports the following sign-in methods:

- **Personal access tokens:** Allows users to create long-lived authentication tokens for improved security, auditing, and automation of migration plans. Personal access tokens let users sign in without requiring interactive login in the Content Migration Tool. For more information, see [Personal Access Tokens](#).
- **Browser-based sign-in:** Users enter their credentials and complete authentication

through an embedded web browser. This option may be similar to how you usually authenticate to Tableau.

- **Username and password sign-in:** Users authenticate through the Content Migration Tool instead of an embedded browser window. This option passes credentials to the server using Tableau Server REST APIs. You can use username and password sign-in to authenticate to migration plans created before version 2020.3 and when troubleshooting issues that prevent the use of browser-based sign-in.

Add or edit saved connections

A link to **Add or edit saved connections** is displayed at the bottom of the Content Migration Tool, and when signing in to the source and destination servers. Clicking this link will open the **Manage Tableau Server Connections** window.

The screenshot shows a window titled "Manage Tableau Server Connections". On the left side, there is a list of connections: "Connection 1", "Connection 2", and "Connection 3". Above this list is a button labeled "New Connection". On the right side, there are several input fields and a group of radio buttons. The fields are: "Connection name", "Server URL", "Personal access token name", "Personal access token secret", and "Site name (from URL)". The radio buttons are: "Use personal access token" (which is selected), "Use browser-based sign-in", and "Use username/password sign-in". At the bottom right of the window is a "Close" button.

Use the steps below to add a saved connection:

1. On the **Manage Tableau Server Connections** window, click **New Connection**, or select an existing connection to make changes.
2. Enter a **Connection Name** (name to describe your server) and the **Server URL**.

If you don't include a prefix for the Server URL, the Content Migration Tool will use `http://`.

3. Select the sign-in method for your connection.

If you're using personal access tokens, see [Add saved connections with personal access tokens](#).

4. Click **Save**.

After you create a saved connection, it's listed in the Select a Server window next time you sign in to the source and destination servers.

Add saved connections with personal access tokens

Adding a saved connection with a personal access token requires more information than other sign-in methods. You will need to create a new personal access token on the source and destination servers to begin. Personal access tokens should not be shared between applications. For more information, see [Personal Access Tokens](#).

Creating personal access tokens

1. In a web browser, sign in to Tableau Online or Tableau Server.
2. At the top of the page, click your profile image or initials, and then select **My Account Settings**.
3. Under **Personal Access Tokens**, enter a descriptive name for your token in the

Token Name field, and then click **Create new token**.

4. In the resulting window, click **Copy to clipboard** and then close the window.
5. Paste the token secret to a file. Store the file in a safe location.

Adding personal access tokens

1. In the Content Migration Tool, click **Add or edit saved connections**.
2. On the **Manage Tableau Server Connections** window, enter a **Connection name** and the **Server URL**.

If you are connecting to Tableau Online, you must enter the full pod URL of your site. For example, `https://10ay.online.tableau.com`. Your pod is shown in the first portion of the site URL after signing in to Tableau Online.

3. Enter the **Personal access token name** and **Personal access token secret**, obtained when creating a personal access token in the previous section.
4. In the **Site name** field, enter the site name as it appears in the URL, without spaces. This is different than the friendly site name. For example, “Site A” would be “sitea” in a browser URL.
5. Click **Save**.

Step 3: Continue to the next step

After successfully signing in to both source and destination servers, click **Next** to continue to the Migration Plans: Source Projects section of the planning phase.

Who can do this

Tableau Server user with an Explorer role or higher. To migrate content, you must have View and Download Workbook/Save a Copy permissions on the source site, and Publishing rights for the destination site. For more information, see Permissions.

Migration Plans: Source Projects

The next step in creating a migration plan in the Tableau Content Migration Tool is to select the source projects. Source projects are the projects the workbooks and published data sources will be migrated from. The project or projects you choose impact which workbooks are available to migrate in the next step of the migration plan.

Step 1: Select your source project

There are two options when selecting source projects, **All Projects**, and **Specific Projects**:

Source ProjectsNeed help? 

All Projects Specific Projects

 Refresh

Workbooks and data sources from **all projects** will be available for migration.

The **All Projects** option selects all projects from the source site you specified in the Servers step. The **Specific Projects** option allows you to select specific projects from the source site.

Source ProjectsNeed help? 

All Projects Specific Projects

 Refresh

Select All (3 of 4 selected)

- Default
- Mkt-Q3
- Mkt-Q4
- Tableau Samples

You can select each project individually or use the **Select All** button and then clear selections for the projects you don't want to include. If you make any changes on the source site while on this step, you can use the **Refresh** button to update the projects list.

 Refresh

Tableau Server on Linux Administrator Guide

Step 2: Continue to the next step

Once you have selected source projects, you are ready to continue to the next step of the your migration plan, Workbooks. Click **Next**.



Migration Plans: Workbooks

Who can do this

Tableau Server user with an Explorer role or higher. To migrate content, you must have View and Download Workbook/Save a Copy permissions on the source site, and Publishing rights for the destination site. For more information, see Permissions.

Migration Plans: Workbooks

You have successfully signed into both your source server and your destination server, selected projects, and chosen sites where your workbooks reside and where you want them migrated. The next step is to prepare your workbooks for migration.

Note: If your workbooks or data sources include extracts, be sure you read and understand the information in Migrating Workbooks and Data Sources that use Extracts.

Step 1: Workbook Selection

All of the workbooks in the source site and projects appear on the **Workbook Selection** screen.

Workbook SelectionNeed help? 
 Specific Workbooks
 Rule Based
 All Workbooks

 Refresh

 Unselect All (9 of 9 selected)


Regional 18.1
 Superstore [Project: Tabl... 18.1
 Test Data - 2019 18.1

Superstore 18.1
 marketing test q3 18.1
 marketing test - sales b... 18.1

Back Next

If you make any changes to the workbooks in the source site while on this step, you can click **Refresh** to update the workbook listings. There are several different ways to select these workbooks.

Specific Workbooks Selection

There are three buttons in the **Specific** section. Any choices from the Basic section will immediately include the specifically selected workbook in the migration plan. Alternately, you can individually select specific workbooks by clicking on each one.

Select All

This button will select or clear selection of all the workbooks in the site. If additional workbooks are added to the site after the plan is saved, they will not be automatically added the next time the plan is used.

Display:

Thumbnails

The default view shows your workbooks in thumbnail previews to help you differentiate each of them. In this view, mousing over the thumbnail will show previews of the other worksheets and dashboards within that workbook.

List

The list view is a more succinct listing that also provides additional information, including Workbook Name, Project, Tableau Version, and Last Modified.

Clicking on any of the column headers will sort the workbooks appropriately. Also, mousing over any of the workbooks will also provide a floating preview of the worksheets and dashboards within that workbook. List view is particularly useful if you have a large amount of workbooks in a site.

Rule Based Selection

You can use **Rule Based** selection to choose workbooks based on specific criteria. Rule-based options will create workbook selection criteria to be used when the migration plan is run. Be aware that selecting "all" in any of the **Rule Based** options is different than the **Specific Workbooks** selection. A rule-based "all" selection will always include all workbooks, so any newly added workbooks are included in future migrations.

In projects

Tagged with [Click to add tag...](#)

Published by

The **Rule Based** radio button allows you to select workbooks by using the following options:

Workbooks in projects

This menu allows you to select workbooks from specific projects.

Workbooks tagged with

This menu allows allow you to select workbooks by their tags.

Workbooks published by

This menu allows you to select workbooks by their author.

With each option, you can select individually or multiple by clicking on the option next to each entry. All selected workbooks will appear in the **Selection Description** box.

All Workbooks Selection

The last option is to select the **All Workbooks** radio button, which selects all workbooks in all projects in the site.

Again, using the **All Workbooks** radio button is different than selected all of the workbooks using the **Specific Workbook** method, because it will use every workbook in the source site each time the migration plan is used in the future, as well.

Specific Workbooks Rule Based All Workbooks

 Refresh

All workbooks in all projects

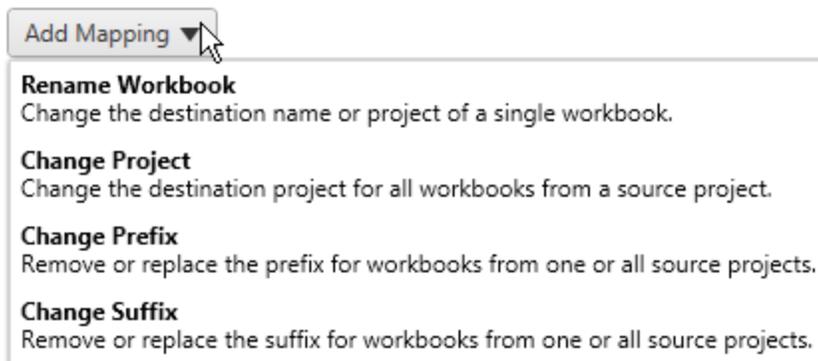
When you are satisfied with your workbook selections, click **Next**.



Step 2: Workbook Mapping

You can now map your selected workbooks from the source file to the destination file. Mapping allows you to rename source workbooks as they are migrated and choose different destinations. You can also add mapping to change the project, prefix, or suffix for the workbooks as well. Projects can be added to the Destination in this section as well.

If you make no changes here, then the selected workbooks will simply be migrated with the same name and into the same project as the source. If you have not defined in projects in your destination site, then they will be migrated into the Default project. To add workbook mapping click the **Add Mapping** button. The following options will appear in the mapping area.



Add Mapping ▼

- Rename Workbook**
Change the destination name or project of a single workbook.
- Change Project**
Change the destination project for all workbooks from a source project.
- Change Prefix**
Remove or replace the prefix for workbooks from one or all source projects.
- Change Suffix**
Remove or replace the suffix for workbooks from one or all source projects.

Rename Workbook

The screenshot shows the 'Edit Mapping' dialog box. It is divided into two main sections: 'Source' and 'Destination', connected by a right-pointing arrow. In the 'Source' section, there is a 'Project' dropdown menu and a 'Workbook Name' text input field. In the 'Destination' section, there is a 'Project' dropdown menu with a blue 'Add New' link to its right, and a 'Workbook Name' text input field. At the bottom right of the dialog, there are 'OK' and 'Cancel' buttons.

To rename the workbook(s) that are being migrated, this transformation will allow you to filter by project in the **Source** site and provide a drop-down list to select the desired workbook. In the **Destination** field, select which project you would like the workbook to be directed as well as entering the desired new name.

Change Project

The screenshot shows the 'Edit Mapping' dialog box. In the 'Source' section, the 'Project' dropdown menu is open and shows '(All Projects)' selected. The 'Destination' section has a 'Project' dropdown menu with a blue 'Add New' link to its right. At the bottom right of the dialog, there are 'OK' and 'Cancel' buttons.

By default, the workbooks are migrated to the same project in the destination. This mapping allows you to change the destination project for all workbooks from a source project.

Add Project

When renaming the workbook or changing the project, the **Add New** option allows you to create a destination project without having to sign in to the destination Tableau Server and

create the project manually. You can create projects and nested project using the **Add New** dialog box.

The 'New Project' dialog box is shown with the following fields:

- Destination:** http://admin@win-vj23dhvudie:80/#/site/Accounting
- Project Name:** (empty text input)
- Parent Project:** (dropdown menu)
- Description:** (empty text area)

Buttons: OK, Cancel

Change Prefix

The 'Edit Mapping' dialog box is shown with the following options:

- Source:**
 - Project:** (All Projects)
 - Prefix:** (empty text input)
- Options:**
 - Remove Prefix
 - Replace Prefix With (empty text input)

Buttons: OK, Cancel

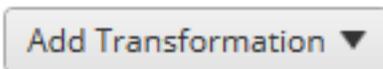
This allows you to remove or replace the prefix for workbooks from one or all source projects.

Change Suffix

Like the prefix mapping, you can remove or replace the suffix for workbooks from one or all source projects.

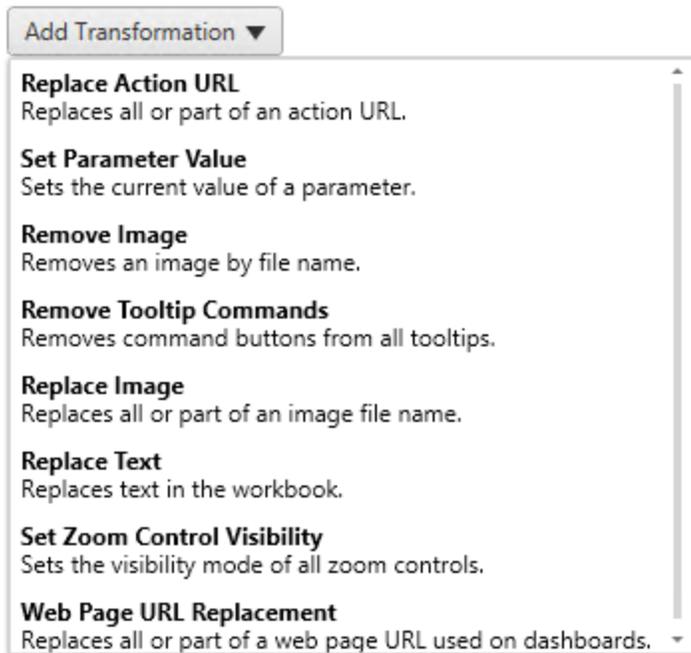
Step 3: Workbook Transformations

You can change and modify your workbooks by using the Transformation step.



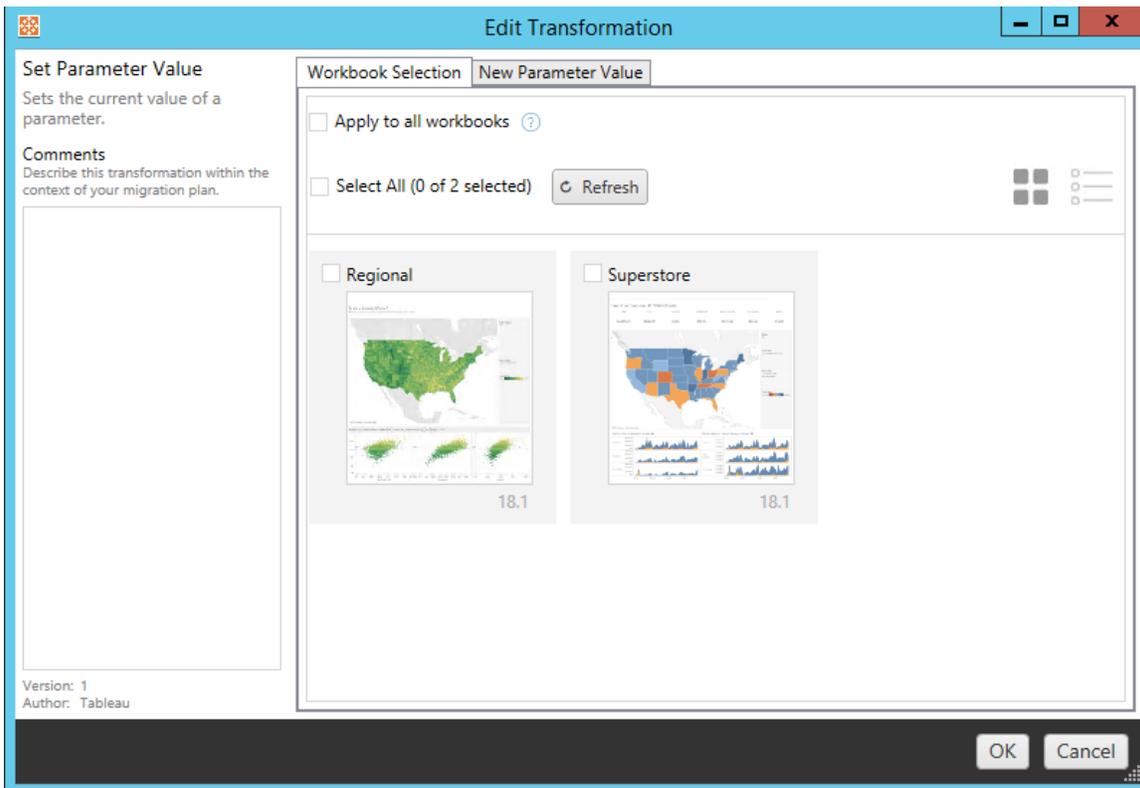
Transformations modify your workbooks in a specified manner. Additional transformations can be included via plug-ins or will be added in future versions of the application. Click on the **Add Transformation** drop-down menu to see the selection of Transformations currently available.

Workbook Transformations

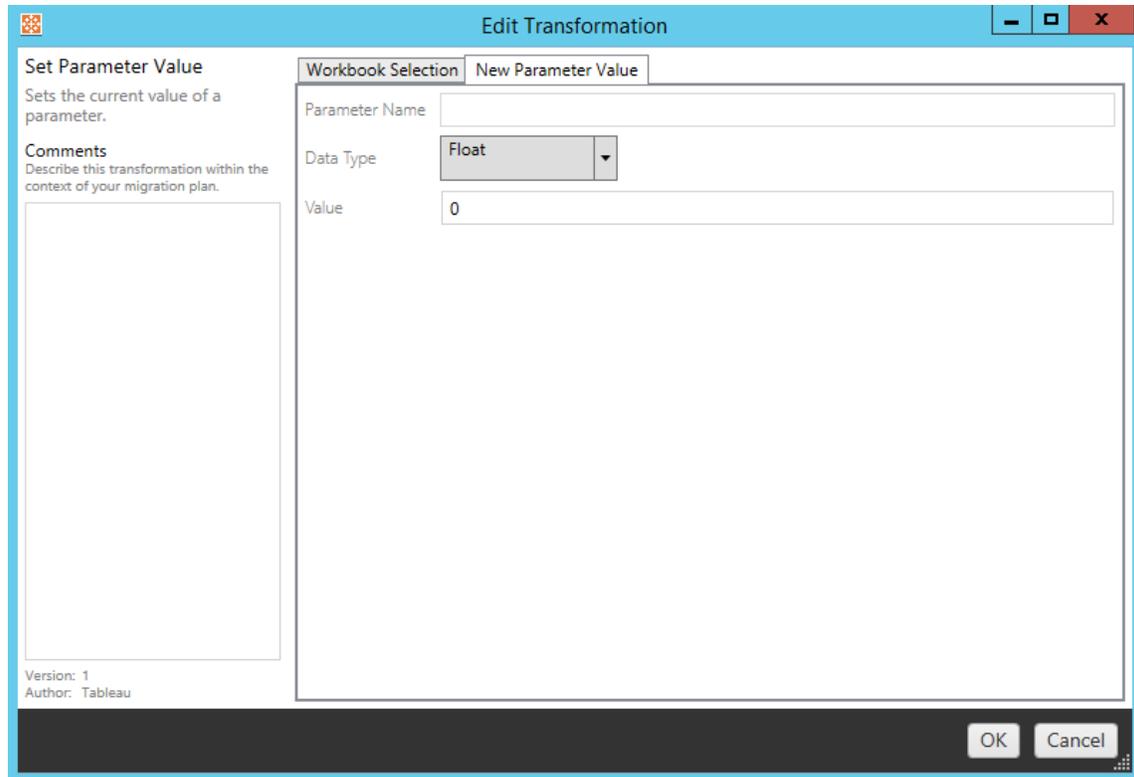


Selecting any of the transformations will bring up the Edit Transformation window, which will allow you to customize it to your selected workbooks. All transformations will be completed in the order that they are listed from top to bottom.

For all of the different types of transformations, there are two basic steps. The first step is to make your selection for the transformation. In this case, select the workbook(s) you want to transform. The selection area is similar to the Workbook Selection section of the Planning phase with all of the features of the Basic selection radio button: **Select/Unselect All**, **Refresh**, **Thumbnail Display**, and **List Display**. At the top of the list, you can select **Select All** workbooks, which is an option to automatically select all workbooks for future transformations. You can also **Refresh** the workbook display window to reflect any changes or updates to the source site.



The second step is to use the options tab to enter the specific selections for whichever transformation you select.



Each of the workbook transformations have different values to be entered on the options tab, and the tab will have different names, depending on the transformation you're editing:

Action URL Replacement

Replace part or all of an URL action inside of the workbook using this transformation. On the options tab, enter the text that should be matched and its replacement value.

Match
<input type="text"/>
Replacement
<input type="text"/>

Example:

URL: `www.exampledev.com`

Match: `dev`

Replacement: `Prod`

Result: `www.exampleProd.com`

Set Parameter Value

Define a new parameter. On the options tab, enter the name of the Parameter, the data type from the drop-down menu, and the value.

Parameter Name	<input type="text"/>
Data Type	<input type="text" value="Float"/>
Value	<input type="text" value="0"/>

Remove Images

Remove any images (such as a watermark) in the selected workbooks by entering in the file name on the options tab. There is an additional check box to receive a warning during migration if no image is found.

File Name	<input type="text"/>
<input checked="" type="checkbox"/>	Warn when no matching images are found in a workbook.

Remove Tooltip Commands

Remove all of the tooltip commands from the selected workbooks. There are no additional options to define for this transformation.

Replace Images

Replace any images in the selected workbooks. On the options tab, enter the file name of the old image and file name of the replacement image. You can use a local file path or URL to replace images.

File Name	<input type="text"/>
Replacement Image URL	<input type="text"/>
<input checked="" type="checkbox"/> Warn when no matching images are found in a workbook.	

Example:

File Name: C:\file\image.png

Replacement Image URL: <http://www.exampledev.com/replacementImage.png>

Zoom Control Visibility

Set the visibility mode from the drop-down menu: **Automatic**, **Show on Hover**, or **Hide** on the options tab.

Visibility Mode	Automatic ▼
-----------------	-------------

Web Page URL Replacement

Replace part or all of a web page URL used on dashboards using this transformation. On the options tab, enter the text that should be matched and its replacement value.

Match	<input type="text"/>
Replacement	<input type="text"/>

Example:

URL: `www.exampledev.com`

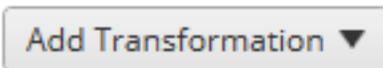
Match: `dev`

Replacement: `Prod`

Result: `www.exampleProd.com`

Step 4: Data Source Transformations

The next step in planning your workbooks for your enterprise migration are your data source transformations. It is similar in function to the Workbook Transformations step. These are for data sources that are packaged within the workbooks. Published data sources are handled in a different step in the process.



Click on the **Add Transformation** drop-down menu and the following options will appear:

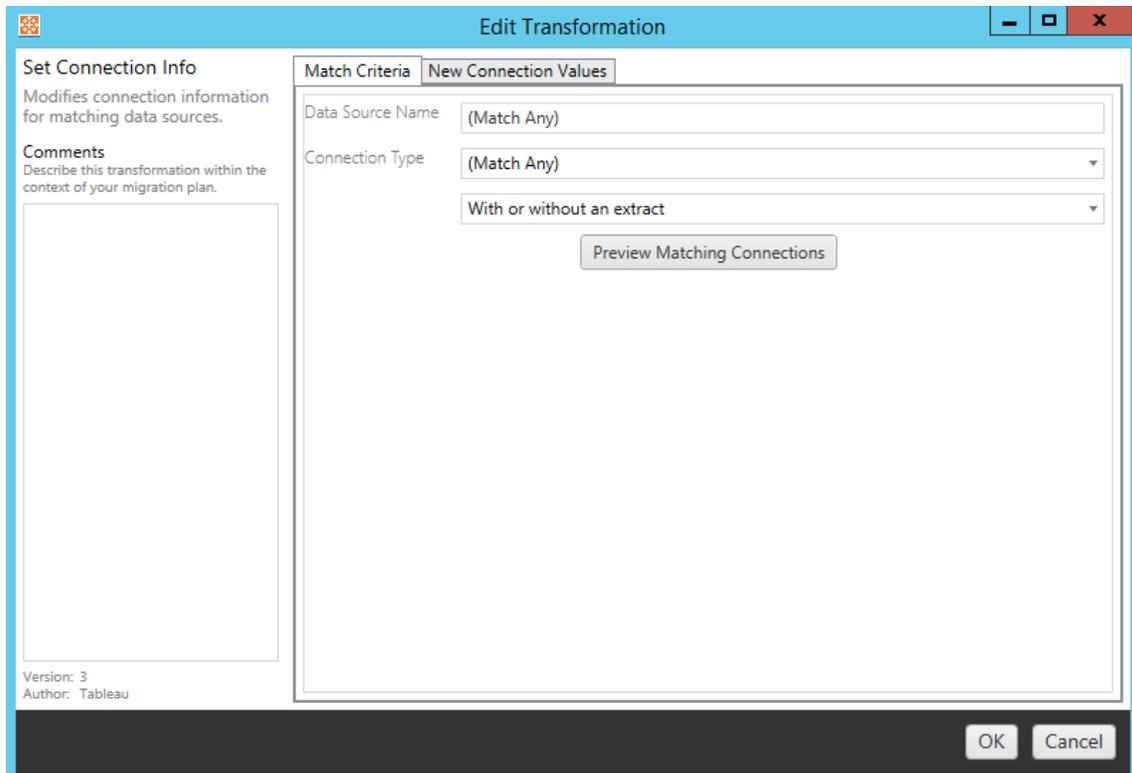
Data Source Transformations

A screenshot of the "Add Transformation" drop-down menu. The menu is displayed as a white box with a thin border, containing several options. At the top left of the menu is a smaller version of the "Add Transformation" button. The options listed are:

- Replace Table/Schema Name**
Replaces all or part of a table or schema name.
- Set Calculation Formula**
Overwrites the formula for a calculated field.
- Set Data Source Display Name**
Change the display name on a data source
- Set Connection Info**
Modifies connection information for matching data sources.
- Set Custom SQL**
Modifies the custom SQL for matching data sources.
- Remove Extract**
Removes extracts from matching data sources.

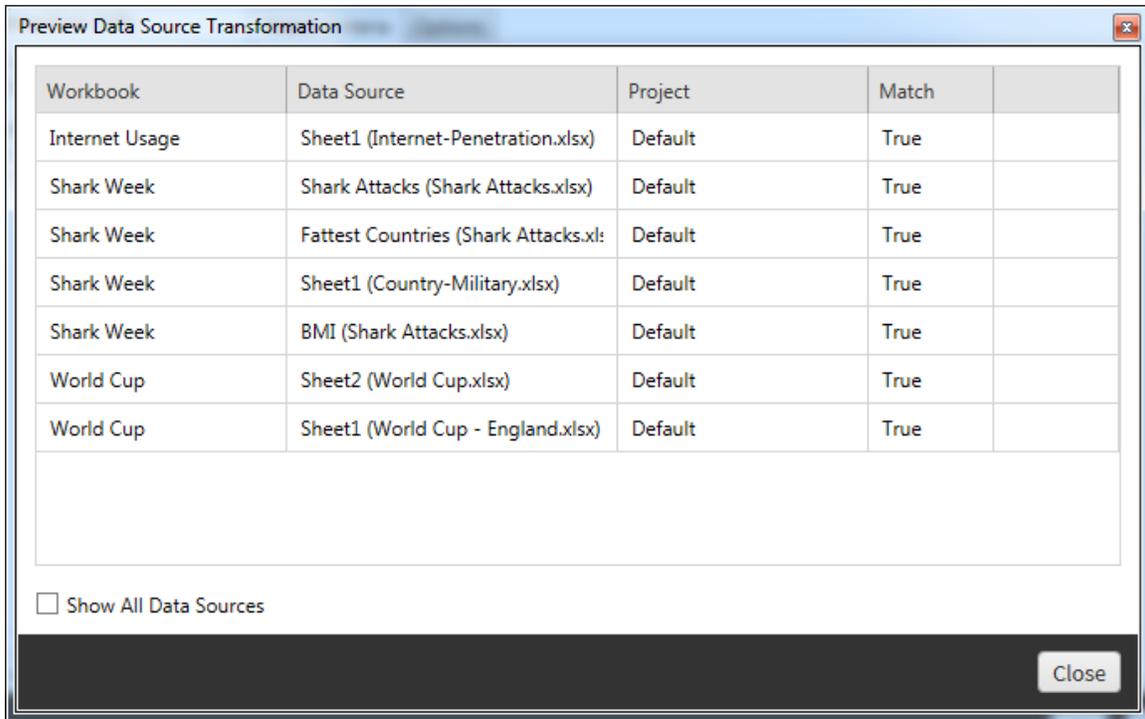
Selecting any of the data source transformations will bring up the Edit Transformation window, which will allow you to customize it to your selected data sources. All transformations will be completed in the order that they are listed from top to bottom.

For all of the different types of data source transformations, there are two basic steps. The first step is to enter in the match criteria for the desired data source. Depending on which connection type you select, more fields will appear on the **Match Criteria** tab.



Click on the **Preview Source Connections** to find any connections that match the criteria entered.

Tableau Server on Linux Administrator Guide

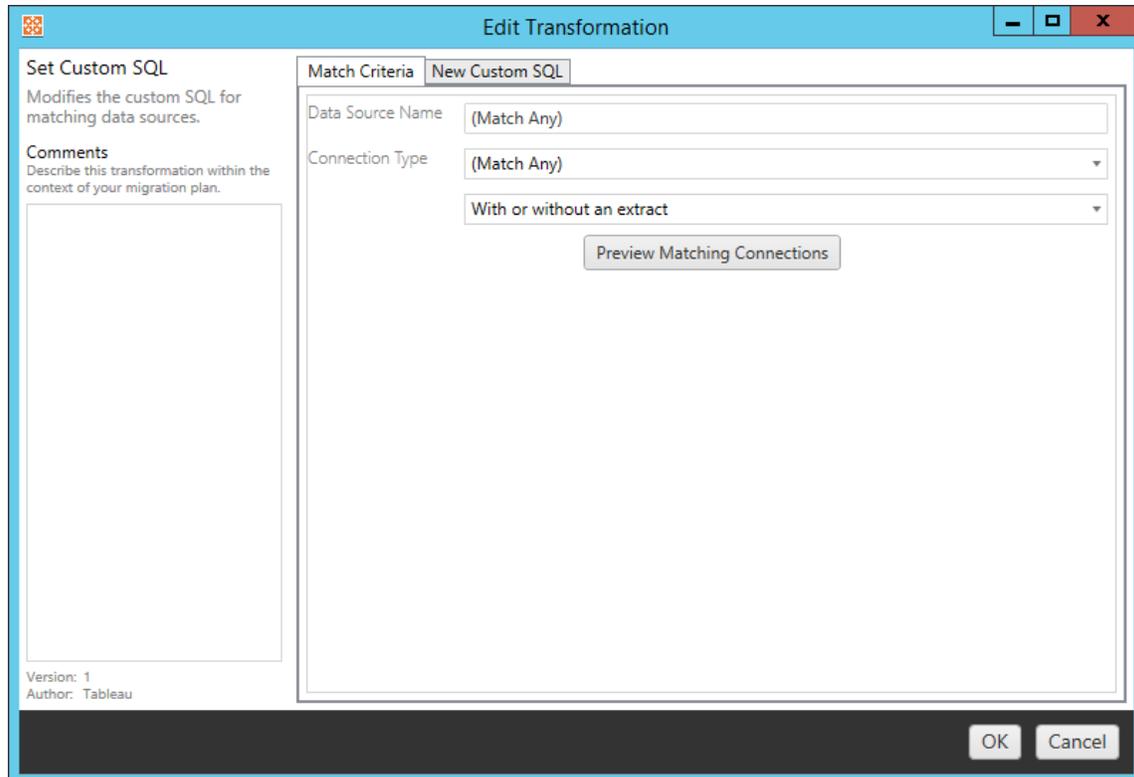


Workbook	Data Source	Project	Match	
Internet Usage	Sheet1 (Internet-Penetration.xlsx)	Default	True	
Shark Week	Shark Attacks (Shark Attacks.xlsx)	Default	True	
Shark Week	Fattest Countries (Shark Attacks.xl:	Default	True	
Shark Week	Sheet1 (Country-Military.xlsx)	Default	True	
Shark Week	BMI (Shark Attacks.xlsx)	Default	True	
World Cup	Sheet2 (World Cup.xlsx)	Default	True	
World Cup	Sheet1 (World Cup - England.xlsx)	Default	True	

Show All Data Sources

Close

The second step is to use the options tab to enter the specific selections for whichever transformation you select.



Each of the data source transformations have different values to be entered on the options tab:

Set Calculation Formula

On the options tab, you can replace the calculation for a column.

Column Name	
Formula	

Set Connection Info

On the options tab, enter in the file path for the new data source.

File Path

Set Custom SQL

On the New Custom SQL tab, enter the name of the custom SQL query you want to modify for **Match Query Name**. The query name must match the custom SQL query name from the physical layer of the data source. If these names don't match, the transformation will fail. For more information about data modeling and the physical layer, see The Tableau Data Model.

After entering the query name, enter the desired **Custom SQL** in the text field. Be aware that custom SQL can negatively impact the performance of your workbooks if improperly used.



Match Query Name

Custom SQL

Remove Extract

There is no options tab for this transformation, simply enter in the **Match Criteria** information and the extract will be removed during migration.

In addition, on each of the transformations you can enter notes in the **Comments** section on the left-hand side of the **Edit Transformation** window.

Step 5: Publish Options

The final step in the Workbooks phase is to select publish options and create transformations for tags, extract refresh schedules, and permissions.

Workbook Publish Options

- Reset Dashboard Selections [?](#)
- Overwrite Newer Workbooks [?](#)
- Copy Workbook Permissions [?](#)
- Copy Extract Refresh Schedules [?](#)

Add Option ▼

No additional publish options.

Reset Dashboard Selections

This option deselects all objects on dashboards.

Overwrite Newer Workbooks

If checked, a workbook will be migrated even if it will overwrite a workbook that has been created at the same time or more recently than the moved workbook

Copy Workbook Permissions

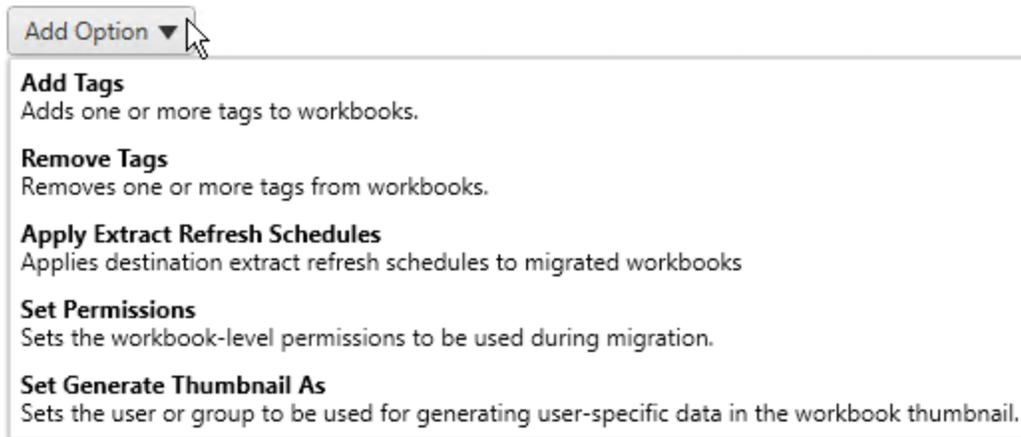
When selected, the migration tool will attempt to match source workbook permissions as closely as possible.

Copy Extract Refresh Schedules

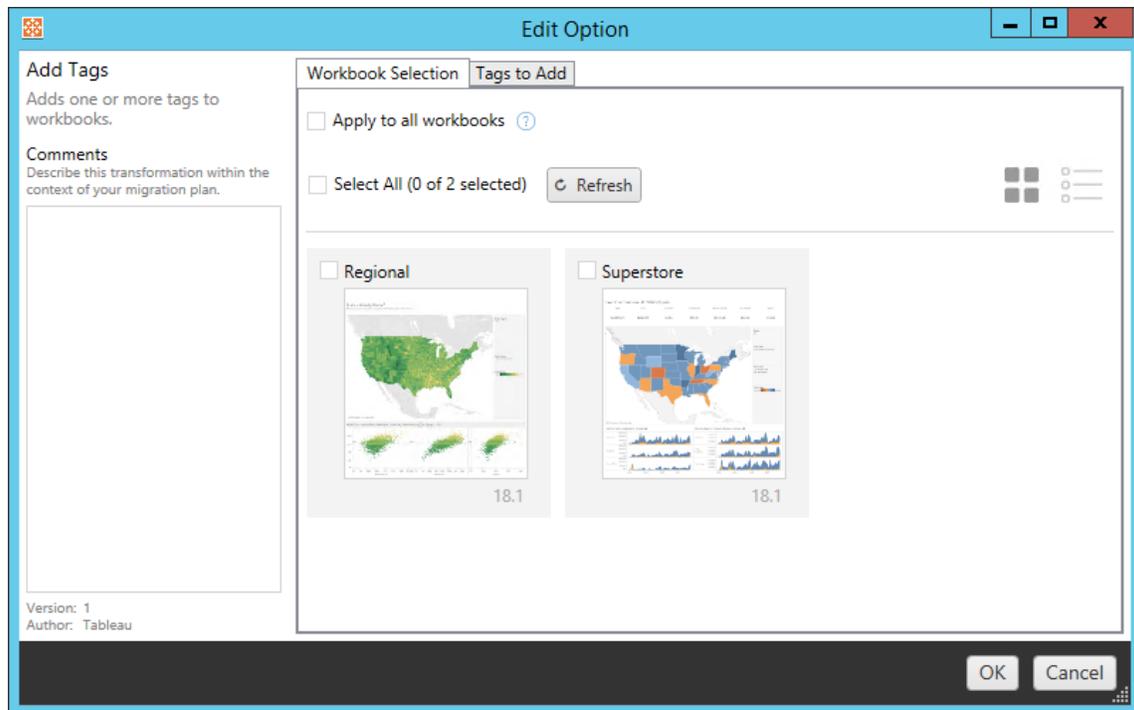
When selected, the migration tool will attempt to set the destination workbook extract refresh schedule(s) to schedules matching the source's name.

