



Red Hat Enterprise Linux 8

Installing Identity Management

Methods of installing IdM servers and clients

Red Hat Enterprise Linux 8 Installing Identity Management

Methods of installing IdM servers and clients

Legal Notice

Copyright © 2024 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

Depending on your environment, you can install Red Hat Identity Management (IdM) to provide DNS and Certificate Authority (CA) services, or you configure IdM to use an existing DNS and CA infrastructure. You can install IdM servers, replicas, and clients manually or by using Ansible Playbooks. Additionally, you can use a Kickstart file to automatically join a client to an IdM domain during the system installation.

Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE	7
PROVIDING FEEDBACK ON RED HAT DOCUMENTATION	8
CHAPTER 1. HOW TO USE THIS GUIDE	9
PART I. INSTALLING IDENTITY MANAGEMENT	10
CHAPTER 2. PREPARING THE SYSTEM FOR IDM SERVER INSTALLATION	11
2.1. PREREQUISITES	11
2.2. HARDWARE RECOMMENDATIONS	11
2.3. CUSTOM CONFIGURATION REQUIREMENTS FOR IDM	11
2.4. FIPS COMPLIANCE	13
Migration to FIPS-compliant IdM	14
2.5. SUPPORT FOR CROSS-FOREST TRUST WITH FIPS MODE ENABLED	14
2.6. TIME SERVICE REQUIREMENTS FOR IDM	15
2.6.1. How IdM uses chronyd for synchronization	15
2.6.2. List of NTP configuration options for IdM installation commands	15
2.6.3. Ensuring IdM can reference your NTP time server	16
2.6.4. Additional resources	17
2.7. HOST NAME AND DNS REQUIREMENTS FOR IDM	17
2.8. PORT REQUIREMENTS FOR IDM	20
2.9. OPENING THE PORTS REQUIRED BY IDM	21
2.10. INSTALLING PACKAGES REQUIRED FOR AN IDM SERVER	22
2.11. SETTING THE CORRECT FILE MODE CREATION MASK FOR IDM INSTALLATION	24
2.12. ENSURING THAT FAPOLICYD RULES DO NOT BLOCK IDM INSTALLATION AND OPERATION	24
2.13. OPTIONS FOR THE IDM INSTALLATION COMMANDS	24
CHAPTER 3. INSTALLING AN IDM SERVER: WITH INTEGRATED DNS, WITH AN INTEGRATED CA AS THE ROOT CA	27
3.1. INTERACTIVE INSTALLATION	27
3.2. NON-INTERACTIVE INSTALLATION	29
CHAPTER 4. INSTALLING AN IDM SERVER: WITH INTEGRATED DNS, WITH AN EXTERNAL CA AS THE ROOT CA	31
4.1. INTERACTIVE INSTALLATION	31
4.2. TROUBLESHOOTING: EXTERNAL CA INSTALLATION FAILS	34
What this means:	35
To fix the problem:	35
CHAPTER 5. INSTALLING AN IDM SERVER: WITH INTEGRATED DNS, WITHOUT A CA	36
5.1. CERTIFICATES REQUIRED TO INSTALL AN IDM SERVER WITHOUT A CA	36
5.2. INTERACTIVE INSTALLATION	37
CHAPTER 6. INSTALLING AN IDM SERVER: WITHOUT INTEGRATED DNS, WITH AN INTEGRATED CA AS THE ROOT CA	41
6.1. INTERACTIVE INSTALLATION	41
6.2. NON-INTERACTIVE INSTALLATION	43
6.3. IDM DNS RECORDS FOR EXTERNAL DNS SYSTEMS	44
CHAPTER 7. INSTALLING AN IDM SERVER: WITHOUT INTEGRATED DNS, WITH AN EXTERNAL CA AS THE ROOT CA	45
7.1. OPTIONS USED WHEN INSTALLING AN IDM CA WITH AN EXTERNAL CA AS THE ROOT CA	45
7.2. INTERACTIVE INSTALLATION	46

7.3. NON-INTERACTIVE INSTALLATION	48
7.4. IDM DNS RECORDS FOR EXTERNAL DNS SYSTEMS	50
CHAPTER 8. INSTALLING AN IDM SERVER OR REPLICA WITH CUSTOM DATABASE SETTINGS FROM AN LDIF FILE	52
CHAPTER 9. TROUBLESHOOTING IDM SERVER INSTALLATION	53
9.1. REVIEWING IDM SERVER INSTALLATION ERROR LOGS	53
9.2. REVIEWING IDM CA INSTALLATION ERRORS	54
9.3. REMOVING A PARTIAL IDM SERVER INSTALLATION	55
9.4. ADDITIONAL RESOURCES	56
CHAPTER 10. UNINSTALLING AN IDM SERVER	57
CHAPTER 11. RENAMING AN IDM SERVER	60
CHAPTER 12. UPDATING AND DOWNGRADING IDM	61
12.1. UPDATING IDM PACKAGES	61
12.2. DOWNGRADING IDM PACKAGES	61
12.3. ADDITIONAL RESOURCES	62
CHAPTER 13. PREPARING THE SYSTEM FOR IDM CLIENT INSTALLATION	63
13.1. SUPPORTED VERSIONS OF RHEL FOR INSTALLING IDM CLIENTS	63
13.2. DNS REQUIREMENTS FOR IDM CLIENTS	63
13.3. PORT REQUIREMENTS FOR IDM CLIENTS	63
13.4. IPV6 REQUIREMENTS FOR IDM CLIENTS	64
13.5. INSTALLING IDM CLIENT PACKAGES FROM THE IDM:CLIENT STREAM	64
13.6. INSTALLING IDM CLIENT PACKAGES FROM THE IDM:DL1 STREAM	65
CHAPTER 14. INSTALLING AN IDM CLIENT	66
14.1. PREREQUISITES	66
14.2. INSTALLING A CLIENT BY USING USER CREDENTIALS: INTERACTIVE INSTALLATION	66
14.3. INSTALLING A CLIENT BY USING A ONE-TIME PASSWORD: INTERACTIVE INSTALLATION	67
14.4. INSTALLING A CLIENT: NON-INTERACTIVE INSTALLATION	69
14.5. REMOVING PRE-IDM CONFIGURATION AFTER INSTALLING A CLIENT	71
14.6. TESTING AN IDM CLIENT	71
14.7. CONNECTIONS PERFORMED DURING AN IDM CLIENT INSTALLATION	71
14.8. IDM CLIENT'S COMMUNICATIONS WITH THE SERVER DURING POST-INSTALLATION DEPLOYMENT	72
14.9. SSSD COMMUNICATION PATTERNS	73
14.10. CERTMONGER COMMUNICATION PATTERNS	74
CHAPTER 15. INSTALLING AN IDM CLIENT WITH KICKSTART	76
15.1. INSTALLING A CLIENT WITH KICKSTART	76
15.2. KICKSTART FILE FOR CLIENT INSTALLATION	76
15.3. TESTING AN IDM CLIENT	77
CHAPTER 16. TROUBLESHOOTING IDM CLIENT INSTALLATION	78
16.1. REVIEWING IDM CLIENT INSTALLATION ERRORS	78
16.2. RESOLVING ISSUES IF THE CLIENT INSTALLATION FAILS TO UPDATE DNS RECORDS	78
16.3. RESOLVING ISSUES IF THE CLIENT INSTALLATION FAILS TO JOIN THE IDM KERBEROS REALM	79
16.4. ADDITIONAL RESOURCES	80
CHAPTER 17. RE-ENROLLING AN IDM CLIENT	81
17.1. CLIENT RE-ENROLLMENT IN IDM	81
17.2. RE-ENROLLING A CLIENT BY USING USER CREDENTIALS: INTERACTIVE RE-ENROLLMENT	81

17.3. RE-ENROLLING A CLIENT BY USING THE CLIENT KEYTAB: NON-INTERACTIVE RE-ENROLLMENT	82
17.4. TESTING AN IDM CLIENT	82
CHAPTER 18. UNINSTALLING AN IDM CLIENT	84
18.1. UNINSTALLING AN IDM CLIENT	84
18.2. UNINSTALLING AN IDM CLIENT: ADDITIONAL STEPS AFTER MULTIPLE PAST INSTALLATIONS	85
CHAPTER 19. RENAMING IDM CLIENT SYSTEMS	87
19.1. PREPARING AN IDM CLIENT FOR ITS RENAMING	87
19.2. UNINSTALLING AN IDM CLIENT	88
19.3. UNINSTALLING AN IDM CLIENT: ADDITIONAL STEPS AFTER MULTIPLE PAST INSTALLATIONS	89
19.4. RENAMING THE HOST SYSTEM	90
19.5. RE-INSTALLING AN IDM CLIENT	90
19.6. RE-ADDING SERVICES, RE-GENERATING CERTIFICATES, AND RE-ADDING HOST GROUPS	90
CHAPTER 20. PREPARING THE SYSTEM FOR IDM REPLICA INSTALLATION	91
20.1. REPLICA VERSION REQUIREMENTS	91
20.2. METHODS FOR DISPLAYING IDM SOFTWARE VERSION	91
20.3. AUTHORIZING THE INSTALLATION OF A REPLICA ON AN IDM CLIENT	92
20.4. AUTHORIZING THE INSTALLATION OF A REPLICA ON A SYSTEM THAT IS NOT ENROLLED INTO IDM	93
CHAPTER 21. INSTALLING AN IDM REPLICA	95
21.1. INSTALLING AN IDM REPLICA WITH INTEGRATED DNS AND A CA	95
21.2. INSTALLING AN IDM REPLICA WITH INTEGRATED DNS AND NO CA	97
21.3. INSTALLING AN IDM REPLICA WITHOUT INTEGRATED DNS AND WITH A CA	97
21.4. INSTALLING AN IDM REPLICA WITHOUT INTEGRATED DNS AND WITHOUT A CA	98
21.5. INSTALLING AN IDM HIDDEN REPLICA	99
21.6. TESTING AN IDM REPLICA	99
21.7. CONNECTIONS PERFORMED DURING AN IDM REPLICA INSTALLATION	100
CHAPTER 22. TROUBLESHOOTING IDM REPLICA INSTALLATION	101
22.1. IDM REPLICA INSTALLATION ERROR LOG FILES	101
22.2. REVIEWING IDM REPLICA INSTALLATION ERRORS	101
22.3. IDM CA INSTALLATION ERROR LOG FILES	103
22.4. REVIEWING IDM CA INSTALLATION ERRORS	104
22.5. REMOVING A PARTIAL IDM REPLICA INSTALLATION	104
22.6. RESOLVING INVALID CREDENTIAL ERRORS	105
22.7. ADDITIONAL RESOURCES	106
CHAPTER 23. UNINSTALLING AN IDM REPLICA	107
CHAPTER 24. INSTALLING DNS ON AN EXISTING IDM SERVER	108
CHAPTER 25. UNINSTALLING THE INTEGRATED IDM DNS SERVICE FROM AN IDM SERVER	110
CHAPTER 26. ADDING THE IDM CA SERVICE TO AN IDM SERVER IN A DEPLOYMENT WITHOUT A CA	111
26.1. INSTALLING THE FIRST IDM CA AS THE ROOT CA INTO AN EXISTING IDM DOMAIN	111
26.2. INSTALLING THE FIRST IDM CA WITH AN EXTERNAL CA AS THE ROOT CA INTO AN EXISTING IDM DOMAIN	111
CHAPTER 27. ADDING THE IDM CA SERVICE TO AN IDM SERVER IN A DEPLOYMENT WITH A CA	113
CHAPTER 28. UNINSTALLING THE IDM CA SERVICE FROM AN IDM SERVER	114
CHAPTER 29. MANAGING REPLICATION TOPOLOGY	115
29.1. EXPLAINING REPLICATION AGREEMENTS, TOPOLOGY SUFFIXES AND TOPOLOGY SEGMENTS	115

29.1.1. Replication agreements between IdM replicas	115
29.1.2. Topology suffixes	116
29.1.3. Topology segments	117
29.2. USING THE TOPOLOGY GRAPH TO MANAGE REPLICATION TOPOLOGY	118
29.3. SETTING UP REPLICATION BETWEEN TWO SERVERS USING THE WEB UI	120
29.4. STOPPING REPLICATION BETWEEN TWO SERVERS USING THE WEB UI	122
29.5. SETTING UP REPLICATION BETWEEN TWO SERVERS USING THE CLI	123
29.6. STOPPING REPLICATION BETWEEN TWO SERVERS USING THE CLI	124
29.7. REMOVING SERVER FROM TOPOLOGY USING THE WEB UI	125
29.8. REMOVING SERVER FROM TOPOLOGY USING THE CLI	126
29.9. VIEWING SERVER ROLES ON AN IDM SERVER USING THE WEB UI	127
29.10. VIEWING SERVER ROLES ON AN IDM SERVER USING THE CLI	127
29.11. PROMOTING A REPLICA TO A CA RENEWAL SERVER AND CRL PUBLISHER SERVER	128
29.12. DEMOTING OR PROMOTING HIDDEN REPLICAS	128
CHAPTER 30. INSTALLING AND RUNNING THE IDM HEALTHCHECK TOOL	130
30.1. HEALTHCHECK IN IDM	130
30.2. INSTALLING IDM HEALTHCHECK	131
30.3. RUNNING IDM HEALTHCHECK	131
30.4. ADDITIONAL RESOURCES	131
CHAPTER 31. INSTALLING AN IDENTITY MANAGEMENT SERVER USING AN ANSIBLE PLAYBOOK	133
31.1. ANSIBLE AND ITS ADVANTAGES FOR INSTALLING IDM	133
Advantages of using Ansible to install IdM	133
31.2. INSTALLING THE ANSIBLE-FREEIPA PACKAGE	133
31.3. ANSIBLE ROLES LOCATION IN THE FILE SYSTEM	134
31.4. SETTING THE PARAMETERS FOR A DEPLOYMENT WITH AN INTEGRATED DNS AND AN INTEGRATED CA AS THE ROOT CA	135
31.5. SETTING THE PARAMETERS FOR A DEPLOYMENT WITH EXTERNAL DNS AND AN INTEGRATED CA AS THE ROOT CA	138
31.6. DEPLOYING AN IDM SERVER WITH AN INTEGRATED CA AS THE ROOT CA USING AN ANSIBLE PLAYBOOK	140
31.7. SETTING THE PARAMETERS FOR A DEPLOYMENT WITH AN INTEGRATED DNS AND AN EXTERNAL CA AS THE ROOT CA	141
31.8. SETTING THE PARAMETERS FOR A DEPLOYMENT WITH EXTERNAL DNS AND AN EXTERNAL CA AS THE ROOT CA	145
31.9. DEPLOYING AN IDM SERVER WITH AN EXTERNAL CA AS THE ROOT CA USING AN ANSIBLE PLAYBOOK	148
31.10. UNINSTALLING AN IDM SERVER USING AN ANSIBLE PLAYBOOK	149
31.11. USING AN ANSIBLE PLAYBOOK TO UNINSTALL AN IDM SERVER EVEN IF THIS LEADS TO A DISCONNECTED TOPOLOGY	150
31.12. ADDITIONAL RESOURCES	152
CHAPTER 32. INSTALLING AN IDENTITY MANAGEMENT REPLICA USING AN ANSIBLE PLAYBOOK	153
32.1. SPECIFYING THE BASE, SERVER AND CLIENT VARIABLES FOR INSTALLING THE IDM REPLICA	153
32.2. SPECIFYING THE CREDENTIALS FOR INSTALLING THE IDM REPLICA USING AN ANSIBLE PLAYBOOK	157
32.3. DEPLOYING AN IDM REPLICA USING AN ANSIBLE PLAYBOOK	158
32.4. UNINSTALLING AN IDM REPLICA USING AN ANSIBLE PLAYBOOK	159
CHAPTER 33. INSTALLING AN IDENTITY MANAGEMENT CLIENT USING AN ANSIBLE PLAYBOOK	160
33.1. SETTING THE PARAMETERS OF THE INVENTORY FILE FOR THE AUTODISCOVERY CLIENT INSTALLATION MODE	160
33.2. SETTING THE PARAMETERS OF THE INVENTORY FILE WHEN AUTODISCOVERY IS NOT POSSIBLE DURING CLIENT INSTALLATION	162

33.3. CHECKING THE PARAMETERS IN THE INSTALL-CLIENT.YML FILE	165
33.4. AUTHORIZATION OPTIONS FOR IDM CLIENT ENROLLMENT USING AN ANSIBLE PLAYBOOK	165
33.5. DEPLOYING AN IDM CLIENT USING AN ANSIBLE PLAYBOOK	167
33.6. TESTING AN IDENTITY MANAGEMENT CLIENT AFTER ANSIBLE INSTALLATION	168
33.7. UNINSTALLING AN IDM CLIENT USING AN ANSIBLE PLAYBOOK	168
PART II. INTEGRATING IDM AND AD	170
CHAPTER 34. INSTALLING TRUST BETWEEN IDM AND AD	171
34.1. SUPPORTED VERSIONS OF WINDOWS SERVER	171
34.2. HOW THE TRUST WORKS	172
34.3. AD ADMINISTRATION RIGHTS	172
34.4. ENSURING SUPPORT FOR COMMON ENCRYPTION TYPES IN AD AND RHEL	173
34.4.1. Enabling AES encryption in AD (recommended)	173
34.4.2. Enabling the AES encryption type in Active Directory using a GPO	173
34.4.3. Enabling RC4 support in RHEL	174
34.4.4. Additional resources	175
34.5. PORTS REQUIRED FOR COMMUNICATION BETWEEN IDM AND AD	175
34.6. CONFIGURING DNS AND REALM SETTINGS FOR A TRUST	178
34.6.1. Unique primary DNS domains	179
34.6.2. Configuring a DNS forward zone in the IdM Web UI	180
34.6.3. Configuring a DNS forward zone in the CLI	183
34.6.4. Configuring DNS forwarding in AD	184
34.6.5. Verifying the DNS configuration	185
34.7. CONFIGURING IDM CLIENTS IN AN ACTIVE DIRECTORY DNS DOMAIN	186
34.7.1. Configuring an IdM client without Kerberos single sign-on	186
34.7.2. Requesting SSL certificates without single sign-on	187
34.7.3. Configuring an IdM client with Kerberos single sign-on	187
34.7.4. Requesting SSL certificates with single sign-on	188
34.8. SETTING UP A TRUST	188
34.8.1. Preparing the IdM server for the trust	189
34.8.2. Setting up a trust agreement using the command line	191
34.8.3. Setting up a trust agreement in the IdM Web UI	192
34.8.4. Setting up a trust agreement using Ansible	195
34.8.5. Verifying the Kerberos configuration	198
34.8.6. Verifying the trust configuration on IdM	198
34.8.7. Verifying the trust configuration on AD	199
34.8.8. Creating a trust agent	201
34.8.9. Enabling automatic private group mapping for a POSIX ID range on the CLI	201
34.8.10. Enabling automatic private group mapping for a POSIX ID range in the IdM WebUI	202
34.9. TROUBLESHOOTING SETTING UP A CROSS-FOREST TRUST	204
34.9.1. Sequence of events when establishing a cross-forest trust with AD	204
34.9.2. Checklist of prerequisites for establishing an AD trust	206
34.9.3. Gathering debug logs of an attempt to establish an AD trust	208
34.10. TROUBLESHOOTING CLIENT ACCESS TO SERVICES IN THE OTHER FOREST	209
34.10.1. Flow of information when a host in the AD forest root domain requests services from an IdM server	209
34.10.2. Flow of information when a host in an AD child domain requests services from an IdM server	210
34.10.3. Flow of information when an IdM client requests services from an AD server	212
34.11. REMOVING THE TRUST USING THE COMMAND LINE	213
34.12. REMOVING THE TRUST USING THE IDM WEB UI	213
34.13. REMOVING THE TRUST USING ANSIBLE	215
34.14. REMOVING AN ID RANGE AFTER REMOVING A TRUST TO AD	217

MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see [our CTO Chris Wright's message](#).

In Identity Management, planned terminology replacements include:

- ***block list*** replaces *blacklist*
- ***allow list*** replaces *whitelist*
- ***secondary*** replaces *slave*
- The word *master* is being replaced with more precise language, depending on the context:
 - ***IdM server*** replaces *IdM master*
 - ***CA renewal server*** replaces *CA renewal master*
 - ***CRL publisher server*** replaces *CRL master*
 - ***multi-supplier*** replaces *multi-master*

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your feedback on our documentation. Let us know how we can improve it.

Submitting feedback through Jira (account required)

1. Log in to the [Jira](#) website.
2. Click **Create** in the top navigation bar.
3. Enter a descriptive title in the **Summary** field.
4. Enter your suggestion for improvement in the **Description** field. Include links to the relevant parts of the documentation.
5. Click **Create** at the bottom of the dialogue.

CHAPTER 1. HOW TO USE THIS GUIDE

An Identity Management (IdM) domain includes IdM servers, also called replicas, and IdM clients. While [installing an IdM deployment](#) always starts with installing the primary IdM server, the order of the next installation steps depends on the targeted topology. For example, you can install an IdM replica before or after installing an IdM client. Additionally, certain IdM deployments require [a trust with Active Directory](#), while others do not.

Additional resources

- [Planning Identity Management](#)

PART I. INSTALLING IDENTITY MANAGEMENT

CHAPTER 2. PREPARING THE SYSTEM FOR IDM SERVER INSTALLATION

The following sections list the requirements to install an Identity Management (IdM) server. Before the installation, verify your system meets these requirements.

2.1. PREREQUISITES

- You need **root** privileges to install an Identity Management (IdM) server on your host.

2.2. HARDWARE RECOMMENDATIONS

RAM is the most important hardware feature to size properly. Make sure your system has enough RAM available. Typical RAM requirements are:

- For 10,000 users and 100 groups: at least 4 GB of RAM and 4 GB swap space
- For 100,000 users and 50,000 groups: at least 16 GB of RAM and 4 GB of swap space

For larger deployments, it is more effective to increase the RAM than to increase disk space because much of the data is stored in cache. In general, adding more RAM leads to better performance for larger deployments due to caching.



NOTE

A basic user entry or a simple host entry with a certificate is approximately 5–10 kB in size.

2.3. CUSTOM CONFIGURATION REQUIREMENTS FOR IDM

Install an Identity Management (IdM) server on a clean system without any custom configuration for services such as DNS, Kerberos, Apache, or Directory Server.

The IdM server installation overwrites system files to set up the IdM domain. IdM backs up the original system files to **/var/lib/ipa/sysrestore/**. When an IdM server is uninstalled at the end of the lifecycle, these files are restored.

IPv6 requirements in IdM

The IdM system must have the IPv6 protocol enabled in the kernel and localhost (::1) is able to use it. If IPv6 is disabled, then the CLDAP plug-in used by the IdM services fails to initialize.



NOTE

IPv6 does not have to be enabled on the network. It is possible to enable IPv6 stack without enabling IPv6 addresses if required.

Support for encryption types in IdM

Red Hat Enterprise Linux (RHEL) uses Version 5 of the Kerberos protocol, which supports encryption types such as Advanced Encryption Standard (AES), Camellia, and Data Encryption Standard (DES).

List of supported encryption types

While the Kerberos libraries on IdM servers and clients might support more encryption types, the IdM Kerberos Distribution Center (KDC) only supports the following encryption types:

- **aes256-cts:normal**
- **aes256-cts:special** (default)
- **aes128-cts:normal**
- **aes128-cts:special** (default)
- **aes128-sha2:normal**
- **aes128-sha2:special**
- **aes256-sha2:normal**
- **aes256-sha2:special**
- **camellia128-cts-cmac:normal**
- **camellia128-cts-cmac:special**
- **camellia256-cts-cmac:normal**
- **camellia256-cts-cmac:special**

RC4 encryption types are disabled by default

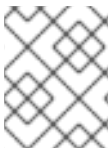
The following RC4 encryption types have been deprecated and disabled by default in RHEL 8, as they are considered less secure than the newer AES-128 and AES-256 encryption types:

- **arcfour-hmac:normal**
- **arcfour-hmac:special**

For more information about manually enabling RC4 support for compatibility with legacy Active Directory environments, see [Ensuring support for common encryption types in AD and RHEL](#) .

Support for DES and 3DES encryption has been removed

Due to security reasons, support for the DES algorithm was deprecated in RHEL 7. The recent rebase of Kerberos packages in RHEL 8.3.0 removes support for single-DES (DES) and triple-DES (3DES) encryption types from RHEL 8.



NOTE

Standard RHEL 8 IdM installations do not use DES or 3DES encryption types by default and are unaffected by the Kerberos upgrade.

If you manually configured any services or users to **only** use DES or 3DES encryption (for example, for legacy clients), you might experience service interruptions after updating to the latest Kerberos packages, such as:

- Kerberos authentication errors
- **unknown enctype** encryption errors

- KDCs with DES-encrypted Database Master Keys (**K/M**) fail to start

Red Hat recommends you do not use DES or 3DES encryption in your environment.



NOTE

You only need to disable DES and 3DES encryption types if you configured your environment to use them.

Support for system-wide cryptographic policies in IdM

IdM uses the **DEFAULT** system-wide cryptographic policy. This policy offers secure settings for current threat models. It allows the TLS 1.2 and 1.3 protocols, as well as the IKEv2 and SSH2 protocols. The RSA keys and Diffie-Hellman parameters are accepted if they are at least 2048 bits long. This policy does not allow DES, 3DES, RC4, DSA, TLS v1.0, and other weaker algorithms.



NOTE

You cannot install an IdM server while using the **FUTURE** system-wide cryptographic policy. When installing an IdM server, ensure you are using the **DEFAULT** system-wide cryptographic policy.

Additional Resources

- [System-wide cryptographic policies](#)
- `man IPV6(7)`

2.4. FIPS COMPLIANCE

With RHEL 8.3.0 or later, you can install a new IdM server or replica on a system with the Federal Information Processing Standard (FIPS) 140 mode enabled.

To install IdM in FIPS mode, first enable FIPS mode on the host, then install IdM. The IdM installation script detects if FIPS is enabled and configures IdM to only use encryption types that are compliant with the FIPS 140 standard:

- **aes256-cts:normal**
- **aes256-cts:special**
- **aes128-cts:normal**
- **aes128-cts:special**
- **aes128-sha2:normal**
- **aes128-sha2:special**
- **aes256-sha2:normal**
- **aes256-sha2:special**

For an IdM environment to be FIPS-compliant, **all** IdM replicas must have FIPS mode enabled.

Red Hat recommends that you enable FIPS mode in IdM clients as well, especially if you might promote those clients to IdM replicas. Ultimately, it is up to administrators to determine how they meet FIPS requirements; Red Hat does not enforce FIPS criteria.

Migration to FIPS-compliant IdM

You cannot migrate an existing IdM installation from a non-FIPS environment to a FIPS-compliant installation. This is not a technical problem but a legal and regulatory restriction.

To operate a FIPS-compliant system, all cryptographic key material must be created in FIPS mode. Furthermore, the cryptographic key material must never leave the FIPS environment unless it is securely wrapped and never unwrapped in non-FIPS environments.

If your scenario requires a migration of a non-FIPS IdM realm to a FIPS-compliant one, you must:

1. create a new IdM realm in FIPS mode
2. perform data migration from the non-FIPS realm to the new FIPS-mode realm with a filter that blocks all key material

The migration filter must block:

- KDC master key, keytabs, and all related Kerberos key material
- User passwords
- All certificates including CA, service, and user certificates
- OTP tokens
- SSH keys and fingerprints
- DNSSEC KSK and ZSK
- All vault entries
- AD trust-related key material

Effectively, the new FIPS installation is a different installation. Even with rigorous filtering, such a migration may not pass a FIPS 140 certification. Your FIPS auditor may flag this migration.

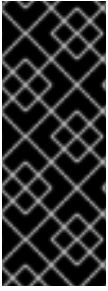
Additional Resources

- For more information about the FIPS 140 implementation in the RHEL operating system, see [Federal Information Processing Standards 140 and FIPS mode](#) in the *RHEL Security Hardening* document.

2.5. SUPPORT FOR CROSS-FOREST TRUST WITH FIPS MODE ENABLED

To establish a cross-forest trust with an Active Directory (AD) domain while FIPS mode is enabled, you must meet the following requirements:

- IdM servers are on RHEL 8.4.0 or later.
- You must authenticate with an AD administrative account when setting up a trust. You cannot establish a trust using a shared secret while FIPS mode is enabled.



IMPORTANT

RADIUS authentication is not FIPS-compliant as the RADIUS protocol uses the MD5 hash function to encrypt passwords between client and server and, in FIPS mode, OpenSSL disables the use of the MD5 digest algorithm. However, if the RADIUS server is running on the same host as the IdM server, you can work around the problem and enable MD5 by performing the steps described in [How to configure FreeRADIUS authentication in FIPS mode](#).

Additional Resources

- For more information about FIPS mode in the RHEL operating system, see [Installing the system in FIPS mode](#) in the *Security Hardening* document.
- For more details about the FIPS 140-2 standard, see the [Security Requirements for Cryptographic Modules](#) on the National Institute of Standards and Technology (NIST) web site.

2.6. TIME SERVICE REQUIREMENTS FOR IDM

The following sections discuss using **chronyd** to keep your IdM hosts in sync with a central time source:

2.6.1. How IdM uses chronyd for synchronization

You can use **chronyd** to keep your IdM hosts in sync with a central time source as described here.

Kerberos, the underlying authentication mechanism in IdM, uses time stamps as part of its protocol. Kerberos authentication fails if the system time of an IdM client differs by more than five minutes from the system time of the Key Distribution Center (KDC).

To ensure that IdM servers and clients stay in sync with a central time source, IdM installation scripts automatically configure **chronyd** Network Time Protocol (NTP) client software.

If you do not pass any NTP options to the IdM installation command, the installer searches for **_ntp._udp** DNS service (SRV) records that point to the NTP server in your network and configures **chrony** with that IP address. If you do not have any **_ntp._udp** SRV records, **chronyd** uses the configuration shipped with the **chrony** package.