Add Option

Click on the **Add Option** drop-down menu for the different types of transformations you can add:



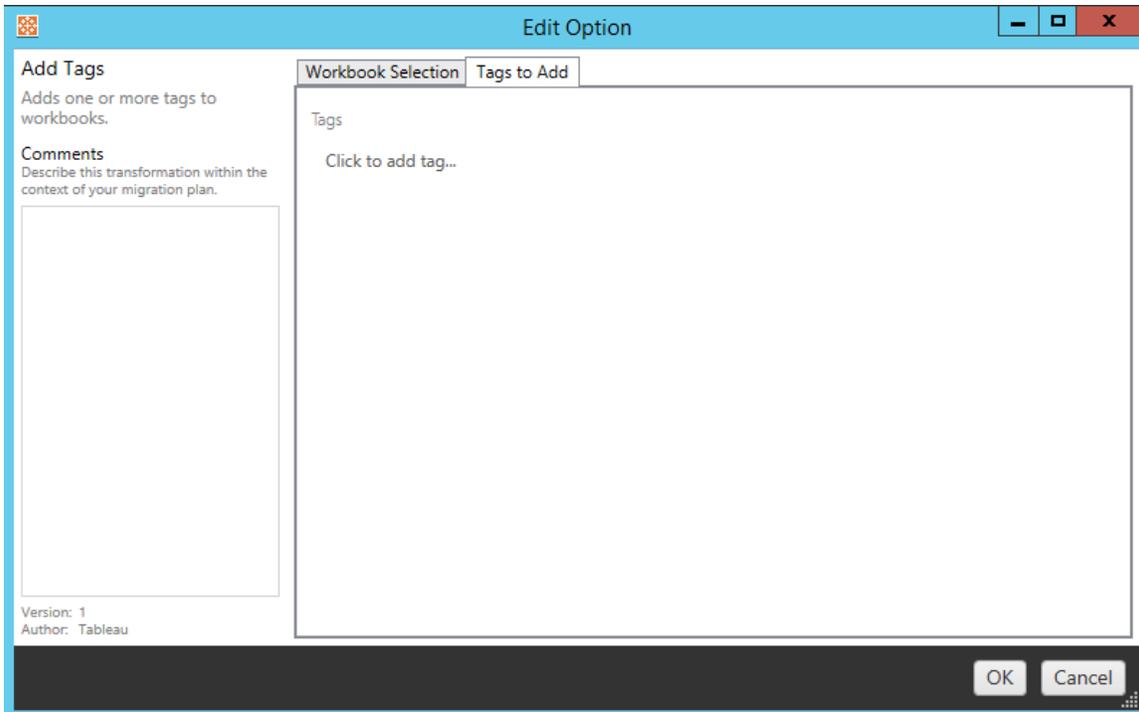
For all of the different types of transformations, there are two basic steps. The first step is to make your selection for the transformation. In this case, select the workbook(s) you want to transform. The selection area is similar to the Workbook Selection section of the Planning phase with all of the features of the Basic selection radio button: **Select/Unselect All**, **Refresh**, **Thumbnail Display**, and **List Display**. At the top of the list, you can select **Select All** workbooks, which is an option to automatically select all workbooks for future transformations. You can also **Refresh** the workbook display window to reflect any changes or updates to the source site.



The second step is to use the options tab to enter the specific selections for whichever transformation you select. **Note:** The options tab will have different names, depending on which transformation you are editing.

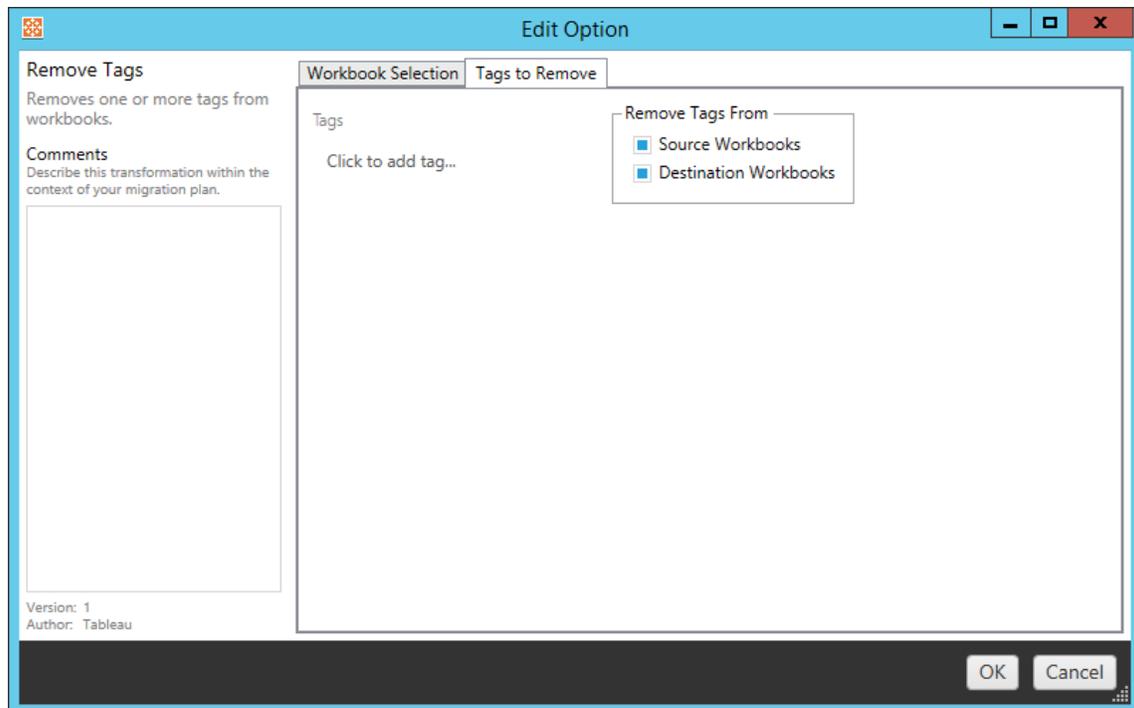
Add Tags

This allows you to add one or more tags to the workbook. If you hover your mouse over a previously entered tag, a blue “X” will appear to allow deletion.



Remove Tags

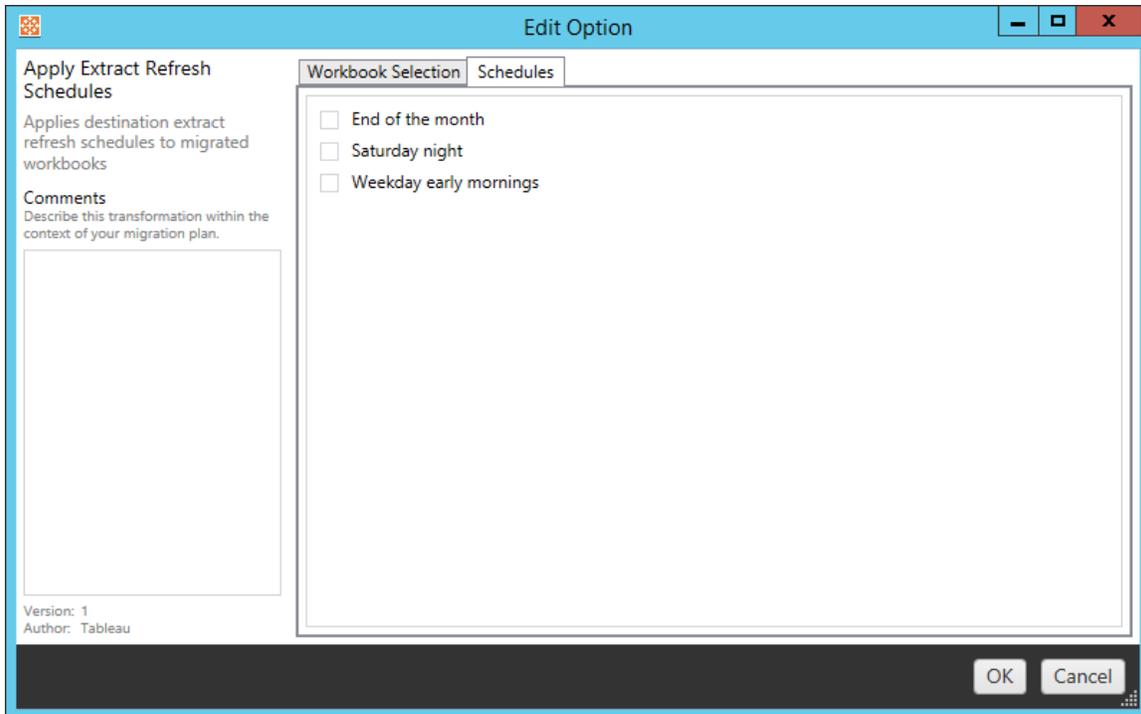
This allows you to add one or more tags to the workbook. If you hover your mouse over a previously entered tag, a blue “X” will appear to allow deletion. You can also choose to remove the tag from the source or destination workbooks.



Apply Extract Refresh Schedules

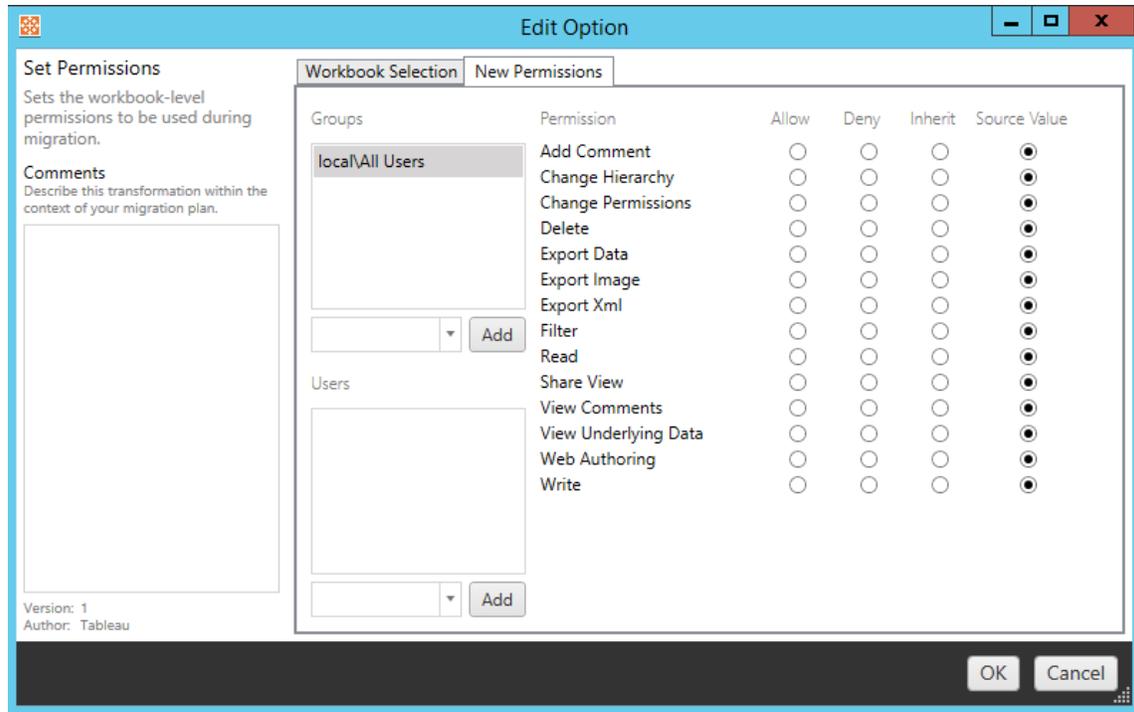
Here you can apply destination extract refresh schedules to migrated workbooks.

Note: The list of schedules generated are from the destination.



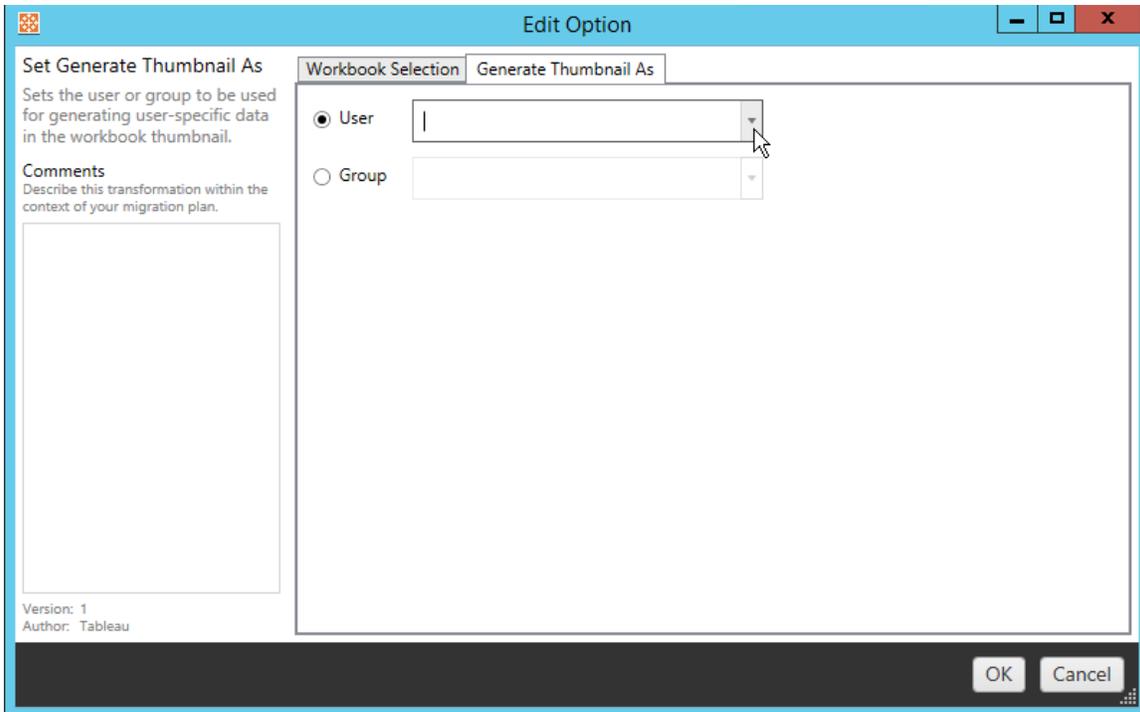
Set Permissions

This transformation is to edit the permissions for the selected workbooks. Enter in a Group or User and then click **Add**. Adjust the permissions as desired. The four different options are to **Allow** the permission, **Deny** the permission, **Inherit**, or to keep the **Source Value**.



Set Generate Thumbnail As

This allows you to set the **User** or **Group** to be used for generating user-specific data in the workbook thumbnail after being migrated. Each option has a drop down to select the desired user or group.



Step 6: Continue to Next Step

You have completed the workbooks section of the planning phase of your migration. Click **Next** to continue.



You will now move on to the Migration Plans: Published Data Sources section of the planning phase.

Who can do this

Tableau Server user with an Explorer role or higher. To migrate content, you must have View and Download Workbook/Save a Copy permissions on the source site, and Publishing rights for the destination site. For more information, see Permissions.

Migration Plans: Published Data Sources

The next step of creating a migration plan in the Tableau Content Migration Tool is to select, map, and add any transformations to your published data sources. The process is very similar to the Workbooks step of the planning phase, particularly the data source mapping step.

Note: If your workbooks or data sources include extracts, be sure you read and understand the information in *Migrating Workbooks and Data Sources that use Extracts*.

Step 1: Selection

Starting the Published Data Sources phase of the migration plan, you'll select any data sources you want to include in the migration plan:

Data Source Selection Need help? 

Specific Data Sources
 Rule Based
 All Data Sources
 Refresh 

Unselect All (1 of 1 selected)

	Name	Project
<input checked="" type="checkbox"/>	Sheet1 (state_plates)	Mkt-Q3

The data sources will only be selected at the moment of migration. You have two methods of selection. You can use the **Specific Data Sources** radio button to click on one or more published data sources. You can use the **Refresh** button to reload the list of published data sources available.

The second option is to select the **All Data Sources** radio button, which true to its name, selects every data source in the selected source server project.

Step 2: Mapping

The next step is to map your source data sources to the new destination. This is similar in functionality to mapping workbooks.

Data Source Mapping

Need help? 

Add Mapping ▼

No changes to data source names or projects.

If you make no changes here, then the selected data sources will simply be deployed with the same name and project as the source. To add data source mapping click **Add Mapping**. The following options will appear in the mapping area.

	Name	Project	Destination Name	Destination Project
Delete	(All Selected Data S ▼	<input type="text"/>	(Same As Source)	<input type="text"/>

The entry has the following options:

Delete

Clicking the **Delete** link will delete this mapping entry.

Name

Use the **Name** menu to select the data source you wish to map. You can select **(All Selected Data Sources)** to choose all of the data sources.

Project

The **Project** is the project of the associated data source names.

Destination Name

By default, the Content Migration Tool will use the same **Destination Name(Same As Source)**, keeping the original name in the Source file, but you can type in a new name here for the destination folder.

Destination Project

If projects have been defined in your destination Site, then you can choose which project to place your migrated workbooks or you can click on the **Add New** option to create a new project within the destination site. You can create different project destinations for individual data sources.

	Name	Project	Destination Name	Destination Project
Delete	(:d Data Sources) ▾	Default ▾	(Same As Source)	Default ▾
				Add New Accounting Default Sales

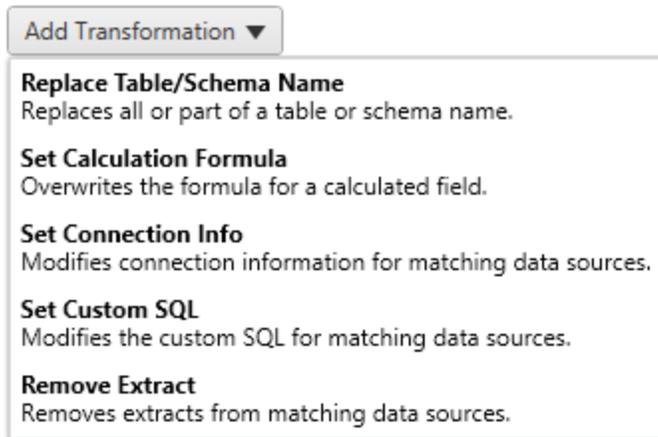
In most situations if multiple mapping entries exist for a single data source, a validation error will be displayed and must be fixed to continue. There is one important exception to this – a data source may match both a specific selection and a project-wide mapping entry. In this instance, the more specific entry will be used.

When you have completed all of the data source mapping necessary, click **Next** to continue.



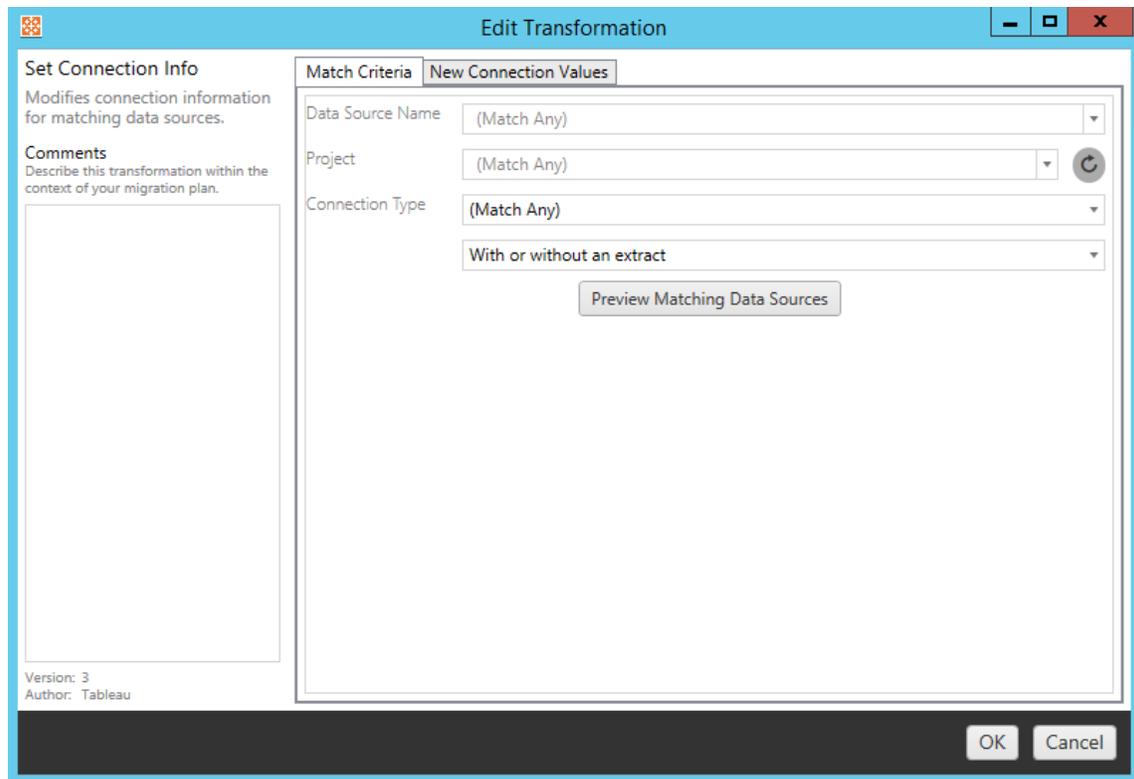
Step 3: Data Source Transformations

You can modify your data sources by using the transformation option. Click **Add Transformation** to see the list of transformations available.



Selecting any of the data source transformations opens the **Edit Transformation** window. Use this to customize your selected data sources. Transformations are executed in the order they are listed, from top to bottom.

For most data source transformations, there are two basic steps. The first step is to enter in the **Match Criteria** for the desired data source. Depending on which connection type you select, more fields will appear on the **Match Criteria** tab.



The second step depends on which transformation type you are adding. Each of the data source transformations have different values to be entered on the second tab.

You can add notes in the **Comments** section on the left side of the **Edit Transformation** window for each of the transformations you add.

Replace Table/Schema Name

On the **Options** tab, you can replace all or part of a table or schema name.

Set Calculation Formula

On the **New Calculation Formula** tab, you can replace the calculation for a column.

Column Name	<input type="text"/>
Formula	<input type="text"/>

Set Connection Info

On the **New Connection Values** tab, there will be different fields here based on the Connection Type you selected and perhaps even no fields needed at all.

File Path	<input type="text"/>
-----------	----------------------

Set Custom SQL

On the **New Custom SQL** tab, enter in any custom SQL in the text area desired. You should be aware that custom SQL can negatively impact the performance of your workbooks if improperly used.

Custom SQL	<input type="text"/>
------------	----------------------

Remove Extract

There is no **Options** tab for this transformation. Type the **Match Criteria** information and the extract will be removed during migration.

Use Tableau Bridge

There is no **Options** tab for this transformation. Type the **Match Criteria** information and data sources that are within a private network (inaccessible to the public internet) will be allowed to refresh using Tableau Bridge.

The destination Tableau Online site must have Tableau Bridge configured before migrating data sources. For information about Tableau Bridge, see [Use Tableau Bridge](#) in Tableau Online help. After the migration, data sources will need to be assigned a refresh schedule through Tableau Online.

Step 4: Publish Options

The final step in the Published Data Source phase is to create transformations for permissions and tags and finalize the publish options specific to the data sources.

Data Source Publish Options

Need help? 

Overwrite Newer Data Sources 

Copy Data Source Permissions 

Copy Extract Refresh Schedules 

Add Option ▼

No additional publish options.

The following checkboxes offer the following additional options:

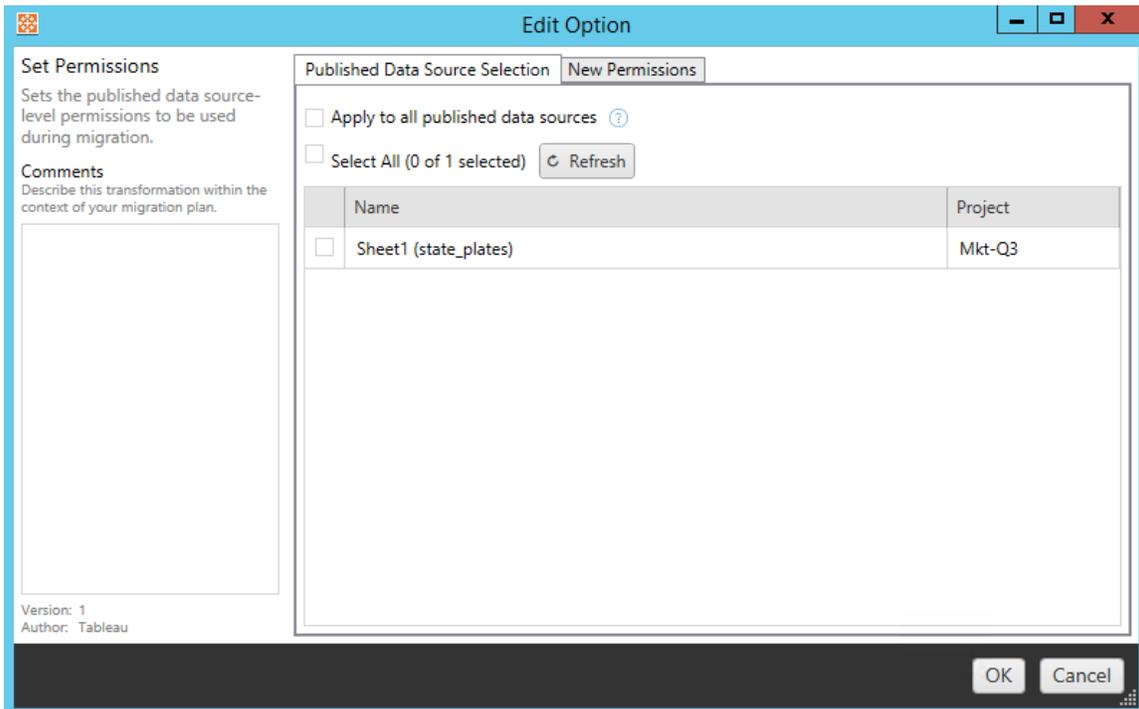
- **Overwrite Newer Data Sources:** If checked, a data source will be deployed even if it will overwrite a data source that has been updated more recently.
- **Copy Data Source Permissions:** If checked, deployment will attempt to make source published data source permissions as closely as possible.
- **Copy Extract Refresh Schedules:** If checked, deployment will attempt to set the destination data source extract refresh schedules to schedules matching the source's name.

Click on **Add Options** for the four different types of transformations you can add:

Add Option ▼

- Remove Tags**
Removes one or more tags from published data sources.
- Add Tags**
Adds one or more tags to published data sources.
- Apply Extract Refresh Schedules**
Applies destination extract refresh schedules to migrated data sources
- Set Permissions**
Sets the published data source-level permissions to be used during migration.

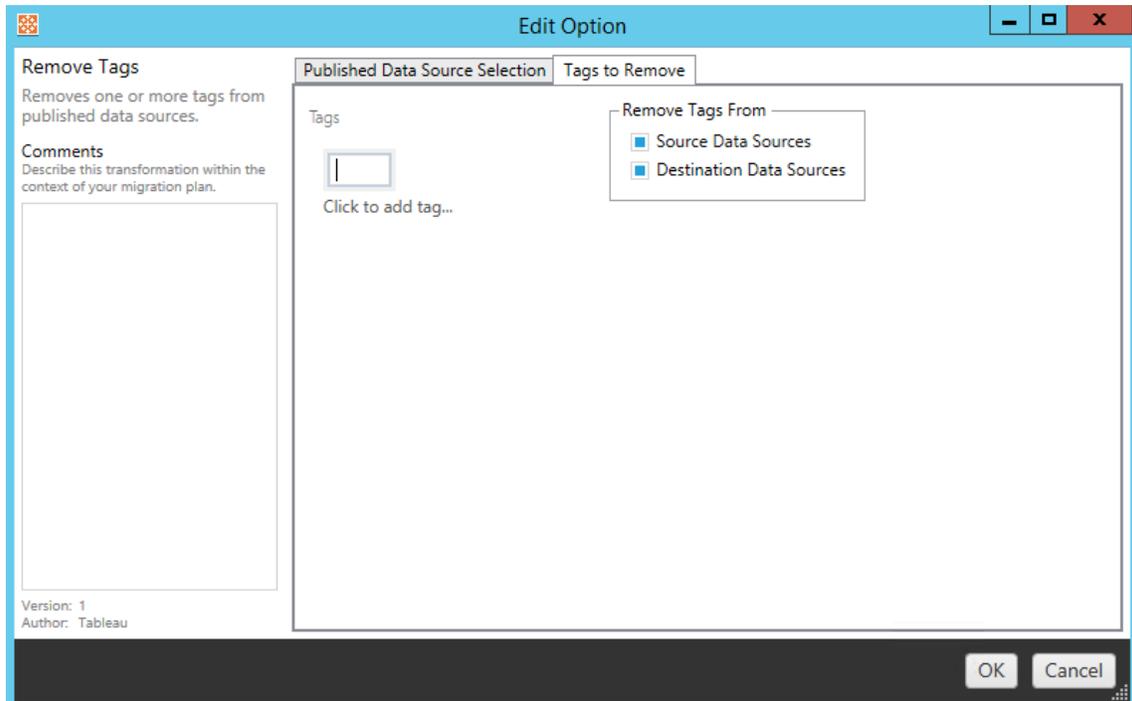
For all of the different types of transformations, there are two basic steps. The first step is to make your selection for the transformation. In this case, select the data source(s) you want to transform. At the top of the list, you can select **Apply to all published data sources**, which is an option to automatically select all data sources for future transformations. You can also **Refresh** the data source display window to reflect any changes or updates to the source site.



The second step is to enter the specific selections for the transformation you select.

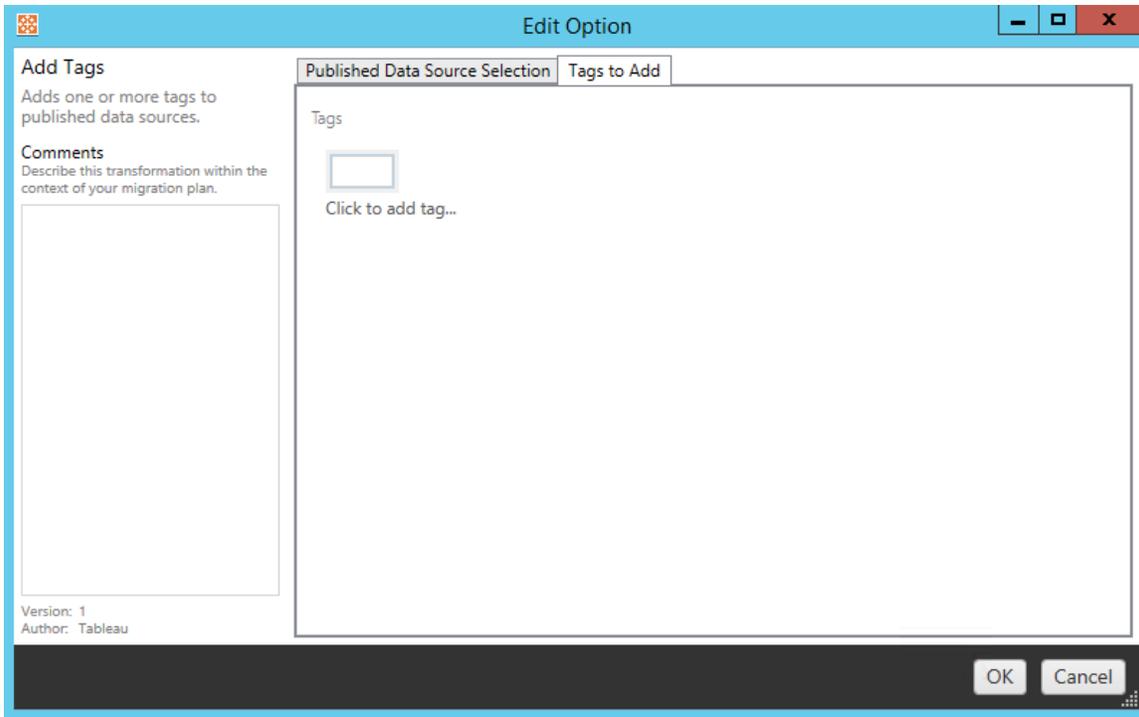
Remove Tags

Once the data sources you would like to remove tags from are selected, enter any tags you want to remove by entering them into the field at the bottom and click **Add**. From this screen, you can also select to remove from the source or destination data sources. If you want to remove a previously entered tag, click on it and press the delete key.



Add Tags

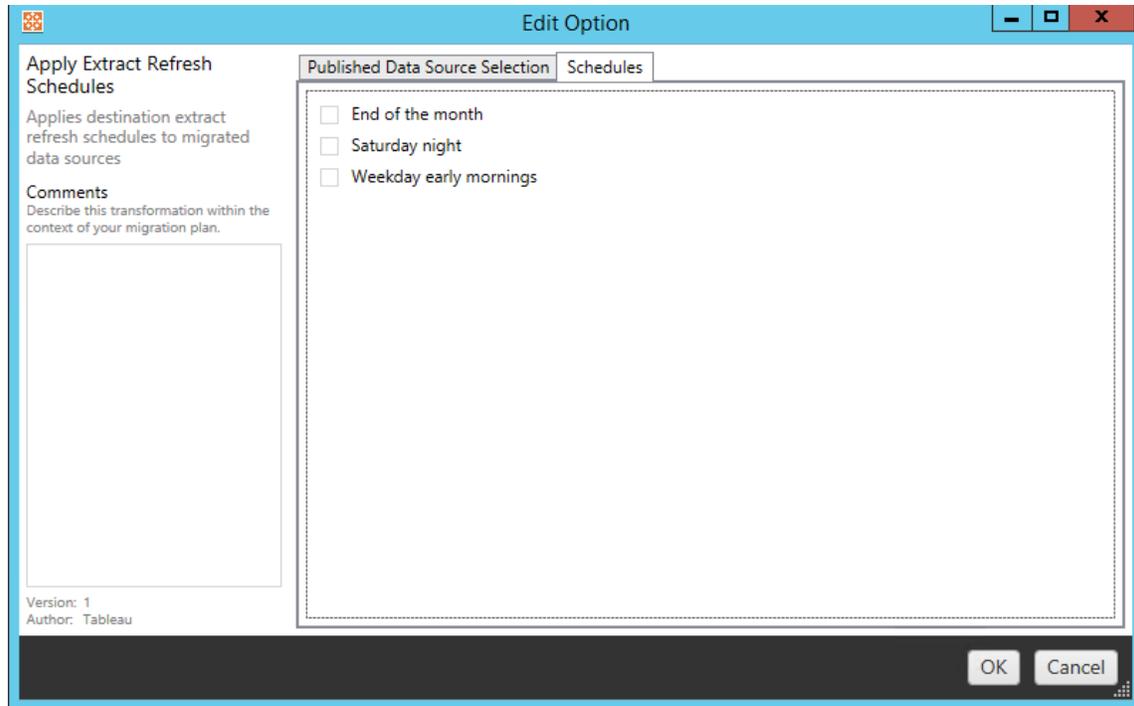
After selecting the data sources desired, enter any tags you want to assign by entering them into the field at the bottom and click **Add**. If you want to remove a tag, click on it and press the delete key.



In addition, on each of the transformations you can enter notes in the **Comments** section on the left-hand side of the Edit Transformation window.