NOTE

Because **ntpd** has been deprecated in favor of **chronyd** in RHEL 8, IdM servers are no longer configured as Network Time Protocol (NTP) servers and are only configured as NTP clients. The RHEL 7 **NTP Server** IdM server role has also been deprecated in RHEL 8.

Additional resources

- [Implementation of NTP](#)
- [Using the Chrony suite to configure NTP](#)

2.6.2. List of NTP configuration options for IdM installation commands

You can use **chronyd** to keep your IdM hosts in sync with a central time source.

You can specify the following options with any of the IdM installation commands (**ipa-server-install**, **ipa-replica-install**, **ipa-client-install**) to configure **chronyd** client software during setup.

Table 2.1. List of NTP configuration options for IdM installation commands

Option	Behavior
--ntp-server	Use it to specify one NTP server. You can use it multiple times to specify multiple servers.
--ntp-pool	Use it to specify a pool of multiple NTP servers resolved as one hostname.
-N, --no-ntp	Do not configure, start, or enable chronyd .

Additional resources

- [Implementation of NTP](#)
- [Using the Chrony suite to configure NTP](#)

2.6.3. Ensuring IdM can reference your NTP time server

This procedure verifies you have the necessary configurations in place for IdM to be able to synchronize with your Network Time Protocol (NTP) time server.

Prerequisites

- You have configured an NTP time server in your environment. In this example, the hostname of the previously configured time server is **ntpserver.example.com**.

Procedure

1. Perform a DNS service (SRV) record search for NTP servers in your environment.

```
[user@server ~]$ dig +short -t SRV _ntp._udp.example.com
0 100 123 ntpserver.example.com.
```

2. If the previous **dig** search does not return your time server, add a **_ntp._udp** SRV record that points to your time server on port **123**. This process depends on your DNS solution.

Verification steps

- Verify that DNS returns an entry for your time server on port **123** when you perform a search for **_ntp._udp** SRV records.

```
[user@server ~]$ dig +short -t SRV _ntp._udp.example.com
0 100 123 ntpserver.example.com.
```

Additional resources

- [Implementation of NTP](#)

- [Using the Chrony suite to configure NTP](#)

2.6.4. Additional resources

- [Implementation of NTP](#)
- [Using the Chrony suite to configure NTP](#)

2.7. HOST NAME AND DNS REQUIREMENTS FOR IDM

The host name and DNS requirements for server and replica systems are outlined below and also how to verify that the systems meet the requirements.

These requirements apply to all Identity Management (IdM) servers, those with integrated DNS and those without integrated DNS.



WARNING

DNS records are vital for nearly all IdM domain functions, including running LDAP directory services, Kerberos, and Active Directory integration. Be extremely cautious and ensure that:

- You have a tested and functional DNS service available
- The service is properly configured

This requirement applies to IdM servers with **and** without integrated DNS.

Verify the server host name

The host name must be a fully qualified domain name, such as ***server.idm.example.com***.



IMPORTANT

Do not use single-label domain names, for example **.company**: the IdM domain must be composed of one or more subdomains and a top level domain, for example **example.com** or **company.example.com**.

The fully qualified domain name must meet the following conditions:

- It is a valid DNS name, which means only numbers, alphabetic characters, and hyphens (-) are allowed. Other characters, such as underscores (_), in the host name cause DNS failures.
- It is all lower-case. No capital letters are allowed.
- It does not resolve to the loopback address. It must resolve to the system's public IP address, not to **127.0.0.1**.

To verify the host name, use the **hostname** utility on the system where you want to install:

```
# hostname
server.idm.example.com
```

The output of **hostname** must not be **localhost** or **localhost6**.

Verify the forward and reverse DNS configuration

1. Obtain the IP address of the server.
 - a. The **ip addr show** command displays both the IPv4 and IPv6 addresses. In the following example, the relevant IPv6 address is **2001:DB8::1111** because its scope is global:

```
[root@server ~]# ip addr show
...
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
state UP group default qlen 1000
link/ether 00:1a:4a:10:4e:33 brd ff:ff:ff:ff:ff:ff
inet 192.0.2.1/24 brd 192.0.2.255 scope global dynamic eth0
    valid_lft 106694sec preferred_lft 106694sec
inet6 2001:DB8::1111/32 scope global dynamic
    valid_lft 2591521sec preferred_lft 604321sec
inet6 fe80::56ee:75ff:fe2b:def6/64 scope link
    valid_lft forever preferred_lft forever
...
```

2. Verify the forward DNS configuration using the **dig** utility.
 - a. Run the command **dig +short server.idm.example.com A**. The returned IPv4 address must match the IP address returned by **ip addr show**:

```
[root@server ~]# dig +short server.idm.example.com A
192.0.2.1
```

- b. Run the command **dig +short server.idm.example.com AAAA**. If it returns an address, it must match the IPv6 address returned by **ip addr show**:

```
[root@server ~]# dig +short server.idm.example.com AAAA
2001:DB8::1111
```



NOTE

If **dig** does not return any output for the AAAA record, it does not indicate incorrect configuration. No output only means that no IPv6 address is configured in DNS for the system. If you do not intend to use the IPv6 protocol in your network, you can proceed with the installation in this situation.

3. Verify the reverse DNS configuration (PTR records). Use the **dig** utility and add the IP address.

If the commands below display a different host name or no host name, the reverse DNS configuration is incorrect.

 - a. Run the command **dig +short -x IPv4_address**. The output must display the server host name. For example:

```
[root@server ~]# dig +short -x 192.0.2.1
server.idm.example.com
```

- b. If the command **dig +short -x server.idm.example.com AAAA** in the previous step returned an IPv6 address, use **dig** to query the IPv6 address too. The output must display the server host name. For example:

```
[root@server ~]# dig +short -x 2001:DB8::1111
server.idm.example.com
```



NOTE

If **dig +short server.idm.example.com AAAA** in the previous step did not display any IPv6 address, querying the AAAA record does not output anything. In this case, this is normal behavior and does not indicate incorrect configuration.



WARNING

If a reverse DNS (PTR record) search returns multiple host names, **httpd** and other software associated with IdM may show unpredictable behavior. Red Hat strongly recommends configuring only one PTR record per IP.

Verify the standards-compliance of DNS forwarders (required for integrated DNS only)

Ensure that all DNS forwarders you want to use with the IdM DNS server comply with the Extension Mechanisms for DNS (EDNS0) and DNS Security Extensions (DNSSEC) standards. To do this, inspect the output of the following command for each forwarder separately:

```
$ dig +dnssec @IP_address_of_the_DNS_forwarder . SOA
```

The expected output displayed by the command contains the following information:

- Status: **NOERROR**
- Flags: **ra**
- EDNS flags: **do**
- The **RRSIG** record must be present in the **ANSWER** section

If any of these items is missing from the output, inspect the documentation for your DNS forwarder and verify that EDNS0 and DNSSEC are supported and enabled. In the latest versions of the BIND server, the **dnssec-enable yes**; option must be set in the **/etc/named.conf** file.

Example of the expected output produced by **dig**:

```
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 48655
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1
```

```

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096

;; ANSWER SECTION:
. 31679 IN SOA a.root-servers.net. nstld.verisign-grs.com. 2015100701 1800 900 604800 86400
. 31679 IN RRSIG SOA 8 0 86400 20151017170000 20151007160000 62530 . GNVz7SQs [...]

```

Verify the `/etc/hosts` file

Verify that the `/etc/hosts` file fulfills one of the following conditions:

- The file does not contain an entry for the host. It only lists the IPv4 and IPv6 localhost entries for the host.
- The file contains an entry for the host and the file fulfills all the following conditions:
 - The first two entries are the IPv4 and IPv6 localhost entries.
 - The next entry specifies the IdM server IPv4 address and host name.
 - The **FQDN** of the IdM server comes before the short name of the IdM server.
 - The IdM server host name is not part of the localhost entry.

The following is an example of a correctly configured `/etc/hosts` file:

```

127.0.0.1 localhost localhost.localdomain \
localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain \
localhost6 localhost6.localdomain6
192.0.2.1 server.idm.example.com server
2001:DB8::1111 server.idm.example.com server

```

2.8. PORT REQUIREMENTS FOR IDM

Identity Management (IdM) uses several [ports](#) to communicate with its services. These ports must be open and available for incoming connections to the IdM server for IdM to work. They must not be currently used by another service or blocked by a [firewall](#).

Table 2.2. IdM ports

Service	Ports	Protocol
HTTP/HTTPS	80, 443	TCP
LDAP/LDAPS	389, 636	TCP
Kerberos	88, 464	TCP and UDP
DNS	53	TCP and UDP (optional)



NOTE

IdM uses ports 80 and 389. This is a secure practice because of the following safeguards:

- IdM normally redirects requests that arrive on port 80 to port 443. Port 80 (HTTP) is only used to provide Online Certificate Status Protocol (OCSP) responses and Certificate Revocation Lists (CRL). Both are digitally signed and therefore secured against man-in-the-middle attacks.
- Port 389 (LDAP) uses STARTTLS and Generic Security Services API (GSSAPI) for encryption.

In addition, ports 8080, 8443, and 749 must be free as they are used internally. Do not open these ports and instead leave them blocked by a firewall.

Table 2.3. firewalld services

Service name	For details, see:
freeipa-ldap	/usr/lib/firewalld/services/freeipa-ldap.xml
freeipa-ldaps	/usr/lib/firewalld/services/freeipa-ldaps.xml
dns	/usr/lib/firewalld/services/dns.xml

2.9. OPENING THE PORTS REQUIRED BY IDM

Procedure

1. Verify that the **firewalld** service is running.

- To find out if **firewalld** is currently running:

```
# systemctl status firewalld.service
```

- To start **firewalld** and configure it to start automatically when the system boots:

```
# systemctl start firewalld.service
# systemctl enable firewalld.service
```

2. Open the required ports using the **firewall-cmd** utility. Choose one of the following options:

- a. Add the individual ports to the firewall by using the **firewall-cmd --add-port** command. For example, to open the ports in the default zone:

```
# firewall-cmd --permanent --add-port={80/tcp,443/tcp,389/tcp,636/tcp,88/tcp,88/udp,464/tcp,464/udp,53/tcp,53/udp}
```

- b. Add the **firewalld** services to the firewall by using the **firewall-cmd --add-service** command. For example, to open the ports in the default zone:

```
# firewall-cmd --permanent --add-service={freeipa-4,dns}
```

For details on using **firewall-cmd** to open ports on a system, see the **firewall-cmd(1)** man page.

3. Reload the **firewall-cmd** configuration to ensure that the change takes place immediately:

```
# firewall-cmd --reload
```

Note that reloading **firewalld** on a system in production can cause DNS connection time outs. If required, to avoid the risk of time outs and to make the changes persistent on the running system, use the **--runtime-to-permanent** option of the **firewall-cmd** command, for example:

```
# firewall-cmd --runtime-to-permanent
```

4. **Optional.** To verify that the ports are available now, use the **nc**, **telnet**, or **nmap** utilities to connect to a port or run a port scan.



NOTE

Note that you also have to open network-based firewalls for both incoming and outgoing traffic.

2.10. INSTALLING PACKAGES REQUIRED FOR AN IDM SERVER

In RHEL8, the packages necessary for installing an Identity Management (IdM) server are shipped as a module. The IdM server module stream is called the **DL1** stream, and you need to enable this stream before downloading packages from this stream. The following procedure shows how to download the packages necessary for setting up the IdM environment of your choice.

Prerequisites

- You have a newly installed RHEL system.
- You have made the required repositories available:
 - If your RHEL system is not running in the cloud, you have registered your system with the Red Hat Subscription Manager (RHSM). For details, see [Registration, attaching, and removing subscriptions in the Subscription Manager command line](#). You have also enabled the **BaseOS** and **AppStream** repositories that IdM uses:

```
# subscription-manager repos --enable=rhel-8-for-x86_64-baseos-rpms
# subscription-manager repos --enable=rhel-8-for-x86_64-appstream-rpms
```

For details on how to enable and disable specific repositories using RHSM, see [Configuring options in Red Hat Subscription Manager](#).

- If your RHEL system is running in the cloud, skip the registration. The required repositories are already available via the Red Hat Update Infrastructure (RHUI).
- You have not previously enabled an IdM module stream.

Procedure

1. Enable the **idm:DL1** stream:

```
# yum module enable idm:DL1
```

- Switch to the RPMs delivered through the **idm:DL1** stream:

```
# yum distro-sync
```

- Choose one of the following options, depending on your IdM requirements:

- To download the packages necessary for installing an IdM server without an integrated DNS:

```
# yum module install idm:DL1/server
```

- To download the packages necessary for installing an IdM server with an integrated DNS:

```
# yum module install idm:DL1/dns
```

- To download the packages necessary for installing an IdM server that has a trust agreement with Active Directory:

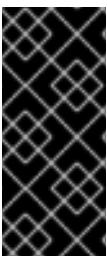
```
# yum module install idm:DL1/adtrust
```

- To download the packages from multiple profiles, for example the **adtrust** and **dns** profiles:

```
# yum module install idm:DL1/{dns,adtrust}
```

- To download the packages necessary for installing an IdM client:

```
# yum module install idm:DL1/client
```



IMPORTANT

When switching to a new module stream once you have already enabled a different stream and downloaded packages from it, you need to first explicitly remove all the relevant installed content and disable the current module stream before enabling the new module stream. Trying to enable a new stream without disabling the current one results in an error. For details on how to proceed, see [Switching to a later stream](#).



WARNING

While it is possible to install packages from modules individually, be aware that if you install any package from a module that is not listed as "API" for that module, it is only going to be supported by Red Hat in the context of that module. For example, if you install **bind-dyndb-ldap** directly from the repository to use with your custom 389 Directory Server setup, any problems that you have will be ignored unless they occur for IdM, too.

2.11. SETTING THE CORRECT FILE MODE CREATION MASK FOR IDM INSTALLATION

The Identity Management (IdM) installation process requires that the file mode creation mask (**umask**) is set to **0022** for the **root** account. This allows users other than **root** to read files created during the installation. If a different **umask** is set, the installation of an IdM server will display a warning. If you continue with the installation, some functions of the server will not perform properly. For example, you will be unable to install an IdM replica from this server. After the installation, you can set the **umask** back to its original value.

Prerequisites

- You have **root** privileges.

Procedure

1. (Optional) Display the current **umask**:

```
# umask
0027
```

2. Set the **umask** to **0022**:

```
# umask 0022
```

3. (Optional) After the IdM installation is complete, set the **umask** back to its original value:

```
# umask 0027
```

2.12. ENSURING THAT FAPOLICYD RULES DO NOT BLOCK IDM INSTALLATION AND OPERATION

If you are using the **fapolicyd** software framework on your RHEL host to control the execution of applications based on a user-defined policy, the installation of the Identity Management (IdM) server can fail. As the installation and operation requires the Java program to complete successfully, ensure that Java and Java classes are not blocked by any **fapolicyd** rules.

For more information, see the [fapolicy restrictions causing IdM installation failures](#) KCS solution.