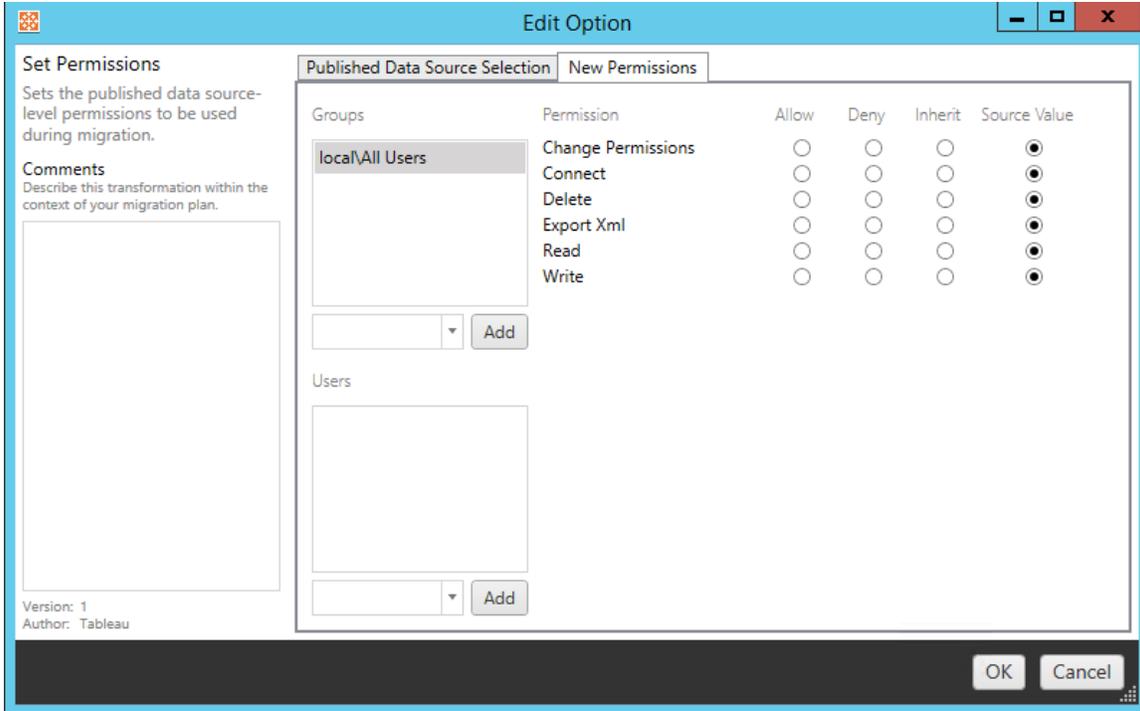
Apply Extract Refresh Schedules

This transformation allows the destination extract schedules to be selected to be added to the selected migrated data sources.



Set Permissions

The last type of transformation is to edit the permissions for the selected data sources. Enter in a Group or User and click **Add**. Adjust the permissions as desired. The four different options are to **Allow** the permission, **Deny** the permission, **Inherit**, or to keep the **Source Value**.



Step 5: Continue to Next Step

When you are ready, click **Next** to continue to the Migration Plans: User Permissions Mapping section of the planning phase.

Who can do this

Tableau Server user with an Explorer role or higher. To migrate content, you must have View and Download Workbook/Save a Copy permissions on the source site, and Publishing rights for the destination site. For more information, see Permissions.

Migration Plans: User Permissions Mapping

The Content Migration Tool allows you to replicate workbook and data source permissions to Tableau Server environments that are in different network domains or have differences in username or group syntax. You can create user permissions mappings to customize and secure content after it has been published to the destination location if **Copy Workbook Permissions** or **Copy Data Source Permissions** has been selected earlier in the planning phase.

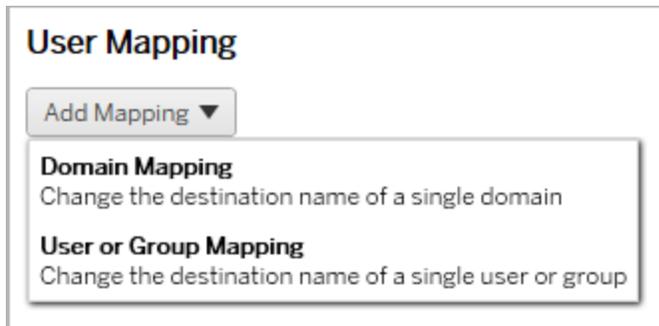
For more information, see [Migration Plans: Workbooks](#) and [Migration Plans: Published Data Sources](#).

Mapping Limitations

- Content Migration Tool will stop the migration process if it fails to find the user or group in the destination location. Subsequent user or group permissions mappings are not checked after the first failure and the plan must be run again.
- Content Migration Tool cannot replicate permissions if the source content has permissions for multiple users and groups with identical names. This only occurs when there are duplicate user or group names sourced from separate domains.

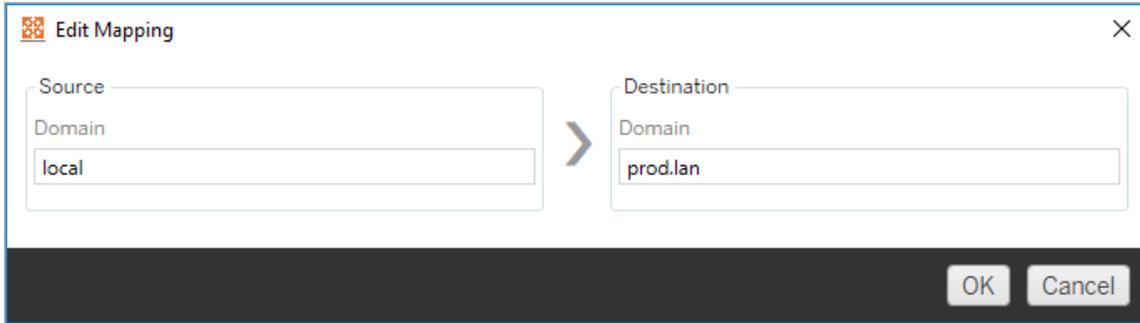
Step 1: Add Mapping

To add user permissions mapping, click **Add Mapping** and select whether to change the name of a domain, user, or group. If Content Migration Tool is unable to match a permission in the destination location, the source content will not be migrated.



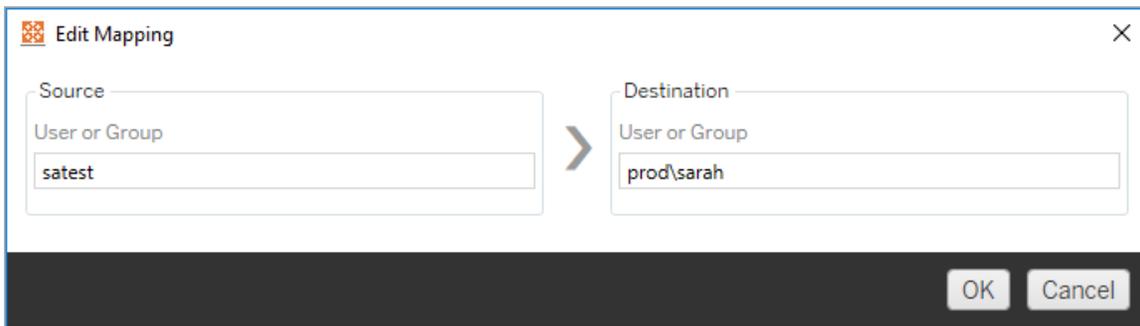
Domain Mapping

Domain permissions mapping applies to all users and groups in the destination location. If you are unsure about the source or destination domain, you can check the user and group pages in Tableau Server. If local user provisioning has been selected, the domain must be specified as `local`.



User or Group Mapping

User or group permissions mapping requires the domain to be listed for the destination location using the syntax `domain\user` or `domain\group`. Content Migration Tool will not be able to find the destination user or group if no domain is specified.



When creating permissions mappings, the destination domain you specify for a user or group can override a previously applied domain mapping. In the example below, the first permissions mapping associates the domain for all users to `prod`, while the second permissions mapping associates the domain of `User_A` to `dev`.

	Mapping	Description
Edit Delete Up Down	Domain Mapping	Match "local" to "prod"
Edit Delete Up Down	User or Group Mapping	Match "User_A" to "dev\User_B"

Step 2: Change Mapping Order

After a permissions mapping is created, you can change the order using the **Up** or **Down** options to determine when it will be handled during the migration. When a domain, user, or group is handled in a permissions mapping, any subsequent permissions mappings for the source domain, user, or group will be ignored.

In the example below, permissions for `User_A` are mapped to `User_B`. Content Migration Tool will ignore the second permissions mapping because `User_A` has already been handled.

	Mapping	Description
Edit Delete Up Down	User or Group Mapping	Match "User_A" to "local\User_B"
Edit Delete Up Down	User or Group Mapping	Match "User_A" to "local\User_C"

Step 3: Continue to Next Step

When you are ready, click **Next** to continue to the Migration Plans: Migration Scripts section of the planning phase.

Who can do this

Tableau Server user with a Site Administrator Explorer role or higher. To migrate content, you must have View and Download Workbook/Save a Copy permissions on the source site, and Publishing rights for the destination site. For more information, see Permissions.

Migration Plans: Migration Scripts

The next step of creating a migration plan in the Tableau Content Migration Tool is to create any scripts you want to run with your plan before or after migration.

Step 1: Pre-Migration

The **Run Pre Migration** section of the screen is dedicated to scripts that will run prior to migration.

The screenshot shows a configuration panel titled "Run Pre Migration". At the top left is a checkbox labeled "Enable" with a help icon. Below it are four rows of fields, each with a help icon: "Working Directory" (text input, browse button, and "Reset" button), "Run" (dropdown menu showing "Executable with parameters"), "Command Executable" (text input, browse button), and "Command Parameters" (text input).

Each field has a help icon you can get information from by moving your cursor over it. To start with your pre-migration scripts, select **Enable**, which will then activate the fields below.

Working Directory

This is the working directory for the script. The default directory is the same folder as the migration plan. Click on the browse button to select a different folder. The **Reset** button will restore the current folder as the plan as the working directory.

Run

This drop down allows you to choose either to run a custom script or an executable with parameters.

Command Executable

If you selected **Executable with Parameters** from the **Run** menu, this field will appear. This is the file path to the command executable to run prior to migration. Type it in directly or use the browse button to find the executable. This is a required field.

Command Parameters

If you selected **Executable with Parameters** from the Run drop-down menu, this field will appear. Enter in command line parameters here to use with the command executable.

Script

If you selected **Custom script** from the **Run** menu, enter in your pre-migration script here. It will be executed as a *.cmd file. This is a required field.

Step 2: Post-Migration

The **Run Post Migration** half of the screen is dedicated to scripts that will run after migration.

The screenshot shows a configuration panel titled "Run Post Migration". At the top left is an "Enable" checkbox with a help icon. Below it are four rows of input fields, each with a help icon: "Working Directory" (text field with a browse button and a "Reset" button), "Run" (dropdown menu showing "Executable with parameters"), "Command Executable" (text field with a browse button), and "Command Parameters" (text field).

Each field has a help icon you can get information from by moving your cursor over it. To start with your post-migration scripts, select **Enable**, which will then activate the fields below.

Working Directory

This is the working directory for the script. The default directory is the same folder as the migration plan. Click on the browse button to select a different folder. The **Reset** button will restore the current folder as the plan as the working directory.

Run

This drop down allows you to choose either to run a custom script or an executable with parameters.

Command Executable

If you selected **Executable with Parameters** from the **Run** menu, this field will appear. This is the file path to the command executable to run prior to migration. Type it in directly or use the browse button to find the executable. This is a required field.

Command Parameters

If you selected **Executable with Parameters** from the **Run** menu, this field displays. Enter in command line parameters here to use with the command executable.

Script

If you selected **Custom script** from the **Run** menu, enter in your post-migration script here. It will be executed as a *.cmd file. This is a required field.

Step 3: Continue to Next Step

When you are ready, click **Next**.



Who can do this

Tableau Server user with an Explorer role or higher. To migrate content, you must have View and Download Workbook/Save a Copy permissions on the source site, and Publishing rights for the destination site. For more information, see Permissions.

Migration Plans: Plan Options

The last step of creating a migration plan in the Tableau Content Migration Tool is configuring the **Options** section.

Plan OptionsNeed help? Plan Name 

- Automatically Create Destination Projects that do not Exist 
- Refresh Extracts After Migration  [Filter...](#)
- Continue Migration if Workbook or Data Source Fails 
- Copy Owner Settings 
- Automatically Create Extract Refresh Schedules that do not Exist 

Version Control

- Enable 
 - Archive Workbooks
 - Archive Published Data Sources

Archive To 

Each option on this screen has a question mark that will provide contextual assistance for that particular option.

Step 1: Options

There are five options.

Plan OptionsNeed help? Plan Name 

- Automatically Create Destination Projects that do not Exist 
- Refresh Extracts After Migration  [Filter...](#)
- Continue Migration if Workbook or Data Source Fails 
- Copy Owner Settings 
- Automatically Create Extract Refresh Schedules that do not Exist 

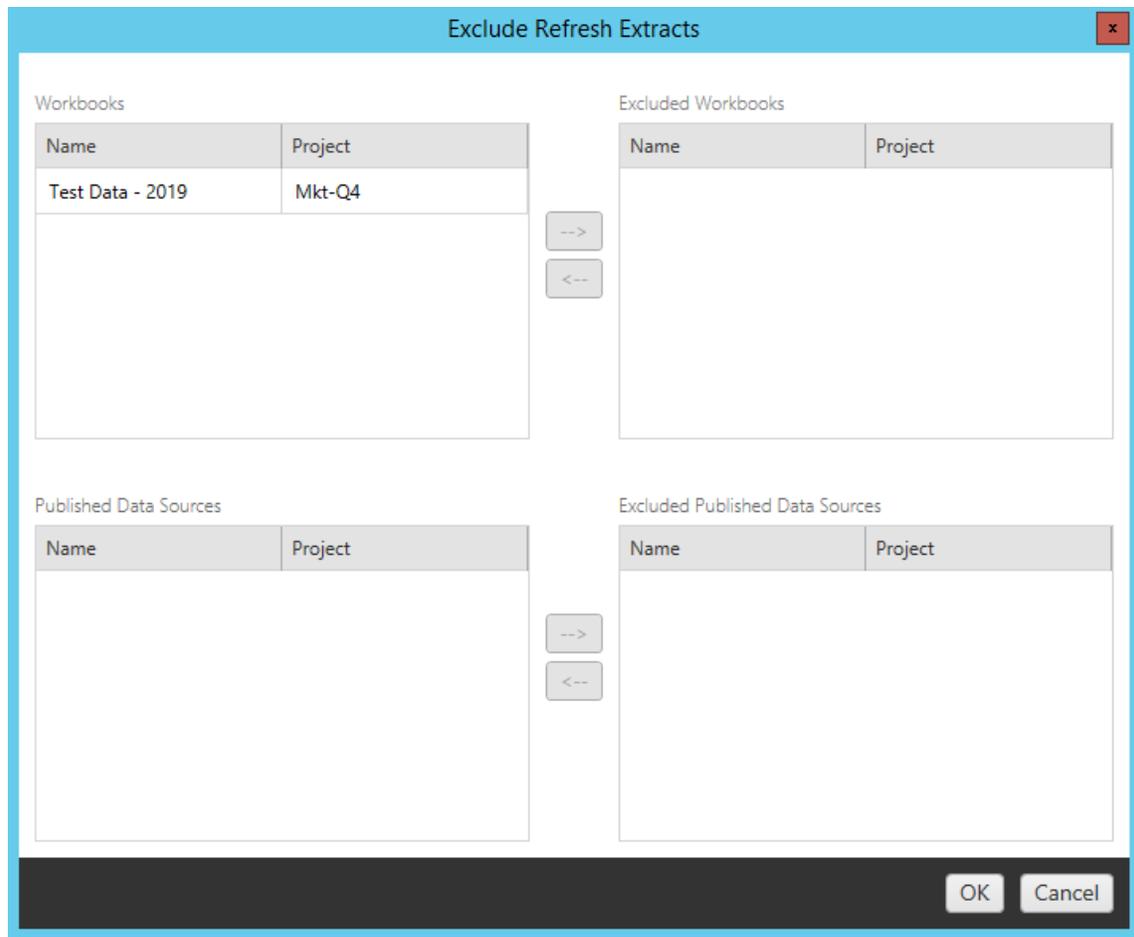
The **Plan Name** is the name of the plan as it will appear in Content Migration Tool. We recommend using a user-friendly name for your plan name.

The following checkboxes offer the following additional options:

- **Automatically Create Destination Projects that do not Exist:** By default, migrations will fail if a project does not exist on the destination server. Check this option to create those projects during migration.
- **Refresh Extracts After Migration:** If selected, any data extracts will be refreshed immediately after migration if Content Migration Tool detects that they might have been modified during migration. Click the **Filter** link for more options.
- **Continue Migration if Workbook or Data Source Fails:** If checked, errors migrating a workbook or data source will not cause the migration to stop. The errors will be logged and the migration will continue. Errors during version control will always stop the migration.
- **Copy Owner Settings:** This will attempt to set the workbook and data source owner of the destination server to the user with a matching username and domain of the source owner.
- **Automatically create Extract Refresh Schedules that do not Exist:** Automatically creates destination extract schedules that do not exist. If not checked, source schedules that do not exist on the destination server will not be copied.

Exclude Extract Refreshes

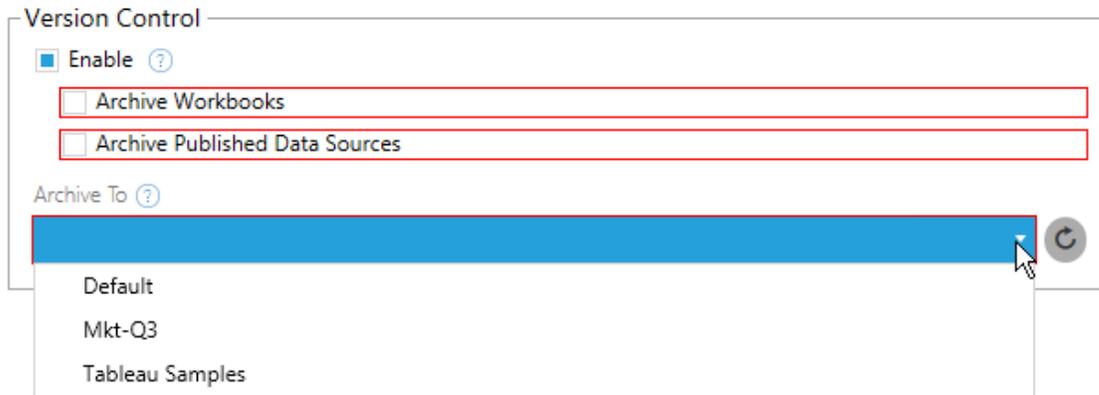
By clicking on the **Filter** link after the **Refresh Extracts After Migration** option, you will be able to choose which workbooks or published data sources will be excluded from being refreshed.



Use the back and forth arrow buttons to select which items you want excluded and then click **OK**.

Step 2: Version Control

These options allow you to avoid losing the existing workbooks in the destination site that might be replaced by the migrated workbooks.



Click **Enable** to save your previous versions. You can choose to archive workbooks and/or published data sources. You'll need to select an option from the **Archive To** menu which lists all of the projects in your destination server. We recommend creating a special archive project just to store your versioned workbooks.

There is a refresh button at the end of the **Archive To** box to update any projects that have been added on Tableau Server.

Step 3: Save Plan

Once you have completed all of your selections on the **Plan Options** screen, you can save your plan for future use. Click **Save Plan**.



Your plan will be saved to Tableau Content Migration Tool folder in the My Documents folder on your local machine.

Step 4: Continue to Next Step

When you are ready, click on **Verify & Run** to end the Planning phase of your migration and prepare to run your plan.



Who can do this

Tableau Server user with an Explorer role or higher. To migrate content, you must have View and Download Workbook/Save a Copy permissions on the source site, and Publishing rights for the destination site. For more information, see [Permissions](#).

Migrating Workbooks and Data Sources that use Extracts

Tableau Server users can publish extracts which are copies, or subsets of the original data. These extracts may be embedded in a workbook or a data source. By default, when you use the Tableau Content Migration Tool to migrate a workbook or data source that contains an extract, that extract is migrated along with the workbook or data source that contains it. The Content Migration Tool gives you a couple options for controlling this behavior:

- **Switching to a Live Connection**

You can add the **Remove Extract** transformation to your migration plan to remove the extract from your workbook or data source during migration. As always, the source workbook or data source will not be modified. The copy of the workbook or data source migrated to the destination server will have the extract removed from it. This effectively switches the data connection back to a live connection.

Refreshing Extracts after Migration

You can enable the **Refresh Extracts After Migration** option in your migration plan to have an immediate extract refresh task scheduled after the workbook or data source is migrated.

We don't recommend using the **Refresh Extracts After Migration** option if your migration plan also uses the **Set Connection Info** transformation to change the data connection's to point to a different set of data (for example, a different database server or database). When

you change the connection information to point to different data and use the **Refresh Extracts After Migration** option, this can unintentionally expose data in a way that is a potential security issue.

Changing data connections that use extracts

Tableau data connections are either live connections that directly query a data source, or they are extracts of a data source. Extracts are copies or subsets of the original data and can be embedded in a workbook or data source. When present, the views will query data from the extract instead of the underlying data source.

Commonly, you'll want to modify the data source connection during the migration so that it points to a different database on the destination server than it did on the source server.

For example, if you are migrating a workbook from your staging server to your production server, you will likely want to update the data connections inside the workbook to connect to your production database. You can implement this by using the Content Migration Tool **Set Connection Info** transformation in your migration plan. Now you have a migration plan which copies a workbook from staging to production and updates the data connections to point to the production database.

If your workbook's data source uses an extract then you need to do a bit more work. In the current scenario, the workbook will be migrated and the live data connection updated, but the workbook's views will still be showing the data from the staging database because the workbook still contains same extract of the staging database which it was using when it was copied from the source (staging) server. There are a few ways to address this.

Option 1: Use Published Data Sources

You can change your workbooks so that they use published data sources instead. This way, the extract will be managed as part of the published data source and migrating updates to the workbooks that use that data source can be simplified by not having to worry about the connection to the live database or the data extract.

Option 2: Remove the Extract During Migration

You can add a **Remove Extract** transformation to your migration plan. This will remove the extract from your workbook, effectively switching the data source to a live connection.

Option 3: Refresh the Extract After Migration

You can use the **Refresh Extracts After Migration** option in your migration plan. This will migrate the extract along with workbook but will schedule an immediate extract refresh task for that workbook after the migration is complete.

This option is usually not recommended when used in combination with a **Set Connection Info** transformation because of potential security issues that it can introduce.

The issue is that the migrated workbook on your destination server will still show the old (source) extract data for the period of time between the completion of migration and the completion of the extract refresh task. If the extract refresh task fails, then the old/source extract data will remain until the extract is refreshed.

In a scenario like we've outlined above, migrating from a staging to production environment, this may be acceptable but you should be aware that the users of your workbooks may not be aware that the workbook is showing old/staging data due it being recently migrated and the extract not being refreshed yet.

In other scenarios where you may be using **Set Connection Info** to change data connections to point to a different set of customer or client data, this could introduce serious security issues where the workbook's extract contains data from a different client or customer until the extract has been refreshed post-migration.

One way to mitigate this issue is to implement a 2-stage migration. This approach requires you to create two migration plans, one for each step described below and ensures the workbooks and data sources have an up-t-o-date extract before they are accessible.

- **Stage 1:** Migrate your content to a project on your destination site that only administrators have access to. This migration allows you to use the **Refresh the Extract**

After Migration option along with the **Set Connection Info** transformation to update the data connection, because no unauthorized users will have an opportunity to see the old data, even if the extract refresh fails.

- **Stage 2:** After stage 1 is complete and you confirm there is a successful extract refresh, run a second migration plan to migrate the content from the stage 1 destination to the final destination where it is visible to end-users.

Who can do this

Tableau Server user with an Explorer role or higher. To migrate content, you must have View and Download Workbook/Save a Copy permissions on the source site, and Publishing rights for the destination site. For more information, see Permissions.

Consolidate Sites

If you need to combine the content of multiple sites into a single site (if, for example, organizational restructuring has changed how your sites should be arranged), you can use Tableau Content Migration Tool to do this.

Consolidating sites using the Content Migration Tool

You can copy all the workbooks and data sources from one site to another by following the following steps:

1. Start the Content Migration Tool.
2. Select **Create New Plan**
3. On the left, under **Source**, enter the Server URL and server administrator credentials, and select the source site (the site you will be copying the workbooks and data sources from).

Source

Import from Saved Connection ▼

Server URL

http://tableauserver.mydomain.com

Username

john.smith

Password

•••••

Save password [?](#)

Site:

Sign In and Select a Site

4. On the right, under **Destination**, enter the server information and select the target site (the site you are copying the workbooks and data sources to).

Destination

Import from Saved Connection ▼

Server URL

Username

Password

Save password [?](#)

Site:

Sign In and Select a Site

Note: The source server and destination server might be the same, but you can also migrate content from a site on one server to a site on another server.

5. Click **Next** and confirm the **All Projects** option is selected on the **Source Projects** page.
6. Continue to the next step by clicking **Next**.
7. On the **Workbook Selection** screen, confirm the **All Workbooks** option is selected and click **Next**.
8. If you are copying published data sources:

Click **Selection** under **Published Data Sources** from the menu on the left.

Make sure that the **All Data Sources** option is selected.

9. Click **Run Migration** from the menu on the left to review the migration plan. Once you're satisfied, click **Run** at the bottom of the screen to run the migration.
10. Repeat these steps until you've consolidated all sites

For more information on setting up migration plans, see [Using Tableau Content Migration Tool](#).

Who can do this

Tableau Server user with an Explorer role or higher. To migrate content, you must have View and Download Workbook/Save a Copy permissions on the source site, and Publishing rights for the destination site. For more information, see [Permissions](#).

Migration Rollback

You can use the rollback feature of the Tableau Content Migration Tool when you have workbooks or data sources that have been archived during migration (see [Migration Plans: Plan Options](#) for more information on archiving). To initiate a rollback, click **Rollback** on the Start screen.



You will be guided through four preparatory steps before starting the rollback.

Step 1: Server

The first step is to sign into server.

Tableau Server

Server

Import from Saved Connection ▼

Server URL

Username

Password

Save password [?](#)

Site:

Sign In and Select a Site

[Add or edit saved connections](#)

Unlike the Planning phase of creating or editing a migration plan, here you only need to enter the connection information once, for the server on which you want to roll back the migration.

Click **Import from Saved Connection** to select a stored connection. Click **Add or edit saved connections** to create or update your sign in information. When you are ready, click **Sign In and Select a Site**.

If you used stored connection information, the default site will be automatically listed. You can change the site by clicking **Change Site**. Click **Next** when you are ready to continue.



Step 2: Migration

The next step is to choose which migration and archived workbooks and data sources you want to restore.

Migration

Date	Summary
7/16/2019 5:22 PM	ACCT-archive: 2 workbooks, 0 data sources

The historical migration are listed by date with a summary of each. The summary shows how many workbooks and published data sources were archived. Select the migration you want to roll back to and click **Next**.



Step 3: Workbooks

The next step is to select which of the archived workbooks you want to restore from the selected migration.

Workbooks

Need help? 

Showing 2 workbook(s) from the 7/16/2019 5:22 PM migration

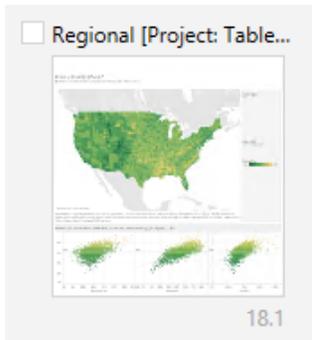
Select the workbooks you want to roll back

 Refresh

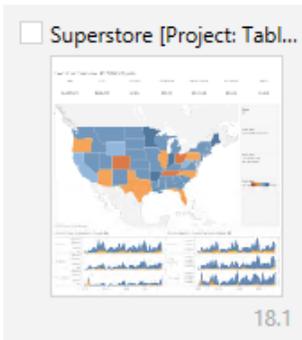
Select All (0 of 2 selected)



Regional [Project: Table...]



Superstore [Project: Tabl...]



This screen is similar to the Workbooks step of the Planning phase. You can view your workbooks in thumbnail or list mode by toggling the different buttons on the right-hand side of the screen. You can reload the workbooks in the screen by clicking **Refresh**. You can individually select workbooks by clicking on each of them or **Select All/Unselect All** by clicking on the button.

Once you have made your selections, click **Next**.



Step 4: Published Data Sources

For this step, choose the archived published data sources that you want to restore from your selected migration. The selection process is identical to the data source selection screen from the migration plan process.

Published Data SourcesNeed help? 

Showing 1 published data source(s) from the 7/17/2019 11:18 PM migration

Select the published data sources you want to roll back

 Refresh Select All (0 of 1 selected)

	Name	Project
<input type="checkbox"/>	Sheet1 (state_plates)	Mkt-Q3

When you have made your selections, click **Next**.

Next

Step 5: Run Rollback

The final step of the rollback process is to verify that all of your selections are accurate for the workbooks and published data sources that you want to restore.

Verify**Source:** http://win-vj23dhvudie /#/site/Accounting-sandbox**Archive Date:** 7/16/2019 5:22:35 PM**Workbooks:** Regional [Project: Tableau Samples]
Superstore [Project: Tableau Samples]**Published Data Sources:** No selectionsWhen you are ready to continue, click **Roll Back** run the rollback.

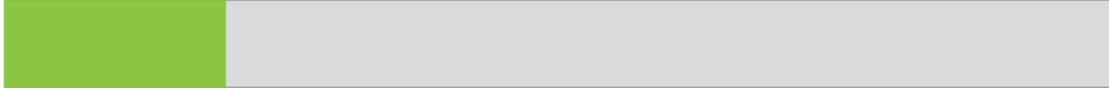
Roll Back

A status bar indicator shows the progress of each step of the rollback. Depending on how many files you are restoring, this process may take several minutes.

Running...

Need help? 

Discovering Source Published Data Sources



When the rollback is complete, you will be alerted with a completed status bar.

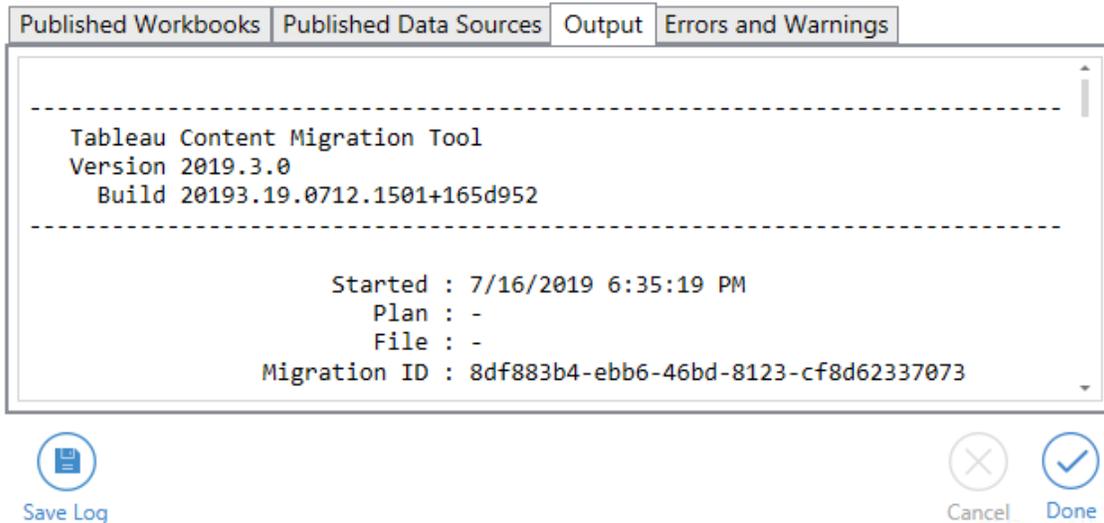
Finished

Need help? 

Migration Complete



Beneath the status bar is a multi-tabbed text area with more information. This is similar to the finish screen of an actual migration. The first tab will indicate archived workbooks that were successfully restored. The second tab will list published data sources. The third tab is an output log that details the rollback. Finally, any errors or warnings will be listed in the final fourth tab.

A multi-tabbed text area with four tabs: "Published Workbooks", "Published Data Sources", "Output", and "Errors and Warnings". The "Output" tab is selected. The text content is as follows:

```
-----  
Tableau Content Migration Tool  
Version 2019.3.0  
Build 20193.19.0712.1501+165d952  
-----  
  
Started : 7/16/2019 6:35:19 PM  
Plan : -  
File : -  
Migration ID : 8df883b4-ebb6-46bd-8123-cf8d62337073
```

At the bottom of the text area are four icons: a document icon labeled "Save Log", a close icon labeled "Cancel", and a checkmark icon labeled "Done".

When you are finished, click **Done** to return back to the main screen.



Who can do this

Tableau Server user with an Explorer role or higher. To migrate content, you must have View and Download Workbook/Save a Copy permissions on the source site, and Publishing rights for the destination site. For more information, see Permissions.

Using the Tableau Content Migration Tool Console Runner

The Tableau Content Migration Tool includes a command-line utility for running migrations, `tabcmt-runner.exe`, located in the installation folder. The default installation folder is `%PROGRAMFILES%\Tableau\Tableau Content Migration Tool`.

Note: The `tabcmt-runner.exe` utility is not the same as the `tabcmt.cmd` command line utility which is used to configure the Content Migration Tool graphical application. For more information about `tabcmt.cmd`, see [Using the Tableau Content Migration Tool Command Line Interface](#).

Usage:

- `tabcmt-runner [options] <plan_file.tcmx>`
- `tabcmt-runner license --remove`
- `tabcmt-runner license <new license key>`
- `tabcmt-runner license <license file path> [--passphrase=<license file passphrase>]`
- `tabcmt-runner encryption --reset`
- `tabcmt-runner encryption <new_key>`
- `tabcmt-runner improvement [on|off]`
- `tabcmt-runner --help`
- `tabcmt-runner --version`

Options:

Tableau Server on Linux Administrator Guide

- --version
- --help
- --quiet
- --info
- --logfile=VALUE
- --src-user=VALUE
- --src-password=VALUE
- --dest-user=VALUE
- --dest-password=VALUE
- --https=VALUE

Run Plan

Executes a migration plan immediately.

```
tabcmt-runner [options] <plan file>
```

Available options:

- --logfile=<file name> sets the file name to log output to
- --https=<secure|legacy> sets the HTTPS mode
- --quiet disables logging to stdout
- --src-user=<username> sets the username of the source Tableau Server connection
- --src-password=<password> sets the password of the source Tableau Server connection
- --dest-user=<username> sets the username of the destination Tableau Server connection
- --dest-password=<password> sets the password of the destination Tableau Server connection

Exit codes:

- 0 indicates that the migration was successful.
- 1 indicates that the migration was successful but warning messages were logged.
- 2 indicates that the migration failed. Specific errors will be included in the log output.

Show Plan Summary

Shows a summary of the migration plan and then exits.

```
tabcmt-runner --info <plan file>
```

help

Shows usage information for the command line utility.

```
tabcmt-runner --help
```

version

Shows the current application version information.

```
tabcmt-runner --version
```

encryption

Reset the encryption key, or specify a new one. You must specify the encryption key before using the `tabcmt-runner` utility, even if you already done so from the Content Migration Tool UI.

```
tabcmt-runner encryption <new_key> | --reset
```

improvement

Enables or disables collection of anonymous usage information by the application. This information is completely anonymous and is sent periodically to Tableau to help us improve Content Migration Tool.

Examples

Show whether the improvement program is enabled or disabled:

```
tabcmt-runner improvement
```

Enable or disable the improvement program:

```
tabcmt-runner improvement <on|off>
```

license

Note: This command is only applicable for legacy licenses.

Manages a legacy application license for the current user. When using a legacy key, to use the `tabcmt-runner` utility you must activate the license using this command, even if you already activated it from the Content Migration Tool UI.

Examples

Show the current license information:

```
tabcmt-runner license
```

Set/activate a serial key or offline license key:

```
tabcmt-runner license <key>
```

Remove/deactivate the current license:

```
tabcmt-runner license --remove
```

Set/activate using a license file:

```
tabcmt-runner license <file path> [--passphrase=<password>]
```

Who can do this

To use the console runner, you must have all the following:

- Administrator permissions on the Content Migration Tool machine.
- Tableau Server user account with an Explorer role or higher.
- View and Download Workbook/Save a Copy permissions on the source site.
- Publishing rights for the destination site.

Example: Scripting Migration Plans

Note: This topic includes a sample script you can use as the basis for scripting a multi-plan migration that satisfies your needs and environment. This script is intended to be used as a sample only, and not to be run as-is. For detailed instructions on using the console runner, see [Using the Tableau Content Migration Tool Console Runner](#).

Tableau Content Migration Tool command line utility for running migrations can be used to automate the running of a migration plan from an external scheduler (such as Windows Task Scheduler) or from a custom script. The console runner only runs one migration plan (stored in a .edt file) at a time. If you have a group of migration plans you want to run as a group, then you can use a custom script in combination with the Content Migration Tool console runner.

The example below is written in PowerShell and uses the console runner to execute a list of migration plans as a group.