2.13. OPTIONS FOR THE IDM INSTALLATION COMMANDS

Commands such as **ipa-server-install**, **ipa-replica-install**, **ipa-dns-install** and **ipa-ca-install** have numerous options you can use to supply additional information for an interactive installation. You can also use these options to script an unattended installation.

The following tables display some of the most common options for different components. Options for a specific component are shared across multiple commands. For example, you can use the **--ca-subject** option with both the **ipa-ca-install** and **ipa-server-install** commands.

For an exhaustive list of options, see the **ipa-server-install(1)**, **ipa-replica-install(1)**, **ipa-dns-install(1)** and **ipa-ca-install(1)** man pages.

Table 2.4. General options: available for **ipa-server-install** and **ipa-replica-install**

Argument	Description
-d, --debug	Enables debug logging for more verbose output.
-U, --unattended	Enables an unattended installation session that does not prompt for user input.
--hostname=<i>server.idm.example.com</i>	The fully-qualified domain name of the IdM server machine. Only numbers, lowercase alphabetic characters, and hyphens (-) are allowed.
--ip-address <i>127.0.0.1</i>	Specifies the IP address of the server. This option only accepts IP addresses associated with the local interface.
--dirsrv-config-file <LDIF_file_name>	The path to an LDIF file used to modify the configuration of the directory server instance.
-n <i>example.com</i>	The name of the LDAP server domain to use for the IdM domain. This is usually based on the IdM server's hostname.
-p <directory_manager_password>	The password of the superuser, cn=Directory Manager , for the LDAP service.
-a <ipa_admin_password>	The password for the admin IdM administrator account to authenticate to the Kerberos realm. For ipa-replica-install , use -w instead.
-r <KERBEROS_REALM_NAME>	The name of the Kerberos realm to create for the IdM domain in uppercase, such as EXAMPLE.COM . For ipa-replica-install , this specifies the name of a Kerberos realm of an existing IdM deployment.
--setup-dns	Tells the installation script to set up a DNS service within the IdM domain.
--setup-ca	Install and configure a CA on this replica. If a CA is not configured, certificate operations are forwarded to another replica with a CA installed. For ipa-server-install , a CA is installed by default and you do not need to use this option.

Table 2.5. CA options: available for **ipa-ca-install** and **ipa-server-install**

Argument	Description
--ca-subject=<SUBJECT>	Specifies the CA certificate subject Distinguished Name (default: CN=Certificate Authority,O=REALM.NAME). Relative Distinguished Names (RDN) are in LDAP order, with the most specific RDN first.

Argument	Description
--subject-base=<SUBJECT>	Specifies the subject base for certificates issued by IdM (default O=REALM.NAME). Relative Distinguished Names (RDN) are in LDAP order, with the most specific RDN first.
--external-ca	Generates a certificate signing request to be signed by an external CA.
--ca-signing-algorithm=<ALGORITHM>	Specifies the signing algorithm of the IdM CA certificate. Possible values are SHA1withRSA, SHA256withRSA, SHA512withRSA. The default is SHA256withRSA. Use this option with --external-ca if the external CA does not support the default signing algorithm.

Table 2.6. DNS options: available for **ipa-dns-install**, or for **ipa-server-install** and **ipa-replica-install** when using **--setup-dns**

Argument	Description
--forwarder=192.0.2.1	Specifies a DNS forwarder to use with the DNS service. To specify more than one forwarder, use this option multiple times.
--no-forwarders	Uses root servers with the DNS service instead of forwarders.
--no-reverse	Does not create a reverse DNS zone when the DNS domain is set up. If a reverse DNS zone is already configured, then that existing reverse DNS zone is used. If this option is not used, then the default value is true . This instructs the installation script to configure reverse DNS.

Additional resources

- **ipa-server-install(1)** man page
- **ipa-replica-install(1)** man page
- **ipa-dns-install(1)** man page
- **ipa-ca-install(1)** man page

CHAPTER 3. INSTALLING AN IDM SERVER: WITH INTEGRATED DNS, WITH AN INTEGRATED CA AS THE ROOT CA

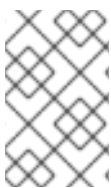
Installing a new Identity Management (IdM) server with integrated DNS has the following advantages:

- You can automate much of the maintenance and DNS record management using native IdM tools. For example, DNS SRV records are automatically created during the setup, and later on are automatically updated.
- You can configure global forwarders during the installation of the IdM server for a stable external internet connection. Global forwarders are also useful for trusts with Active Directory.
- You can set up a DNS reverse zone to prevent emails from your domain to be considered spam by email servers outside of the IdM domain.

Installing IdM with integrated DNS has certain limitations:

- IdM DNS is not meant to be used as a general-purpose DNS server. Some of the advanced DNS functions are not supported. For more information, see [DNS services available in an IdM server](#).

This chapter describes how you can install a new IdM server with an integrated certificate authority (CA) as the root CA.



NOTE

The default configuration for the `ipa-server-install` command is an integrated CA as the root CA. If no CA option, for example `--external-ca` or `--ca-less` is specified, the IdM server is installed with an integrated CA.

3.1. INTERACTIVE INSTALLATION

During the interactive installation using the `ipa-server-install` utility, you are asked to supply basic configuration of the system, for example the realm, the administrator's password and the Directory Manager's password.

The `ipa-server-install` installation script creates a log file at `/var/log/ipaserver-install.log`. If the installation fails, the log can help you identify the problem.

Procedure

1. Run the `ipa-server-install` utility.

```
# ipa-server-install
```

2. The script prompts to configure an integrated DNS service. Enter **yes**.

```
Do you want to configure integrated DNS (BIND)? [no]: yes
```

3. The script prompts for several required settings and offers recommended default values in brackets.
 - To accept a default value, press **Enter**.
 - To provide a custom value, enter the required value.

Server host name [server.idm.example.com]:
 Please confirm the domain name [idm.example.com]:
 Please provide a realm name [IDM.EXAMPLE.COM]:

**WARNING**

Plan these names carefully. You will not be able to change them after the installation is complete.

4. Enter the passwords for the Directory Server superuser (**cn=Directory Manager**) and for the Identity Management (IdM) administration system user account (**admin**).

Directory Manager password:
 IPA admin password:

5. The script prompts for per-server DNS forwarders.

Do you want to configure DNS forwarders? [yes]:

- To configure per-server DNS forwarders, enter **yes**, and then follow the instructions on the command line. The installation process will add the forwarder IP addresses to the IdM LDAP.
 - For the forwarding policy default settings, see the **--forward-policy** description in the **ipa-dns-install(1)** man page.
 - If you do not want to use DNS forwarding, enter **no**.
 With no DNS forwarders, hosts in your IdM domain will not be able to resolve names from other, internal, DNS domains in your infrastructure. The hosts will only be left with public DNS servers to resolve their DNS queries.
6. The script prompts to check if any DNS reverse (PTR) records for the IP addresses associated with the server need to be configured.

Do you want to search for missing reverse zones? [yes]:

If you run the search and missing reverse zones are discovered, the script asks you whether to create the reverse zones along with the PTR records.

Do you want to create reverse zone for IP 192.0.2.1 [yes]:
 Please specify the reverse zone name [2.0.192.in-addr.arpa.]:
 Using reverse zone(s) 2.0.192.in-addr.arpa.

**NOTE**

Using IdM to manage reverse zones is optional. You can use an external DNS service for this purpose instead.

7. Enter **yes** to confirm the server configuration.

Continue to configure the system with these values? [no]: yes

8. The installation script now configures the server. Wait for the operation to complete.
9. After the installation script completes, update your DNS records in the following way:
 - a. Add DNS delegation from the parent domain to the IdM DNS domain. For example, if the IdM DNS domain is **idm.example.com**, add a name server (NS) record to the **example.com** parent domain.



IMPORTANT

Repeat this step each time after an IdM DNS server is installed.

- b. Add an **_ntp._udp** service (SRV) record for your time server to your IdM DNS. The presence of the SRV record for the time server of the newly-installed IdM server in IdM DNS ensures that future replica and client installations are automatically configured to synchronize with the time server used by this primary IdM server.

3.2. NON-INTERACTIVE INSTALLATION

The **ipa-server-install** installation script creates a log file at **/var/log/ipaserver-install.log**. If the installation fails, the log can help you identify the problem.

Procedure

1. Run the **ipa-server-install** utility with the options to supply all the required information. The minimum required options for non-interactive installation are:
 - **--realm** to provide the Kerberos realm name
 - **--ds-password** to provide the password for the Directory Manager (DM), the Directory Server super user
 - **--admin-password** to provide the password for **admin**, the Identity Management (IdM) administrator
 - **--unattended** to let the installation process select default options for the host name and domain name

To install a server with integrated DNS, add also these options:

- **--setup-dns** to configure integrated DNS
- **--forwarder** or **--no-forwarders**, depending on whether you want to configure DNS forwarders or not
- **--auto-reverse** or **--no-reverse**, depending on whether you want to configure automatic detection of the reverse DNS zones that must be created in the IdM DNS or no reverse zone auto-detection

For example:

```
# ipa-server-install --realm IDM.EXAMPLE.COM --ds-password DM_password --admin-  
password admin_password --unattended --setup-dns --forwarder 192.0.2.1 --no-  
reverse
```

2. After the installation script completes, update your DNS records in the following way:
 - a. Add DNS delegation from the parent domain to the IdM DNS domain. For example, if the IdM DNS domain is *idm.example.com*, add a name server (NS) record to the **example.com** parent domain.



IMPORTANT

Repeat this step each time after an IdM DNS server is installed.

- b. Add an **_ntp._udp** service (SRV) record for your time server to your IdM DNS. The presence of the SRV record for the time server of the newly-installed IdM server in IdM DNS ensures that future replica and client installations are automatically configured to synchronize with the time server used by this primary IdM server.

Additional resources

- For a complete list of options accepted by **ipa-server-install**, run the **ipa-server-install --help** command.

CHAPTER 4. INSTALLING AN IDM SERVER: WITH INTEGRATED DNS, WITH AN EXTERNAL CA AS THE ROOT CA

Installing a new Identity Management (IdM) server with integrated DNS has the following advantages:

- You can automate much of the maintenance and DNS record management using native IdM tools. For example, DNS SRV records are automatically created during the setup, and later on are automatically updated.
- You can configure global forwarders during the installation of the IdM server for a stable external internet connection. Global forwarders are also useful for trusts with Active Directory.
- You can set up a DNS reverse zone to prevent emails from your domain to be considered spam by email servers outside of the IdM domain.

Installing IdM with integrated DNS has certain limitations:

- IdM DNS is not meant to be used as a general-purpose DNS server. Some of the advanced DNS functions are not supported. For more information, see [DNS services available in an IdM server](#) .

This chapter describes how you can install a new IdM server with an external certificate authority (CA) as the root CA.

4.1. INTERACTIVE INSTALLATION

During the interactive installation using the **ipa-server-install** utility, you are asked to supply basic configuration of the system, for example the realm, the administrator's password and the Directory Manager's password.

The **ipa-server-install** installation script creates a log file at `/var/log/ipaserver-install.log`. If the installation fails, the log can help you identify the problem.

Follow this procedure to install a server:

- With integrated DNS
- With an external certificate authority (CA) as the root CA

Prerequisites

- You have determined the type of the external CA to specify with the **--external-ca-type** option. See the **ipa-server-install(1)** man page for details.
- If you are using a Microsoft Certificate Services certificate authority (MS CS CA) as your external CA: you have determined the certificate profile or template to specify with the **--external-ca-profile** option. By default, the **SubCA** template is used. For more information about the **--external-ca-type** and **--external-ca-profile** options, see [Options used when installing an IdM CA with an external CA as the root CA](#) .

Procedure

1. Run the **ipa-server-install** utility with the **--external-ca** option.

```
# ipa-server-install --external-ca
```

- If you are using the Microsoft Certificate Services (MS CS) CA, also use the **--external-ca-type** option and, optionally, the **--external-ca-profile** option:

```
[root@server ~]# ipa-server-install --external-ca --external-ca-type=ms-cs --
external-ca-profile=<oid>/<name>/default
```

- If you are not using MS CS to generate the signing certificate for your IdM CA, no other option may be necessary:

```
# ipa-server-install --external-ca
```

2. The script prompts to configure an integrated DNS service. Enter **yes** or **no**. In this procedure, we are installing a server with integrated DNS.

```
Do you want to configure integrated DNS (BIND)? [no]: yes
```



NOTE

If you want to install a server without integrated DNS, the installation script will not prompt you for DNS configuration as described in the steps below. See [Chapter 6, Installing an IdM server: Without integrated DNS, with an integrated CA as the root CA](#) for details on the steps for installing a server without DNS.

3. The script prompts for several required settings and offers recommended default values in brackets.

- To accept a default value, press **Enter**.
- To provide a custom value, enter the required value.

```
Server host name [server.idm.example.com]:
Please confirm the domain name [idm.example.com]:
Please provide a realm name [IDM.EXAMPLE.COM]:
```



WARNING

Plan these names carefully. You will not be able to change them after the installation is complete.

4. Enter the passwords for the Directory Server superuser (**cn=Directory Manager**) and for the Identity Management (IdM) administration system user account (**admin**).

```
Directory Manager password:
IPA admin password:
```

5. The script prompts for per-server DNS forwarders.

```
Do you want to configure DNS forwarders? [yes]:
```

- To configure per-server DNS forwarders, enter **yes**, and then follow the instructions on the command line. The installation process will add the forwarder IP addresses to the IdM LDAP.
 - For the forwarding policy default settings, see the **--forward-policy** description in the **ipa-dns-install(1)** man page.
 - If you do not want to use DNS forwarding, enter **no**.
With no DNS forwarders, hosts in your IdM domain will not be able to resolve names from other, internal, DNS domains in your infrastructure. The hosts will only be left with public DNS servers to resolve their DNS queries.
6. The script prompts to check if any DNS reverse (PTR) records for the IP addresses associated with the server need to be configured.

Do you want to search for missing reverse zones? [yes]:

If you run the search and missing reverse zones are discovered, the script asks you whether to create the reverse zones along with the PTR records.

Do you want to create reverse zone for IP 192.0.2.1 [yes]:
Please specify the reverse zone name [2.0.192.in-addr.arpa.]:
Using reverse zone(s) 2.0.192.in-addr.arpa.



NOTE

Using IdM to manage reverse zones is optional. You can use an external DNS service for this purpose instead.

7. Enter **yes** to confirm the server configuration.

Continue to configure the system with these values? [no]: yes

8. During the configuration of the Certificate System instance, the utility prints the location of the certificate signing request (CSR): **/root/ipa.csr**:

...

Configuring certificate server (pki-tomcatd): Estimated time 3 minutes 30 seconds

[1/8]: creating certificate server user

[2/8]: configuring certificate server instance

The next step is to get **/root/ipa.csr** signed by your CA and re-run **/sbin/ipa-server-install** as:
/sbin/ipa-server-install --external-cert-file=/path/to/signed_certificate --external-cert-file=/path/to/external_ca_certificate

When this happens:

- a. Submit the CSR located in **/root/ipa.csr** to the external CA. The process differs depending on the service to be used as the external CA.
- b. Retrieve the issued certificate and the CA certificate chain for the issuing CA in a base 64-encoded blob (either a PEM file or a Base_64 certificate from a Windows CA). Again, the process differs for every certificate service. Usually, a download link on a web page or in the notification email allows the administrator to download all the required certificates.

**IMPORTANT**

Be sure to get the full certificate chain for the CA, not just the CA certificate.

- c. Run **ipa-server-install** again, this time specifying the locations and names of the newly-issued CA certificate and the CA chain files. For example:

```
# ipa-server-install --external-cert-file=/tmp/servercert20170601.pem --external-cert-file=/tmp/cacert.pem
```

9. The installation script now configures the server. Wait for the operation to complete.
10. After the installation script completes, update your DNS records in the following way:
 - a. Add DNS delegation from the parent domain to the IdM DNS domain. For example, if the IdM DNS domain is **idm.example.com**, add a name server (NS) record to the **example.com** parent domain.

**IMPORTANT**

Repeat this step each time after an IdM DNS server is installed.

- b. Add an **_ntp._udp** service (SRV) record for your time server to your IdM DNS. The presence of the SRV record for the time server of the newly-installed IdM server in IdM DNS ensures that future replica and client installations are automatically configured to synchronize with the time server used by this primary IdM server.

**NOTE**

The **ipa-server-install --external-ca** command can sometimes fail with the following error:

```
ipa      : CRITICAL failed to configure ca instance Command '/usr/sbin/pkispawn -s CA -f /tmp/configuration_file' returned non-zero exit status 1
Configuration of CA failed
```

This failure occurs when the ***_proxy** environmental variables are set. For a solution of the problem, see [Troubleshooting: External CA installation fails](#).

4.2. TROUBLESHOOTING: EXTERNAL CA INSTALLATION FAILS

The **ipa-server-install --external-ca** command fails with the following error:

```
ipa      : CRITICAL failed to configure ca instance Command '/usr/sbin/pkispawn -s CA -f /tmp/configuration_file' returned non-zero exit status 1
Configuration of CA failed
```

The **env|grep proxy** command displays variables such as the following:

```
# env|grep proxy
http_proxy=http://example.com:8080
ftp_proxy=http://example.com:8080
https_proxy=http://example.com:8080
```

-

What this means:

The `*_proxy` environmental variables are preventing the server from being installed.

To fix the problem:

1. Use the following shell script to unset the `*_proxy` environmental variables:

```
# for i in ftp http https; do unset ${i}_proxy; done
```

2. Run the `pkidestroy` utility to remove the unsuccessful certificate authority (CA) subsystem installation:

```
# pkidestroy -s CA -i pki-tomcat; rm -rf /var/log/pki/pki-tomcat /etc/sysconfig/pki-tomcat /etc/sysconfig/pki/tomcat/pki-tomcat /var/lib/pki/pki-tomcat /etc/pki/pki-tomcat /root/ipa.csr
```

3. Remove the failed Identity Management (IdM) server installation:

```
# ipa-server-install --uninstall
```

4. Retry running `ipa-server-install --external-ca`.

CHAPTER 5. INSTALLING AN IDM SERVER: WITH INTEGRATED DNS, WITHOUT A CA

Installing a new Identity Management (IdM) server with integrated DNS has the following advantages:

- You can automate much of the maintenance and DNS record management using native IdM tools. For example, DNS SRV records are automatically created during the setup, and later on are automatically updated.
- You can configure global forwarders during the installation of the IdM server for a stable external internet connection. Global forwarders are also useful for trusts with Active Directory.
- You can set up a DNS reverse zone to prevent emails from your domain to be considered spam by email servers outside of the IdM domain.

Installing IdM with integrated DNS has certain limitations:

- IdM DNS is not meant to be used as a general-purpose DNS server. Some of the advanced DNS functions are not supported. For more information, see [DNS services available in an IdM server](#).

This chapter describes how you can install a new IdM server without a certificate authority (CA).

5.1. CERTIFICATES REQUIRED TO INSTALL AN IDM SERVER WITHOUT A CA

You need to provide the certificates required to install an Identity Management (IdM) server without a certificate authority (CA). By using the command-line options described, you can provide these certificates to the **ipa-server-install** utility.



IMPORTANT

You cannot install a server or replica using self-signed third-party server certificates because the imported certificate files must contain the full CA certificate chain of the CA that issued the LDAP and Apache server certificates.

The LDAP server certificate and private key

- **--dirsrv-cert-file** for the certificate and private key files for the LDAP server certificate
- **--dirsrv-pin** for the password to access the private key in the files specified in **--dirsrv-cert-file**

The Apache server certificate and private key

- **--http-cert-file** for the certificate and private key files for the Apache server certificate
- **--http-pin** for the password to access the private key in the files specified in **--http-cert-file**

The full CA certificate chain of the CA that issued the LDAP and Apache server certificates

- **--dirsrv-cert-file** and **--http-cert-file** for the certificate files with the full CA certificate chain or a part of it

You can provide the files specified in the **--dirsrv-cert-file** and **--http-cert-file** options in the following formats:

- Privacy-Enhanced Mail (PEM) encoded certificate (RFC 7468). Note that the Identity Management installer accepts concatenated PEM-encoded objects.
- Distinguished Encoding Rules (DER)
- PKCS #7 certificate chain objects
- PKCS #8 private key objects
- PKCS #12 archives

You can specify the **--dirsrv-cert-file** and **--http-cert-file** options multiple times to specify multiple files.

The certificate files to complete the full CA certificate chain (not needed in some environments)

- **--ca-cert-file** for the file or files containing the CA certificate of the CA that issued the LDAP, Apache Server, and Kerberos KDC certificates. Use this option if the CA certificate is not present in the certificate files provided by the other options.

The files provided using **--dirsrv-cert-file** and **--http-cert-file** combined with the file provided using **--ca-cert-file** must contain the full CA certificate chain of the CA that issued the LDAP and Apache server certificates.

The Kerberos key distribution center (KDC) PKINIT certificate and private key

- If you have a PKINIT certificate, use the following 2 options:
 - **--pkinit-cert-file** for the Kerberos KDC SSL certificate and private key
 - **--pkinit-pin** for the password to access the Kerberos KDC private key in the files specified in **--pkinit-cert-file**
- If you do not have a PKINIT certificate and want to configure the IdM server with a local KDC with a self-signed certificate, use the following option:
 - **--no-pkinit** for disabling pkinit setup steps

Additional resources

- For details on what the certificate file formats these options accept, see the **ipa-server-install(1)** man page.
- For details on PKINIT extensions required to create a RHEL IdM PKINIT certificate, see [RHEL IdM PKINIT KDC certificate and extensions](#).

5.2. INTERACTIVE INSTALLATION

During the interactive installation using the **ipa-server-install** utility, you are asked to supply basic configuration of the system, for example the realm, the administrator's password and the Directory Manager's password.

The **ipa-server-install** installation script creates a log file at **/var/log/ipaserver-install.log**. If the installation fails, the log can help you identify the problem.

Procedure

1. Run the **ipa-server-install** utility and provide all the required certificates. For example:

```
[root@server ~]# ipa-server-install \
  --http-cert-file /tmp/server.crt \
  --http-cert-file /tmp/server.key \
  --http-pin secret \
  --dirsrv-cert-file /tmp/server.crt \
  --dirsrv-cert-file /tmp/server.key \
  --dirsrv-pin secret \
  --ca-cert-file ca.crt
```

See [Certificates required to install an IdM server without a CA](#) for details on the provided certificates.

2. The script prompts to configure an integrated DNS service. Enter **yes** or **no**. In this procedure, we are installing a server with integrated DNS.

```
Do you want to configure integrated DNS (BIND)? [no]: yes
```



NOTE

If you want to install a server without integrated DNS, the installation script will not prompt you for DNS configuration as described in the steps below. See [Installing an IdM server: Without integrated DNS, with an integrated CA as the root CA](#) for details on the steps for installing a server without DNS.

3. The script prompts for several required settings and offers recommended default values in brackets.
 - To accept a default value, press **Enter**.
 - To provide a custom value, enter the required value.

```
Server host name [server.idm.example.com]:
Please confirm the domain name [idm.example.com]:
Please provide a realm name [IDM.EXAMPLE.COM]:
```



WARNING

Plan these names carefully. You will not be able to change them after the installation is complete.

4. Enter the passwords for the Directory Server superuser (**cn=Directory Manager**) and for the Identity Management (IdM) administration system user account (**admin**).

Directory Manager password:
IPA admin password:

5. The script prompts for per-server DNS forwarders.

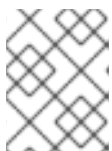
Do you want to configure DNS forwarders? [yes]:

- To configure per-server DNS forwarders, enter **yes**, and then follow the instructions on the command line. The installation process will add the forwarder IP addresses to the IdM LDAP.
 - For the forwarding policy default settings, see the **--forward-policy** description in the **ipa-dns-install(1)** man page.
 - If you do not want to use DNS forwarding, enter **no**.
With no DNS forwarders, hosts in your IdM domain will not be able to resolve names from other, internal, DNS domains in your infrastructure. The hosts will only be left with public DNS servers to resolve their DNS queries.
6. The script prompts to check if any DNS reverse (PTR) records for the IP addresses associated with the server need to be configured.

Do you want to search for missing reverse zones? [yes]:

If you run the search and missing reverse zones are discovered, the script asks you whether to create the reverse zones along with the PTR records.

Do you want to create reverse zone for IP 192.0.2.1 [yes]:
Please specify the reverse zone name [2.0.192.in-addr.arpa]:
Using reverse zone(s) 2.0.192.in-addr.arpa.



NOTE

Using IdM to manage reverse zones is optional. You can use an external DNS service for this purpose instead.

7. Enter **yes** to confirm the server configuration.

Continue to configure the system with these values? [no]: yes

8. The installation script now configures the server. Wait for the operation to complete.
9. After the installation script completes, update your DNS records in the following way:
 - a. Add DNS delegation from the parent domain to the IdM DNS domain. For example, if the IdM DNS domain is **idm.example.com**, add a name server (NS) record to the **example.com** parent domain.



IMPORTANT

Repeat this step each time after an IdM DNS server is installed.

- b. Add an **_ntp._udp** service (SRV) record for your time server to your IdM DNS. The

presence of the SRV record for the time server of the newly-installed IdM server in IdM DNS ensures that future replica and client installations are automatically configured to synchronize with the time server used by this primary IdM server.

CHAPTER 6. INSTALLING AN IDM SERVER: WITHOUT INTEGRATED DNS, WITH AN INTEGRATED CA AS THE ROOT CA

This chapter describes how you can install a new Identity Management (IdM) server without integrated DNS.



NOTE

Red Hat strongly recommends installing IdM-integrated DNS for basic usage within the IdM deployment: When the IdM server also manages DNS, there is tight integration between DNS and native IdM tools which enables automating some of the DNS record management.

For more details, see [Planning your DNS services and host names](#).

6.1. INTERACTIVE INSTALLATION

During the interactive installation using the **ipa-server-install** utility, you are asked to supply basic configuration of the system, for example the realm, the administrator's password and the Directory Manager's password.

The **ipa-server-install** installation script creates a log file at `/var/log/ipaserver-install.log`. If the installation fails, the log can help you identify the problem.

This procedure installs a server:

- Without integrated DNS
- With integrated Identity Management (IdM) certificate authority (CA) as the root CA, which is the default CA configuration

Procedure

1. Run the **ipa-server-install** utility.

```
# ipa-server-install
```

2. The script prompts to configure an integrated DNS service. Press **Enter** to select the default **no** option.

```
Do you want to configure integrated DNS (BIND)? [no]:
```

3. The script prompts for several required settings and offers recommended default values in brackets.
 - To accept a default value, press **Enter**.
 - To provide a custom value, enter the required value.

```
Server host name [server.idm.example.com]:
Please confirm the domain name [idm.example.com]:
Please provide a realm name [IDM.EXAMPLE.COM]:
```

**WARNING**

Plan these names carefully. You will not be able to change them after the installation is complete.

4. Enter the passwords for the Directory Server superuser (**cn=Directory Manager**) and for the IdM administration system user account (**admin**).

```
Directory Manager password:
IPA admin password:
```

5. The script prompts for several required settings and offers recommended default values in brackets.

- To accept a default value, press **Enter**.
- To provide a custom value, enter the required value.

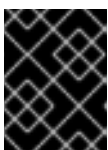
```
NetBIOS domain name [EXAMPLE]:
Do you want to configure chrony with NTP server or pool address? [no]:
```

6. Enter **yes** to confirm the server configuration.

```
Continue to configure the system with these values? [no]: yes
```

7. The installation script now configures the server. Wait for the operation to complete.
8. The installation script produces a file with DNS resource records: **the /tmp/ipa.system.records.UFRBto.db** file in the example output below. Add these records to the existing external DNS servers. The process of updating the DNS records varies depending on the particular DNS solution.

```
...
Restarting the KDC
Please add records in this file to your DNS system:
/tmp/ipa.system.records.UFRBto.db
Restarting the web server
...
```

**IMPORTANT**

The server installation is not complete until you add the DNS records to the existing DNS servers.

Additional resources

- For more information about the DNS resource records you must add to your DNS system, see [IdM DNS records for external DNS systems](#) .

6.2. NON-INTERACTIVE INSTALLATION

This procedure installs a server without integrated DNS or with integrated Identity Management (IdM) certificate authority (CA) as the root CA, which is the default CA configuration.



NOTE

The **ipa-server-install** installation script creates a log file at `/var/log/ipaserver-install.log`. If the installation fails, the log can help you identify the problem.

Procedure

1. Run the **ipa-server-install** utility with the options to supply all the required information. The minimum required options for non-interactive installation are:
 - **--realm** to provide the Kerberos realm name
 - **--ds-password** to provide the password for the Directory Manager (DM), the Directory Server super user
 - **--admin-password** to provide the password for **admin**, the IdM administrator
 - **--unattended** to let the installation process select default options for the host name and domain name

For example:

```
# ipa-server-install --realm IDM.EXAMPLE.COM --ds-password DM_password --admin-  
password admin_password --unattended
```

2. The installation script produces a file with DNS resource records: **the `/tmp/ipa.system.records.UFRPto.db`** file in the example output below. Add these records to the existing external DNS servers. The process of updating the DNS records varies depending on the particular DNS solution.

```
...  
Restarting the KDC  
Please add records in this file to your DNS system:  
/tmp/ipa.system.records.UFRBto.db  
Restarting the web server  
...
```



IMPORTANT

The server installation is not complete until you add the DNS records to the existing DNS servers.

Additional resources

- For more information about the DNS resource records you must add to your DNS system, see [IdM DNS records for external DNS systems](#).
- For a complete list of options accepted by **ipa-server-install**, run the **ipa-server-install --help** command.

6.3. IDM DNS RECORDS FOR EXTERNAL DNS SYSTEMS

After installing an IdM server without integrated DNS, you must add LDAP and Kerberos DNS resource records for the IdM server to your external DNS system.