The following example code demonstrates:

- Running multiple migration plans as a group using the console runner.
- Optionally halting deployment of the group of plans immediately when any single migration in the group fails.
- Using the console runner's exit code to determine whether the migration failed or logged warnings.

```
# List of migration plans to execute as a group.
$planFiles = @(
    'customer 1.tcmx',
    'customer 2.tcmx'
)

# True or false whether to continue with the next plan if a
migration fails.
```

Tableau Server on Linux Administrator Guide

```
$continueOnFailure = $false

# Path to the CMT console runner executable
$runnerExe = 'C:\Program Files (x86)\Tableau\Tableau Content
Migration Tool\tabcmt-runner.exe'

# Store the exit code from the previously run migration plan.
$lastResult = -1

# Loop through and run each migration plan one at a time.
$planFiles | % {
    $file = $_

    if ($lastResult -ge 2 -and -not($continueOnFailure)) {
        Write-Warning "Skipping plan because previous migration
failed. `nSkipped plan: $file"
        return
    }

    Write-Verbose "Running migration plan: $file"
    & $runnerExe $file
    $lastResult = $LASTEXITCODE

    if ($lastResult -ge 2) {
        Write-Error "Migration failed. See output or log file for
error details.`nPlan: $file" -ErrorAction 'Continue'
    }
    elseif ($lastResult -eq 1) {
        Write-Warning "Migration completed with warnings. See output
or log file for warning details.`nPlan: $file"
    }
}
}
```

Who can do this

To script migration plans, you must have all the following:

- Administrator permissions on the Content Migration Tool machine.
- Tableau Server user account with an Explorer role or higher.
- View and Download Workbook/Save a Copy permissions on the source site.
- Publishing rights for the destination site.

Using the Tableau Content Migration Tool Command Line Interface

The Tableau Content Migration Tool includes a command line interface, `tabcmt.cmd`, located in the installation folder. The default installation folder is `%PROGRAMFILES%\Tableau\Tableau Content Migration Tool (32-bit Windows)` or `%PROGRAMFILES(x86)%\Tableau\Tableau Content Migration Tool (64-bit Windows)`.

Note: The `tabcmt.cmd` utility is not the same as the Content Migration Tool console runner, `tabcmt-runner.exe`. The console runner is a separate command line utility used for running migrations from the command line. For information on using the Content Migration Tool console runner, see [Using the Tableau Content Migration Tool Console Runner](#).

Here are the commands that can be used with the `tabcmt` command line:

- `migrate`
- `help`
- `license`
- `update`
- `version`

`migrate`

Opens a migration plan file to the migrate step in the GUI:

```
tabcmt migrate <plan file>
```

`help`

Shows general help about the command line interface and the available commands.

Tableau Server on Linux Administrator Guide

Examples

Show all commands available:

```
tabcmt help
```

Show help and usage information for a specific command:

```
tabcmt help <command>
```

license

Manages the application license for the current user.

Note: This command is only applicable for legacy licenses.

Examples

Show the current license information:

```
tabcmt license
```

Remove/deactivate the current license:

```
edt license remove
```

Set/activate a serial key or offline license key:

```
tabcmt license <key>
```

Set/activate using a license file:

```
tabcmt license <file path> [--passphrase=<password>]
```

update

Manages the options for application updates.

Examples

Show the current update settings:

```
tabcmt update
```

Enable or disable the automatic update notifications:

```
tabcmt update --disabled=<true|false>
```

Set the URL to detect/download updates from:

```
tabcmt update --url=<url>
```

Enable or disable showing beta updates. Set to false to only show stable release updates.

```
tabcmt update --beta=<true|false>
```

version

Shows the current application version information.

```
tabcmt version
```

Who can do this

To use the command line interface, you must have all the following:

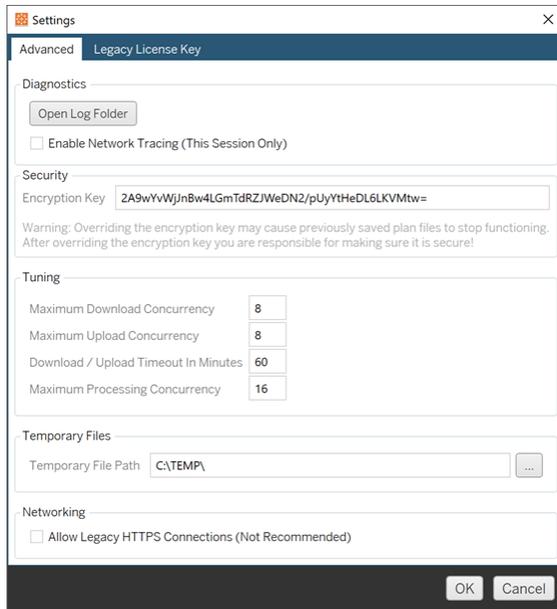
- Administrator permissions on the Content Migration Tool machine.
- Tableau Server user account with an Explorer role or higher.
- View and Download Workbook/Save a Copy permissions on the source site.
- Publishing rights for the destination site.

Tableau Content Migration Tool Settings

The Tableau Content Migration Tool default settings work in most cases, but you can change these if you need to, or if you are working with Tableau Support and they ask you to make changes.

To view or update the Content Migration Tool settings:

1. Open Content Migration Tool.
2. Click **Help** and **Settings**. The **Settings** dialog opens:



3. **Diagnostics**—Click **Open Log Folder** to open the logs location. Here you can view the logs, and zip them up if you need to send them to Tableau. For more information, see [Tableau Content Migration Tool Log Files](#).

Select **Enable Network Tracing** if you are working with Support and they ask you to include a network trace in the logs. This applies until you clear the option or restart the Content Migration Tool.

4. **Security**—The encryption key is automatically generated on installation. If you change the encryption key, any migration plans with embedded passwords that were created with the previous key cannot be opened. If you have multiple installations of Tableau Content Migration Tool and want to share migration plans, you need to make sure the encryption key used by each instance of the tool is the same.
5. **Tuning**—In almost all cases you can leave these set to the defaults. If you are working with Support, they may ask you to change these settings.
6. **Temporary Files**—Select a location for temporary files if you want to change the default. This is the location where content is copied during a migration. You may want

to change this if the default location does not have enough space to temporarily hold migrated content.

7. **Networking**—Selecting **Allow Legacy HTTPS Connections** gives you the ability to connect to Tableau Server installations running with older HTTPS configurations (for example, SSL v3). This is not recommended.

Who can do this

Typically, the tasks listed above can only be done by a user with Administrator access on the machine.

Tableau Content Migration Tool Log Files

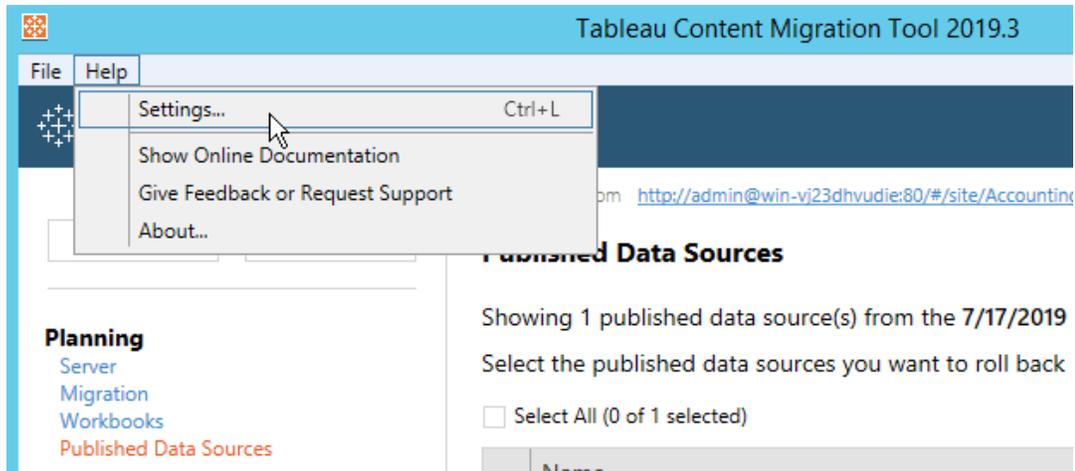
Tableau Content Migration Tool generates log files when you run migrations. These can be helpful for troubleshooting problems.

Note: For information on all the Content Migration Tool settings, see [Tableau Content Migration Tool Settings](#).

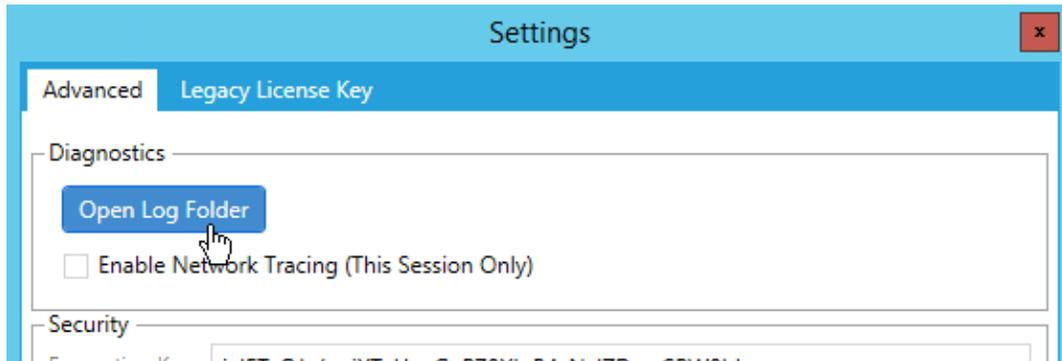
Content Migration Tool Log File Location

To find the Content Migration Tool log files from within the Content Migration Tool:

1. Start Content Migration Tool.
2. Click **Help** and **Settings**:



3. In the **Settings** dialog, click **Open Log Folder**:



A window opens with the log files.

If you are working with Tableau Support and they ask you to send log files, zip the files up before you send them. For more information on sending log files to Tableau, see the [Tableau Knowledge Base](#).

Who can do this

Typically, the tasks listed above can only be done by a user with Administrator access on the machine.

Tableau Server Key Management System

Tableau Server has three Key Management System (KMS) options that allow you to enable

encryption at rest. One is a local option that is available with all installations of Tableau Server. Two additional options require the Server Management add-on, but allow you to use a different KMS.

Beginning in version 2019.3, Tableau Server added these KMS options:

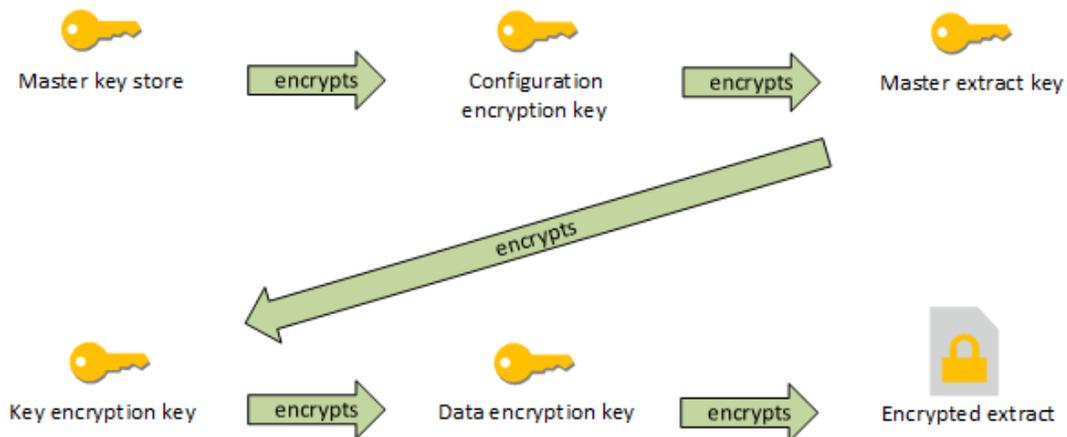
- A local KMS that is available with all installations. This is described below.
- An AWS-based KMS that comes as part of the Server Management add-on. For details, see [AWS Key Management System](#).

Beginning in version 2021.1, Tableau Server added another KMS option:

- An Azure-based KMS that comes as part of the Server Management add-on. For details, see [Azure Key Vault](#).

Tableau Server local KMS

The Tableau Server local KMS uses the secret storage capability described in [Manage Server Secrets](#) to encrypt and store the master extract key. In this scenario, the Java key-store serves as the root of the key hierarchy. The Java keystore is installed with Tableau Server. Access to the master key is managed by native file system authorization mechanisms by the operating system. In the default configuration, the Tableau Server local KMS is used for encrypted extracts. The key hierarchy for local KMS and encrypted extracts is illustrated here:



Troubleshoot configuration

Multi-node misconfiguration

In a multi-node setup for AWS KMS, the `tsm security kms status` command may report healthy (OK) status, even if another node in the cluster is misconfigured. The KMS status check only reports on the node where the Tableau Server Administration Controller process is running and does not report on the other nodes in the cluster. By default the Tableau Server Administration Controller process runs on the initial node in the cluster.

Therefore, if another node is misconfigured such that Tableau Server is unable to access the AWS CMK, those nodes may report Error states for various services, which will fail to start.

If some services fail to start after you have set KMS to the AWS mode, then run the following command to revert to local mode: `tsm security kms set-mode local`.

Regenerate RMK and MEK on Tableau Server

To regenerate the root master key and the master encryption keys on Tableau Server, run the `tsm security regenerate-internal-tokens` command.

AWS Key Management System

Tableau Server has three Key Management System (KMS) options that allow you to enable encryption at rest. Two of these require the Server Management add-on, while a local one is available with all installations of Tableau Server.

Beginning in version 2019.3, Tableau Server added these KMS options:

- A local KMS that is available with all installations. For details, see Tableau Server Key Management System.
- An AWS-based KMS that comes as part of the Server Management add-on. This is described below.

Beginning in version 2021.1, Tableau Server added another KMS option:

- An Azure-based KMS that comes as part of the Server Management add-on. For details, see Azure Key Vault.

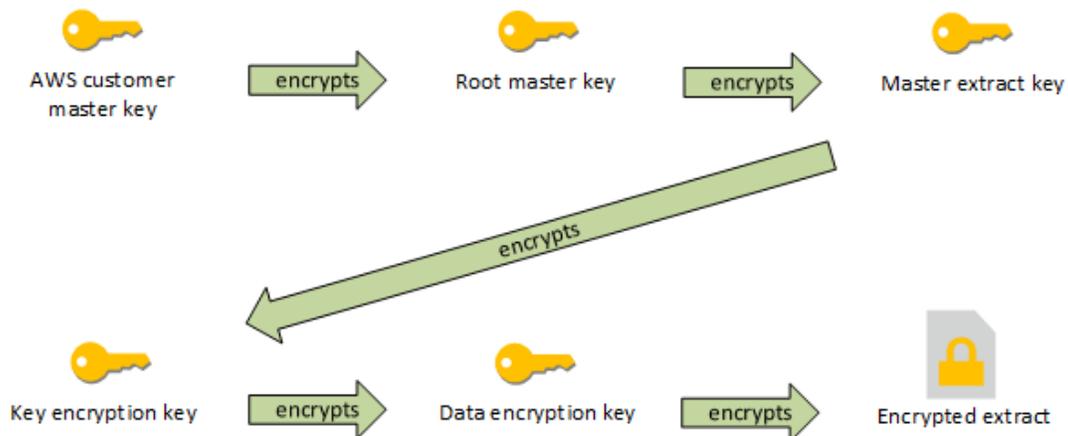
With the release of version 2019.3, Tableau Server supports the AWS key management system (KMS) as part of the Server Management add-on.

AWS KMS for encryption at rest

AWS KMS is available as part of the Server Management Add-on to Tableau Server. For more information see [About Tableau Advanced Management](#).

If your organization is deploying Data Extract Encryption at Rest, then you may optionally configure Tableau Server to use AWS as the KMS for extract encryption. To enable AWS KMS, you must deploy Tableau Server in AWS EC2. In the AWS scenario, Tableau Server uses the AWS KMS customer master key (CMK) to generate an **AWS data key**. Tableau Server uses the AWS data key as the root master key for all encrypted extracts. However, even when configured for AWS KMS, the native Java keystore and local KMS are still used for secure storage of secrets on Tableau Server. The AWS KMS is only used to encrypt the root master key for encrypted extracts.

Using AWS to encrypt the master root key provides better security properties by not storing the master key under the same permissions as the extracts.



The key hierarchy when Tableau Server is configured with AWS KMS

Tableau Server on Linux Administrator Guide

Configure AWS KMS for Tableau Server encrypted extracts

To use the AWS customer master key (CMK) to encrypt the root key in the Tableau Server KMS hierarchy, you must configure Tableau Server as described in this section.

Before you begin, verify that you meet the following requirements:

- Tableau Server must be deployed in AWS EC2
- Tableau Server must be configured with a Advanced Management license. See [About Tableau Advanced Management](#).
- You must have administrative control of a customer master key (CMK) created in AWS Key Management Service

Step 1: Create CMK and set key policy for Tableau Server in AWS

The following procedures are performed in the AWS KMS service. References are included to AWS documentation.

1. Create the CMK that you will use for Tableau Server. See the AWS topic, [Creating Keys](#).
2. Update the server instance's IAM role.

Tableau Server needs to be able to authenticate with AWS KMS using the instance's IAM role. The role needs to have a policy attached to it. The policy should be giving the instance permissions to call the "GenerateDataKey" and "Decrypt" actions on the CMK. See [IAM Roles for Amazon EC2](#).

In a multinode deployment of Tableau Server, all nodes of the server must be running under roles that have this policy (or equivalent) attached. You can assign the same role to all nodes in the cluster.

3. At a minimum, the CMK must have a key policy where the `Effect` is set to `Allow` the `Principal` (the IAM role that is attached to the server instances) the `Action: GenerateDataKey` and `Decrypt`. See [Using Key Policies in AWS KMS](#).

Step 2: Collect AWS configuration parameters

You will need the full ARN string from AWS KMS. This string is in the "General configuration" section of the AWS KMS management pages. The ARN is presented in this format:

arn:aws:kms:<region>:<account>:key/<CMK_ID>, for example, arn:aws:kms:us-west-2:867530990073:key/1abc23de-fg45-6hij-7k89-110mn1234567.

You will also need to specify the AWS region, which is also included in the ARN string. In the example above, the region is `us-west-2`. The region is where your KMS instance resides. In the next step, you will need to specify a region as shown in the Region column in the [Amazon API Gateway table](#).

Step 3: Configure Tableau Server for AWS KMS

Run the following command on Tableau Server. This command will restart the server:

- `tsm security kms set-mode aws --aws-region "<region>" --key-arn "arn:aws:kms:<region>:<account_number>:key/<CMK_ID>"`

The `--key-arn` option takes a direct string copy from the ARN in the "General configuration" section of the AWS KMS management pages.

For example, if your AWS KMS instance is running in `us-west-2` region, your account number is `867530990073`, and your CMK key is `1abc23de-fg45-6hij-7k89-110mn1234567`, then the command would be as follows:

```
tsm security kms set-mode aws --aws-region "us-west-2" --
key-arn "arn:aws:kms:us-west-2:867530990073:key/1abc23de-
fg45-6hij-7k89-110mn1234567"
```

Step 4: Enable encryption at rest

See [Extract Encryption at Rest](#).

Step 5: Validate installation

1. Run the following command:

```
tsm security kms status
```

The following information may be returned:

Tableau Server on Linux Administrator Guide

- The ARN (ID) of the customer master key (CMK)
- The region the CMK is in
- The ID of the root master key (RMK) in use. The RMK is a key that is encrypted by the CMK. Tableau Server decrypts the CMK by making calls to AWS KMS. The RMK is then used to encrypt/decrypt the master extract key (MEK). The RMK can change, but there will be only one at a time.
- KMS stores a collection of master extract keys (MEKs). Each MEK has:
 - An ID, for example, 8ddd70df-be67-4dbf-9c35-1f0aa2421521
 - Either a “encrypt or decrypt key” or “decrypt-only key” status. If a key is “encrypt or decrypt”, Tableau Server will encrypt new data with it. Otherwise, the key will only be used for decryption
 - A creation timestamp, for example, "Created at: 2019-05-29T23:46:54Z."
 - First transition to encrypt and decrypt: a timestamp indicating when the key became an encrypt or decrypt key.
 - Transition to decrypt-only: a timestamp indicating when the key transitioned to decrypt-only.

2. View logs after you encrypt and decrypt extracts:

- Publish extracts to your site and then encrypt them. See [Extract Encryption at Rest](#).
- Access the extracts with Tableau Desktop or with Web Authoring on a browser (this will decrypt the extracts for use).
- Search the vizqlserver_node log files for the `AwsKmsEncryptionEnvelopeAccessor` and `AwsKmsEncryptionEnvelope` strings. The default location of the logs are at `/var/opt/tableau/tableau_server/data/tabsvc/logs/`

Log entry examples that indicate successful configuration include the following:

- Decrypted the RMK with ID 1abc23de-fg45-6hij-7k89-110mn1234567 using the CMK with ARN `arn:aws:kms:us-west-2:867530990073:key/1234567d-a6ba-451b-adf6-3179911b760f`

- Using RMK with ID 1abc23de-fg45-6hij-7k89-1l0mn1234567 to decrypt KMS store

For publishing and extract refreshes related to KMS, search the background logs. For more information about logs, see [Tableau Server Logs and Log File Locations](#).

Troubleshoot configuration

Multi-node misconfiguration

In a multi-node setup for AWS KMS, the `tsm security kms status` command may report healthy (OK) status, even if another node in the cluster is misconfigured. The KMS status check only reports on the node where the Tableau Server Administration Controller process is running and does not report on the other nodes in the cluster. By default the Tableau Server Administration Controller process runs on the initial node in the cluster.

Therefore, if another node is misconfigured such that Tableau Server is unable to access the AWS CMK, those nodes may report Error states for various services, which will fail to start.

If some services fail to start after you have set KMS to the AWS mode, then run the following command to revert to local mode: `tsm security kms set-mode local`.

Refresh AWS CMK

Refreshing the AWS CMK is a task that you perform with AWS. By default, the AWS CMK will refresh once a year. See the AWS topic, [How Automatic Key Rotation Works](#). Since the ARN and region do not change, you do not need to update the KMS configuration on Tableau Server for normal CMK refresh scenarios.

After AWS CMK refreshes, you must regenerate the internal RMK and MEKs on Tableau Server. You should also re-encrypt all extracts with the new CMK:

1. Run the `tsm security regenerate-internal-tokens` command to regenerate all internal keys on Tableau Server, including the RMK and MEKs used for extract encryption.

Tableau Server on Linux Administrator Guide

2. Run `tabcmd reencryptextracts <site-name>` to re-encrypt extracts on a given site. Run this command on every site where you are storing encrypted extracts. Depending on the number of encrypted extracts on the site, this operation could consume significant server processing load. Consider running this operation outside of business hours. See [Extract Encryption at Rest](#).

Regenerate RMK and MEK on Tableau Server

To regenerate the root master key and the master encryption keys on Tableau Server, run the `tsm security regenerate-internal-tokens` command.

Back up and restore with AWS KMS

A server backup can be taken in AWS mode with no additional configurations or procedures. The backup contains encrypted copies of the RMK and MEKs. Decrypting the keys requires access and control of the AWS CMK.

For the restore scenario, the server being restored to can be in either KMS mode, including Local. The only requirement is that the server the backup is being restored to has decrypt access to the CMK the backup itself used.

Upon restore, the MEKs from the backup are imported as decrypt-only keys. The RMK is not migrated over. A new RMK is generated as part of the installation/restore process.

Azure Key Vault

Tableau Server has three Key Management System (KMS) options that allow you to enable encryption at rest. Two of these require the Server Management add-on, while a local one is available with all installations of Tableau Server.

Beginning in version 2019.3, Tableau Server added these KMS options:

- A local KMS that is available with all installations. For details, see [Tableau Server Key Management System](#).
- An AWS-based KMS that comes as part of the Server Management add-on. For details, see [AWS Key Management System](#).

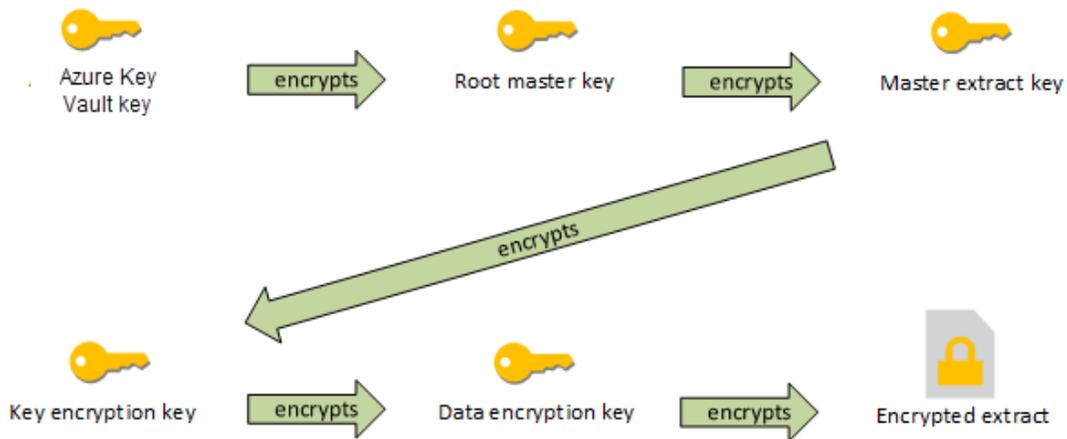
Beginning in version 2021.1, Tableau Server added another KMS option:

- An Azure-based KMS that comes as part of the Server Management add-on. This is described below.

Azure Key Vault for encryption at rest

Azure Key Vault is available as part of the Server Management Add-on to Tableau Server beginning in version 2021.1.0. For more information see [About Tableau Advanced Management](#).

If your organization is deploying Data Extract Encryption at Rest, then you may optionally configure Tableau Server to use Azure Key Vault as the KMS for extract encryption. To enable Azure Key Vault, you must deploy Tableau Server in Azure. In the Azure scenario, Tableau Server uses the Azure Key Vault to encrypt the root master key (RMK) for all encrypted extracts. However, even when configured for Azure Key Vault, the Tableau Server native Java keystore and local KMS are still used for secure storage of secrets on Tableau Server. The Azure Key Vault is only used to encrypt the root master key for encrypted extracts.



The key hierarchy when Tableau Server is configured with Azure Key Vault

Tableau Server on Linux Administrator Guide

Configure Azure Key Vault for Tableau Server encrypted extracts

To use the Azure Key Vault to encrypt the root key in the Tableau Server KMS hierarchy, you must configure Tableau Server as described in this section.

Before you begin, verify that you meet the following requirements:

- Tableau Server must be deployed in Azure.
- Tableau Server must be configured with a Advanced Management license. See [About Tableau Advanced Management](#).
- You must have administrative control over the key vault in Azure where the key resides.

Step 1: Create a key vault and key for Tableau Server in Azure

The following procedures are performed in the Azure Key Vault service. References are included to Azure documentation.

1. Create the key vault that you will use for Tableau Server. See the Azure topic, [Create a key vault](#).
2. Create a key in the vault. See the Azure topic, [Manage keys and secrets](#).

The key must be an asymmetric, RSA type, but can be any size (Tableau Server does not care about the key size). We recommend you use the Principle of Least Privilege to have maximum security.

Tableau requires permissions to perform the GET, UNWRAP KEY, and WRAP KEY commands operations and we recommend you allow access for only these operations for least privilege. Assign the access policy to the VM you are running Tableau Server on.

In a multi-node deployment of Tableau Server, the access policy must be assigned to all nodes of the server cluster.

Step 2: Collect Azure configuration parameters

You will need the key vault name and the key name from Azure.

Step 3: Configure Tableau Server for Azure Key Vault

Run the following command on Tableau Server. This command will restart the server:

- `tsm security kms set-mode azure --vault-name "<vault name>" --key-name "<key name>"`

The `--vault-name` and `--key-name` options are direct string copies from your Azure key vault.

For example, if your Azure key vault is named `tabsrv-keyvault` and your key is `tabsrv-sandbox-key01`, then the command would be as follows:

```
tsm security kms set-mode azure --vault-name "tabsrv-keyvault" --key-name "tabsrv-sandbox-key01"
```

Step 4: Enable encryption at rest

See [Extract Encryption at Rest](#).

Step 5: Validate installation

1. Run the following command:

```
tsm security kms status
```

The following information may be returned:

- Status: OK (indicates the Key Vault is accessible by the controller node):
 - Mode: Azure Key Vault
 - Vault name: `<key_vault_name>`
 - Azure Key Vault key name: `<key_name>`
 - List of available UUIDs for MEKs indicating which key is active
 - Error information if the KMS data is not accessible
2. View logs after you encrypt and decrypt extracts:
 - Publish extracts to your site and then encrypt them. See [Extract Encryption at Rest](#).

Tableau Server on Linux Administrator Guide

- Access the extracts with Tableau Desktop or with Web Authoring on a browser (this will decrypt the extracts for use).
- Search the `vizqlserver_node` log files for the `AzureKeyVaultEnvelopeAccessor` and `AzureKeyVaultEnvelope` strings. The default location of the logs are at `/var/opt/tableau/tableau_server/data/tabsvc/logs/`

For publishing and extract refreshes related to the Azure Key Vault, search the backgrounder logs. For more information about logs, see [Tableau Server Logs and Log File Locations](#).

Troubleshoot configuration

Multi-node misconfiguration

In a multi-node setup for Azure Key Vault, the `tsm security kms status` command may report healthy (OK) status, even if another node in the cluster is misconfigured. The KMS status check only reports on the node where the Tableau Server Administration Controller process is running. It does not report on the other nodes in the cluster. By default the Tableau Server Administration Controller process runs on the initial node in the cluster.

Therefore, if another node is misconfigured so that Tableau Server is unable to access the Azure key, those nodes may report Error states for various services, which will fail to start.

If some services fail to start after you have set KMS to the "azure" mode, then run the following command to revert to local mode: `tsm security kms set-mode local`.

Refresh Azure Key

You refresh the Azure key in Azure. There is no required or scheduled key refresh period. You can refresh your key by creating a new key version in Azure. Because the Key Vault name and Key Name do not change, you do not need to update the KMS configuration on Tableau Server for normal Azure Key refresh scenarios.

Back up and restore with Azure Key Vault

A server backup can be taken in Azure Key Vault mode with no additional configurations or procedures. The backup contains encrypted copies of the RMK and MEKs. Decrypting the keys requires access and control of the Azure Key Vault.

For the restore scenario, the server being restored to can be in either Azure Key Vault or Local KMS mode. The only requirement is that the server to which the backup is being restored has access to the Azure Key Vault the backup itself used.

Tableau Server External File Store

This topic provides an overview of Tableau Server External File Store.

Tableau Server File Store stores extracts and workbook revisions. Typically, Tableau Server File Store is a built-in Tableau Server process that is installed locally on Tableau Server. Starting in Tableau Server 2020.1, you can configure Tableau Server to use an external storage to store the File Store data. The external storage must be a network share, a dedicated file storage that enables multiple users and heterogeneous client devices to retrieve data from a centralized disk capacity. This can be a Server Message Block (SMB) for Windows or a Network File System (NFS) for Linux installations. Users on a local area network (LAN) access the shared storage via a standard Ethernet connection.

With this new feature, Tableau Server can now be configured in two ways:

- Install File Store locally, meaning File Store is installed on the Tableau Server nodes.
- Use External File Store (Beginning in 2020.1).

Why use External File Store?

Using external storage has the following key benefits over installing File Store locally:

- **Centralized location:** When File Store is installed locally, the data needs to be replicated across multiple File Store nodes, which will consume network bandwidth. Moving data to a centralized location will eliminate the need for running File Store on multiple nodes in a Tableau cluster and replication between the nodes. This also reduces the disk space requirements on an individual node and also reduce the

network bandwidth usage since the data will not be replicated on multiple nodes.

- **Improving backup time:** Snapshot backup technologies are efficient, and by using snapshot backup of the Tableau Data, you can expect to significantly reduce the amount of time it takes to do a Tableau backup.

Managing External File Store

License Management

To configure External File Store, you must first enable this feature using Server Management Add-on product keys. For more information, see [About Tableau Advanced Management](#). If you don't have the Server Management Add-on keys activated or if the licenses is expired, you will see the following behavior:

- If you try to configure Tableau Server with External File Store during installation, you will see an error message, but you will be able to continue the installation and Tableau Server File Store will be installed locally.
- If you are already using External File Store, and the Server Management Add-on license expires, you will see the following behavior:
 - The server will fail on restart.
 - Backups will fail.
 - If you no longer have a valid Server Management Add-on license, but have a valid Tableau Server license, you can migrate the External File Store to local File Store, to get your server up and running again. For more information on how to migrate from external repository to local repository, see [Reconfigure File Store](#).

Supported Migration Scenarios

- Moving File Store installed locally on Tableau Server to an external managed storage (network attached storage).
- Move File Store from external managed storage to Tableau Server.

Backup and Restore

Backup on Tableau Server with External File Store is different than how you would create backup when File Store is installed locally. For more information on how to do backup and

restore on Tableau Server with External File Store, see [Backup and Restore with External File Store](#).

High Availability Considerations

Tableau Server does not manage or setup high availability for the External File Store. Your managed storage may have solutions to support redundancy and high availability.

Topology

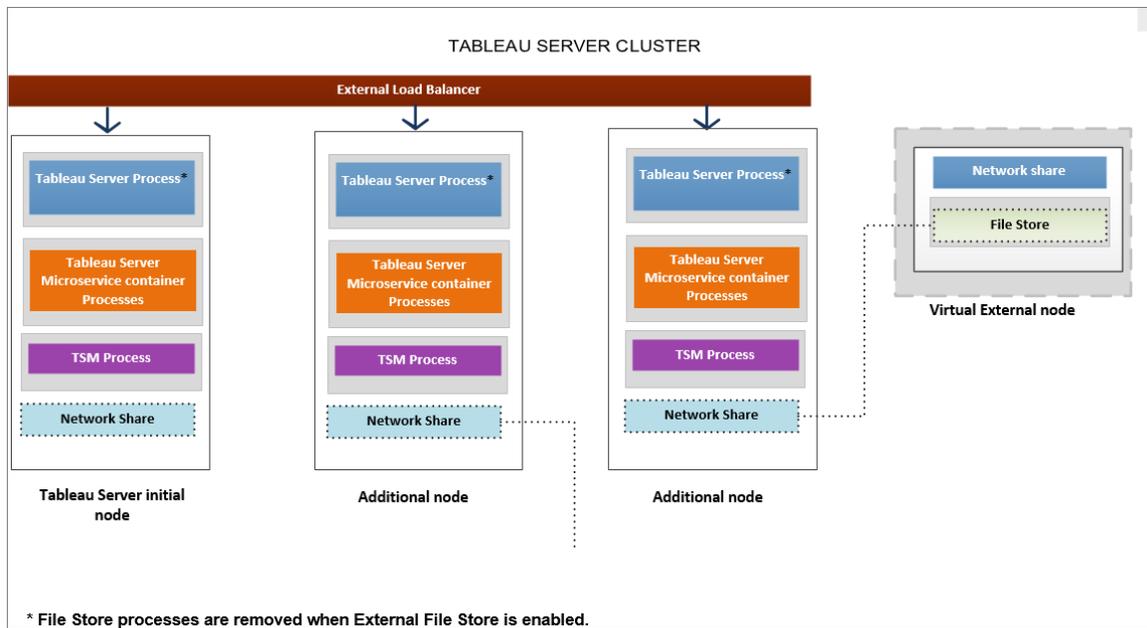
When you configure Tableau Server with External File Store, you will no longer run File Store locally. The Server status page will indicate that the File Store process is on an external node.

When File Store is configured external to Tableau Server, Data Engine and File Store are no longer co-located. During setup, Data Engine will continue to be automatically installed with other processes as described in [Tableau Server Data Engine](#), except for File Store. However, when you have Tableau Server configured with External File Store, you will be able to install Data Engine on a separate node without any other processes.