The **ipa-server-install** installation script generates a file containing the list of DNS resource records with a file name in the format **/tmp/ipa.system.records.<random_characters>.db** and prints instructions to add those records:

Please add records in this file to your DNS system: **/tmp/ipa.system.records.6zjqxh3.db**

This is an example of the contents of the file:

```
_kerberos-master._tcp.example.com. 86400 IN SRV 0 100 88 server.example.com.  
_kerberos-master._udp.example.com. 86400 IN SRV 0 100 88 server.example.com.  
_kerberos._tcp.example.com. 86400 IN SRV 0 100 88 server.example.com.  
_kerberos._udp.example.com. 86400 IN SRV 0 100 88 server.example.com.  
_kerberos.example.com. 86400 IN TXT "EXAMPLE.COM"  
_kpasswd._tcp.example.com. 86400 IN SRV 0 100 464 server.example.com.  
_kpasswd._udp.example.com. 86400 IN SRV 0 100 464 server.example.com.  
_ldap._tcp.example.com. 86400 IN SRV 0 100 389 server.example.com.
```



NOTE

After adding the LDAP and Kerberos DNS resource records for the IdM server to your DNS system, ensure that the DNS management tools have not added PTR records for **ipa-ca**. The presence of PTR records for **ipa-ca** in your DNS could cause subsequent IdM replica installations to fail.

CHAPTER 7. INSTALLING AN IDM SERVER: WITHOUT INTEGRATED DNS, WITH AN EXTERNAL CA AS THE ROOT CA

This chapter describes how you can install a new Identity Management (IdM) server, without integrated DNS, that uses an external certificate authority (CA) as the root CA.



NOTE

Red Hat strongly recommends installing IdM-integrated DNS for basic usage within the IdM deployment: When the IdM server also manages DNS, there is tight integration between DNS and native IdM tools which enables automating some of the DNS record management.

For more details, see [Planning your DNS services and host names](#).

7.1. OPTIONS USED WHEN INSTALLING AN IDM CA WITH AN EXTERNAL CA AS THE ROOT CA

You may want to install an Identity Management IdM certificate authority (CA) with an external CA as the root CA if one of the following conditions applies:

- You are installing a new IdM server or replica by using the **ipa-server-install** command.
- You are installing the CA component into an existing IdM server by using the **ipa-ca-install** command.

You can use following options for both commands that you can use for creating a certificate signing request (CSR) during the installation of an IdM CA with an external CA as the root CA.

--external-ca-type=TYPE

Type of the external CA. Possible values are **generic** and **ms-cs**. The default value is **generic**. Use **ms-cs** to include a template name required by Microsoft Certificate Services (MS CS) in the generated CSR. To use a non-default profile, use the **--external-ca-profile** option in conjunction with **--external-ca-type=ms-cs**.

--external-ca-profile=PROFILE_SPEC

Specify the certificate profile or template that you want the MS CS to apply when issuing the certificate for your IdM CA.

Note that the **--external-ca-profile** option can only be used if **--external-ca-type** is **ms-cs**.

You can identify the MS CS template in one of the following ways:

- **<oid>:<majorVersion>[:<minorVersion>]**. You can specify a certificate template by its object identifier (OID) and major version. You can optionally also specify the minor version.
- **<name>**. You can specify a certificate template by its name. The name cannot contain any `:` characters and cannot be an OID, otherwise the OID-based template specifier syntax takes precedence.
- **default**. If you use this specifier, the template name **SubCA** is used.

In certain scenarios, the Active Directory (AD) administrator can use the **Subordinate Certification Authority (SCA)** template, which is a built-in template in AD CS, to create a unique template to better suit the needs of the organization. The new template can, for example, have a customized validity period

and customized extensions. The associated Object Identifier (OID) can be found in the AD **Certificates Template** console.

If the AD administrator has disabled the original, built-in template, you must specify the OID or name of the new template when requesting a certificate for your IdM CA. Ask your AD administrator to provide you with the name or OID of the new template.

If the original SCA AD CS template is still enabled, you can use it by specifying **--external-ca-type=ms-cs** without additionally using the **--external-ca-profile** option. In this case, the **subCA** external CA profile is used, which is the default IdM template corresponding to the SCA AD CS template.

7.2. INTERACTIVE INSTALLATION

During the interactive installation using the **ipa-server-install** utility, you are asked to supply basic configuration of the system, for example the realm, the administrator's password and the Directory Manager's password.

The **ipa-server-install** installation script creates a log file at **/var/log/ipaserver-install.log**. If the installation fails, the log can help you identify the problem.

Follow this procedure to install a server:

- Without integrated DNS
- With an external certificate authority (CA) as the root CA

Prerequisites

- You have determined the type of the external CA to specify with the **--external-ca-type** option. See the **ipa-server-install(1)** man page for details.
- If you are using a Microsoft Certificate Services certificate authority (MS CS CA) as your external CA: you have determined the certificate profile or template to specify with the **--external-ca-profile** option. By default, the **SubCA** template is used. For more information about the **--external-ca-type** and **--external-ca-profile** options, see [Options used when installing an IdM CA with an external CA as the root CA](#).

Procedure

1. Run the **ipa-server-install** utility with the **--external-ca** option.
 - If you are using the Microsoft Certificate Services (MS CS) CA, also use the **--external-ca-type** option and, optionally, the **--external-ca-profile** option:

```
[root@server ~]# ipa-server-install --external-ca --external-ca-type=ms-cs --external-ca-profile=<oid>/<name>/default
```

- If you are not using MS CS to generate the signing certificate for your IdM CA, no other option may be necessary:

```
# ipa-server-install --external-ca
```

2. The script prompts to configure an integrated DNS service. Press **Enter** to select the default **no** option.

Do you want to configure integrated DNS (BIND)? [no]:

- The script prompts for several required settings and offers recommended default values in brackets.

- To accept a default value, press **Enter**.
- To provide a custom value, enter the required value.

Server host name [**server.idm.example.com**]:
Please confirm the domain name [**idm.example.com**]:
Please provide a realm name [**IDM.EXAMPLE.COM**]:



WARNING

Plan these names carefully. You will not be able to change them after the installation is complete.

- Enter the passwords for the Directory Server superuser (**cn=Directory Manager**) and for the IdM administration system user account (**admin**).

Directory Manager password:
IPA admin password:

- Enter **yes** to confirm the server configuration.

Continue to configure the system with these values? [no]: **yes**

- During the configuration of the Certificate System instance, the utility prints the location of the certificate signing request (CSR): **/root/ipa.csr**:

...

```
Configuring certificate server (pki-tomcatd): Estimated time 3 minutes 30 seconds
[1/8]: creating certificate server user
[2/8]: configuring certificate server instance
The next step is to get /root/ipa.csr signed by your CA and re-run /sbin/ipa-server-install as:
/sbin/ipa-server-install --external-cert-file=/path/to/signed_certificate --external-cert-
file=/path/to/external_ca_certificate
```

When this happens:

- Submit the CSR located in **/root/ipa.csr** to the external CA. The process differs depending on the service to be used as the external CA.
- Retrieve the issued certificate and the CA certificate chain for the issuing CA in a base 64-encoded blob (either a PEM file or a Base_64 certificate from a Windows CA). Again, the process differs for every certificate service. Usually, a download link on a web page or in the notification email allows the administrator to download all the required certificates.

**IMPORTANT**

Be sure to get the full certificate chain for the CA, not just the CA certificate.

- c. Run **ipa-server-install** again, this time specifying the locations and names of the newly-issued CA certificate and the CA chain files. For example:

```
# ipa-server-install --external-cert-file=/tmp/servercert20170601.pem --external-cert-
file=/tmp/cacert.pem
```

7. The installation script now configures the server. Wait for the operation to complete.
8. The installation script produces a file with DNS resource records: **the /tmp/ipa.system.records.UFRPto.db** file in the example output below. Add these records to the existing external DNS servers. The process of updating the DNS records varies depending on the particular DNS solution.

```
...
Restarting the KDC
Please add records in this file to your DNS system:
/tmp/ipa.system.records.UFRBto.db
Restarting the web server
...
```

**IMPORTANT**

The server installation is not complete until you add the DNS records to the existing DNS servers.

Additional resources

- For more information about the DNS resource records you must add to your DNS system, see [IdM DNS records for external DNS systems](#) .
- The **ipa-server-install --external-ca** command can sometimes fail with the following error:

```
ipa      : CRITICAL failed to configure ca instance Command '/usr/sbin/pkispawn -s CA -f
/tmp/pass:quotes[configuration_file]' returned non-zero exit status 1
Configuration of CA failed
```

This failure occurs when the ***_proxy** environmental variables are set. For a solution of the problem, see [Troubleshooting: External CA installation fails](#) .

7.3. NON-INTERACTIVE INSTALLATION

This procedure installs a server:

- Without integrated DNS
- With an external certificate authority (CA) as the root CA



NOTE

The **ipa-server-install** installation script creates a log file at `/var/log/ipaserver-install.log`. If the installation fails, the log can help you identify the problem.

Prerequisites

- You have determined the type of the external CA to specify with the **--external-ca-type** option. See the **ipa-server-install(1)** man page for details.
- If you are using a Microsoft Certificate Services certificate authority (MS CS CA) as your external CA: you have determined the certificate profile or template to specify with the **--external-ca-profile** option. By default, the **SubCA** template is used. For more information about the **--external-ca-type** and **--external-ca-profile** options, see [Options used when installing an IdM CA with an external CA as the root CA](#).

Procedure

1. Run the **ipa-server-install** utility with the options to supply all the required information. The minimum required options for non-interactive installation of an IdM server with an external CA as the root CA are:

- **--external-ca** to specify an external CA is the root CA
 - **--realm** to provide the Kerberos realm name
 - **--ds-password** to provide the password for the Directory Manager (DM), the Directory Server super user
 - **--admin-password** to provide the password for **admin**, the IdM administrator
 - **--unattended** to let the installation process select default options for the host name and domain name
- For example:

```
# ipa-server-install --external-ca --realm IDM.EXAMPLE.COM --ds-password
DM_password --admin-password admin_password --unattended
```

If you are using a Microsoft Certificate Services (MS CS) CA, also use the **--external-ca-type** option and, optionally, the **--external-ca-profile** option. For more information, see [Options used when installing an IdM CA with an external CA as the root CA](#).

2. During the configuration of the Certificate System instance, the utility prints the location of the certificate signing request (CSR): **/root/ipa.csr**:

```
...
Configuring certificate server (pki-tomcatd). Estimated time: 3 minutes
[1/11]: configuring certificate server instance
The next step is to get /root/ipa.csr signed by your CA and re-run /usr/sbin/ipa-server-install
as:
/usr/sbin/ipa-server-install --external-cert-file=/path/to/signed_certificate --external-cert-
file=/path/to/external_ca_certificate
The ipa-server-install command was successful
```

When this happens:

- a. Submit the CSR located in **/root/ipa.csr** to the external CA. The process differs depending on the service to be used as the external CA.
- b. Retrieve the issued certificate and the CA certificate chain for the issuing CA in a base 64-encoded blob (either a PEM file or a Base_64 certificate from a Windows CA). Again, the process differs for every certificate service. Usually, a download link on a web page or in the notification email allows the administrator to download all the required certificates.



IMPORTANT

Be sure to get the full certificate chain for the CA, not just the CA certificate.

- c. Run **ipa-server-install** again, this time specifying the locations and names of the newly-issued CA certificate and the CA chain files. For example:

```
# ipa-server-install --external-cert-file=/tmp/servercert20170601.pem --external-cert-
file=/tmp/cacert.pem --realm IDM.EXAMPLE.COM --ds-password DM_password --
admin-password admin_password --unattended
```

3. The installation script now configures the server. Wait for the operation to complete.
4. The installation script produces a file with DNS resource records: the **/tmp/ipa.system.records.UFRPto.db** file in the example output below. Add these records to the existing external DNS servers. The process of updating the DNS records varies depending on the particular DNS solution.

```
...
Restarting the KDC
Please add records in this file to your DNS system:
/tmp/ipa.system.records.UFRBto.db
Restarting the web server
...
```



IMPORTANT

The server installation is not complete until you add the DNS records to the existing DNS servers.

Additional resources

- For more information about the DNS resource records you must add to your DNS system, see [IdM DNS records for external DNS systems](#).

7.4. IDM DNS RECORDS FOR EXTERNAL DNS SYSTEMS

After installing an IdM server without integrated DNS, you must add LDAP and Kerberos DNS resource records for the IdM server to your external DNS system.

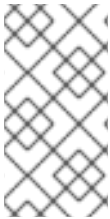
The **ipa-server-install** installation script generates a file containing the list of DNS resource records with a file name in the format **/tmp/ipa.system.records.<random_characters>.db** and prints instructions to add those records:

-

Please add records in this file to your DNS system: **/tmp/ipa.system.records.6zdjqxh3.db**

This is an example of the contents of the file:

```
_kerberos-master._tcp.example.com. 86400 IN SRV 0 100 88 server.example.com.  
_kerberos-master._udp.example.com. 86400 IN SRV 0 100 88 server.example.com.  
_kerberos._tcp.example.com. 86400 IN SRV 0 100 88 server.example.com.  
_kerberos._udp.example.com. 86400 IN SRV 0 100 88 server.example.com.  
_kerberos.example.com. 86400 IN TXT "EXAMPLE.COM"  
_kpasswd._tcp.example.com. 86400 IN SRV 0 100 464 server.example.com.  
_kpasswd._udp.example.com. 86400 IN SRV 0 100 464 server.example.com.  
_ldap._tcp.example.com. 86400 IN SRV 0 100 389 server.example.com.
```



NOTE

After adding the LDAP and Kerberos DNS resource records for the IdM server to your DNS system, ensure that the DNS management tools have not added PTR records for **ipa-ca**. The presence of PTR records for **ipa-ca** in your DNS could cause subsequent IdM replica installations to fail.

CHAPTER 8. INSTALLING AN IDM SERVER OR REPLICA WITH CUSTOM DATABASE SETTINGS FROM AN LDIF FILE

You can install an IdM server and IdM replicas with custom settings for the Directory Server database. The following procedure shows you how to create an LDAP Data Interchange Format (LDIF) file with database settings, and how to pass those settings to the IdM server and replica installation commands.

Prerequisites

- You have determined custom Directory Server settings that improve the performance of your IdM environment. See [Adjusting IdM Directory Server performance](#).

Procedure

1. Create a text file in LDIF format with your custom database settings. Separate LDAP attribute modifications with a dash (-). This example sets non-default values for the idle timeout and maximum file descriptors.

```
dn: cn=config
changetype: modify
replace: nsslapd-idletimeout
nsslapd-idletimeout=1800
-
replace: nsslapd-maxdescriptors
nsslapd-maxdescriptors=8192
```

2. Use the **--dirsrv-config-file** parameter to pass the LDIF file to the installation script.

- a. To install an IdM server:

```
# ipa-server-install --dirsrv-config-file filename.ldif
```

- b. To install an IdM replica:

```
# ipa-replica-install --dirsrv-config-file filename.ldif
```

Additional resources

- Options for the [ipa-server-install](#) and [ipa-replica-install](#) commands

CHAPTER 9. TROUBLESHOOTING IDM SERVER INSTALLATION

The following sections describe how to gather information about a failing IdM server installation, and how to resolve common installation issues.