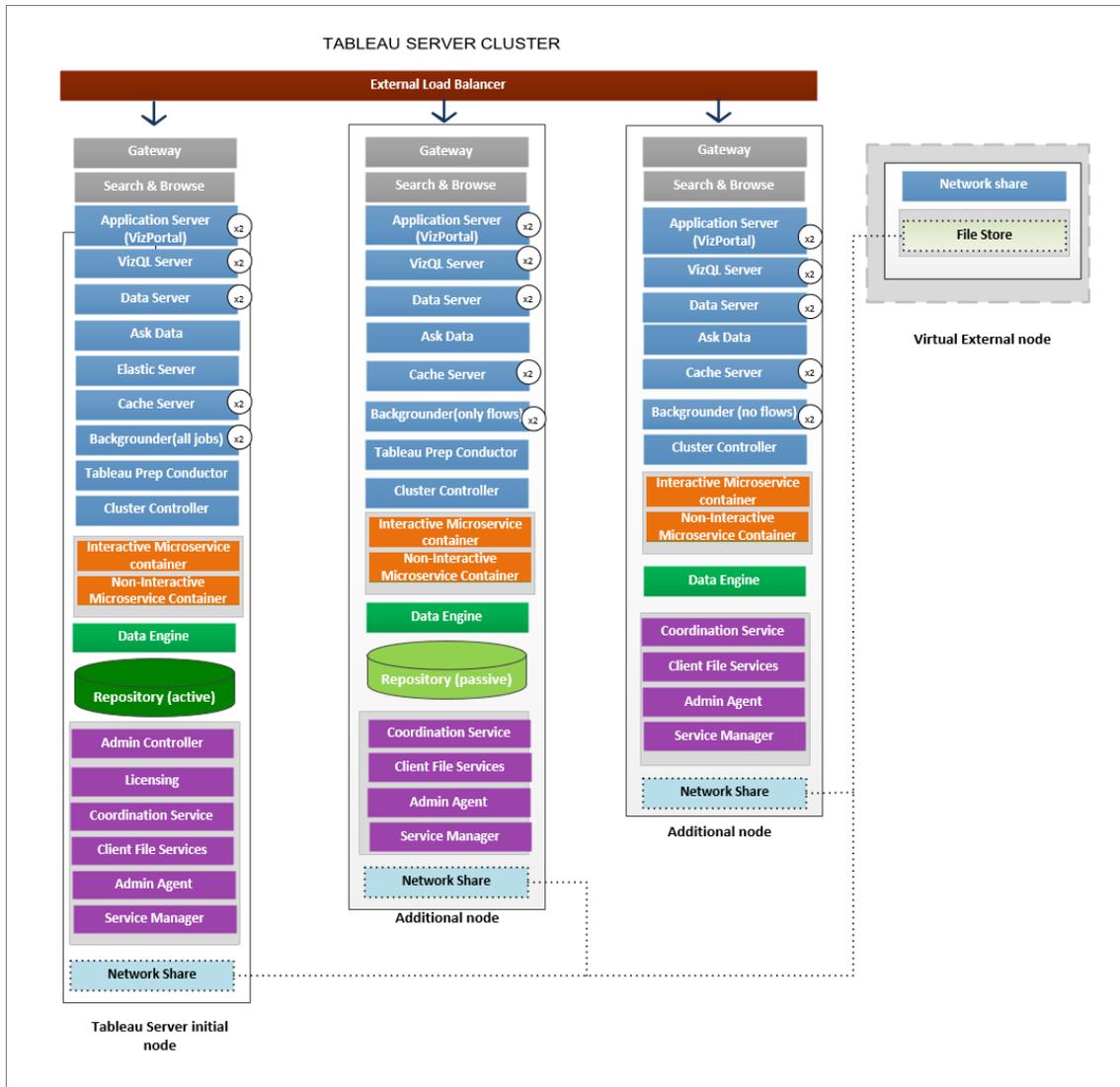
When File Store is configured externally, Data Engine will access the File Store data (extracts) on the storage system across the network. To make sure that your overall system performs to your requirements, there are a few things you must consider for your network and storage system. For more information, see [Performance Considerations for External File Store](#).

The diagram below is a summarized version of the Tableau Server topology with External File Store.

Tableau Server on Linux Administrator Guide



The diagram below is a detailed version of the Tableau Server topology with External File Store and shows all the processes installed on each node.



Next

Install Tableau Server with External File Store

Install Tableau Server with External File Store

This topic walks you through the process of configuring network share as your Tableau Server File Store for a new installation. If you are trying to do this on an existing installation

Tableau Server on Linux Administrator Guide

of Tableau Server that has File Store running locally, see [Reconfigure File Store](#) .

Prerequisites

- You must use Tableau Server 2020.1 or later.
- You must have network share that you can use as your storage option. For recommendations on the storage solution, see [Performance Considerations for External File Store](#).

Use **NFS** for Linux installations.

Estimating the storage size: You must take into account the amount of storage needed for publishing and refreshing extracts. In addition, you must also take into account the repository backup size unless you specifically choose the option to do your repository backup separately as described in the [Option 2: Backup Repository Separately](#) topic.

- **Extracts:**
 - Consider the number of extracts that will be published to Tableau Server and the size of each extract. Test your needs by publishing several extracts to Tableau Server, and then checking the disk space used. You can use this amount of disk space to help you figure out how many extracts will be published to Tableau Server over time as well as how each existing extract will increase in size.
 - Consider the space needed by the temp directory during an extract refresh. The temp directory, which is where an extract is stored to during a refresh, may require up to three times the final file size of the extract.
- **Repository Backup:**
 - To obtain an estimate of the repository data, check the size of `<data directory>/pgsql/data/base` directory.
 - To obtain the exact size of the repository data, open the backup file and use the size of the `workgroup.pg_dump` file.
- You must have a valid Advanced Management license activated on your server. To learn more about Advanced Management, see [About Tableau Advanced Management](#).

Install Tableau Server with External File Store

You can install Tableau Server with File Store using a network share to store Tableau Server data. This solution replaces the need for running the File Store process locally. To learn more about this solution and its benefits, see [Tableau Server External File Store](#).

Use the following steps to install your Tableau Server with External File Store during install:

Step 1: Configure a network share

On your file server:

1. Create and share a directory to use as the Tableau Server External File Store.
2. Make sure the network share is accessible as a directory in the same location on all the Tableau Server nodes.
3. Create the `tableau` directory in your network share and give full access to **tableau user** and **tableau group**. The tableau user will need read and write permissions to the directory on the network share. We recommend calling the directory '**tableau**'.

```
/mnt/<network share>/tableau/
```

4. **Validate that the network share is configured properly:** From Tableau Server run a command to write to a network share and confirm that you are able to write to it.

Step 2: Download and install TSM

1. Download the appropriate installer based on the distribution of Linux you are using.
2. Log on as a user with sudo access to the computer where you want to install Tableau Server.
3. Download the `.rpm` or `.deb` installer package.
4. Navigate to the directory where you copied the `.rpm` or `.deb` package.
5. Use the package manager to install the Tableau Server.
 - On **RHEL-like** distributions, including CentOS, you have the option to install Tableau Server to a non-default location.
 - Default location—To install to the default location (`/opt/tableau/tableau_server`), run

Tableau Server on Linux Administrator Guide

the following commands:

```
sudo yum update
```

```
sudo yum install tableau-server-<version>.x86_64.rpm
```

- **Non-default location**—to install to a non-default location, you must use `rpm -i`. You will also need to install all dependent packages. See the note below.

Run the following command:

```
sudo rpm -i--prefix/pREFERRED/INSTALL/PATH tableau-server.rpm
```

Note: When you use `yum` to install Tableau Server, all dependent packages are automatically downloaded and installed. This is the preferred method for installing Tableau Server. If you want to install to a non-default location, or your organization does not allow you to use `yum` and you must install using `rpm -i`, you must also install all dependent packages separately. For information about installing dependent packages, see [Installing Tableau Server on an Air-Gapped Computer Running Linux](#).

- On **Ubuntu and Debian**, run the following commands:

```
sudo apt-get update
```

```
sudo apt-get -y install gdebi-core
```

```
sudo gdebi -n tableau-server-<version>_amd64.deb
```

Step 3: Initialize TSM

1. Run the following script to start TSM:

```
sudo ./initialize-tsm --accepteula --<optional_parameters>
```

The only required parameter for the initialize-tsm script is `--accepteula`. You must include this parameter to accept the Tableau Server End User License Agreement (EULA). The EULA is available in the following location:

```
/opt/tableau/tableau_server/packages/docs.<version_
code>/Commercial_EULA.txt
```

2. Log off and log on again to the terminal before you configure Tableau Server.

When you log on again, you create a new session in which group membership changes have taken effect. The new session also has access to the environment variables added by the initialize-tsm script.

Alternatively, you can run the following command to update your path for the current session (but not to update your group membership):

```
source /etc/profile.d/tableau_server.sh
```

Step 4: Activate and register Tableau Server

Provide the Tableau Server Key and the Server Management Add-on key in the activate step. You will need to run the following command twice, first with the Tableau Server product key and then with the Server Management Add-on product key:

```
tsm licenses activate -k <product key>
```

Step 5. Enable External File Store

Configuring Tableau Server with external repository can only be done using TSM CLI.

1. Enable the network storage feature using the following tsm commands:

```
tsm topology external-services storage enable -network-
share /mnt/<network share name>/tableau
```

The setup program automatically creates the following directory structure in the share:

PostgreSQL data backups:

tableau_data/tabsvc/pg_backups

Note: This directory will be created the first time you create a backup.

Extracts and workbook revisions:

tableau_data tabsvc/dataengine/extracts

tableau_data/tabsvc/dataengine/revisions

Step 6: Configure the initial node settings

Follow the instructions provided in the Configure Initial Node Settings topic.

Step 7: Complete the install

You must create the initial administrative account for Tableau Server.

- If you configured a local identity store during setup, then specify a name and password that you want to use.
- If you configured a LDAP or Active Directory identity store during setup, then you must specify a user account that is a member of the directory.

To create the initial user, run the following tabcmd command:

```
tabcmd initialuser --server localhost:80 --username '<new-admin-username>'
```

After you run the command, the shell will prompt for an administrative password.

Step 8: Post-installation tasks

After you have created the Tableau Server administrator account, continue your deployment by working through the configuration steps described in the Post Installation Tasks topics.

Who can do this

Tableau Server Administrators can install and configure Tableau Server and External File Store. In addition, you must have permissions and access to configure the network share to use with Tableau Server.

Next

Backup and Restore with External File Store

Reconfigure File Store

Your Tableau Server may be configured with a locally running File Store or an External File Store. This topic describes the steps needed to reconfigure your existing Tableau Server:

- [Reconfigure your Tableau Server to use External File Store](#). This will move your File Store to a network share.
- [Reconfigure your Tableau Server to run File Store locally](#). This will move your File Store from the external storage to your Tableau Server.
- [Configure your Tableau Server to use a different storage](#). An example of this might be when your current network share is at the end of life and you need to use a new network share with new hardware.

Reconfigure Tableau Server with External File Store

Prerequisites

- Tableau Server should be version 2020.1 or later.
- You must have a network share to use as the external storage.

Use **NFS** for Linux installations.

Storage and network considerations: See Performance Considerations for External File Store.

Estimating the storage size: You must take into account the amount of storage needed for publishing and refreshing extracts. In addition, you must also take into account the repository backup size unless you specifically choose the option to do your repository backup separately as described in the Option 2: Backup Repository Separately topic.

- Extracts:
 - Consider the number of extracts that will be published to Tableau Server and the size of each extract. Test your needs by publishing several extracts to Tableau Server, and then checking the disk space used. You can use this amount of disk space to help you figure out how many extracts will be published to Tableau Server over time as well as how each existing extract will increase in size.
 - Consider the space needed by the temp directory during an extract refresh. The temp directory, which is where an extract is stored to during a refresh, may require up to three times the final file size of the extract.
- Repository Backup:
 - To obtain an estimate of the repository data, check the size of `<data directory>/pgsql/data/base` directory.
 - To obtain the exact size of the repository data, open the backup file and use the size of the `workgroup.pg_dump` file.
- Tableau Server should have a valid and activated Advanced Management license.

Step 1: Upgrade Tableau Server

Upgrade your Tableau Server to 2020.1 or later: Upgrading from 2018.1 and Later (Linux) .
If your Tableau Server is already on version 2020.1 or later, you can skip this step.

Step 2: Activate the Advanced Management license

1. View Server Licenses to make sure you have a Advanced Management license activated on your Tableau Server.
2. If you don't have a Advanced Management on your Tableau Server, use the following `tsm` command to activate the license. Provide the Advanced Management key as your

product key:

```
tsm licenses activate -k <product key>
```

Step 3: Configure File Store to use an external storage

After completing the upgrade and verifying the licenses, configure Tableau Server with External File Store. This will move any existing data from your local File Store to the external storage of your choice.

1. Configure a network share. On the File Server:
 - Create and share a directory to host the files
 - On all the Tableau Server nodes, map the network share or use a UNC path.
2. Create a tableau directory in your network share and give full access to **tableau user** and **tableau group**. The tableau user will need read and write permissions to the directory on the network share. We recommend calling the directory **tableau**.
3. Enable the network share feature using the following tsm commands:

```
tsm topology external-services storage enable -network-  
share /mnt/<network share name>/tableau
```

The setup program automatically creates the following directory structure and will move the data from the local File Store to the external storage. The local File Stores will be automatically decommissioned during this process.

Extracts and workbook revisions:

```
tableau_data tabsvc/dataengine/extracts
```

```
tableau_data/tabsvc/dataengine/revisions
```

PostgreSQL data backups:

tableau_data/tabsvc/pg_backups

Note: This directory will be created the first time you create a backup.

Reconfigure Tableau Server to use local File Store

1. Stop Tableau Server by running the following command:

```
tsm stop
```

2. Run the following command to move the File Store data from the external storage to Tableau Server:

```
tsm topology external-services storage disable -fsn <node1,  
node2>
```

3. Start Tableau Server by running the following command:

```
tsm start
```

For a Tableau Server cluster, specify the nodes where File Store should be installed. The data is copied to the first node specified in the command and then replicated to the other nodes.

Note: When moving File Store from external to local, make sure Data Engine process is not installed by itself on a separate node and is installed along with one of the core services which include File Store, Application Server (VizPortal), VizQL Server, Data Server, and Backgrounder.

Typically, when you install Tableau Server Data Engine installation happens automatically and is installed on the nodes that have one or more of the core services. However, when you configure Tableau Server to use external storage, you will have the ability to manually install Data Engine on a node on its own without co-locating with the

core processes.

If you currently have a Data Engine process installed on a separate node, you can either choose to install File Store on that node or remove Data Engine from that node, before running the disable command. If you install File Store on a node that currently does not have Data Engine installed, Data Engine will be added automatically.

If you have a Data Engine only node when you run the disable command, it will result in an error.

Configure Tableau Server to use a different external storage

1. Configure the new network share. On the File Server:
 1. Create and share a directory to host the files.
 2. On all the Tableau Server nodes, map the network share or use a UNC path.
2. Create a tableau directory in your network share and give full access to *Network Service* account. We recommend calling the directory **tableau**.
3. Stop Tableau Server by running the following command:

```
tsm stop
```

4. Run the following command to configure Tableau Server to use the new network share:

```
tsm topology external-services filestore --switch-share --
network-share \\newshare\tableau
```

5. Start Tableau Server by running the following command:

```
tsm start
```

Who can do this

Tableau Server Administrators can move File Store locations. In addition, you will need access to the external storage that is used for the External File Store.

Backup and Restore with External File Store

If your Tableau Server is configured to run File Store externally, the backup and restore process is different than if you were running File Store locally. This topic describes the process of backup and restore when:

- **You have Tableau Server configured with External File Store.**
- **You have Tableau Server configured with External File Store and External Repository.**

Backup Strategies:

The backup strategy you use depends on your recovery plan. The snapshot backup process may or may not be sufficient as it only creates a backup of the File Store and repository data. There are other configurations and settings that you might need to do a full recovery. Here are some scenarios that illustrate where the snapshot backup may or may not be enough:

- You maintain a standby Tableau Server to use in case your production Server is down. In this scenario, creating a snapshot backup and restoring it to your standby server on a regular schedule might be sufficient. Your backup schedule should be according to your recovery point objective.
- You plan to use a new Tableau Server installation in case of a disaster, but don't necessarily need to use the configurations and settings from your existing Tableau Server installation. In this case, you can install a new Tableau Server, and use the snapshot to restore the data.
- You don't have a standby Tableau Server, but want to be able to create a new Tableau Server installation using the configurations and settings from your existing Tableau Server and restore the data. In this scenario, you would need more than the

snapshot backup to install a new Tableau Server. To do a full backup including all the configurations and settings, follow these instructions:

1. Export topology and configuration data. This exports majority of Tableau Server and the topology. For more information, see [Perform a Full Backup and Restore of Tableau Server](#)
2. Create a snapshot of the network share to create a backup of the File Store and Repository data as described earlier in the [Backup and Restore with External File Store](#) section of this topic.
3. Document the settings that are not included in the export. These are settings like system user accounts, coordination serviced deployment configuration, customized settings, and such. For more information, see [Perform a Full Backup and Restore of Tableau Server](#).

Tableau Server with External File Store

When you have External File Store enabled on your Tableau Server, you cannot use the *tsm maintenance backup command* to do a backup of the Tableau Server Repository and File Store Data. Instead, use the storage snapshot technology to create a point in time snapshot of your network share. However, before you can create the snapshot backup, you must first prepare Tableau Server for the snapshot backup and on completion resume Tableau Server normal processes.

When you have Tableau Server that is configured with External File Store but is using a local repository, Tableau Server backup operations include a backup of the Repository.

Creating a Snapshot Backup

Use the following steps to create a snapshot backup.

1. **Prepare for snapshot backup:** Run the following command to create a repository backup file , and temporarily copy it to the network share. Server continues to operate normally during the backup process. Some internal processes will be paused during this time to ensure a consistent snapshot.

```
tsm maintenance snapshot-backup prepare
```

You should see a message that says '**Preparation for snapshot backup succeeded**' to indicate that the preparation step is done.

Check to make sure that the repository backup file was created on the network share.

Note: The prepare step includes creating a backup file of the repository and KMS and Asset keys. If you are using a cloud solution for your encryption key solution, you will need access to the CMK used to decrypt the keys which is necessary for restore. For more information on encryption key solution with AWS, see [AWS Key Management System](#). For more information on encryption key solution with Azure, see [Azure Key Vault](#).

2. **Create a snapshot of the network share:** Use the snapshot technology to create a snapshot of the network share. **This snapshot will include both the File Store data and the repository backup file that was created in the previous step.**
3. **Complete the backup process:** Run the following command to complete the backup process in restart the paused internal processes. This also removes the repository backup file that was copied to your network share.

```
tsm maintenance snapshot-backup complete
```

Check to make sure that the temporary repository backup file was removed from the network share.

Restoring a snapshot backup

The steps described below apply to both stand-alone and Tableau Server cluster.

1. Stop Tableau Server. At the command prompt, run the following command:

```
tsm stop
```

2. Restore File Store data. Use the point in time snapshot that you want to recover to,

and restore it to your shared file server. Once the restore is complete, Tableau Server is using the restored File Store data.

3. Restore Repository Data. Use the following restore command to restore the repository data.

```
tsm maintenance snapshot-backup restore
```

Note: If you are using a cloud solution for your encryption key solution, make sure the server where the backup is being restored has decrypt access to cloud instance where CMK is deployed.

4. Restart Tableau Server. At a command prompt, run the following command:

```
tsm start
```

Tableau Server with External File Store and External Repository

When you have External File Store enabled on your Tableau Server, you cannot use the `tsm maintenance backup` command to do a backup of the Tableau Server Repository and File Store Data. Instead, use the storage snapshot technology to create a point in time snapshot of your network share.

When you have both External File Store and External Repository enabled, you have two options when you do your backup: You can either choose to backup the External Repository separately or include it as part of the initial Prepare step.

Here are some of the reasons why you may choose to back up the Repository separately:

- Doing a backup of the Database instance of the External Repository separately can be faster, especially if you are using a cloud solution that allows for snapshot backup of the instance.
- The size of the backup file impacts the time taken in the preparation step of the backup process since it has to be copied to the network share.

On the other hand the advantage of including the Repository backup as part of the Tableau Server backup for the following reasons:

- **Ease of backup management:** By including the backup of the Repository, you are not having to do a separate backup and making sure that the backup is completed in the time that is necessary to keep the File Store and Repository backups in sync.
- **Restore:** Restore operations have to be done separately and typically requires additional steps of creating a new instance and configuring the Tableau Server to use the new restored instance.

Option 1: Include Repository in the Backup

This option describes how to include Repository as part of the Tableau Server Backup process. If you prefer to create the backup file of the repository separately, see [Backup and Restore with External File Store](#).

Create a Snapshot Backup

1. **Prepare for snapshot backup:** Run the following command to start the process so you can take a snapshot backup of the network share. Some internal processes will be paused during this time to ensure a consistent snapshot, but Tableau Server continues to operate normally during the backup process. **In order to make sure that the repository backup is included you must use the `--include-pg-backup` option.**

Note: Starting in 2021.1, if you have both External File Store and External Repository enabled, the prepare step no longer includes creating a backup file of the repository. If you are upgrading from a version earlier than 2021.1, and you have scripts to run or schedule your backups, and want to continue to include the repository backup, add the `--include-pg-backup` option in the prepare command as shown below.

```
tsm maintenance snapshot-backup prepare --include-pg-backup
```

You should see a message that says '**Preparation for snapshot backup succeeded**' to indicate that the preparation step is done.

Check to make sure that the repository backup file was created on the network share.

Note: The prepare step includes creating a backup file of the repository and KMS and Asset keys. If you are using a cloud solution for your encryption key solution, you will need access to the CMK used to decrypt the keys which is necessary for restore. For more information on encryption key solution with AWS, see [AWS Key Management System](#). For more information on encryption key solution with Azure, see [Azure Key Vault](#).

2. **Create a snapshot of the network share:** Use the snapshot technology to create a snapshot of the network share. **This snapshot will include both the File Store data and the repository backup file that was created in the previous step.**
3. **Complete the backup process:** Run the following command to complete the backup process in restart the paused internal processes. This also removes the repository backup file that was copied to your network share.

```
tsm maintenance snapshot-backup complete
```

Check to make sure that the temporary repository backup file was removed from the network share.

Restoring a Snapshot Backup

The steps described below apply to both stand-alone and Tableau Server cluster.

1. Stop Tableau Server. At the command prompt, run the following command:

```
tsm stop
```

2. Restore File Store data. Use the point in time snapshot that you want to recover to, and restore it to your shared file server. Once the restore is complete, Tableau Server

is using the restored File Store data.

3. Restore Repository Data. Use the following restore command to restore the repository data:

```
tsm maintenance snapshot-backup restore
```

Note: If you are using a cloud solution for your encryption key solution, make sure the server where the backup is being restored has decrypt access to cloud instance where CMK is deployed.

4. Restart Tableau Server. At a command prompt, run the following command:

```
tsm start
```

Option 2: Backup Repository Separately

This option is recommended only when the host platform allows you to do snapshot backup. If you are using Azure as your host platform, we recommend using Option 1.

Create Snapshot Backups

1. Prepare for snapshot backup: Run the following command to create a repository backup file, and temporarily copy it to the network share. Server continues to operate normally during the backup process. Some internal processes will be paused during this time to ensure a consistent snapshot.

```
tsm maintenance snapshot-backup prepare
```

Note that the Repository is not included in the backup in this case.

You should see a message that says '**Preparation for snapshot backup succeeded**' to indicate that the preparation step is done.

Note: The prepare step includes creating a backup file of the repository and KMS and Asset keys. If you are using a cloud solution for your encryption key solution, you will need access to the CMK used to decrypt the keys which is necessary for restore. For more information on encryption key solution with AWS, see [AWS Key Management System](#). For more information on encryption key solution with Azure, see [Azure Key Vault](#).

2. **Create a snapshot of the network share:** Use the snapshot technology to create a snapshot of the network share. This snapshot will include both the File Store data that was created in the previous step.
3. **Create a backup of the repository:** Use the backup technology of platform where you are hosting your external repository to create a backup.

The backup of the network share and the repository **must be completed within 3 hours and 30 minutes after completing the Prepare step**. This is to make sure that the File Store and the Repository backups are in sync and for the restore to work properly.

For more information on creating a snapshot of AWS DB instance, see [Creating a DB snapshot](#).

For more information on creating a backup of Azure DB instance, see [Backup and Restore on Flexible Server](#) (PostgreSQL 12 and later), [Backup and Restore on Single Server](#) (PostgreSQL 11 or earlier).

Note: If you are using a cloud solution for your encryption key solution, make sure the server where the backup is being restored has decrypt access to cloud instance where CMK is deployed.

4. **Complete the backup process:** Run the following command to complete the

backup process in restart the paused internal processes. This also removes the repository backup file that was copied to your network share:

```
tsm maintenance snapshot-backup complete
```

Restoring a Snapshot Backup

The steps described below apply to both stand-alone and Tableau Server cluster.

1. Use the DB snapshot backup to your external repository. If you are using a cloud platform to host your Repository, typically this requires that you create a new DB instance to restore the backup. Following the steps described in these articles to create a new instance:

[AWS: Create a PostgreSQL DB Instance on AWS Relational Database Service \(RDS\)](#)

2. Create a configuration file for the new instance. **Use instructions described in Step 1** of the [Install Tableau Server with External PostgreSQL Repository](#) topic.
3. Stop Tableau Server. At the command prompt, run the following command:

```
tsm stop
```

4. If the restore of the external repository creates a new DB instance, use the following command to point Tableau Server to the new DB instance:

```
tsm topology external-services repository replace-host -f  
<filename>.json -c <ssl certificate file>.pem
```

The .json file is the configuration file you created in Step 2. The certificate file is the SSL certificate you downloaded from the new DB instance.

5. Restore File Store data. Use the point in time snapshot that you want to recover to, and restore it to your shared file server. Once the restore is complete, Tableau Server is using the restored File Store data.

Note: Some technologies require you to create a new network share when doing a restore. If this applies to your network attached storage, you can do your restore **before** stopping the Tableau Server. Also, If you are restoring the File Store data to a new network share, you must configure Tableau Server to use the new network share. For more information, see [Configure Tableau Server to use a different external storage](#).

6. Run the following command to restore the KMS and Asset keys:

```
tsm maintenance snapshot-backup restore
```

Note: If you are using a cloud solution for your encryption key solution, make sure the server where the backup is being restored has decrypt access to cloud instance where CMK is deployed.

7. Run the following command to restart Tableau Server:

```
tsm start
```

Who can do this

Tableau Server Administrators can backup and restore Tableau Server. In addition you must have permission to access and perform snapshot backups on the external storage.

Performance Considerations for External File Store

This topic lists the factors you must consider when you have External File Store to make sure you have optimal performance.

Tableau Server on Linux Administrator Guide

In this scenario where File Store is configured external to Tableau Server, you are storing the extracts on a network share. This means that Tableau Server will be accessing this data across the network. To ensure optimal performance, we recommend the following:

- Use Enterprise grade storage system to ensure reliability and high data access performance.
- The storage system supports enough read IOPS:
 - Use Solid State Drives. If using spinning disks is the only option, use the fastest and as many as possible.
- The network infrastructure supports the following:
 - At least 10 GB Ethernet to support high speed data transfers between Tableau Server and the storage system.
 - No more than 10 millisecond storage latency between Tableau Server and the storage system.

The above recommendations are based on testing done by the Tableau team. Your requirements and performance may vary. We highly recommend that you create your own benchmarks to assess the performance and determine resource requirements.

When creating benchmarks, consider overall performance of Tableau Server with workbook load times as one of the key metrics. This is especially relevant to this configuration since External File Store mostly impacts extract based workbooks.

You can use [Tabjolt](#) to do your benchmarks.

Who can do this

Tableau Server Administrators monitor Tableau Server performance. However, there are network, hardware, and storage considerations that might either need access to make configuration changes to these resources. You may also choose to work with your network administrator to make any changes if you don't have access to these resources.

Tableau Server External Repository

The Tableau Server Repository is a PostgreSQL database that stores data about all user interactions, extract refreshes, and more.

The repository can be installed as locally on the same nodes as the Tableau Server or installed externally:

Local repository: The PostgreSQL Database is installed and deployed locally, meaning it is deployed along with Tableau Server.

External repository: The PostgreSQL Database is deployed externally. The external repository can be installed on Amazon RDS, Azure Database, or as a stand-alone installation.

For more information about what Tableau Server Repository is in general, see:

- [Workgroups Database](#)
- [Collect Data with the Tableau Server Repository](#)

The supported hosts for the external repository are:

- Amazon RDS (Beginning in version 2019.3)
- Azure Database (Beginning in version 2020.4)
- Stand-alone PostgreSQL Instance (Beginning in version 2021.2).

This topic is an overview of the Tableau Server external repository.

External Repository Considerations

Both Amazon RDS and Azure Database offer better scalability, reliability, high availability and security built-in for PostgreSQL. By integrating more closely with these cloud offerings, you will be able to take advantage of these additional benefits.

If you are setting up a stand-alone PostgreSQL instance, you must set up and manage high availability and scale as needed.

Requirements

- Your Tableau Server must be using the following versions:
 - 2019.3 or later to use with AWS
 - 2020.4 or later to use with Azure.

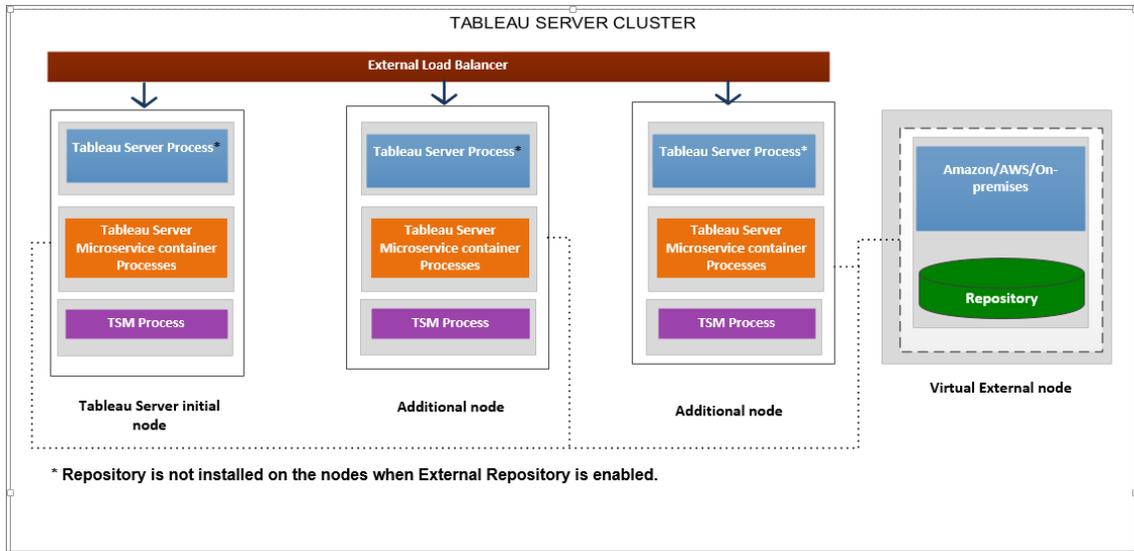
For the right version of PostgreSQL to use, see [Azure Database for PostgreSQL Flexible Server](#) .

Note: Azure Database for PostgreSQL version 12 is currently in public preview.

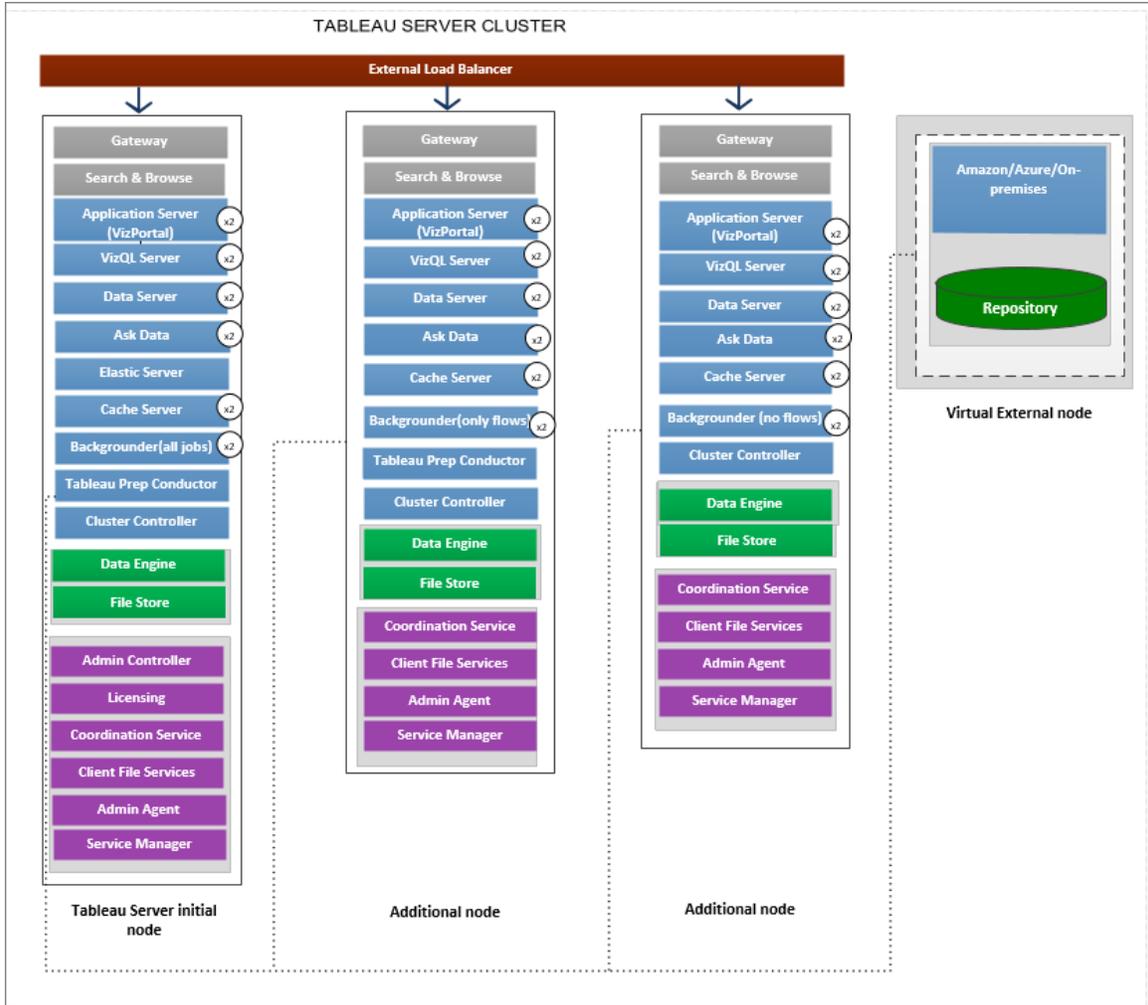
- 2021.2 or later to use with stand-alone PostgreSQL instance. (Can be used for on-premises, Azure VM, or AWS EC2 installations).
- Your Tableau Server must have the Advanced Management keys activated.
- Depending on where you are planning to host the External Repository, you will need to be familiar with one of the following:
 - Amazon RDS Database setup and management.
 - Azure Database setup and management.
 - PostgreSQL database setup and management as a standalone installation.

Topology

The diagram below is a summarized version of the Tableau Server topology with External Repository.



The diagram below is a detailed version of the Tableau Server topology with External Repository and shows all the processes installed on each node.



Managing the External Repository

License Management

To enable this feature you must first activate the Advanced Management product keys on Tableau Server. For more information, see [About Tableau Advanced Management](#). If you don't have the Advanced Management keys activated or if the licenses is expired, you will see the following behavior:

- If you try to configure Tableau Server to use external repository during installation, you will see an error message, but you will be able to continue the installation and Tableau

Server Repository will be installed locally. For single server installations, the repository is installed on the same machine as Tableau Server. For multi-node installations, the repository is installed on one of the nodes of your Tableau Server cluster.

- If you are already using an external Tableau Server Repository on your Tableau Server installation, and the Advanced Management license expires, the server will fail on restart. If you no longer have a valid Advanced Management license, but have a valid Tableau Server license, you will still be able to create a backup. You can also migrate the external repository to local repository which does not require the Advanced Management license, to get your server up and running again. For more information on how to migrate from external repository to local repository, see [Re-Configure Tableau Server Repository](#).

Supported Migration Scenarios

- Moving your repository from local to external.
- Moving your repository from external to local.

Backup and Restore

If you have only External Repository configured:

The backup and restore process remains the same for both local and external repository and as described in the [Back up Tableau Server data](#) topic.

- The backup and restore commands work in the same manner for both local and external repository. Backup in the case where Tableau Server uses external repository requires more disk space for backup, so you should keep that in mind when configuring your disk space.
- The default and other supported locations for storing the backup file are the same whether it the repository is local or external.

The following exceptions apply to Tableau Servers using external repository:

- Any custom user accounts that are created in the external Tableau Server repository will be included in the restore, but the passwords for the custom user accounts will not be. The passwords will have to be reconfigured after restore is complete. Custom user accounts are PostgreSQL database user accounts, used by SQL or other database client software to connect to the PostgreSQL database.

Note: These custom accounts will be disabled as a security measure, but this can be reconfigured.

- Configuration and topology should not be included in the backup. For more information on how to export configuration and topology settings, see [Perform a Full Backup and Restore of Tableau Server](#).

If you have both External Repository and External File Store configured:

You have a couple of options if you have both External Repository and External File Store configured for your Tableau Server. These options allow you to take advantage of the snapshot backup capabilities of the cloud platforms you might be using to host the External Repository and File Store. For more information, and detailed step by step instructions, see [Backup and Restore with External File Store](#).

SSL Connections

You have the option to require or not require TLS/SSL connections from Tableau Server to the External Repository. If you do not need to have secure connections between Tableau Server and the External Repository, you will need to configure the RDS, Azure Database, or the stand-alone PostgreSQL instance to not use TLS/SSL connections, and configure Tableau Server to use the External Repository with the `--no-ssl` option. For more information, see `tsm topology external-services repository enable`.

Updating the SSL Certificate

If as part of a planned expiration of the SSL certificate of the RDS or Azure Database, or a stand-alone PostgreSQL instance, you need to update the instance with the new certificate

file, you will also need to update Tableau Server settings to use the new certificate file. You can do this by downloading the latest file and running the `tsm topology external-services repository replace-host` command and providing the new certificate file.

High Availability Considerations

Tableau Server does not manage or setup high availability for the external repository.

- **AWS:** Amazon RDS offers high availability features that can be used to provide high availability and manage failover. For more information, see [Amazon RDS High Availability](#).
- **Azure:** Azure offers high availability features that can be used to provide high availability and manage failover. For more information, see [Azure Database High Availability](#).
- **Stand-alone PostgreSQL instance:** PostgreSQL offers several features that can be used to provide high availability and manage failover. For more information, see [PostgreSQL High Availability](#).