9.1. REVIEWING IDM SERVER INSTALLATION ERROR LOGS

When you install an Identity Management (IdM) server, debugging information is appended to the following log files:

- `/var/log/ipaserver-install.log`
- `/var/log/httpd/error_log`
- `/var/log/dirsrv/slapd-INSTANCE-NAME/access`
- `/var/log/dirsrv/slapd-INSTANCE-NAME/errors`

The last lines of the log files report success or failure, and the **ERROR** and **DEBUG** entries provide additional context.

To troubleshoot a failing IdM server installation, review the errors at the end of the log files and use this information to resolve any corresponding issues.

Prerequisites

- You must have **root** privileges to display the contents of IdM log files.

Procedure

1. Use the **tail** command to display the last lines of a log file. The following example displays the last 10 lines of `/var/log/ipaserver-install.log`.

```
[user@server ~]$ sudo tail -n 10 /var/log/ipaserver-install.log
[sudo] password for user:
value = gen.send(prev_value)
File "/usr/lib/python3.6/site-packages/ipapython/install/common.py", line 65, in _install
for unused in self._installer(self.parent):
File "/usr/lib/python3.6/site-packages/ipaserver/install/server/init.py", line 564, in main
master_install(self)
File "/usr/lib/python3.6/site-packages/ipaserver/install/server/install.py", line 291, in decorated
raise ScriptError()

2020-05-27T22:59:41Z DEBUG The ipa-server-install command failed, exception:
ScriptError:
2020-05-27T22:59:41Z ERROR The ipa-server-install command failed. See
/var/log/ipaserver-install.log for more information
```

2. To review a log file interactively, open the end of the log file using the **less** utility and use the **↑** and **↓** arrow keys to navigate. The following example opens the `/var/log/ipaserver-install.log` file interactively.

```
[user@server ~]$ sudo less -N +G /var/log/ipaserver-install.log
```

- Gather additional troubleshooting information by repeating this review process with the remaining log files.

```
[user@server ~]$ sudo less -N +G /var/log/httpd/error_log
```

```
[user@server ~]$ sudo less -N +G /var/log/dirsrv/slapd-INSTANCE-NAME/access
```

```
[user@server ~]$ sudo less -N +G /var/log/dirsrv/slapd-INSTANCE-NAME/errors
```

Additional resources

- If you are unable to resolve a failing IdM server installation, and you have a Red Hat Technical Support subscription, open a Technical Support case at the [Red Hat Customer Portal](#) and provide an **sosreport** of the server.
- The **sosreport** utility collects configuration details, logs and system information from a RHEL system. For more information about the **sosreport** utility, see [What is an sosreport and how to create one in Red Hat Enterprise Linux?](#).

9.2. REVIEWING IDM CA INSTALLATION ERRORS

When you install the Certificate Authority (CA) service on an Identity Management (IdM) server, debugging information is appended to the following locations (in order of recommended priority):

Location	Description
/var/log/pki/pki-ca-spawn.\$<i>TIME_OF_INSTALLATION</i>.log	High-level issues and Python traces for the pkispawn installation process
journalctl -u pki-tomcatd@pki-tomcat output	Errors from the pki-tomcatd@pki-tomcat service
/var/log/pki/pki-tomcat/ca/debug.\$<i>DATE</i>.log	Large JAVA stacktraces of activity in the core of the Public Key Infrastructure (PKI) product
/var/log/pki/pki-tomcat/ca/signedAudit/ca_audit log file	Audit log of the PKI product
<ul style="list-style-type: none"> /var/log/pki/pki-tomcat/ca/system /var/log/pki/pki-tomcat/ca/transactions /var/log/pki/pki-tomcat/catalina.\$<i>DATE</i>.log 	Low-level debug data of certificate operations for service principals, hosts, and other entities that use certificates



NOTE

If a full IdM server installation fails while installing the optional CA component, no details about the CA are logged; a message is logged in the `/var/log/ipaserver-install.log` file indicating that the overall installation process failed. Red Hat recommends reviewing the log files listed above for details specific to the CA installation failure.

The only exception to this behavior is when you are installing the CA service and the root CA is an external CA. If there is an issue with the certificate from the external CA, errors are logged in `/var/log/ipaserver-install.log`.

To troubleshoot a failing IdM CA installation, review the errors at the end of these log files and use this information to resolve any corresponding issues.

Prerequisites

- You must have **root** privileges to display the contents of IdM log files.

Procedure

- To review a log file interactively, open the end of the log file using the **less** utility and use the `↑` and `↓` arrow keys to navigate, while searching for **ScriptError** entries. The following example opens `/var/log/pki/pki-ca-spawn.$TIME_OF_INSTALLATION.log`.

```
[user@server ~]$ sudo less -N +G /var/log/pki/pki-ca-spawn.20200527185902.log
```

- Gather additional troubleshooting information by repeating this review process with all the log files listed above.

Additional resources

- If you are unable to resolve a failing IdM server installation, and you have a Red Hat Technical Support subscription, open a Technical Support case at the [Red Hat Customer Portal](#) and provide an **sosreport** of the server.
- The **sosreport** utility collects configuration details, logs and system information from a RHEL system. For more information about the **sosreport** utility, see [What is an sosreport and how to create one in Red Hat Enterprise Linux?](#).

9.3. REMOVING A PARTIAL IDM SERVER INSTALLATION

If an IdM server installation fails, some configuration files can be left behind. Additional attempts to install the IdM server fail and the installation script reports that IPA is already configured.

Example system with existing partial IdM configuration

```
[root@server ~]# ipa-server-install
```

The log file for this installation can be found in `/var/log/ipaserver-install.log`

IPA server is already configured on this system.

If you want to reinstall the IPA server, **please uninstall it first using 'ipa-server-install --uninstall'**.

The `ipa-server-install` command failed. See `/var/log/ipaserver-install.log` for more information

To resolve this issue, uninstall the partial IdM server configuration and retry the installation process.

Prerequisites

- You must have **root** privileges.

Procedure

1. Uninstall the IdM server software from the host you are trying to configure as an IdM server.

```
[root@server ~]# ipa-server-install --uninstall
```

2. If you continue to experience difficulty installing an IdM server because of repeated failed installations, reinstall the operating system.

One of the requirements for installing an IdM server is a clean system without any customization. Failed installations may have compromised the integrity of the host by unexpectedly modifying system files.

Additional resources

- For additional details on uninstalling an IdM server, see [Uninstalling an IdM server](#).
- If installation attempts fail after repeated uninstallation attempts, and you have a Red Hat Technical Support subscription, open a Technical Support case at the [Red Hat Customer Portal](#) and provide an **sosreport** of the server.
- The **sosreport** utility collects configuration details, logs and system information from a RHEL system. For more information about the **sosreport** utility, see [What is an sosreport and how to create one in Red Hat Enterprise Linux?](#).

9.4. ADDITIONAL RESOURCES

- [Troubleshooting IdM replica installation](#)
- [Troubleshooting IdM client installation](#)
- [Backing up and restoring IdM](#)

CHAPTER 10. UNINSTALLING AN IDM SERVER

Follow this procedure to uninstall an Identity Management (IdM) server named **server123.idm.example.com** (server123). In the procedure, you first ensure that other servers are running critical services and that the topology will continue to be redundant before performing the uninstallation.

Prerequisites

- You have **root** access to server123.
- You have an IdM administrator's credentials.

Procedure

1. If your IdM environment uses integrated DNS, ensure that server123 is not the only **enabled** DNS server:

```
[root@server123 ~]# ipa server-role-find --role 'DNS server'
-----
2 server roles matched
-----
Server name: server456.idm.example.com
Role name: DNS server
Role status: enabled
[...]
-----
Number of entries returned 2
-----
```

If server123 is the only remaining DNS server in the topology, add the DNS server role to another IdM server. For more information, see the **ipa-dns-install(1)** man page.

2. If your IdM environment uses an integrated certificate authority (CA):
 - a. Ensure that server123 is not the only **enabled** CA server:

```
[root@server123 ~]# ipa server-role-find --role 'CA server'
-----
2 server roles matched
-----
Server name: server123.idm.example.com
Role name: CA server
Role status: enabled

Server name: r8server.idm.example.com
Role name: CA server
Role status: enabled
-----
Number of entries returned 2
-----
```

If server123 is the only remaining CA server in the topology, add the CA server role to another IdM server. For more information, see the **ipa-ca-install(1)** man page.

- b. If you have enabled vaults in your IdM environment, ensure that `server123.idm.example.com` is not the only **enabled** Key Recovery Authority (KRA) server:

```
[root@server123 ~]# ipa server-role-find --role 'KRA server'
-----
2 server roles matched
-----
Server name: server123.idm.example.com
Role name: KRA server
Role status: enabled

Server name: r8server.idm.example.com
Role name: KRA server
Role status: enabled
-----
Number of entries returned 2
-----
```

If `server123` is the only remaining KRA server in the topology, add the KRA server role to another IdM server. For more information, see **man ipa-kra-install(1)**.

- c. Ensure that `server123.idm.example.com` is not the CA renewal server:

```
[root@server123 ~]# ipa config-show | grep 'CA renewal'
IPA CA renewal master: r8server.idm.example.com
```

If `server123` is the CA renewal server, see [Changing and resetting IdM CA renewal server](#) for more information about how to move the CA renewal server role to another server.

- d. Ensure that `server123.idm.example.com` is not the current certificate revocation list (CRL) publisher:

```
[root@server123 ~]# ipa-crlgen-manage status
CRL generation: disabled
```

If the output shows that CRL generation is enabled on `server123`, see [Generating CRL on an IdM CA server](#) for more information about how to move the CRL publisher role to another server.

3. Connect to another IdM server in the topology:

```
$ ssh idm_user@server456
```

4. On the server, obtain the IdM administrator's credentials:

```
[idm_user@server456 ~]$ kinit admin
```

5. View the DNA ID ranges assigned to the servers in the topology:

```
[idm_user@server456 ~]$ ipa-replica-manage dnrange-show
server123.idm.example.com: 1001-1500
server456.idm.example.com: 1501-2000
[...]
```

The output shows that a DNA ID range is assigned to both server123 and server456.

- If server123 is the only IdM server in the topology with a DNA ID range assigned, create a test IdM user on server456 to ensure that the server has a DNA ID range assigned:

```
[idm_user@server456 ~]$ ipa user-add test_idm_user
```

- Delete server123.idm.example.com from the topology:

```
[idm_user@server456 ~]$ ipa server-del server123.idm.example.com
```



IMPORTANT

If deleting server123 would lead to a disconnected topology, the script warns you about it. For information about how to create a replication agreement between the remaining replicas so that the deletion can proceed, see [Setting up replication between two servers using the CLI](#).



NOTE

Running the **ipa server-del** command removes all replication data and agreements related to server123 for both the **domain** and **ca** suffixes. This is in contrast to Domain Level 0 IdM topologies, where you initially had to remove these data by using the **ipa-replica-manage del server123** command. Domain Level 0 IdM topologies are those running on RHEL 7.2 and earlier. Use the **ipa domainlevel-get** command to view the current domain level.

- Return to server123.idm.example.com and uninstall the existing IdM installation:

```
[root@server123 ~]# ipa-server-install --uninstall
...
Are you sure you want to continue with the uninstall procedure? [no]: yes
```

- Ensure that all name server (NS) DNS records pointing to server123.idm.example.com are deleted from your DNS zones. This applies regardless of whether you use integrated DNS managed by IdM or external DNS. For more information about how to delete DNS records from IdM, see [Deleting DNS records in the IdM CLI](#).

Additional resources

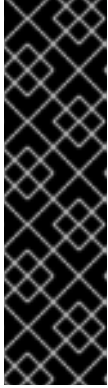
- [Displaying and raising the domain level](#) in RHEL 7 documentation
- [Planning the replica topology](#)
- [Explanation of IdM CA renewal server](#)
- [Generating CRL on an IdM CA server](#)

CHAPTER 11. RENAMING AN IDM SERVER

You cannot change the host name of an existing Identity Management (IdM) server. However, you can replace the server with a replica of a different name.

Procedure

1. Install a new replica that will replace the existing server, ensuring the replica has the required host name and IP address. For details, see [Installing an IdM replica](#) .



IMPORTANT

If the server you are uninstalling is the certificate revocation list (CRL) publisher server, make another server the CRL publisher server before proceeding.

For details on how this is done in the context of a migration procedure, see the following sections:

- [Stopping CRL generation on a RHEL 7 IdM CA server](#)
- [Starting CRL generation on the new RHEL 8 IdM CA server](#)

2. Stop the existing IdM server instance.

```
[root@old_server ~]# ipactl stop
```

3. Uninstall the existing server as described in [Uninstalling an IdM server](#) .

CHAPTER 12. UPDATING AND DOWNGRADING IDM

12.1. UPDATING IDM PACKAGES

You can use the **yum** utility to update the Identity Management (IdM) packages on the system.

Prerequisites

- Ensure you have applied all previously released errata relevant to the RHEL system. For more information, see the [How do I apply package updates to my RHEL system?](#) KCS article.

Procedure

- Select one of the following options:
 - To update all IdM packages that are relevant for your profile and that have updates available:

```
# yum upgrade ipa-*
```
 - To install or update packages to match the latest version available for your profile from any enabled repository:

```
# yum distro-sync ipa-*
```

After you update the IdM packages on at least one server, all other servers in the topology receive the updated schema, even if you do not update their packages. This ensures that any new entries which use the new schema can be replicated among the other servers.



WARNING

When updating multiple IdM servers, wait at least 10 minutes after updating one server before updating another server. However, the actual time required for a server's successful update depends on the topology deployed, the latency of the connections, and the number of changes generated by the update.

When two or more servers are updated simultaneously or with only short intervals between the upgrades, there is not enough time to replicate the post-upgrade data changes throughout the topology, which can result in conflicting replication events.



IMPORTANT

Red Hat recommends upgrading to the next version only. For example, if you want to upgrade to IdM for RHEL 8.8, we recommend upgrading from IdM for RHEL 8.7. Upgrading from earlier versions can cause problems.

12.2. DOWNGRADING IDM PACKAGES

Red Hat does not support downgrading Identity Management.

12.3. ADDITIONAL RESOURCES

- **yum(8)** man page

CHAPTER 13. PREPARING THE SYSTEM FOR IDM CLIENT INSTALLATION

This chapter describes the conditions your system must meet to install an Identity Management (IdM) client.

13.1. SUPPORTED VERSIONS OF RHEL FOR INSTALLING IDM CLIENTS

An Identity Management deployment in which IdM servers are running on the latest minor version of Red Hat Enterprise Linux 8 supports clients that are running on the latest minor versions of:

- RHEL 7
- RHEL 8
- RHEL 9

NOTE

While other client systems, for example Ubuntu, can work with IdM 8 servers, Red Hat does not provide support for these clients.

IMPORTANT

If you are planning to make your IdM deployment FIPS-compliant, Red Hat strongly recommends migrating your environment to RHEL 9. RHEL 9 is the first major RHEL version that is planned to be compliant with FIPS 140-3.