Monitoring the Status of the Repository

TSM status page will show the Tableau Server external service as an additional node for your Tableau Server installation.

Tableau Server on Linux Administrator Guide

Process	node1	external
Gateway	<input checked="" type="checkbox"/>	
Application Server	<input checked="" type="checkbox"/>	
Interactive Microservice Container	<input checked="" type="checkbox"/>	
VirQL Server	<input checked="" type="checkbox"/>	
Cache Server	<input checked="" type="checkbox"/>	
Cluster Controller	<input checked="" type="checkbox"/>	
Search & Browse	<input checked="" type="checkbox"/>	
Backgrounder	<input checked="" type="checkbox"/>	
Non-interactive Microservice Container	<input checked="" type="checkbox"/>	
Data Server	<input checked="" type="checkbox"/>	
Data Engine	<input checked="" type="checkbox"/>	
File Store	<input checked="" type="checkbox"/>	
Repository		<input checked="" type="checkbox"/>
Tableau Prep Conductor		
Ask Data	<input checked="" type="checkbox"/>	
Elastic Server	<input checked="" type="checkbox"/>	
TSM Controller	<input checked="" type="checkbox"/>	
License Server	<input checked="" type="checkbox"/>	

The topology tab indicates whether there are Tableau Server external services configured:

External Services
The following services are set to external and will not be present in the nodes. [Learn more](#)

- Repository

Add a Node

Step 1
Download the node bootstrap configuration file and locate your Tableau Server installer. The same installer can be used to install multiple nodes. Having trouble finding the installer?
[Download Bootstrap File](#)

Step 2
Run the node installer on the new node, and when prompted, provide the configuration file. Tableau Services Manager will detect the new node and display it on the Topology page.
[Learn more about adding, removing, and managing nodes in Tableau Services Manager.](#)

Getting Logs

Tableau Server logs will not include logs from the external repository. Use the following ways to get the logs specific to your instance:

- **AWS:** For information on setting up logging for your Amazon RDS PostgreSQL instance, see [PostgreSQL Database Log Files](#).
- **Azure:** For information on setting up logging for your Azure Database for PostgreSQL instance see [Logs in Azure Database for PostgreSQL](#).
- **Stand alone PostgreSQL Instance:** For information on setting up logging for your stand alone PostgreSQL instance, see [Error Reporting and Logging](#).

Next Steps

- You can use one of the following to create a PostgreSQL database instance:
 - Create a PostgreSQL DB Instance on AWS Relational Database Service (RDS)
 - Create a Azure Database PostgreSQL Instance on Azure
 - Create a PostgreSQL Database as a Stand-alone Installation
- Install Tableau Server with External PostgreSQL Repository
- Re-Configure Tableau Server Repository
 - Migrate from local to external
 - Migrate from external to local

Create a PostgreSQL DB Instance on AWS Relational Database Service (RDS)

Beginning in version 2019.3, you can host your External Repository on the AWS Cloud Platform. This topic describes how to create a AWS RDS PostgreSQL DB instance to use as your Tableau Server external repository.

To see a full list of hosts that you can use for the your external repository, see [Tableau Server External Repository](#).

Tableau Server on Linux Administrator Guide

Requirements and Recommendations

- At a minimum use db.m4.2xlarge Amazon RDS memory optimized instance type. This is the minimum recommended AWS RDS instance size to use for Tableau Server external repository, but the exact requirements will vary with your requirements and usage. We recommend db.r4.xlarge (or db.r5.xlarge) Amazon RDS memory optimized instance types for maximum performance.

You can start with a smaller instance type and if you find later that you need a larger instance type, you can upgrade your existing RDS instance. For more information, see [Upgrading your RDS Instance](#) .

Secure communications between Tableau Server and the external PostgreSQL DB instance using SSL is required for Tableau Server versions 2021.2 and 2021.2.1. Starting from version 2021.2.2, SSL connections are optional but recommended.

- The PostgreSQL DB instance must be reachable by all nodes in the Tableau Server cluster. One of the ways you can do this is by making the PostgreSQL DB instance a member of a security group that has the necessary permissions to be accessed by all the nodes in the Tableau Server cluster.
- The version of PostgreSQL should match the version used by Tableau Server when installed locally. Tableau Server 2020.4 uses PostgreSQL version 12.

Create a PostgreSQL DB instance on Amazon RDS

Step 1: Create a parameter group

PostgreSQL parameters that you set for a local PostgreSQL instance in the *postgresql.conf* file are maintained in the DB parameter group for your DB instance. When you create a DB instance, the parameters in the associated DB parameter group are loaded.

You can modify parameter values by changing values in the parameter group. You can also change parameter values, if you have the security privileges to do so, by using the ALTER DATABASE, ALTER ROLE, and SET commands.

There are two types of PostgreSQL parameters, static and dynamic. Static parameters require that the DB instance be rebooted before they are applied. Dynamic parameters can be applied immediately.

The following table shows parameters that you should modify with Tableau recommended values:

Name	Data type	Parameter Type	Recommended Tableau Configuration Values	Description
autovacuum	boolean	Dynamic	1	Starts the autovacuum sub-process.
client_min_messages	string	Dynamic	error	Sets the message levels that are sent to the client.
escape_string_warning	boolean	Dynamic	0	Warns about backslash (\) escapes in ordinary string literals.
lc_messages	string	Dynamic	en_US.UTF-8	Sets the language in which messages are displayed.
lc_monetary	string	Dynamic	en_US.UTF-8	Sets the locale for formatting monetary amounts.
lc_numeric	string	Dynamic	en_US.UTF-8	Sets the locale for formatting numbers.
lc_time	string	Dynamic	en_US.UTF-8	Sets the locale for formatting date and time values.
log_auto-vacuum_min_duration	integer	Dynamic	100	Sets the minimum execution time above which auto-vacuum actions will be logged.

Tableau Server on Linux Administrator Guide

Name	Data type	Parameter Type	Recommended Tableau Configuration Values	Description
log_connections	boolean	Dynamic	1	Logs each successful connection.
log_lock_waits	boolean	Dynamic	1	Logs long lock waits.
log_min_duration_statement	integer	Dynamic	100	Sets the minimum execution time above which statements will be logged.
log_min_messages	string	Dynamic	error	Sets the message levels that are logged.
log_temp_files	integer	Dynamic	128	Logs the use of temporary files larger than this number of kilobytes.
rds.force_ssl	boolean		1	Force SSL connections; requires import of certs on client keystore
rds.logical_replication	boolean		1	
ssl	boolean	Static	1	Set to allow SSL connections
standard_conforming_strings	boolean	Dynamic	0	Causes ... strings to treat backslashes literally.
work_mem	integer	Dynamic	16384	Sets the maximum memory to be used for query workspaces.

For more information and a full list of PostgreSQL Parameters, see [Working with PostgreSQL Parameters](#), on the AWS site.

Step 2: Create a PostgreSQL DB instance on Amazon RDS

To create a new PostgreSQL DB instance, follow the instructions provided on the [Amazon documentation site](#).

Following are configuration options and the recommended values for the new PostgreSQL DB instance:

- **Instance Specifications**

- Use the parameter group created in **Step 1**.
- Use the compatible version of PostgreSQL for the version of Tableau Server you are using. For a full list of PostgreSQL versions compatible with Tableau Server versions, see Product Compatibility.
- Use DB instance class that is db.m4.2xlarge or larger.
- Allocate at least 100GB of storage.
- Storage type and Provisioned IOPS: leave default (recommendations may change depending on load testing).

- **Settings**

- You **must** use **rails** as the master username.

This is a requirement for the external repository to work properly with Tableau Server.

- Pick a password that meets AWS's requirements.

- **Network and Security**

- Make sure that the RDS instance can be reached by all the Tableau Server nodes. This most often involves creating a security group that allows access from the nodes.

- **Database Options**

Tableau Server on Linux Administrator Guide

- Don't create an initial database. The Database name should be left blank, as Tableau Server will create the needed databases in the RDS instance.
- The port can be anything, but we recommend leaving it as the default 5432.
- Set the DB Parameter Group to the one created in **Step 1**.
- Leave the IAM DB authentication as disabled.
- **Encryption**
 - You can choose whether or not you want encryption.
- **Backup**
 - This is for AWS's automated backups, not Tableau Server's backups. You can specify the settings that meets the requirements.
- **Monitoring**
 - You can specify the settings based on your requirements.
- **Log Exports**
 - You can specify the settings based on your requirements.
- **Maintenance**
 - Disable auto minor version upgrade. Tableau Server is built to use a specific version of PostgreSQL. and you will be prompted to upgrade the PostgreSQL version if needed, during Tableau Server Upgrade.
- **Delete Protection**
 - You can specify the settings based on your requirements.

Step 3: Get the PostgreSQL DB Instance Endpoint

After creating the PostgreSQL database instance, you can't use it until it's completed initialization by AWS and this can take several minutes. Once the instance is ready, get the endpoint information that you will use to configure Tableau Server to use this instance for the Tableau Server Repository.

Step 4: Download the SSL certificate file (.pem file)

Starting in version 2021.2.2, SSL is optional but recommended to connect to the DB instance, but required if you are using Tableau Server version 2021.2 and 2021.2.1.

If you want to use secure connections between Tableau Server and the External Repository, you will need the .pem file when you configure Tableau Server to use the external DB instance for your Tableau Server Repository. For more information, see [Using SSL to Encrypt the Connection to a DB Instance](#).

If you do not need to use secure connections between Tableau Server and External Repository, you need to configure the RDS instance to allow unencrypted connections.

Important: If as part of a planned expiration of the SSL certificate of the RDS instance, you need to update your RDS instance with the new certificate file, you also need to update Tableau Server settings to use the new certificate file. You can do this by downloading the latest file and running the `tsm topology external-services repository replace-host` command and providing the new certificate file.

Configuring High Availability for your PostgreSQL DB

Tableau Server does not manage or setup high availability for the external repository. Amazon RDS offers high availability features that can be used to provide high availability, manage failover, etc. For more information, see [Amazon RDS High Availability](#).

Disaster Recovery for your PostgreSQL DB

In the event of a disaster, you may need to setup a new RDS instance. There are other scenarios where you may need to recover from an issue with the RDS instance. For example, when you upgrade your Tableau Server, you might also need to upgrade the PostgreSQL

version on your RDS instance. In the event that your PostgreSQL upgrade is not successful you might have to use a new RDS instance. In such scenarios, to configure your Tableau Server to use the new RDS instance, use the following steps:

1. **Restore the snapshot to a new RDS instance.** AWS does not support restoring a snapshot to an existing RDS instance. For more on RDS snapshot backup and restore, see [Amazon RDS Backup and Restore](#).
2. **Create a new JSON settings file** containing connection information for the new RDS instance. For more information on creating a JSON settings file, see **Step 1** in [Install Tableau Server with External PostgreSQL Repository](#).
3. **Use the `tsm topology external-services repository replace-host` command** to point your Tableau Server to the new RDS instance.

For more information on the `tsm topology external-services repository replace-host` command, see [tsm topology](#).

Who can do this

Only Tableau Server Administrators can configure Tableau Server to use the external repository. You will also need an AWS account to create a RDS instance.

Create a Azure Database PostgreSQL Instance on Azure

Beginning in version 2020.4, you can host your External Repository on the Azure Cloud Platform. This topic describes how to create a Azure Database for PostgreSQL instance to use as your Tableau Server external repository.

Requirements and Recommendations

- We recommend that you use 8 vCore memory optimized server with 50 GB of storage for Tableau Server external repository, but the exact requirements will vary with your requirements and usage. If you already have a Tableau Server, review the usage of your existing repository to determine your storage needs.

You can also scale your resources if you find that you need more. For more information, see [Scaling your PostgreSQL Azure Database resources](#).

- Secure communications between Tableau Server and the external PostgreSQL DB instance using SSL is required if you are using 2021.2 and 2022.2.1.

For versions 2021.2.2 and later, SSL connections between Tableau Server and the External Repository is optional but recommended. If you do not want to use secure connections between Tableau Server and External Repository, you should configure the Azure Database to allow unencrypted connections.

- The PostgreSQL DB instance must be reachable by all nodes in the Tableau Server cluster. The database instance must be set up to allow connections from all the Tableau Server nodes. There are two ways to set this up:
 - This is most secure way: Configure Azure Database for PostgreSQL instance to only allow private access via the Virtual Network service endpoint. For more information, see [Use Virtual Network service endpoints and rules for Azure Database for PostgreSQL](#) and [Create and Manage VNet service endpoints](#).

You may also want to review the [overview topic](#) on Azure virtual networks.

- Alternatively Azure Database for PostgreSQL can be configured to allow connections from a range of public IP addresses. This method exposes the Azure Database endpoint to public access on the internet.
- When setting up the Azure Database instance, we recommend using **postgres** as the Administrator user name. If you choose to use a different user name, make sure that the user name does not start with **pg**, or **azure**. The user name also cannot be **rails**, **tblwadmin**, **tableau**, **readonly**, or **tbladminviews**.
- The version of PostgreSQL should match the version used by Tableau Server when installed locally. Tableau Server 2020.4 uses PostgreSQL version 12.

Tableau Server on Linux Administrator Guide

Create a Database PostgreSQL instance on Azure

Step 1: Create a delegated subnet for the Azure Database for PostgreSQL instance

This step is a prerequisite for setting up private access for your networking option when you create the instance. Setting up private access to the database is a must for secure communications. This let the virtual machines created anywhere in that Virtual Network to connect to the database instance, but none outside of the Virtual Network is able to do so.

On the same virtual network where you are currently hosting your Tableau Server, create a new delegated subnet for the Azure Database instance. For more information on setting up private access see [Networking Options for Azure Database for PostgreSQL - Flexible Server](#) on the Azure website.

Step 2: Create an Azure Database for PostgreSQL instance

To create a new Azure Database for PostgreSQL, follow the instructions provided on the [Azure documentation site](#).

Following are configuration options and the recommended values for the new PostgreSQL DB instance:

- **Server Details**

- Specify None as the Data source to create a new server.
- For Admin user name, we recommend using **postgres** as the Administrator user name. If you choose to use a different user name, make sure that the user name does not start with **pg**, or **azure**. The user name also cannot be **rails**, **tblwgadmin**, **tableau**, **readonly**, or **tbladminviews**.
- Pick a password that meets Azure's requirements.
- Use the compatible version of PostgreSQL for the version of Tableau Server you are using. For a full list of PostgreSQL versions compatible with Tableau Server versions, see Product Compatibility.
- Allocate at least 512GB of storage.

- **Compute and Storage**

- At a minimum, use Flexible Server with General Purpose computer tier, and Standard_D8s_v3 (8 vCores, 32 GB RAM) compute size.
- **Network Options**
 - Select Private Access (Virtual Network). This ensures private and secure communications for the database.
- **High availability**
 - Enable the high availability option per your requirements.
- **Backup**
 - Set the retention period per your requirements. This is for Azure automated backups, not Tableau Server's backups. You can specify the settings that meets the requirements.

Step 3: Configure a server-level firewall rule

Once the database is created, configure a server-level firewall rule to allow access to the Tableau Server nodes.

Make sure that the Database instance can be reached by all the Tableau Server nodes using the dedicated subnet described in Step 1.

Step 4: Configure the Azure Database for PostgreSQL Instance.

The Database instance should have certain server settings to work with Tableau Server. You may need to modify server parameters after the database has been created.

For information on how to configure server parameters, see this [Azure documentation](#).

The following table shows parameters that you should modify with Tableau recommended values:

Name	Required	Data type	Recommended Tableau Configuration Values	Description
autovacuum	Required	boolean	1	Starts the autovacuum subprocess.
client_min_messages	Required	string	error	Sets the message levels that are sent to the client.
escape_string_warning	Required	boolean	0	Warns about backslash (\) escapes in ordinary string literals.
lc_monetary	Required	string	en_US.UTF-8	Sets the locale for formatting monetary amounts.
lc_numeric	Required	string	en_US.UTF-8	Sets the locale for formatting numbers.
log_autovacuum_min_duration	Not required, but recommended.	integer	100	Sets the minimum execution time above which autovacuum actions will be logged.
log_connections	Not required, but recommended.	boolean	1	Logs each successful connection.
log_lock_waits	Not required, but recommended.	boolean	1	Logs long lock waits.
log_min_duration_statement	Not required, but recommended.	integer	100	Sets the minimum execution time above which statements will be logged.
log_min_messages	Not required,	string	error	Sets the message levels

Name	Required	Data type	Recommended Tableau Configuration Values	Description
sages	but recommended.			that are logged.
log_temp_files	Not required, but recommended.	integer	128	Logs the use of temporary files larger than this number of kilobytes.
standard_conforming_strings	Required	boolean	0	Causes ... strings to treat backslashes literally.
work_mem	Required	integer	16384	Sets the maximum memory to be used for query workspaces.

Step 5: Get the PostgreSQL DB Instance Endpoint

Once the instance is ready, get the endpoint information that you will use to configure Tableau Server to use this instance for the Tableau Sever Repository.

Step 6: Download the SSL certificate file

In versions 2021.2 and 2021.2.1, SSL connections are required between Tableau Server and the External Repository. Starting in version 2021.2.2, SSL is optional but recommended to connect to the DB instance.

If you want to use secure connections between Tableau Server and the External Repository, you will need the certificate file when you configure Tableau Server to use this external repository. For more information, see [Configure TLS connectivity for Azure Database for PostgreSQL](#).

Tableau Server on Linux Administrator Guide

If you do not need to use secure connections between Tableau Server and External Repository, you need to configure the Azure Database instance to allow unencrypted connections.

Configuring High Availability for your PostgreSQL DB

Tableau Server does not manage or setup high availability for the external repository. Azure offers high availability features that can be used to provide high availability. For more information, see [Azure Database High Availability](#).

Disaster Recovery for your PostgreSQL DB

In the event of a disaster, you may need to setup a new Azure Database for PostgreSQL instance. There are other scenarios where you may need to recover from an issue with the database instance. For example, when you upgrade your Tableau Server, you might also need to upgrade the PostgreSQL version on your database instance. In the event that your PostgreSQL upgrade is not successful you might have to use a new database instance. In such scenarios, to configure your Tableau Server to use the new Azure Database instance, use the following steps:

1. **Restore the backup to a new Azure Database instance.** In Azure Database for PostgreSQL, performing a restore creates a new server from the original server's backups. For more on Azure Database for PostgreSQL backup and restore, see [Azure Database for PostgreSQL Backup and Restore](#).
2. **Create a new JSON settings file** containing connection information for the new Azure Database for PostgreSQL instance. For more information on creating a JSON settings file, see **Step 1** in [Install Tableau Server with External PostgreSQL Repository](#).
3. **Use the `tsm topology external-services repository replace-host` command** to point your Tableau Server to the new Azure Database for PostgreSQL instance.

For more information on the `tsm topology external-services repository replace-host` command, see [tsm topology](#).

Who can do this

Only Tableau Server Administrators can configure Tableau Server to use the external repository. You will also need an Azure account to create the Azure Database.

Create a PostgreSQL Database as a Stand-alone Installation

Beginning in version 2021.2, you can host the Tableau Server repository separately as a stand alone installation. This is different from using a managed cloud service such as AWS RDS, or Azure Database. This configuration can be done on-premises, on AWS EC2, or on an Azure VM. Such an installation of the Tableau Server Repository will be referred to as stand-alone External Repository.

To see a full list of hosts that you can use for the your external repository, see Tableau Server External Repository.

This topic provides guidance on the requirements and configurations that are necessary for Tableau Server to connect to a PostgreSQL installation and use it as the Tableau Server External Repository. This topic does not provide you with the detailed instructions on how to install PostgreSQL. We recommend that you follow the [documentation on the PostgreSQL site](#) for this information.

Requirements and Recommendations

- **Hardware Recommendations:** CPU and storage depend on your requirements. For smaller installations, you should have at least 50 GB of disk storage, and a quad processor (or 4 virtual cores) system with 32 gig of RAM. Review the guidance in [this topic](#) to calculate the disk space requirements for backup and restore. The general recommendation is to start with more hardware resources and scale back after monitoring.
- **Networking:** The PostgreSQL database instance must be reachable from all nodes in the Tableau Server cluster. One of the ways you can do this is by making the PostgreSQL database instance a member of a security group that has the necessary permissions to be accessed by all the nodes in the Tableau Server cluster.

Tableau Server on Linux Administrator Guide

- **Version Compatibility:** The version of PostgreSQL should match the version of the Tableau Server Repository when installed locally. For more information about compatibility, see [Product Compatibility](#).
- **Security:** The PostgreSQL instance must be set up to require encrypted (SSL/TLS) connections if you are using Tableau Server versions 2021.2 and 2021.2..1.

Starting in version 2021.2.2, SSL is not a requirement, but recommended when connecting from Tableau Server to the External Repository. If you do not want to use secure connections between Tableau Server and External Repository, you should configure the stand-alone PostgreSQL Database to allow unencrypted connections.

Create a stand-alone PostgreSQL Database Instance

Step 1: Install and initialize PostgreSQL

1. Use the [PostgreSQL documentation](#) to install PostgreSQL database instance to serve as the External Repository for Tableau Server. You may want to set up a PostgreSQL database cluster to meet any high availability requirements you may have.
2. Install the contrib package that includes the uuid-oss extension. This module is used to generate the UUIDS that Tableau Server uses for keys in the database.
3. Initialize the PostgreSQL instance.

Step 1: Configure your PostgreSQL Instance

You will be using two configuration files to configure your PostgreSQL instance:

- `pg_hba` : This is the general server configuration file.
- `postgresql.conf`: This is the configuration file for host-based authentication.

By default they are located at:

`/var/lib/pgsql/12/data` (This may be different depending on the distribution)

Super User Settings

Choose a user name that meets your requirements. We recommend using **postgres** as the Administrator user name. If you choose to use a different user name, make sure that the user

name does not start with **pg**. The user name also cannot be **rails**, **tblwgadmin**, **tableau**, **readonly**, or **tbladminviews**.

This is a requirement for the external repository to work properly with Tableau Server.

Network and Security

Make sure that the database instance can be reached by all the Tableau Server nodes. This most often involves creating a security group that allows access from the nodes.

Database Options

The port can be anything, but we recommend leaving it as the default 5432.

Update Parameters

Update the PostgreSQL configuration in the `postgresql.conf` file with Tableau recommended values shown below:

Name	Data type	Parameter Type	Recommended Tableau Configuration Values	Description
<code>autovacuum</code>	boolean	Dynamic	1	Starts the autovacuum sub-process.
<code>client_min_messages</code>	string	Dynamic	error	Sets the message levels that are sent to the client.
<code>escape_string_warning</code>	boolean	Dynamic	0	Warns about backslash (\) escapes in ordinary string literals.
<code>lc_monetary</code>	string	Dynamic	en_US.UTF-8	Sets the locale for formatting monetary amounts.

Name	Data type	Parameter Type	Recommended Tableau Configuration Values	Description
lc_numeric	string	Dynamic	en_US.UTF-8	Sets the locale for formatting numbers.
log_auto-vacuum_min_duration	integer	Dynamic	100	Sets the minimum execution time above which auto-vacuum actions will be logged.
log_connections	boolean	Dynamic	1	Logs each successful connection.
log_lock_waits	boolean	Dynamic	1	Logs long lock waits.
log_min_duration_statement	integer	Dynamic	100	Sets the minimum execution time above which statements will be logged.
log_min_messages	string	Dynamic	error	Sets the message levels that are logged.
log_temp_files	integer	Dynamic	128	Logs the use of temporary files larger than this number of kilobytes.
standard_conforming_strings	boolean	Dynamic	0	Causes ... strings to treat backslashes literally.
work_mem	integer	Dynamic	16384	Sets the maximum memory to be used for query workspaces.

Configure remote connections

Use the following steps to make updates to the configuration files:

1. By default, the configuration in the `postgresql.conf` is configured to only listen to local connections. Enable remote connections by making the following changes in the connections and authentication section of the `postgresql.conf` file:

Add this line to allow remote connections:

```
listen_addresses = '*'
```

2. Restart the PostgreSQL instance.

Configure SSL

SSL connections between Tableau Server and the External Repository is required in versions 2021.2 and 2021.2.1. Starting in version 2021.2.2, SSL is not a requirement, but recommended when connecting from Tableau Server to the External Repository.

To configure encrypted connections between Tableau Server and the External Repository, follow the guidance and the detailed steps described below:

When configuring Tableau Server to use the stand-alone PostgreSQL database instance, you will need to provide a trusted root certificate authority (CA) which is used to verify the connection to the server. Ideally, the stand-alone PostgreSQL instance's server certificate should specify a resolvable hostname so Tableau Server can use `sslmode, verify-full`. This mode verifies that the PostgreSQL server's certificate was signed by a trusted CA and that the hostname in the PostgreSQL Server's certificate matches the hostname used to connect to the PostgreSQL instance. However, if that is not possible, `sslmode, verify-ca` will just verify that the Postgres server's certificate was signed by a trusted CA.

The following procedure provides the general steps to generating a root CA certificate on the PostgreSQL Server. For more detailed information, read the [SSL documentation](#) on the PostgreSQL website (The link points to version 12):

Tableau Server on Linux Administrator Guide

1. Generate signing root certificate authority (CA) key.
2. Create the root CA certificate.
3. Create the certificate and related key (for example - server.csr and server.key) for the PostgreSQL Server. The subject name for the certificate must match the DNS name of the PostgreSQL Server. The subject name is set with the -subj option with the format "/CN=<private DNS name>".
4. Sign the new certificate with the CA certificate that you created in step 2.
5. Copy the crt and key files to the data directory (/pgsql/<version>/data).
6. The pg_hba.conf file controls the connections to the database. Add the following line to allow remote connections. For example:

```
host all all 10.0.0.0/8 md5
```

7. To enable SSL add or update the postgresql.conf file with:

```
ssl = on
```

To restrict connections to only SSL only, use *hostssl* instead of *host*.

High Availability and Disaster Recovery

Tableau Server does not manage or setup high availability for the external repository. PostgreSQL database supports several solutions for these purposes including replication and log-shipping. For more information, see the [high availability documentation](#) on the PostgreSQL website.

In the event of a disaster, if you need to setup a new PostgreSQL instance, make sure to follow these steps to configure Tableau Server to use the new instance.

1. **Create a new JSON settings file** containing connection information for the new RDS instance. For more information on creating a JSON settings file, see **Step 1** in **Install Tableau Server with External PostgreSQL Repository**.
2. **Use the tsm topology external-services repository replace-**

`host` command to point your Tableau Server to the new PostgreSQL instance.

For more information on the `tsm topology external-services repository replace-host` command, see [tsm topology](#).

Who can do this

Only Tableau Server Administrators can configure Tableau Server to use the external repository. If you are using AWS EC2 or Azure VM to setup a stand-alone External Repository, you need to have accounts to access these platforms.

Install Tableau Server with External PostgreSQL Repository

This topic describes how to install and configure Tableau Server to use an external service for Tableau Server Repository.

Before you install

- You must have the Advanced Management license activated on your Tableau Server to make sure your environment has the appropriate license to use external repository. For more information about Advanced Management, see [About Tableau Advanced Management](#).
- Your Tableau Server environment must be one of the following:
 - Public Cloud Services:
 - AWS cloud services.
 - Azure cloud services. For more information on Tableau Server installation on Azure, see [Install Tableau Server on Microsoft Azure](#).
 - On-premises: This is Tableau Server running on the hardware located in your organization or company and not on a public cloud.
- You must have an instance of PostgreSQL database install and ready. You will also need the endpoint of your PostgreSQL DB instance.
 - If you are installing Tableau Server on **AWS**, see:
 - RDS: Create a PostgreSQL DB Instance on AWS Relational Database Service (RDS).

Tableau Server on Linux Administrator Guide

- EC2: Create a PostgreSQL Database as a Stand-alone Installation
- If you are installing Tableau Server on **Azure**, see:
 - Azure DB: Create a Azure Database PostgreSQL Instance on Azure.
 - Azure VM:
- If you are installing this on-premises, see Create a PostgreSQL Database as a Stand-alone Installation
- Download the SSL certificate:

Encrypted connections are required between Tableau Server and the External Repository if you are using Tableau Server versions 2021.2 and 2021.2.1. Starting in version 2021.2.2 and later, SSL connections are not required but recommended.

If you want to setup SSL connections for communications between Tableau Server and the External Repository, do the following:

- Amazon RDS: See [Using SSL to Encrypt the Connection to a DB Instance](#).
- Azure Database for PostgreSQL: See [Configure TLS connectivity for Azure Database for PostgreSQL](#).
- Stand alone PostgreSQL database: The CA certificate that you used to configure SSL for the database should be copied to the Tableau Server initial node. For more information on configuring SSL for your PostgreSQL database, see [Configure SSL](#).

Install and Configure Tableau Server

Step 1: Create a configuration file

Create a json file with the following configuration settings:

```
{
  "flavor": "<flavor name>",
  "masterUsername": "<admin user name>",
  "masterPassword": "<password>",
  "host": "<instance host name>",
  "port": 5432
}
```

- **flavor:** This is the type of external service you are going to use for Tableau Server repository.
 - Amazon RDS: use "rds".
 - Azure Database: use "azure"
 - Stand alone PostgreSQL database: use "generic"

- **masterUsername:**
 - Amazon RDS: Use "rails" for the user name. This is the user that you specified when creating the RDS instance.

You must use "rails" as the masterUsername. This is required for the external repository to work with Tableau Server properly.
 - Azure Database or Standalone PostgreSQL Database: Choose a user name that meets your requirements. We recommend using **postgres** as the Administrator user name. If you choose to use a different user name, make sure that the user name does not start with **pg**, or **azure**. The user name also cannot be **rails**, **tblwgadmin**, **tableau**, **readonly**, or **tbladminviews**.

- **masterPassword:** This is the same password you specified when creating the PostgreSQL database instance.

- **host:** This is the endpoint of your PostgreSQL database instance.

- **port:** The database port you specified when creating the PostgreSQL DB instance.

Step 2: Install Tableau Server and Configure the External Repository

Using TSM CLI:

1. Install and Initialize TSM: Follow the instructions provided in this topic and complete steps 1-5 which runs the setup program and installs TSM.
2. Activate and Register Tableau Server: Provide the Tableau Server Key and the IT Management Add-on key in the activate step. You will need to run the following command twice, first with the Tableau Server product key and then with the Advanced

Tableau Server on Linux Administrator Guide

Management product key:

```
tsm licenses activate -k <product key>
```

3. Configure Initial Node Settings: Follow the instructions provided in the topic to configure the initial node settings.

Important! Do not run the **Initialize and Start Tableau Server** step when you configure the initial node. After completing the other steps in the Configure Initial Node Settings topic, return to this page and follow the rest of the instructions.

4. Configure Tableau Server to use the external repository by using the following commands:

- Specify the external repository settings using the json file that you created in the previous step:

```
tsm topology external-services repository enable -f <filename>.json -c <ssl certificate file>
```

The json file is the file that you created in the first step with the configuration settings.

Note: The SSL certificate is needed only if you are using encrypted connections between Tableau Server and the External Repository. If this is not a must for you, you must specify the `--no-ssl` option. In this case, the tsm command would look like this:

```
tsm topology external-services repository enable -f  
<filename>.json --no-ssl
```

- Apply the changes:

```
tsm pending-changes apply
```

Step 3: Complete tsm Initialize

To initialize and start Tableau Server:

```
tsm initialize --start-server --request-timeout 1800
```

Step 4: Complete the install

Add an Administrator Account and complete the installation.

Who can do this

Tableau Server Administrators can install and configure Tableau Server.

Re-Configure Tableau Server Repository

Your Tableau Server may be configured to use either a local or an external repository. This topic describes the steps needed to reconfigure your existing Tableau Server with one of the following options:

- Move a local Tableau Server Repository to an external repository and configure your Tableau Server to use an external repository.
- Move the external Tableau Server Repository to your local Tableau Server installation, and configure your Tableau Server to use the local repository. This means that the Tableau Server repository will be installed on the same machine or machines as your Tableau Server.

To learn more about these options and external repositories, see [Tableau Server External Repository](#).

Move a local Tableau Server Repository to external

Use the following steps to move Tableau Server Repository from local to external:

1. Activate the Advanced Management product key on your Tableau Server if it is not already activated. Advanced Management license is required to configure your Tableau Server with an external repository.
2. Configure Amazon PostgreSQL DB instance to use as the external repository.
 1. Amazon: Create a PostgreSQL DB Instance on AWS Relational Database Service (RDS).
 2. Azure Database: Create a Azure Database PostgreSQL Instance on Azure.

3. Stand-alone PostgreSQL Instance: Create a PostgreSQL Database as a Stand-alone Installation .

3. Create a json file with the following configuration settings:

```
{  
  "flavor": "<flavor name",  
  "masterUsername": "<admin user name>",  
  "masterPassword": "<password>",  
  "host": "<instance host name>",  
  "port": 5432  
}
```

- **flavor:** This is the type of external service you are going to use for Tableau Server repository.

- Amazon RDS: use "rds".
- Azure Database: use "azure"
- Stand-alone PostgreSQL instance: use "generic"

- **masterUsername:**

- **Amazon RDS:** Use "rails" for the user name. This is the user that you specified when creating the RDS instance.

You must use "rails" as the masterUsername. This is required for the external repository to work with Tableau Server properly.

- **Azure Database and Stand-alone PostgreSQL instance:** Choose a user name that meets your requirements. We recommend using **postgres** as the Administrator user name. If you choose to use a different user name, make sure that the user name does not start with **pg**, or **azure**. The user name also cannot be **rails**, **tblwgadmin**, **tableau**, **readonly**, or **tbladminviews**.

- **masterPassword:** This is the same password you specified when creating the

PostgreSQL database instance.

- **host:** This is the endpoint of your PostgreSQL database instance.
- **port:** The database port you specified when creating the PostgreSQL DB instance.

4. Run the following TSM CLI command to configure Tableau Server to use external repository:

```
tsm topology external-services repository enable -f
file.json -c <ssl certificate file>.pem
```

Note: The SSL certificate is needed only if you are using encrypted connections between Tableau Server and the External Repository. If this is not a must for you, you must specify the `--no-ssl` option. In this case, the `tsm` command would look like this:

```
tsm topology external-services repository enable -f
<filename>.json --no-ssl
```

The json file is the file that you created in the first step with the configuration settings. The SSL certificate file can be downloaded as described in [this topic](#).

Running the above command will migrate the local repository to your new external PostgreSQL DB instance.

Move the external Tableau Server Repository to the local installation of Tableau Server

You can move the external Tableau Server Repository from to your local installation of Tableau Server.

1. Run the following TSM CLI command to move the repository to a specific node:

```
tsm topology external-services repository disable -n nodeN
```

2. If you are setting up HA for your repository, install the repository on a second node.
For more information, see [Example: Install and Configure a Three-Node HA Cluster](#).

Note: To install the repository on a second node, you must run the command described in the previous step first. The first step migrates your external repository to the local repository. You can then install the repository on a second node on your Tableau Server.

Who can do this

Tableau Server Administrators can reconfigure external repository. You will also need to have access to create PostgreSQL database instance on Amazon or Azure.

Upgrade Tableau Server with External Repository for a New Major Version of PostgreSQL

When there is a change in the PostgreSQL major version requirement for Tableau Server, there are some specific instructions you must follow to ensure that your Tableau Server upgrade is successful. For example, Tableau Server 2020.4 requires that the repository use PostgreSQL version 12. This is a major version change from PostgreSQL version 9.x used in Tableau Server versions earlier than 2020.4. So if you are upgrading from an earlier version of Tableau Server to version 2020.4 or later, you will need to take the steps described in the following sections to complete the upgrade.

This topic also includes the product compatibility between PostgreSQL and Tableau Server.

Before you upgrade

You cannot do an in-place upgrade to update the PostgreSQL version on your existing PostgreSQL DB instance on Amazon RDS or Azure Database for PostgreSQL DB. Instead, you must create a new instance and point the Tableau Server to the new instance during upgrade. Use the following information to create a new instance and prepare for the upgrade:

1. Create a new instance of PostgreSQL Database instance:
 1. Create a PostgreSQL DB Instance on AWS Relational Database Service (RDS)
 2. Create a Azure Database PostgreSQL Instance on Azure
 3. Create a PostgreSQL Database as a Stand-alone Installation

2. Download the SSL certificate to the Tableau Server initial node:

Encrypted connections are required between Tableau Server and the External Repository if you are using Tableau Server versions 2021.2 and 2021.2.1. Starting in version 2021.2.2 and later, SSL connections are not required but recommended.

If you want to setup SSL connections for communications between Tableau Server and the External Repository, do the following:

1. **Amazon RDS:** See [Using SSL to Encrypt the Connection to a DB Instance](#).
 2. **Azure Database:** See [Configure TLS connectivity for Azure Database for PostgreSQL](#).
 3. **Stand-alone PostgreSQL Instance:** See [Configure SSL](#).
3. Create a configuration file for the new instance you created in step 1.

1. Create a json file with the following configuration settings:

```
{
  "flavor": "rds/azure/generic",
  "masterUsername": "<admin user name>",
  "masterPassword": "<password>",
  "host": "<instance host name>",
  "port": "<port number>"
}
```

- **flavor:** This is the type of external service you are going to use for Tableau Server repository.
 - Amazon RDS: use "rds".
 - Azure DB : use "azure"

Tableau Server on Linux Administrator Guide

- Stand-alone PostgreSQL instance: "generic"
- **masterUsername:**
 - **Amazon RDS:** Use "rails" for the user name. This is the user that you specified when creating the RDS instance.

You must use "rails" as the masterUsername. This is required for the external repository to work with Tableau Server properly.
 - **Azure Database or Stand-alone PostgreSQL Instance:**
Choose a user name that meets your requirements. We recommend using **postgres** as the Administrator user name. If you choose to use a different user name, make sure that the user name does not start with **pg**, or **azure**. The user name also cannot be **rails**, **tblwgadmin**, **tableau**, **readonly**, or **tbladminviews**.
- **masterPassword:** This is the same password you specified when creating the PostgreSQL database instance.
- **host:** This is the endpoint of your PostgreSQL database instance.
- **port:** The database port you specified when creating the PostgreSQL DB instance. Default port for PostgreSQL is 5432.

Tableau Server Upgrade

Note: If you are using Tableau Server External Repository, you must upgrade Tableau server using the command line option.

The following are the high level steps of how to specify the external repository parameters when running the upgrade script.

For a full walk-through of Tableau Server upgrade process, see [Upgrading from earlier versions](#)

1. On the Tableau Server, open a command prompt as administrator.

Note: You must open a new command window because the Setup program updates the path for the new installation.

2. Navigate to the scripts folder for your new installation.

By default:

```
/opt/tableau/tableau_server/packages/scripts.<version_
code>/
```

3. Run the upgrade script and specify the configuration file and the SSL certificate:

```
upgrade-tsm --external-repository-config-file=<json config
file> --external-repository-cert-file=<SSL certificate
file>
```

Product Compatibility

For the latest information about version compatibility between Tableau Server and PostgreSQL, see [Product Compatibility](#)

Who can do this

Tableau Server Administrators can upgrade and configure Tableau Server.

Upgrading your RDS Instance

If you find that the current RDS instance you are using to host Tableau repository is a performance bottleneck, you can upgrade your RDS instance to a larger size. This topic describes the steps that you can use to upgrade your RDS instance.

Tableau Server on Linux Administrator Guide

1. Back up Tableau Server data.
2. Stop Tableau Server:

```
tsm stop
```

3. After confirming that the server has shutdown, sign in to the AWS Management Console and open the Amazon RDS console at

```
https://console.aws.amazon.com/rds/.
```

4. In the navigation pane, choose **Databases**, and then choose the DB instance that you want to modify.
5. Choose **Modify**. The Modify DB Instance page appears.
6. Modify your RDS instance by setting the DB instance class to the one you want.
7. Choose **Apply immediately** to make sure that the changes are applied right away.
For more detailed information, see [Modifying a DB Instance Running the PostgreSQL Database Engine](#) on AWS documentation site.
8. Monitor the status of the RDS instance in the AWS console. It may take a while, but when the status shows as Available, you can start Tableau Server and resume normal operations:

```
tsm start
```

Who can do this

Tableau Server Administrator who has have access to the Amazon RDS account can perform all the steps required to upgrade the Amazon RDS instance.

Workload Management through Node Roles

Using node roles, you can configure where certain types of workloads are processed on your Tableau Server installation. The node roles features allows you to dedicate and scale resources to specific workloads. You can configure node roles for Backgrounder and File Store.

The Backgrounder node role specifies the type of background tasks that should run on a node, whereas the File Store node role specifies the type extract workload that should run on a node. Both node roles are specified at a node level. Although these node roles can work independently to optimize selected workload, the two node roles in combination can be used to specialize server nodes to preferentially execute selected workloads to optimize performance extract heavy workloads. This combination is discussed in more detail later in the File Store Node Roles section.

Backgrounder Node Roles

The Backgrounder process runs Tableau Server tasks, including extract refreshes, subscriptions, flow tasks, 'Run Now' tasks, and tasks initiated from *tabcmd*. Running all these tasks can use a lot of machine resources. If you have more than one Backgrounder node in your cluster, you can manage your Backgrounder workload by specifying the type of tasks a Backgrounder can run on a node using the Backgrounder node role feature.

This configuration option is currently available only through TSM CLI commands and is only useful on multi-node clusters. If you have only one node, the Backgrounder is set to run all tasks by default and that cannot be changed.

Using Backgrounder Node Roles

The Backgrounder node role feature is intended to give you more control and governance over where certain type of Backgrounder workloads are processed in your Tableau Server installation and allows you to dedicate and scale resources to specific workloads.

For example, if your deployment is heavy on extract and users are running a lot of extract refreshes or encryption jobs, it could be beneficial to dedicate a node to extract refreshes. Similarly, in the case of subscriptions, if your Tableau Server installation processes a lot of subscriptions and you want to ensure that other jobs do not take resources from subscriptions, then you can dedicate a node to subscriptions. In these cases, you would also want to dedicate other backgrounder nodes to workloads other than extract refreshes or subscriptions.

Tableau Server on Linux Administrator Guide

To support high availability, Tableau recommends having multiple nodes that are dedicated towards a specific workload. For example, if you dedicate a node to extract refreshes, you should also configure a second node to process extract refresh workload. This way if a node dedicated to extract refreshes becomes unavailable, extract refreshes can still be processed by the other node.

Configuration Options

Configuration	Jobs
all-jobs (default)	All Tableau Server jobs
flows	Flow run jobs.
no-flows	All jobs except flows.
extract-refreshes	Jobs that are created for: Incremental refreshes, full refreshes, encryption and decryption of all extracts including extracts that flow outputs create.
subscriptions	Subscription jobs
extract-refreshes-and-subscriptions	Extract-refreshes, encryption and decryption of all extracts including extracts that flow outputs generate, and subscription jobs.
no-extract-refreshes	All jobs except extract-refreshes, extract encryption and decryption of all extracts including extracts created from flow outputs.
no-subscriptions	All jobs except subscriptions.
no-extract-refreshes-and-subscriptions	All jobs except extract-refreshes, encryption and decryption of all extracts including extracts created from flow outputs, and subscriptions.

For more information on how to use the tsm commands to set the node role, see tsm topology.

Note: Making configurations to node roles require a restart of the server and will require some downtime. For more information, see [tsm pending-changes](#).

License Requirements

Configuring a node to do only a specific type of tasks, like, flows, extract refreshes, and subscriptions, you must have one of the following licenses activated on your Tableau Server:

- To configure a node to run flows, you must have a valid Data Management Add-on license activated on your server, and have Tableau Prep Conductor running on that node. To learn more about Tableau Prep Conductor, see [Tableau Prep Conductor](#).
- To configure a node to run all jobs except flows, you must have Tableau Prep Conductor running on the node. Tableau Prep Conductor requires a valid Tableau Data Management Add-on license activated on your server. For more information, see [Licensing Tableau Prep Conductor](#).
- To configure a node to run extract refreshes, subscriptions, and any combination related to extract refreshes and subscriptions you must have a valid Advanced Management license activated on your Tableau Server. If the license expires or is deactivated, you will see an error any time you make a change to the Server configuration. For more information on Advanced Management, see [About Tableau Advanced Management](#).

Important!

While flows, extract refreshes, and subscriptions can be expensive and resource heavy, they are not the only jobs that may require dedicated resources. In the **all jobs** group, there are a variety of System jobs that the Backgrounder executes, such as thumbnail generation for workbooks. Make sure that the nodes that run jobs other than extract refreshes, subscriptions, or flows have enough machine resources.

For more information on configuring node roles using TSM commands, see [tsm topology set-node-role](#).

Considerations

There are some rules you must consider when configuring Backgrounder node roles, which are listed below:

- Only one node role configuration can be set for a node at a time. You cannot configure multiple node roles on a node.
- To configure a node role, there must be at least one Backgrounder process on that node.
- If you have only one Backgrounder node, you must configure this node to run all jobs. This is the default configuration and does not require additional licensing.
- If you have more than one Backgrounder node, combined, they must be configured to handle all jobs. This can be achieved in the following ways:
 - Configure one of the nodes to run all jobs using the all jobs option. This is the easiest and most straightforward way.
 - Using one of the exception configurations on one of the nodes:
 - no-flows
 - no-subscriptions
 - no-extract-refreshes
 - no-extract-refreshes-and-subscriptions

For example, in a cluster where there are three backgrounders, you could have one node configured to run flows, one to run subscriptions and extract refreshes, and one to run all jobs except flows, subscription and extract refreshes.

Note: The ability to specify node roles to run flows, or run all jobs except flows, or run all jobs was introduced in 2019.1.

File Store Node Roles

The Tableau Server File Store controls the storage of extracts. There are three broad categories of workloads that are extract dependent.

Extract Workload	Execution Service
Refresh	Backgrounder
Query	Data Engine
Backup/Restore	Backup/Restore

File Store node role management in combination with Backgrounder node role management gives server admins the ability to specialize server nodes to preferentially execute selected workloads to optimize performance of all categories of extract heavy workloads.

It is possible to specialize a node to execute extract query workloads through a topology that has only stand-alone Data Engine nodes. For more information, see [Optimize for Extract Query Heavy Environments](#). However, this is at the expense of extract refresh workloads, which are executed by Backgrounder nodes. With the topology-based isolation approach, extract refresh heavy Backgrounder workloads can get slower as none of the Backgrounder nodes have a File Store and thus all extract refresh traffic goes over the network.

With the File Store Node Role configuration option, it is possible to designate certain server nodes that process extract queries to be preferentially selected from the list of server nodes that can do so. This helps speed up workloads such as backup and extract refreshes by allowing server admins to enable File Store on Backgrounder server nodes, which prevents extract queries from running on these nodes. This feature is useful if you have an extract-heavy query workload and an extract-heavy refresh workload and want to achieve optimal extract query and refresh performance.

Tableau Server on Linux Administrator Guide

Guidelines to Optimize for Extract Refresh and Backup/Restore Workloads.

Start from a topology with specialized Data Engine nodes (see Optimize for Extract Query Heavy Environments).

Process	Initial Node	Additional Node 1	Additional Node 2	Additional Node 3
Cluster Controller	✓	✓	✓	✓
Gateway	✓	✓		
Application Server	✓	✓		
VizQL Server	✓ ✓	✓ ✓		
Cache Server	✓ ✓	✓ ✓		
Search & Browse	✓	✓		
Backgrounder	✓ ✓ ✓ ✓	✓ ✓ ✓ ✓		
Data Server	✓ ✓	✓ ✓		
Data Engine	✓	✓	✓	✓
File Store			✓	✓
Repository	✓	✗		

Topology 1 - Dedicated Data Engine Nodes

1. Add File Store to Node 0.

```
tsm topology set-process -n node0 -pr filestore -c 1
```

2. Designate Node 2 and Node 3 to preferentially execute extract-query workloads

```
tsm topology set-node-role -n node2, node3 -r extract-queries
```

3. Designate Node 0 to preferentially execute extract-refresh workloads.

```
tsm topology set-node-role -n node0 -r extract-refreshes
```

4. Designate Node 1 to preferentially execute non-extract-refresh workloads.

```
tsm topology set-node-role -n node1 -r no-extract-refreshes
```

5. Apply pending changes.

```
tsm pending-changes apply
```

Process	Initial Node	Additional Node 1	Additional Node 2	Additional Node 3
Cluster Controller	✓	✓	✓	✓
Gateway	✓	✓		
Application Server	✓	✓		
VizQL Server	✓ ✓	✓ ✓		
Cache Server	✓ ✓	✓ ✓		
Search & Browse	✓	✓		
Backgrounder	✓ ✓ ✓ ✓	✓ ✓ ✓ ✓		
Data Server	✓ ✓	✓ ✓		
Data Engine	✓	✓	✓	✓
File Store	✓		✓	✓
Repository	✓	✗		

Topology 2 - Extra File Store Node

Note: In your Tableau Server deployment, adding File Store roles to existing nodes will temporarily increase network I/O between all File Store nodes while the new File Store is being synchronized. The duration of this operation is dependent on the volume of data on the File Store and the network bandwidth capacity. The status of synchronization can be monitored using the TSM Web GUI. If you are adding more than one File Store to your deployment, it is recommended to add them consecutively and wait for the initial synchronization to complete in between each File Store addition.

Fine Tune Extract Query Workload Management

When extract queries for email subscriptions and metric alerts are running at the same time that users are interactively viewing extract-based visualizations, users may experience slower than normal viz load times. Use the following node roles to fine tune how these workloads are prioritized.

Node role to use	Type of extract query workload	Example
<code>extract-queries</code>	scheduled	email subscriptions and metric alerts

<code>extract-queries-interactive</code>	<code>interactive</code>	users viewing an extract-based visualization
--	--------------------------	--

If your server deployment is seeing growth in email subscriptions and metric alerts, you can add nodes and assign the `extract-queries` node role, which makes them more available to handle subscriptions and alerts.

If your server deployment is seeing growth in users viewing extract-based visualizations, you can add nodes and assign the `extract-queries-interactive` node role, which makes them prioritize interactive extract queries to reduce extract-based viz load times. The `extract-queries-interactive` node role is a preference and not strict isolation. This means that queries will be routed to nodes that have the `extract-queries-interactive` node role assigned. If you have multiple nodes with the `extract-queries-interactive` role, queries will be routed based on node health.

For example, add a node and designate it to preferentially execute `extract-queries-interactive` workloads.

- `tsm topology set-node-role -n node4 -r extract-queries-interactive`

Configuration Options

Configuration	Jobs
<code>all-jobs</code> (default)	All Tableau Server jobs
<code>extract-queries</code>	Jobs that are created for extract queries. The nodes selected will run as <code>all-jobs</code> and will prioritize the processing of extract queries.
<code>extract-queries-interactive</code>	Jobs that are created for extract queries. The nodes selected will run as <code>all-jobs</code> and will prioritize the processing of interactive extract queries, such as those that run when a user is looking at their screen and waiting for an extract-based dashboard to load. This is an advanced setting and it should only be used if the cluster has a heavy subscription and

	alert job workload that causes users to experience degraded performance on viz load times that run around the same time as scheduled loads.
--	---

For more information on configuring node roles using TSM commands, see `tsm topology set-node-role`.

License Requirements

To configure a node to run extract queries you must have a valid Advanced Management license activated on your Tableau Server.

Who can do this

Tableau Server Administrators can configure node roles and activate any required product keys.

About Data Management Add-on

The Data Management Add-on is a collection of features and functionality that helps customers manage Tableau content and data assets in their Tableau Server or Tableau Online environment.

Starting in Tableau Server version 2019.1, Tableau Prep Conductor is available for on-premise Tableau Server deployments, and in version 2019.3, Tableau Prep Conductor is available for Tableau Online deployments. You can use Tableau Prep Conductor to schedule and monitor flows.

Starting in version 2019.3, Tableau Catalog is included in the Data Management Add-on, making a variety of additional features available to you in the data management space. You can use Tableau Catalog to discover data, curate data assets, communicate data quality, perform impact analysis, and trace the lineage of data used in Tableau content.

Both Tableau Catalog and Tableau Prep Conductor are licensed through the Data Management Add-on. For more information about licensing, see License the Data Management Add-on.

To purchase the Data Management Add-on, contact your account manager.

Data Management Add-on Features

The following table lists the features for the Data Management Add-on, which includes:

- Tableau Catalog
- Tableau Prep Conductor

Tableau Catalog

These features require the Data Management Add-on with **Tableau Catalog enabled**.

Feature	Description
Permissions on metadata	Tableau Catalog enables you to control who can see and manage external assets and what metadata is shown through lineage by setting permissions.
Expanded connect experience - data discovery	Whether you author in the web or in Tableau Desktop , you can now search for and connect to the specific databases and tables used by published data sources and workbooks on your Tableau Server or Tableau Online site.
Expanded search	Tableau Catalog expands search results to include fields, columns, databases, and tables.
Tag external assets	You can categorize items on Tableau Server and Tableau Online with tags, helping users to filter external assets (databases, files, tables, and columns).
Certify databases and	Help users find trusted data that meets the standards you set by certifying databases and tables.

Feature	Description	
tables		
Set data quality warnings	You can set warnings to alert users to data quality issues, such as stale or deprecated data.	
Lineage and impact analysis	The Lineage tool traces the source of your data. You can use it to analyze the impact of changes to your data, identify which users might be impacted, and email owners of a workbook, data source, or flow, or contacts for a database or table, about data-related updates.	
Data Details	Enable users to better understand a published visualization by seeing information about the data used.	
Add descriptions to assets	Help users find the data they're looking for by adding descriptions to databases, tables, and columns.	
Developer resources	Tableau REST API - metadata methods	Programmatically add, update, and remove external assets; and add additional metadata to Tableau content and external assets like descriptions.
	Tableau Metadata API	Programmatically query metadata from the content published to Tableau Server or Tableau Online. Programmatically update certain metadata using the metadata methods in the Tableau Server REST API. Note: The Metadata API does not require the Data Management Add-on.
	GraphiQL	Explore and test queries against the Metadata API schema using an interactive in-browser tool called GraphiQL. Note: GraphiQL does not require the Data Management Add-on.

Tableau Prep Conductor

These features require the Data Management Add-on with Tableau Prep Conductor enabled on [Tableau Server](#) or [Tableau Online](#).

Feature	Description
Schedule flow runs	You can create scheduled flow tasks to run a flow at a specific time or on a recurring basis.
Monitor flow health and performance	Set up email notifications at the site or server level when flows fail, view and resume suspended flow tasks, and view errors and alerts. You can also use Administrative Views to monitor the activities related to flows, performance history, and the disk space used at the server or site level.
Tableau REST API - flow methods	Programmatically schedule flows.

License the Data Management Add-on

The Data Management includes Tableau Catalog and Tableau Prep Conductor. Contact your account manager (or go to the [Tableau pricing](#) page) to purchase the Data Management.

The Data Management can only be activated on a licensed Tableau Server Deployment. A Deployment includes a licensed production Tableau Server installation and licensed non-production Tableau Server installations that support the production installation. For more information on Deployment, see the [EULA Documentation](#).

Tableau Prep Conductor

After you purchase and license the Data Management Add-on, you must enable Prep Conductor on Tableau Server. For more information, see [Enable and Configure Tableau Prep Conductor on Tableau Server](#).

- When the Data Management is active and enabled, you can schedule flows in Tableau Server or Tableau Online and monitor flows.
- When the Data Management is removed or deactivated, or if the Data Management expires, then the ability to schedule flows is disabled.
- If your Tableau Server or Tableau Online license is still active and valid, you can download the flows using the Tableau Server REST API. For more information, see [Flow Methods](#).

Tableau Catalog

After you purchase and license the Data Management Add-on, you must enable Catalog on Tableau Server. For more information, see [Enable Tableau Catalog](#).

- When the Data Management is active and enabled, you can use Tableau Catalog to discover data, curate data assets, perform impact analysis, and trace the lineage of data used in Tableau content.
- When the Data Management is removed, deactivated, or expired, the information remains on the server. The Tableau Catalog-specific information is then only accessible using the Tableau Metadata API; it no longer appears in the product. For more information, see the [Metadata API](#).
- When the Data Management is removed, deactivated, or expired, the write APIs for all new Tableau Catalog information (for example, table descriptions, data quality warnings, column descriptions) are disabled. You can still read information using the Metadata API, however permissions on tables and databases can't be explicitly managed in the product.

How Data Management Add-on licensing works

A Tableau Server Deployment may be user-based or core-based, depending on which license you purchase.

Tableau Server on Linux Administrator Guide

User-Based

A User-Based license metric allows you to deploy Tableau Server on a single computer or on multiple computers in a cluster. Each user that accesses Tableau Server must be licensed. Administrators add users and license them. The first Creator or Explorer product key that you add to Tableau Server activates Tableau Server and will be used by a Server Administrator.

The Data Management Add-on product key enables the features that are included with the add-on at the deployment level and those features are licensed for all the users that are already licensed for Tableau Server.

Core-Based

A Core-Based license metric imposes no constraints on the number of user accounts in Tableau Server. Instead, the license specifies the maximum number of computer cores on which you can run Tableau Server.

Note: When you purchase and use core-based licensing, you must apply both the Data Management product key and the Resource Core product key to your Tableau Deployment. The first key allows flows to be run on Tableau Server through the Tableau Prep Conductor and the second key adds the additional cores for the Tableau Prep Conductor nodes. All product keys are available through the [Customer Portal](#).

Typically, the total number of cores in all the computers should not exceed the total number that the Tableau Server license allows. When you add the Data Management product key to your Tableau server, it includes a specific number of Tableau Prep Conductor cores. In this scenario, the total number of cores on all computers should not exceed the total number that the Tableau Server license and the Data Management license together allow.

In this topic we will refer to the cores licensed through Tableau Server license as Tableau Server cores, and the cores licensed through Data Management Add-on as Tableau Prep Conductor cores.

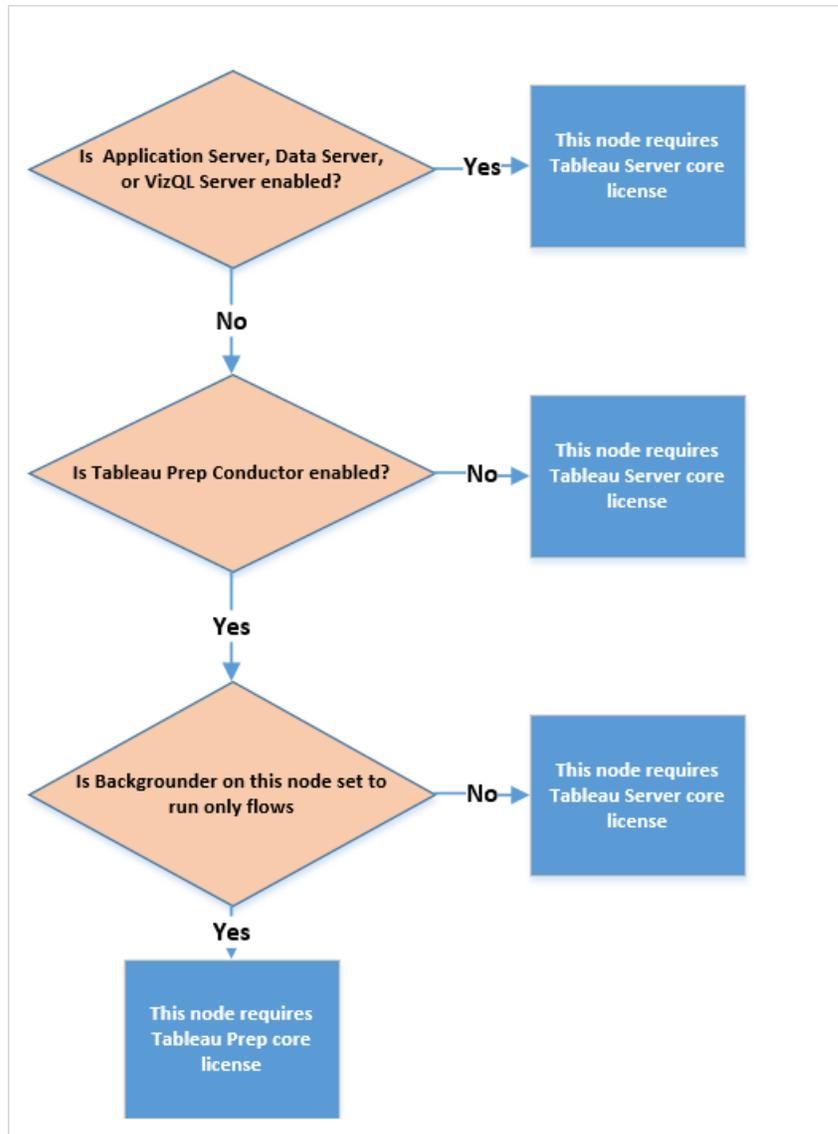
Here are some concepts that apply to how licensing is applied in a Core-Based metric:

- A node can be licensed by only one of type—Tableau Server Cores or Tableau Prep Conductor cores.
- The Tableau Prep Conductor cores are applied to any node that is dedicated to running Tableau Prep Conductor and when the Backgrounder on that node is set to run only flow background jobs. In this case, the total number of cores on this node cannot exceed the number of cores that Data Management Add-on license allows. If this node has any other licensed process besides Tableau Prep Conductor, Backgrounder and Data Engine enabled, then this node will require and use a Tableau Server core license.
- As mentioned above, the Backgrounder node role also affects which license is used by a node. For example, if the Backgrounder node role is set to run jobs of all types (this is the default), then this node will be licensed through the Tableau Server cores. For more information on node roles, see [Node Roles in Tableau Server](#).

See the following table and decision flow to understand how a node is licensed:

If a node has...	the core on the node is counted towards...	the node is licensed using...
one of the following processes enabled: <ul style="list-style-type: none"> • Application Server • Backgrounder (node role is set to run all jobs) • File Store • Data Server • VizQL Server 	Total count of Tableau Server cores.	Tableau Server cores.
only the following processes enabled: <ul style="list-style-type: none"> • Tableau Prep 	Total number of Tableau Prep Cores purchased through the Data Man-	Tableau Prep Cores included in the Data Management Add-on.

<p>Conductor</p> <ul style="list-style-type: none">• Backgrounder (node role is set to run only flows)• Data Engine	<p>agement Add-on.</p> <p>Note: If there are no Tableau Prep Conductor cores available, but Tableau Server cores are available, then Tableau Server cores are used.</p>	<p>Note: If there are no Tableau Prep Conductor cores available, but Tableau Server cores are available, then Tableau Server cores are used.</p>
--	--	---



For more information about licensed processes, see [Tableau Server Processes](#).

To learn more about Tableau Server licensing, see [Tableau Server Licensing Overview](#).

Tableau Prep Conductor

Overview

If you build flows in Tableau Prep, you can automate running your flows to refresh flow input and output data on a schedule using Tableau Server, instead of opening Tableau Prep Builder to run individual flows manually.

Tableau Prep Conductor runs flows, checks connection credentials, and sends alerts if a flow fails. Tableau Prep Conductor leverages the scheduling and tracking functionality of Tableau Server so you can automate running flows to update the flow output instead of logging into Tableau Prep Builder to manually run individual flows as your data changes.

For more information about the Tableau Prep Conductor process on Tableau Server, see [Tableau Prep Conductor](#).

Enabling Tableau Prep Conductor on Tableau Server

Before you can start publishing flows to your Tableau Server, there are server-level and site-level settings you must configure or verify to prepare your Tableau Server to allow publishing, scheduling and monitoring flows.

Review the following topics to understand Tableau Prep Conductor licensing, and learn how to enable Tableau Prep Conductor:

- [Enable and Configure Tableau Prep Conductor on Tableau Server](#): This topic provides step-by-step instructions on how to enable and configure Tableau Prep Conductor, and preparing your Tableau Server to allow publishing, scheduling, managing, and maintaining flows.
- [License the Data Management Add-on](#)

Enable and Configure Tableau Prep Conductor on Tableau Server

Tableau Prep Conductor is supported only on Tableau Server versions 2019.1 or later.

Tableau Prep Conductor is licensed through the Data Management Add-on, on a per Deployment basis, which is User-Based or Core-Based. A Deployment includes a licensed production Tableau Server installation and licensed non-production Tableau Server installations that support the production installation. For more information on Deployment, see the [EULA Documentation](#).

For more information on how Tableau Prep Conductor licensing works, see [Licensing Tableau Prep Conductor for Tableau Server](#).

Server Topology

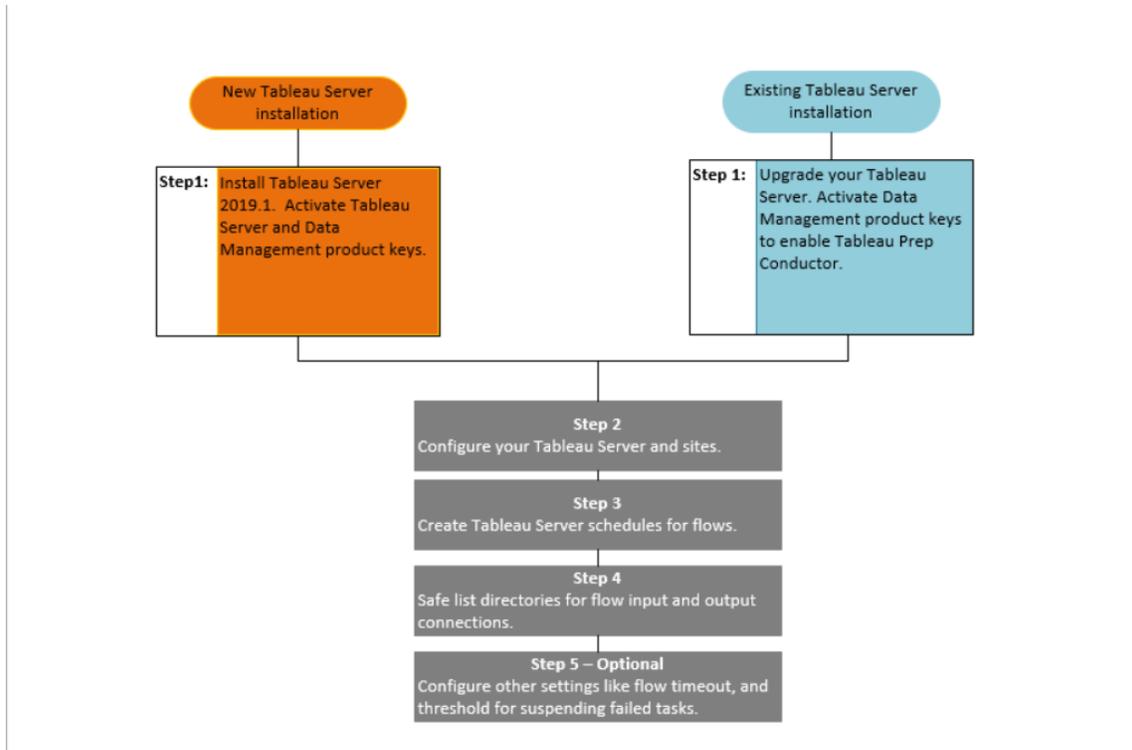
When you install Tableau Server and enable Tableau Prep Conductor, using the Data Management product key, Tableau Prep Conductor is automatically enabled by default by the setup program.

For multi-node installations, by default, one instance of Tableau Prep Conductor is enabled on any node that has backgrounder installed. In the example below, Tableau Prep Conductor is enabled on node 2 and 3 where the backgrounders are also enabled, but not on node 1,4, and 5.

Tableau Server on Linux Administrator Guide

+ a b e a u					
	STATUS	MAINTENANCE	CONFIGURATION		
Process	node1	node2	node3	node4	node5
Gateway	✓				
Application Server	✓				
Interactive Microservice Container	✓				
VizQL Server	✓ ✓ ✓ ✓				
Cache Server	✓ ✓				
Cluster Controller	✓	✓	✓	✓	✓
Search & Browse	✓				
Backgrounder		✓ ✓ ✓ ✓	✓ ✓ ✓ ✓		
Background Microservice Container		✓ ✓ ✓ ✓	✓ ✓ ✓ ✓		
Data Server	✓ ✓				
Data Engine	✓	✓	✓		
File Store	✓				
Repository	✓				
Tableau Prep Conductor		✓	✓		
Ask Data	✓				
Elastic Server	✓				

Below is a visual representation of that work-flow:



Next step:

New Tableau Server Installations: Step 1 (New Install): Install Tableau Server with Tableau Prep Conductor

Existing Tableau Server Installations: Step 1 (Existing Install): Enable Tableau Prep Conductor

Who can do this

Server administrators can install Tableau Server and enable Tableau Prep Conductor.

Server-level settings can be configured by Tableau Server administrators, and site-level settings can be configured by Tableau Server and Site administrators.

Step 1 (New Install): Install Tableau Server with Tableau Prep Conductor

This topic describes how to Tableau Prep conductor on a new installation of Tableau Server.

Tableau Server on Linux Administrator Guide

Tableau Prep Conductor is supported only on Tableau Server versions 2019.1 or later.

Tableau Prep Conductor is licensed through the Data Management Add-on, on a per Deployment basis. A Deployment includes a licensed production Tableau Server installation and licensed non-production Tableau Server installations that support the production installation. For more information on Deployment, see the [EULA Documentation](#).

Before you install

The recommended topology for a production Tableau Server installation is a dedicated node for running flows. If you are currently planning to have a single node Tableau Server installation it is recommend that you add a second node and dedicate it to run flows.

- Review the hardware recommendations for Tableau Server and Tableau Prep conductor.
 - [Minimum Hardware Requirements and Recommendations for Tableau Server installation on Windows](#).
 - [Minimum Hardware Requirements and Recommendations for Tableau Server installation on Linux](#).

Install Tableau Server and enable Tableau Prep Conductor

Use the instructions provided in the following topics to install Tableau Server.

[Windows: Install Tableau Server](#)

[Linux: Install Tableau Server topic](#)

When you get to the **Activate** step, use the Tableau Server product keys to activate Tableau Server.

All product keys are available through the [Customer Portal](#).

Configure public gateway settings

If your Tableau Server is set up with one of the following:

- Load balancer to distribute requests across gateways.
- Reverse proxy to authenticate external (internet) client requests and offloading SSL-based encryption.

You must configure the following public gateway settings:

```
tsm configuration set -k gateway.public.host -v <name> (This should be the URL that your users are using to access Tableau Server)
```

```
tsm configuration set -k gateway.public.port -v 443
```

For more information on configuring gateway settings, see [Configuring Proxies for Tableau Server](#).

Enable Tableau Prep Conductor

Use the following steps to add the Data Management product key to your Tableau Server:

Note: This process requires a restart of the Tableau Server.

Note: If you are using core-based licensing, you must apply both the Data Management product key and the Resource Core product key to your Tableau Deployment. The first key allows flows to be run on Tableau Server through the Tableau Prep Conductor and the second key adds the additional cores for the Tableau Prep Conductor nodes. All product keys are available through the [Customer Portal](#).

1. If the computer where you are running Tableau Server has been configured to connect to the internet through a forward proxy, follow the procedure in the topic, [Configure Product Key Operations with Forward Proxy](#), before continuing.
2. Open TSM in a browser:

```
https://<tsm-computer-name>:8850
```

3. Click **Licensing** on the **Configuration** tab and click **Activate License**.
4. Enter or paste your **Data Management product key** and click **Activate**.
5. On the Register page, enter your information into the fields and click **Register**.
6. Follow the prompts and restart Tableau Server after registration is complete.

Verify Tableau Prep Conductor is enabled and running

When you activate the Data Management product key, a single instance of Tableau Prep Conductor is automatically enabled on any node that has Backgrounder enabled.

Use the following steps to verify that it is enabled and running:

1. Open a browser and enter the Tableau Server URL, and append the dedicated TSM web UI port. Here are some examples of what the URL might look like:

`https://localhost:8850/` (if you're working directly on the server computer)

`https://MarketingServer:8850/` (if you know the server's name)

`https://10.0.0.2:8850/` (if you know the server's IP address)

In the sign-in page that appears, enter your administrator user name and password.

Note: Tableau Server creates and configures a self-signed certificate during the installation process. This certificate is used to encrypt traffic to the TSM Web UI. Because it's a self-signed certificate, your browser will not trust it by default. Therefore, your browser will display a warning about the trustworthiness of the certificate before allowing you to connect.

2. In the Tableau Services Manager web interface, click the **Status** tab to see the status.

- If Tableau Prep Conductor is enabled and running, you should see Tableau Prep Conductor in the list of processes as **Active** on at least on one node. If Tableau Prep Conductor is not enabled, you will see Tableau Prep Conductor in the list of processes, but with no status information for any of the nodes.

Tableau Prep Conductor not enabled:

The screenshot shows the Tableau Server Status page with the 'STATUS' tab selected. The 'Tableau Prep Conductor' row is highlighted with a red box, and it shows no status information for any of the three nodes (node1, node2, node3).

Process	node1	node2	node3
Gateway	✓		
Application Server	✓		
Interactive Microservice Container	✓		
VizQL Server	✓ ✓ ✓ ✓		
Cache Server	✓ ✓		
Cluster Controller	✓	✓	✓
Search & Browse	✓		
Backgrounder	✓ ✓		✓ ✓
Non-Interactive Microservice Container	✓		✓
Data Server	✓ ✓		
Data Engine	✓	✓	✓
File Store	✓	✓	
Repository	✓		
Tableau Prep Conductor			
Ask Data	✓		
Elastic Server	✓		
TSM Controller	✓		
License Server	✓		

Legend: ✓ Active, ⚙ Busy, ⚠ Degraded, ❌ Error, ⏹ Stopped

Tableau Prep Conductor enabled and running. In the image below Tableau Prep Conductor is enabled on node1 and node3:

Tableau Server on Linux Administrator Guide

Process	node1	node2	node3
Gateway	✓		
Application Server	✓		
Interactive Microservice Container	✓		
VizQL Server	✓ ✓ ✓ ✓		
Cache Server	✓ ✓		
Cluster Controller	✓	✓	✓
Search & Browse	✓		
Backgrounder	✓ ✓		✓ ✓
Non-Interactive Microservice Container	✓		✓
Data Server	✓ ✓		
Data Engine	✓	✓	✓
File Store	✓	✓	
Repository	✓		
Tableau Prep Conductor	✓		✓
Ask Data	✓		
Elastic Server	✓		
TSM Controller	✓		
License Server	✓		

Refresh Status

Active Busy Degraded Error Stopped

Dedicate a node for Tableau Prep Conductor

On the node you are planning to dedicate to running flows, enable Backgrounder process if it is not already enabled. It is recommended that you do not run other processes like VizQL server on this node.

Because you are dedicating this node to running flows, you must configure Backgrounder to run only flow tasks. By default, the Backgrounder process runs tasks of all types, including flows, extract refreshes, and subscriptions. For more information, see [Node Roles in Tableau Server](#).

Run the following tsm commands on that dedicated node to run only flow tasks:

1. Run the following command to allow Backgrounders on this node to run only flow tasks.

```
tsm topology set-node-role -n node1 -r flows
```

2. Set the node role on the initial node to no flows. The backgrounder on this node will run all jobs except flows:

```
tsm topology set-node-role -n node1 -r no-flows
```

3. Apply the changes and restart Tableau Server:

```
tsm pending-changes apply
```

Multi-node installations

If you have more than 2 nodes in your Tableau Server installation, you can choose to configure other nodes to run all tasks other than flows:

1. Restrict a node to not allow flows. This command removes Tableau Prep Conductor from this node and Backgrounders on this node will not run flow tasks.

```
tsm topology set-node-role -n node1 -r no-flows
```

2. Apply the changes and restart Tableau Server:

```
tsm pending-changes apply
```

Next step

Step 2: Configure Flow Settings for your Tableau Server

Who can do this

Server administrators can install Tableau Server and enable Tableau Prep Conductor.

Step 1 (Existing Install): Enable Tableau Prep Conductor

This topic describes how to enable Tableau Prep conductor on your existing installation of Tableau Server.

Tableau Prep Conductor is supported only on Tableau Server versions 2019.1 or later. If you are using Tableau Server 2018.3 or earlier, you must first upgrade your Tableau Server to 2019.1 before enabling Tableau Prep Conductor on your Tableau Server installation.

Tableau Prep Conductor is licensed through the Data Management Add-on, on a per Deployment basis, which is User-Based or Core-Based. A Deployment includes a licensed production Tableau Server installation and licensed non-production Tableau Server installations that support the production installation. For more information on Deployment, see the [EULA Documentation](#).

This topic describes how to enable Tableau Prep conductor on your existing installation of Tableau Server.

Before you upgrade

Prepare for upgrade:

- [Know before you upgrade](#)
- [Licensing Tableau Prep Conductor](#)
- [Tableau Server Hardware Requirements and Recommendations](#)

Configure public gateway settings

If your Tableau Server is set up with one of the following:

- Load balancer to distribute requests across gateways.
- Reverse proxy to authenticate external (internet) client requests and offloading SSL-based encryption.

You must configure the following public gateway settings:

```
tsm configuration set -k gateway.public.host -v <name> (This should be the URL that your users are using to access Tableau Server)
```

```
tsm configuration set -k gateway.public.port -v 443
```

For more information on configuring gateway settings, see [Configuring Proxies for Tableau Server](#).

Tableau Server Installations using User-Based licenses

The recommended topology for a production Tableau Server installation is a dedicated node for running flows. For more information, see [Minimum Hardware Requirements and Recommendations for Tableau Server](#).

Tableau Server single-node installations

If you currently have a single node Tableau Server installation, it is recommended that you add a second node and dedicate it to running flows.

1. Run upgrade on your current Tableau Server installation using the information in the topics below:
 - [Windows](#)
 - [Linux](#)

When you get to the **Activate** step, use the Tableau Server product keys to activate Tableau Server.

All product keys are available through the [Customer Portal](#).

2. After completing the installation, add the Data Management product key to enable Tableau Prep Conductor on your node. The Data Management product key, like your other server keys, are available through the [Customer Portal](#).
 - In the Tableau Services Manager web interface, click **Licensing** on the **Configuration** tab and click **Activate License**.
 - Enter or paste your new product key and click **Activate**.

Tableau Server on Linux Administrator Guide

- On the **Register** page, enter your information into the fields and click **Register**.
3. You will be prompted to restart the server. Restart the server and verify that Tableau Prep Conductor is enabled and running.
- In the Tableau Services Manager web interface, click the **Status** tab to see the status. If Tableau Prep Conductor is enabled and running, you should see Tableau Prep Conductor in the list of processes as **Active**. If Tableau Prep Conductor is not enabled, you will see Tableau Prep Conductor in the list of processes, but with no status information.

Tableau Prep Conductor not enabled:

The screenshot shows the Tableau Services Manager web interface. The top navigation bar includes 'STATUS', 'MAINTENANCE', and 'CONFIGURATION'. The main content area displays a table of processes for 'node1'. The 'Tableau Prep Conductor' row is highlighted with a red border and shows a green checkmark in the status column, indicating it is active. A legend at the bottom right identifies the status icons: Active (green checkmark), Busy (green square), Degraded (red triangle), Error (red square), and Stopped (grey square).

Process	node1
Gateway	✓
Application Server	✓
Interactive Microservice Container	✓
VisQ Server	✓ ✓ ✓ ✓
Cache Server	✓ ✓
Cluster Controller	✓
Search & Browse	✓
Backgrounder	✓ ✓
Non-Interactive Microservice Container	✓
Data Server	✓ ✓
Data Engine	✓
File Store	✓
Repository	✓
Tableau Prep Conductor	✓
Ask Data	✓
Elastic Server	✓
TSM Controller	✓
License Server	✓

Tableau Prep Conductor enabled and running:

Process	node1
Gateway	✓
Application Server	✓
Interactive Microservice Container	✓
VizQL Server	✓ ✓ ✓ ✓
Cache Server	✓ ✓
Cluster Controller	✓
Search & Browse	✓
Backgrounder	✓ ✓
Non-Interactive Microservice Container	✓
Data Server	✓ ✓
Data Engine	✓
File Store	✓
Repository	✓
Tableau Prep Conductor	✓
Ask Data	✓
Elastic Server	✓
TSM Controller	✓
License Server	✓

Refresh Status

Active | Busy | Degraded | Error | Stopped

- Add a second node to your Tableau Server installation. The installer will enable certain required processes like the Cluster Controller. Enable Backgrounder process on it as it is required to run scheduled flow tasks. When you enable the Backgrounder process, the installer automatically enables a single instance of Data Engine and Tableau Prep Conductor on the node. Do not add any other processes on this node.
- Run the following commands to dedicate this node to do only flow tasks. For more information on node roles, see [Node Roles in Tableau Server](#).

- Get the nodeID for your dedicated node to see the list of services on each node:

```
tsm topology list-nodes -v.
```

- Set the node role for the dedicated node using the nodeID that you got from running the command described above:

```
tsm topology set-node-role -n <nodeID> -r flows.
```

- Apply the changes, and restart the server:

```
tsm pending-changes apply.
```

- Review the status to ensure that all the processes are up and running and configured correctly:

```
tsm status -v.
```

You have successfully added Tableau Prep Conductor to your Tableau Server installation.

Tableau Server multi-node installations

1. Run upgrade on your current Tableau Server Installation using the information in the topics below:

- [Windows](#)
- [Linux](#)

When you get to the **Activate** step, use the Tableau Server product keys to activate Tableau Server.

All product keys are available through the [Customer Portal](#).

2. After completing the installation, add the Data Management product key to enable Tableau Prep Conductor. Tableau Prep Conductor is automatically enabled on the nodes where you already have the Backgrounder process enabled. The Data Management product key, like your other server keys, are available through the [Customer Portal](#).
 - In the Tableau Services Manager web interface, click **Licensing** on the **Configuration** tab and click **Activate License**.
 - Enter or paste your new product key and click **Activate**.
 - On the **Register** page, enter your information into the fields and click **Register**.
3. You will be prompted to restart the server. Restart the server and verify that Tableau Prep Conductor is enabled and is running.

- In the Tableau Services Manager web interface, click the **Status** tab to see the status of all the processes. If Tableau Prep Conductor is enabled and running, you should see Tableau Prep Conductor in the list of processes as **Active**. If Tableau Prep Conductor is not enabled, you will see Tableau Prep Conductor in the list of processes, but with no status information.

Tableau Prep Conductor not enabled:

The screenshot shows the Tableau Services Manager interface with the 'STATUS' tab selected. The 'Tableau Prep Conductor' row is highlighted with a red box, indicating it is not running. The legend at the bottom right shows status icons: Active (green checkmark), Busy (green circle with slash), Degraded (red triangle), Error (red exclamation mark), and Stopped (grey X).

Process	node1	node2	node3
Gateway	✓		
Application Server	✓		
Interactive Microservice Container	✓		
VizQL Server	✓ ✓ ✓ ✓		
Cache Server	✓ ✓		
Cluster Controller	✓	✓	✓
Search & Browse	✓		
Backgrounder	✓ ✓		✓ ✓
Non-Interactive Microservice Container	✓		✓
Data Server	✓ ✓		
Data Engine	✓	✓	✓
File Store	✓	✓	
Repository	✓		
Tableau Prep Conductor			
Ask Data	✓		
Elastic Server	✓		
TSM Controller	✓		
License Server	✓		

Tableau Prep Conductor enabled and running:

Tableau Server on Linux Administrator Guide

The screenshot shows the Tableau Server Administration console with the 'STATUS' tab selected. The interface displays a table of processes and their status across three nodes (node1, node2, node3). The 'Tableau Prep Conductor' process is highlighted with a red box. A legend at the bottom right indicates the status icons: Active (green checkmark), Busy (green circle with slash), Degraded (red triangle), Error (red square), and Stopped (grey square).

Process	node1	node2	node3
Gateway	✓		
Application Server	✓		
Interactive Microservice Container	✓		
VizQL Server	✓ ✓ ✓ ✓		
Cache Server	✓ ✓		
Cluster Controller	✓	✓	✓
Search & Browse	✓		
Backgrounder	✓ ✓		✓ ✓
Non-Interactive Microservice Container	✓		✓
Data Server	✓ ✓		
Data Engine	✓	✓	✓
File Store	✓	✓	
Repository	✓		
Tableau Prep Conductor	✓		✓
Ask Data	✓		
Elastic Server	✓		
TSM Controller	✓		
License Server	✓		

4. Add a new node to your Tableau Server installation. The installer will enable certain required processes like the Cluster Controller. Enable Backgrounder process on it as it is required to run scheduled flow tasks. When you enable the Backgrounder process, the installer automatically enables a single instance of Data Engine and Tableau Prep Conductor on the node. Do not add any other processes on this node.

Note: The dedicated node counts towards the total count of the Coordination Service ensemble. You may need to deploy a Coordination Service on the new node depending on the total number of nodes you have in your cluster including the new dedicated node. For more information, see [Deploy a Coordination Service Ensemble](#).

5. Run the following command to dedicate this node to only doing flow related operations. For more information on node roles, see [Node Roles in Tableau Server](#).

- Get the nodeID for your dedicated node to see the list of services on each node:
 - `tsm topology list-nodes -v.`
 - Set the node role for the dedicated node using the nodeID that you got from running the command described above:
 - `tsm topology set-node-role -n <nodeID> -r flows.`
 - Apply the changes and restart the server:
 - `tsm pending-changes apply.`
 - Review the status to ensure that all the processes are up and running and configured correctly:
 - `tsm status -v.`
6. At this stage, you may have Tableau Prep Conductor enabled on other nodes. By default, the Backgrounder process on a node performs all tasks of all types including flow tasks. To isolate Tableau Prep Conductor and flow tasks to only certain nodes, you can configure the Backgrounders to do one of the following:
- To run only flow tasks: `tsm topology set-node-role -n <nodeID> -r flows.`
 - To run all other tasks except flows: `tsm topology set-node-role -n <nodeID> -r no-flows.`

You have successfully added Tableau Prep Conductor to your Tableau Server installation.

Tableau Server Installations using Core-Based licenses

The recommended topology for a production Tableau Server installation is a dedicated node for running flows. For more information, see [Minimum Hardware Requirements and Recommendations for Tableau Server](#).

The Data Management Add-on for Core-Based licenses includes product keys that enable Tableau Prep Conductor for your Tableau Server, and Tableau Prep Conductor cores that comes in units of four. The Tableau Prep Conductor cores should be applied to the node dedicated to running the flows. These product keys, like your other server keys, are available through the [Customer Portal](#).

To learn more about Tableau Prep Conductor licensing, see [Licensing Tableau Prep Conductor for Tableau Server](#).

Tableau Server single-node installations

If you currently have a single node Tableau Server installation, it is recommended that you add a second node and dedicate it to running flows.

1. Run upgrade on your current Tableau Server Installation using the information in the topics below:
 - [Windows](#)
 - [Linux](#)
2. Activate the product keys. This will enable Tableau Prep Conductor on the nodes where you already have the Backgrounder process enabled. When you are using core-based licensing, you must apply both the Data Management product key and the Resource Core product key to your Tableau Deployment. The first key allows flows to be run on Tableau Server through the Tableau Prep Conductor and the second key adds the additional cores for the Tableau Prep Conductor nodes. All product keys are available through the [Customer Portal](#).
 - In the Tableau Services Manager web interface, click **Licensing** on the **Configuration** tab and click **Activate License**.
 - Enter or paste your new product key and click **Activate**.
 - On the **Register** page, enter your information into the fields and click **Register**.
3. You will be prompted to restart the server. Restart the server and verify that Tableau Prep Conductor is enabled and is running.
 - In the Tableau Services Manager web interface, click the **Status** tab to see the status. If Tableau Prep Conductor is enabled and running, you should see

Tableau Prep Conductor in the list of processes as **Active**. If Tableau Prep Conductor is not enabled, you will see Tableau Prep Conductor in the list of processes, but with no status information.

Tableau Prep Conductor not enabled:

The screenshot shows the Tableau Server status page for 'node1'. The 'Tableau Prep Conductor' process is listed with a red box around it, but it has no status icon next to it, indicating it is not enabled.

Process	Status
Gateway	Active
Application Server	Active
Interactive Microservice Container	Active
VizQL Server	Active
Cache Server	Active
Cluster Controller	Active
Search & Browse	Active
Backgrounder	Active
Non-Interactive Microservice Container	Active
Data Server	Active
Data Engine	Active
File Store	Active
Repository	Active
Tableau Prep Conductor	
Ask Data	Active
Elastic Server	Active
TSM Controller	Active
License Server	Active

Tableau Prep Conductor enabled and running:

The screenshot shows the Tableau Server status page for 'node1'. The 'Tableau Prep Conductor' process is listed with a red box around it and a green checkmark icon next to it, indicating it is enabled and running.

Process	Status
Gateway	Active
Application Server	Active
Interactive Microservice Container	Active
VizQL Server	Active
Cache Server	Active
Cluster Controller	Active
Search & Browse	Active
Backgrounder	Active
Non-Interactive Microservice Container	Active
Data Server	Active
Data Engine	Active
File Store	Active
Repository	Active
Tableau Prep Conductor	Active
Ask Data	Active
Elastic Server	Active
TSM Controller	Active
License Server	Active

4. Add a second node to your Tableau Server installation. The installer will enable certain required processes like the Cluster Controller. Enable Backgrounder process on

it as it is required to run scheduled flow tasks. When you enable the Backgrounder process, the installer automatically enables a single instance of Data Engine and Tableau Prep Conductor on the node. Do not add any other processes on this node.

Important: The number of physical cores on this machine must be equal to, or less than the Tableau Prep Conductor cores you purchased. For example, if you purchased four Tableau Prep Conductor cores, your node can only have up to four physical cores. To understand about how Tableau Prep Conductor licensing works, see [Licensing Tableau Prep Conductor for Tableau Server](#).

5. Run the following commands to dedicate this node to only doing flow tasks. For more information on node roles, see [Node Roles in Tableau Server](#).
 - Get the nodeID for your dedicated node to see the list of services on each node:

```
tsm topology list-nodes -v.
```
 - Set the node role for the dedicated node using the nodeID that you got from running the command described above:

```
tsm topology set-node-role -n <nodeID> -r flows.
```
 - Apply the changes and restart the server: `tsm pending-changes apply`.
 - Review the status to ensure that all the processes are up and running and configured correctly:

```
tsm status -v.
```

You have successfully added Tableau Prep Conductor to your Tableau Server installation.

Tableau Server multi-node installations

1. Run upgrade on your current Tableau Server installation using the information in the topics below:

- [Windows](#)
 - [Linux](#)
2. Activate the product keys. This will enable Tableau Prep Conductor on the nodes where you already have the Backgrounder process enabled. When you are using core-based licensing, you must apply both the Data Management product key and the Resource Core product key to your Tableau Deployment. The first key allows flows to be run on Tableau Server through the Tableau Prep Conductor and the second key adds the additional cores for the Tableau Prep Conductor nodes. All product keys are available through the [Customer Portal](#).
 - In the Tableau Services Manager web interface, click **Licensing** on the **Configuration** tab and click **Activate License**.
 - Enter or paste your new product key and click **Activate**.
 - On the **Register** page, enter your information into the fields and click **Register**.
 3. You will be prompted to restart the server. Restart the server and verify that Tableau Prep Conductor is enabled and is running.
 - In the Tableau Services Manager web interface, click the **Status** tab to see the status. If Tableau Prep Conductor is enabled and running, you should see Tableau Prep Conductor in the list of processes as **Active**. If Tableau Prep Conductor is not enabled, you will see Tableau Prep Conductor in the list of processes, but with no status information.

Tableau Prep Conductor not enabled:

Tableau Server on Linux Administrator Guide

Process	node1	node2	node3
Gateway	✓		
Application Server	✓		
Interactive Microservice Container	✓		
VirQL Server	✓✓✓✓		
Cache Server	✓✓		
Cluster Controller	✓	✓	✓
Search & Browse	✓		
Backgrounder	✓✓		✓✓
Non-Interactive Microservice Container	✓		✓
Data Server	✓✓		
Data Engine	✓	✓	✓
File Store	✓	✓	
Repository	✓		
Tableau Prep Conductor			
Ask Data	✓		
Elastic Server	✓		
TSM Controller	✓		
License Server	✓		

Refresh Status Active Busy Degraded Error Stopped

Tableau Prep Conductor enabled and running:

Process	node1	node2	node3
Gateway	✓		
Application Server	✓		
Interactive Microservice Container	✓		
VirQL Server	✓✓✓✓		
Cache Server	✓✓		
Cluster Controller	✓	✓	✓
Search & Browse	✓		
Backgrounder	✓✓		✓✓
Non-Interactive Microservice Container	✓		✓
Data Server	✓✓		
Data Engine	✓	✓	✓
File Store	✓	✓	
Repository	✓		
Tableau Prep Conductor	✓		✓
Ask Data	✓		
Elastic Server	✓		
TSM Controller	✓		
License Server	✓		

Refresh Status Active Busy Degraded Error Stopped

4. Add a new node to your Tableau Server installation. A dedicated node to run flow related operations is recommended for production Tableau Server installations. The installer will enable certain required processes like the Cluster Controller. Enable Backgrounder process on it as it is required to run scheduled flow tasks. When you enable the Backgrounder process, the installer automatically enables a single instance of Data Engine on the node. Do not add any other processes on this node.

Note: The dedicated node counts towards the total count of the Coordination Service ensemble. You may need to deploy a Coordination Service on the new node depending on the total number of nodes you have in your cluster including the new dedicated node. For more information, see [Deploy a Coordination Service Ensemble](#).

Important:

The number of physical cores on this machine must be equal to, or less than the Tableau Prep Conductor cores you purchased. For example, if you purchased four Tableau Prep Conductor cores, your node can only have up to four physical cores. To understand about how Tableau Prep Conductor licensing works, see [Licensing Tableau Prep Conductor for Tableau Server](#).

5. Run the following commands to dedicate this node to only doing flow tasks. This will enable Tableau Prep Conductor on your new node. For more information, see [Node Roles in Tableau Server](#).

- Get the nodeID for your dedicated node to see the list of services on each node:

```
tsm topology list-nodes -v.
```

- Set the node role for the dedicated node using the nodeID that you got from running the command described above:

Tableau Server on Linux Administrator Guide

```
tsm topology set-node-role -n nodeID -r flows.
```

- Apply the changes and restart the server:

```
tsm pending-changes apply.
```

- Review the status to ensure that all the processes are up and running and configured correctly:

```
tsm status -v.
```

6. At this stage, you may have Tableau Prep Conductor enabled on other nodes that have the Backgrounder process. By default, the Backgrounder process on a node performs all tasks of all types including flow tasks. To isolate Tableau Prep Conductor and flow operations to only certain nodes, you can configure the backgrounders to do one of the following:

- To run only flow tasks:

```
tsm topology set-node-role -n <nodeID> -r flows.
```

- To run all other tasks except flows:

```
tsm topology set-node-role -n <nodeID> -r no-flows.
```

Next step

Step 2: Configure Flow Settings for your Tableau Server.

Who can do this

Tableau Server Administrators can install or upgrade Tableau Server, and enable Tableau Prep Conductor on Tableau Server.

Step 2: Configure Flow Settings for your Tableau Server

This topic describes the various flow settings that you can configure for your Tableau Server.

Publishing, Scheduling, and Credential Settings

When you activate Tableau Prep Conductor using the Data Management product key, Tableau Prep Conductor is enabled for the entire Tableau Server installation. You can further modify and customize the setting for sites.

Use the following instructions to configure settings related to flows for all your sites or for individual sites:

Use the following instructions to sign in to Tableau Server Admin pages:

- Windows: [Tableau Server Admin Pages](#).
- Linux: [Tableau Server Admin Pages](#).

Configure whether publishing and scheduling flow should be allowed for a site:

1. **Enable users to publish and schedule flows:** This setting is enabled by default when you enable Tableau Prep Conductor. If you have multiple sites, you can selectively turn off Tableau Prep for Server for individual sites. If you disable this setting for a site that once allowed flows, see [Implication of disabling Tableau Prep Conductor](#) to learn more.

On the **General** page under **Settings**, scroll to the **Tableau Prep Conductor** section and clear the **Allow users to schedule and monitor flows** check box.

2. **Embed Credentials**

-Allow publishers to embed credentials in a data source, flow or workbook:

This setting allows publishers to attach passwords to published flows that will automatically authenticate web users.

-Allow publishers to schedule flow runs and data extract refreshes: This option is only available if setting above is enabled. When this setting is enabled, publishers will see scheduling options in the Publish dialog box.

Implication of disabling Tableau Prep Conductor

If you disable Tableau Prep Conductor after using it for a while, you will not be able to see the flows, schedules, tasks, and other things related to flows. The following table gives you more information on what you can and cannot see when you disable Tableau Prep for Server completely or only for specific sites:

	Prep not enabled at Server level	Prep enabled at Server level, but disabled for a site	Prep enabled for both Server and site
Show flows	No	No	Yes
Show tasks/schedules in Server view	No	Yes	Yes
Show tasks/schedules in Site view	No	No	Yes
Show Site setting (only for Server Admins)	Yes (disabled)	Yes	Yes
Show TSM status	Yes (Tableau Prep Conductor is not shown)	Yes	Yes
Show TSM settings	Yes (disabled)	Yes	Yes

Important: Scheduled tasks will continue to run even when Tableau Prep Conductor is disabled for that site, but will fail.

Configure notifications for flow failures

You can configure Tableau Server to send email notifications for flow run failures. The notifications are sent for failures that occur when running the flows through either a scheduled

task or a manual run using the **Run now** menu option. You must first enable the server-wide setting, and then configure at the site level.

To enable the server-wide email notification

You can either use the Tableau Services Manager (TSM) web interface or TSM CLI as described below:

Use the TSM web interface

1. Open TSM in a browser:

`https://<tsm-computer-name>:8850.`

2. Click **Notifications** on the **Configuration** tab and click **Email Server**.
3. Enter the email server information.
4. Click the **Events** tab.
5. Under **Content Updates**, select **Send emails for extract refresh failures and flow run failures** if not already turned on by default.
6. Click **Save Pending Changes** after you've entered your configuration information.
7. Click **Apply Changes and Restart**.

Use the TSM CLI

The notification values can be set individually with the `tsm configuration set` command:

Windows: [tsm configuration](#).

Linux: [tsm configuration](#).

Set notification values

Use the `tsm configuration set` command with the following syntax to

to enable flow failure notifications, run the following command:

```
tsm configuration set -k backgrounder.notifications_enabled -v true
```

Note: This will enable email notification for both extract refresh failures and flow failures.

After you are done setting values, you must run the following command:

```
tsm pending-changes apply
```

The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in the case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior.

To configure email notification for a site:

On the **General** page under **Settings**, scroll to the **Flow Run and Refresh Failure Notifications** setting and clear the check box to turn off refresh failure and flow run notifications.

General Extensions

Revert Save

Offline Favorites for Tableau Mobile
Let users access favorite workbooks and views, even if they're disconnected from the server.
 Enable offline favorites

Flow Run and Extract Job Failure Notifications
Enable email notifications to inform content owners when flow or extract jobs fail. Content owners will receive notifications when Tableau is unable to complete a flow run, encryption job, or scheduled refresh. Tableau will only send encryption job failure emails to owners when the site encryption mode is set to Enable.
 Send email to data source, flow, and workbook owners when flow runs, encryption jobs, or scheduled refreshes fail.

Next step

Step 3: Create Schedules for Flow Tasks

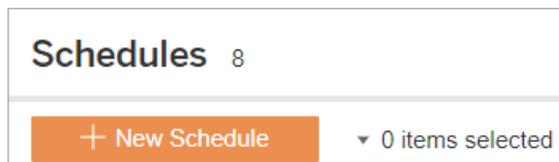
Who can do this

Tableau Server Administrators can configure server and site level settings. Tableau Site Administrators can configure site level settings.

Step 3: Create Schedules for Flow Tasks

Create a new schedule:

1. On the **Schedules** tab, click **New Schedule**.



2. Enter the following information in the **New Schedule** dialog box and click **Create**.
 - **Name:** Enter a descriptive name for the schedule. Typically, this includes the description of the schedule frequency.
 - **Task Type:** Select Flow as the task type.

- **Default Priority:** You can define a default priority from 1 to 100, where 1 is the highest priority. This value will be assigned to the tasks by default. If two tasks are pending in the queue, the one with the higher priority runs first.
- **Execution:** Choose whether a schedule will run in parallel or serially. Schedules that run in parallel run on all available background processes so that they can complete faster.
- **Frequency:** You can define an hourly, daily, weekly, or monthly schedule.

New Schedule

Create a schedule users can choose for running extract refreshes, flows, or subscriptions.

Name

Task type

Default priority
Tasks are executed in priority order from 1 to 100

Execution Parallel: Use all available background processes for this schedule
 Serial: Limit this schedule to one background process

Frequency Hourly
 Daily
 Weekly
 Monthly

every

from :

to :

Next step

Step 4: Safe list Input and Output locations

Who can do this

Tableau Server Administrators can create and modify schedules. Schedules are created at the server level and apply across all the sites on a server.

Step 4: Safe list Input and Output locations

This topic describes the rules that apply to this feature and how to safe list the directories on your network.

Flow input and output connections may need to connect to databases or files in the directories on your network. You must safe list the directories you want to allow access to. Input and Output connections will only be allowed to connect to data in the safe listed locations. By default, no connections are allowed.

Note: You can still publish the flows and any data that is embedded in the flow file (tflx) to Tableau Server, but the flow will fail to run if the directories aren't included in your organization's safe list.

How to safe list input and output locations

The following rules apply and must be considered when configuring this setting:

- The directory paths should be accessible by Tableau Server. These paths are verified during server startup and at flow run time and are **not** verified at the time of publishing the flow to Tableau Server.
- Network directory paths have to be absolute and cannot contain wildcards or other path traversing symbols. For example, `\\myhost\myShare*` **or** `\\myhost\myShare*` are invalid paths and would result in all the paths as disallowed. The correct way to safelist any folder under *myShare* would be `\\myhost\myShare` **or** `\\myhost\myShare\.`

Note: The `\\myhost\myShare` configuration will not allow `\\myhost\myShare1`. In order to safe list both of these folders safe list them as `\\myhost\myShare; \\myhost\myShare1`.

- **Windows:**

Tableau Server on Linux Administrator Guide

- The value can be either *, (for example, `tsm configuration set -k maestro.input.allowed_paths -v "*"`) to allow any network directory, or a specified list of network directory paths, delimited by a semicolon (;). If you specify a list of directory paths, be sure to specify particular directories rather than the root of the file share.
- If the path contains spaces or special characters you will have to either use single or double quotes. Whether you use single or double quotes depends on the shell that you are using.
- No local directory paths are allowed even when the value is set to *.
- To save flow output to a network share, you must first [configure a Run As user](#) service account on Tableau Server. You cannot save flows to a network share using the default system account. Then configure the target directory on the network share for Full Control permissions for the Run As user account you created.

Depending on how your organization manages nested folder permissions, you may need to grant additional permissions in the folder hierarchy, with a minimum of Read, Write, Execute, Delete, and List Folder permission, to allow the Run As user account access to the target folder.

- **Linux:**

- The value can be either *, (for example, `tsm configuration set -k maestro.input.allowed_paths -v "*"`) meaning that any path, including local (with the exception of some system paths configured using “`native_api.internal_disallowed_paths`”), or a list of paths, delimited by a semicolon (;).
- You must be using a kernel version of equal to or later than 4.7. Safe listing is not supported on kernel version earlier than 4.7. To check the kernel version, in the Linux terminal, type the command `uname -r`. This will display the full

version of the kernel you are running on the Linux machine. Note that for Red Hat Enterprise Linux, kernel version 4.7 and later is only available with Red Hat Enterprise Linux version 8, which is not supported by Tableau Server.

- To save flow output to a network share, the local Linux account that has access to Tableau Server resources must be given Full Control permissions to the target directory on the network share. For information about configuring this, see this Tableau Community post: [Tableau Server on Linux - Connecting to a Windows Shared Directory](#).

Note: If a path is both on the flows allowed list and internal_disallowed list, internal_disallowed takes precedence.

Use the following commands to create a list of allowed network directory paths:

For input connections:

```
tsm configuration set -k maestro.input.allowed_paths -v your_
networkdirectory_path_1;your_networkdirectory_path_2
```

```
tsm pending-changes apply
```

For output connections:

```
tsm configuration set -k maestro.output.allowed_paths -v your_
networkdirectory_path_1;your_networkdirectory_path_2
```

```
tsm pending-changes apply
```

Important:

These commands overwrite existing information and replace it with the new information you provided. If you want to add a new location to an existing list, you must provide a list of all the locations, existing, and the new one you want to add. Use the following commands to see the current list of input and output locations:

Tableau Server on Linux Administrator Guide

```
tsm configuration get -k maestro.input.allowed_paths  
tsm configuration get -k maestro.output.allowed_paths
```

Next step

Step 5: Optional Server Configurations

Who can do this

Tableau Server Administrators who also are a member of the **tsmadmin** group can configure settings using *tsm* commands.

Step 5: Optional Server Configurations

The options described in this topic are not required to enable flow publishing and scheduling flows on Tableau Server. They can be used to customizing your environment according to your requirements.

Set the timeout period for flows

You can set time limits for how long a flow can run to make sure that subsequent tasks are not held up due to stalled tasks. The following two *tsm* command options determine how long a flow task can run before the flow background task is canceled. These two commands together determine the total timeout value for flow tasks.

Use the following *tsm* commands to change the default timeout period:

```
backgrounder.extra_timeout_in_seconds
```

(Default value: 1800 seconds or 30 minutes)

```
backgrounder.default_timeout.run_flow
```

(Default value: 14400 seconds or 4 hours)

Set the threshold for suspended flow tasks

By default, a flow task is suspended after 5 consecutive flow task failures. To change the threshold number of flow task failures that can occur before they are suspended, use the following tsm configuration set command:

```
tsm configuration set -k backgrounder.flow_failure_threshold_
for_run_prevention -v <number>
```

This sets the threshold for the number of consecutive failed flow tasks necessary before suspending the tasks. This is a server-wide setting.

Who can do this

Tableau Server administrators can make changes to server configurations.

About Tableau Catalog

Data is increasing in volume, formats, and importance leading to more complex environments. With the rapid pace that data changes, it can be hard to keep track of that data and how it's being used in such complex environments. At the same time, more users need to access more of that data in more places, and it's difficult for users to find the right data. Ultimately, this causes a lack of trust in the data because people question whether they're using the right source or if the source is up to date.

Tableau Catalog integrates features like lineage, impact analysis, data dictionary, data quality warnings, and search into your Tableau applications, helping solve these problems differently from a stand-alone catalog. It focuses on both IT and the end user so that everyone using Tableau Server or Tableau Online has more trust in and visibility into the data, while also enabling more discoverability. Tableau Catalog builds a catalog out of the Tableau content being used by your organization, enabling comprehensive functionality like the following:

- **Impact analysis and lineage.**
 - You can see the workbooks and other Tableau content that depend on particular columns or fields from tables or data sources you manage. When you need to make changes to your data, you can notify the impacted Tableau authors using email.
 - As a workbook author, you can use lineage to trace the fields that your workbook depends on.
 - As a user, when you use a Tableau visualization, you can see where the data came from that was used to create the view.
- **Curation and trust.** As a data steward, you can add helpful metadata, like descriptions and certification, so that users find the right data. You can set data quality warnings, view data details on the Data Details pane, certify assets, and remove assets from the catalog.
- **Data discovery.** You can use Tableau Catalog to search for databases, tables, and data sources to analyze in Tableau and connect to them from the search results.

Starting in 2019.3, Tableau Catalog is available in the Data Management Add-on to Tableau Server and Tableau Online. When the product key is active and enabled, the catalog features described above are integrated into the product you're using, so you can work with the data where you find it.

How Tableau Catalog works

Tableau Catalog discovers and indexes all the content on your site—workbooks, metrics, data sources, sheets, and flows—to gather metadata about the content. From the metadata, databases and tables (also referred to as external assets) are identified. Knowing the relationships between the content and the external assets enables Tableau to display the lineage of the content and external assets. Tableau Catalog also enables users to connect to external assets using Tableau Server or Tableau Online.

For information about how you can use Tableau Catalog to support data governance in your organization, see [Governance in Tableau](#) in the Tableau Blueprint Help.

Key Tableau Catalog terms

- **Metadata.** Information about the data.
- **Tableau content.** Content created in Tableau such as workbooks, data sources, and flows.
- **External assets.** The metadata about the databases and tables used by the Tableau content published to Tableau Server or Tableau Online.

License Tableau Catalog

Tableau Catalog is licensed through the Data Management Add-on. For information about how Data Management Add-on licensing works, see [License the Data Management Add-on](#).

Enable Tableau Catalog

After Tableau Server or Tableau Online is licensed with the Data Management, you can enable Tableau Catalog by doing one of the following tasks:

- **For Tableau Online**, no action necessary. Tableau Catalog is on by default, configured to use derived permissions, and ready to use. For more information about derived permissions, see the [Permissions on metadata](#) topic.
- **For Tableau Server**, the Server admin must first enable the Tableau Metadata API using the `tsm maintenance metadata-services` command. For more information, see [Enable Tableau Catalog](#).

After the Metadata API is enabled, Tableau Catalog is on by default, configured to use derived permissions, and ready to use. For more information about derived permissions, see the [Permissions on metadata](#) topic.

Features and functionality

To learn more about the features you can use with Tableau Catalog, see the following [Help articles](#):

Tableau Server on Linux Administrator Guide

Data discovery

- In the **Connect** pane on Tableau Desktop, under **Search for Data** select **Tableau Server** to [connect to data using Tableau Server or Tableau Online](#). When Tableau Catalog is enabled, in addition to searching for published data sources to connect to, you can now search for and connect to the specific databases and tables used by published data sources and workbooks on your Tableau Server or your Tableau Online site.
- [Search](#) is expanded to include results based on fields, columns, databases, and tables when Tableau Catalog is enabled.
- If you author in the web, you can also connect to databases and tables, as well as published data sources, from [Tableau Server](#) and [Tableau Online](#).

Curation and trust

- [Certify your data assets](#) to help users find trusted and recommended data.
- Set [data quality warnings](#) to alert users to data quality issues, such as stale or deprecated data.
- Categorize items on Tableau Server and Tableau Online using [tags](#) to help users filter external assets.
- Better understand published visualizations by using the [Data Details](#) tab to see information about the data used.
- [Add descriptions](#) to databases, tables, and columns to help users find the data they're looking for.

Lineage and impact analysis

- Use [lineage](#) to trace the source of your data and to analyze the impact of changes to your data and identify which users might be impacted.
- [Email owners](#) of a workbook, data source, or flow or contacts for a database or table about data-related updates.

Developer resources

You can use metadata methods in the Tableau REST API to programmatically update certain metadata. For more information about the metadata methods, see [Metadata Methods](#) in the Tableau Server REST API.

In addition to the REST API, you can use the [Tableau Metadata API](#) to programmatically query metadata from the content published to Tableau Server or Tableau Online. The Metadata API is fast and flexible and is best when you are looking to find out specific information about the relationship between metadata and its structures. Explore and test queries against the Metadata API using an interactive in-browser tool called [GraphiQL](#).

Note: The Data Management Add-on is not required to use the Metadata API or GraphiQL.