13.2. DNS REQUIREMENTS FOR IDM CLIENTS

Client installer by default tries to search for **_ldap._tcp.DOMAIN** DNS SRV records for all domains that are parent to its hostname. For example, if a client machine has a hostname **client1.idm.example.com**, the installer will try to retrieve an IdM server hostname from **_ldap._tcp.idm.example.com**, **_ldap._tcp.example.com** and **_ldap._tcp.com** DNS SRV records, respectively. The discovered domain is then used to configure client components (for example, SSSD and Kerberos 5 configuration) on the machine.

However, the hostnames of IdM clients are not required to be part of the primary DNS domain. If the client machine hostname is not in a subdomain of an IdM server, pass the IdM domain as the **--domain** option of the **ipa-client-install** command. In that case, after the installation of the client, both SSSD and Kerberos components will have the domain set in their configuration files and will use it to autodiscover IdM servers.

Additional resources

- For details on DNS requirements in IdM, see [Host name and DNS requirements for IdM](#).

13.3. PORT REQUIREMENTS FOR IDM CLIENTS

Identity Management (IdM) clients connect to a number of ports on IdM servers to communicate with their services.

On IdM client, these ports must be open *in the outgoing direction*. If you are using a firewall that does not filter outgoing packets, such as **firewalld**, the ports are already available in the outgoing direction.

Additional resources

- For information about which specific ports are used, see [Port requirements for IdM](#).

13.4. IPV6 REQUIREMENTS FOR IDM CLIENTS

Identity Management (IdM) does not require the **IPv6** protocol to be enabled in the kernel of the host that you want to enroll into IdM. For example, if your internal network only uses the **IPv4** protocol, you can configure the System Security Services Daemon (SSSD) to only use **IPv4** to communicate with the IdM server. You can do this by inserting the following line into the **[domain/NAME]** section of the **/etc/sss/sss.conf** file:

```
lookup_family_order = ipv4_only
```

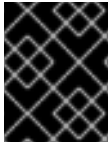
Additional resources

- For more information about the **lookup_family_order** option, see the **sss.conf(5)** man page.

13.5. INSTALLING IDM CLIENT PACKAGES FROM THE IDM:CLIENT STREAM

In RHEL8, the packages necessary for installing an Identity Management (IdM) client are shipped as a module.

The **idm:client** stream is the default stream of the **idm** module. Use this stream to download the IdM client packages if you do not need to install server components on your machine. Using the **idm:client** stream is especially recommended if you need to consistently use IdM client software that is supported long-term, provided you do not need server components, too.

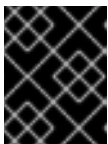


IMPORTANT

Do not use the **idm:client** stream if you are planning to install an IdM replica on the host. In that case, [use the idm:DL1 stream instead](#).

Prerequisites

- When switching to the **idm:client** stream after you previously enabled the **idm:DL1** stream and downloaded packages from it, you need to first explicitly remove all the relevant installed content and disable the **idm:DL1** stream before enabling the **idm:client** stream. For details on how to proceed, see [Switching to a later stream](#).



IMPORTANT

Trying to enable a new stream without disabling the current one results in an error.

Procedure

- To download the packages necessary for installing an IdM client:

```
# yum module install idm
```

13.6. INSTALLING IDM CLIENT PACKAGES FROM THE IDM:DL1 STREAM

In RHEL8, the packages necessary for installing an Identity Management (IdM) client are shipped as a module.

The **idm:DL1** stream needs to be enabled before you can download packages from it. Use this stream to download the IdM client packages if you need to install IdM server components on your machine.

Prerequisites

- When switching to the **idm:DL1** stream after you previously enabled the **idm:client** stream and downloaded packages from it, you need to first explicitly remove all the relevant installed content and disable the **idm:client** stream before enabling the **idm:DL1** stream. For details on how to proceed, see [Switching to a later stream](#).



IMPORTANT

Trying to enable a new stream without disabling the current one results in an error.

Procedure

1. To switch to the RPMs delivered through the **idm:DL1** stream:

```
# yum module enable idm:DL1
# yum distro-sync
```

2. To download the packages necessary for installing an IdM client:

```
# yum module install idm:DL1/client
```

CHAPTER 14. INSTALLING AN IDM CLIENT

The following sections describe how to configure a system as an Identity Management (IdM) client by using the **ipa-client-install** utility. Configuring a system as an IdM client enrolls it into an IdM domain and enables the system to use IdM services on IdM servers in the domain.

To install an Identity Management (IdM) client successfully, you must provide credentials that can be used to enroll the client.

14.1. PREREQUISITES

- You have prepared the system for IdM client installation. For details, see [Preparing the system for IdM client installation](#).

14.2. INSTALLING A CLIENT BY USING USER CREDENTIALS: INTERACTIVE INSTALLATION

Follow this procedure to install an Identity Management (IdM) client interactively by using the credentials of an authorized user to enroll the system into the domain.

Prerequisites

- Ensure you have the credentials of a user authorized to enroll clients into the IdM domain. This could be, for example, a **hostadmin** user with the Enrollment Administrator role.

Procedure

1. Run the **ipa-client-install** utility on the system that you want to configure as an IdM client.

```
# ipa-client-install --mkhomedir
```

Add the **--enable-dns-updates** option to update the DNS records with the IP address of the client system if either of the following conditions applies:

- The IdM server the client will be enrolled with was installed with integrated DNS
- The DNS server on the network accepts DNS entry updates with the GSS-TSIG protocol

```
# ipa-client-install --enable-dns-updates --mkhomedir
```

Enabling DNS updates is useful if your client:

- has a dynamic IP address issued using the Dynamic Host Configuration Protocol
 - has a static IP address but it has just been allocated and the IdM server does not know about it
2. The installation script attempts to obtain all the required settings, such as DNS records, automatically.
 - If the SRV records are set properly in the IdM DNS zone, the script automatically discovers all the other required values and displays them. Enter **yes** to confirm.

```
Client hostname: client.example.com
```

```

Realm: EXAMPLE.COM
DNS Domain: example.com
IPA Server: server.example.com
BaseDN: dc=example,dc=com

```

Continue to configure the system with these values? [no]: yes

- To install the system with different values, enter **no**. Then run **ipa-client-install** again, and specify the required values by adding command-line options to **ipa-client-install**, for example:
 - **--hostname**
 - **--realm**
 - **--domain**
 - **--server**
 - **--mkhomedir**



IMPORTANT

The fully qualified domain name must be a valid DNS name:

- Only numbers, alphabetic characters, and hyphens (-) are allowed. For example, underscores are not allowed and can cause DNS failures.
 - The host name must be all lower-case. No capital letters are allowed.
- If the script fails to obtain some settings automatically, it prompts you for the values.
3. The script prompts for a user whose identity will be used to enroll the client. This could be, for example, a **hostadmin** user with the Enrollment Administrator role:

```

User authorized to enroll computers: hostadmin
Password for hostadmin@EXAMPLE.COM:

```

4. The installation script now configures the client. Wait for the operation to complete.

```

Client configuration complete.

```

Additional resources

- For details on how the client installation script searches for the DNS records, see the **DNS Autodiscovery** section in the **ipa-client-install(1)** man page.

14.3. INSTALLING A CLIENT BY USING A ONE-TIME PASSWORD: INTERACTIVE INSTALLATION

Follow this procedure to install an Identity Management (IdM) client interactively by using a one-time password to enroll the system into the domain.

Prerequisites

- On a server in the domain, add the future client system as an IdM host. Use the **--random** option with the **ipa host-add** command to generate a one-time random password for the enrollment.



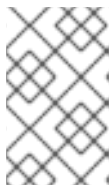
NOTE

The **ipa host-add <client_fqdn>** command requires that the client FQDN is resolvable through DNS. If it is not resolvable, provide the IdM client system's IP address using the **--ip address** option or alternatively, use the **--force** option.

```
$ ipa host-add client.example.com --random
```

```
-----  
Added host "client.example.com"  
-----
```

```
Host name: client.example.com  
Random password: W5YpARI=7M.n  
Password: True  
Keytab: False  
Managed by: server.example.com
```



NOTE

The generated password will become invalid after you use it to enroll the machine into the IdM domain. It will be replaced with a proper host keytab after the enrollment is finished.

Procedure

- Run the **ipa-client-install** utility on the system that you want to configure as an IdM client. Use the **--password** option to provide the one-time random password. Because the password often contains special characters, enclose it in single quotes (').

```
# ipa-client-install --mkhomedir --password=password
```

Add the **--enable-dns-updates** option to update the DNS records with the IP address of the client system if either of the following conditions applies:

- The IdM server the client will be enrolled with was installed with integrated DNS
- The DNS server on the network accepts DNS entry updates with the GSS-TSIG protocol

```
# ipa-client-install --password 'W5YpARI=7M.n' --enable-dns-updates --mkhomedir
```

Enabling DNS updates is useful if your client:

- has a dynamic IP address issued using the Dynamic Host Configuration Protocol
 - has a static IP address but it has just been allocated and the IdM server does not know about it
- The installation script attempts to obtain all the required settings, such as DNS records, automatically.

- If the SRV records are set properly in the IdM DNS zone, the script automatically discovers all the other required values and displays them. Enter **yes** to confirm.

```
Client hostname: client.example.com
Realm: EXAMPLE.COM
DNS Domain: example.com
IPA Server: server.example.com
BaseDN: dc=example,dc=com
```

Continue to configure the system with these values? [no]: **yes**

- To install the system with different values, enter **no**. Then run **ipa-client-install** again, and specify the required values by adding command-line options to **ipa-client-install**, for example:
 - **--hostname**
 - **--realm**
 - **--domain**
 - **--server**
 - **--mkhomedir**



IMPORTANT

The fully qualified domain name must be a valid DNS name:

- Only numbers, alphabetic characters, and hyphens (-) are allowed. For example, underscores are not allowed and can cause DNS failures.
 - The host name must be all lower-case. No capital letters are allowed.
- If the script fails to obtain some settings automatically, it prompts you for the values.

3. The installation script now configures the client. Wait for the operation to complete.

Client configuration complete.

Additional resources

- For details on how the client installation script searches for the DNS records, see the **DNS Autodiscovery** section in the **ipa-client-install(1)** man page.

14.4. INSTALLING A CLIENT: NON-INTERACTIVE INSTALLATION

For a non-interactive installation, you must provide all required information to the **ipa-client-install** utility using command-line options. The following sections describe the minimum required options for a non-interactive installation.

Options for the intended authentication method for client enrollment

The available options are:

- **--principal** and **--password** to specify the credentials of a user authorized to enroll clients
- **--random** to specify a one-time random password generated for the client
- **--keytab** to specify the keytab from a previous enrollment

The option for unattended installation

The **--unattended** option lets the installation run without requiring user confirmation.

If the SRV records are set properly in the IdM DNS zone, the script automatically discovers all the other required values. If the script cannot discover the values automatically, provide them using command-line options, such as:

- **--hostname** to specify a static fully qualified domain name (FQDN) for the client machine.



IMPORTANT

The FQDN must be a valid DNS name:

- Only numbers, alphabetic characters, and hyphens (-) are allowed. For example, underscores are not allowed and can cause DNS failures.
 - The host name must be all lower-case. No capital letters are allowed.
- **--domain** to specify the primary DNS domain of an existing IdM deployment, such as **example.com**. The name is a lowercase version of the IdM Kerberos realm name.
- **--server** to specify the FQDN of the IdM server to connect to. When this option is used, DNS autodiscovery for Kerberos is disabled and a fixed list of KDC and Admin servers is configured. Under normal circumstances, this option is not needed as the list of servers is retrieved from the primary IdM DNS domain.
- **--realm** to specify the Kerberos realm of an existing IdM deployment. Usually it is an uppercase version of the primary DNS domain used by the IdM installation. Under normal circumstances, this option is not needed as the realm name is retrieved from the IdM server.

An example of a basic **ipa-client-install** command for non-interactive installation:

```
# ipa-client-install --password 'W5YpARI=7M.n' --mkhomedir --unattended
```

An example of an **ipa-client-install** command for non-interactive installation with more options specified:

```
# ipa-client-install --password 'W5YpARI=7M.n' --domain idm.example.com --server server.idm.example.com --realm IDM.EXAMPLE.COM --mkhomedir --unattended
```

Additional resources

- For a complete list of options accepted by **ipa-client-install**, see the **ipa-client-install(1)** man page.

14.5. REMOVING PRE-IDM CONFIGURATION AFTER INSTALLING A CLIENT

The **ipa-client-install** script does not remove any previous LDAP and System Security Services Daemon (SSSD) configuration from the **/etc/openldap/ldap.conf** and **/etc/sss/sss.conf** files. If you modified the configuration in these files before installing the client, the script adds the new client values, but comments them out. For example:

```
BASE dc=example,dc=com
URI ldap://ldap.example.com

#URI ldaps://server.example.com # modified by IPA
#BASE dc=ipa,dc=example,dc=com # modified by IPA
```

To apply the new Identity Management (IdM) configuration values:

1. Open **/etc/openldap/ldap.conf** and **/etc/sss/sss.conf**.
2. Delete the previous configuration.
3. Uncomment the new IdM configuration.
4. Server processes that rely on system-wide LDAP configuration might require a restart to apply the changes. Applications that use **openldap** libraries typically import the configuration when started.

14.6. TESTING AN IDM CLIENT

The command-line interface informs you that the **ipa-client-install** was successful, but you can also do your own test.

To test that the Identity Management (IdM) client can obtain information about users defined on the server, check that you are able to resolve a user defined on the server. For example, to check the default **admin** user:

```
[user@client ~]$ id admin
uid=1254400000(admin) gid=1254400000(admins) groups=1254400000(admins)
```

To test that authentication works correctly, **su** to a root user from a non-root user:

```
[user@client ~]$ su -
Last login: Thu Oct 18 18:39:11 CEST 2018 from 192.168.122.1 on pts/0
[root@client ~]#
```

14.7. CONNECTIONS PERFORMED DURING AN IDM CLIENT INSTALLATION

[Requests performed during an IdM client installation](#) lists the operations performed by **ipa-client-install**, the Identity Management (IdM) client installation tool.

Table 14.1. Requests performed during an IdM client installation

Operation	Protocol used	Purpose
DNS resolution against the DNS resolvers configured on the client system	DNS	To discover the IP addresses of IdM servers; (optionally) to add A/AAAA and SSHFP records
Requests to ports 88 (TCP/TCP6 and UDP/UDP6) on an IdM replica	Kerberos	To obtain a Kerberos ticket
JSON-RPC calls to the IdM Apache-based web-service on discovered or configured IdM servers	HTTPS	IdM client enrollment; retrieval of CA certificate chain if LDAP method fails; request for a certificate issuance if required
Requests over TCP/TCP6 to ports 389 on IdM servers, using SASL GSSAPI authentication, plain LDAP, or both	LDAP	IdM client enrollment; identity retrieval by SSSD processes; Kerberos key retrieval for the host principal
Network time protocol (NTP) discovery and resolution (optionally)	NTP	To synchronize time between the client system and an NTP server

14.8. IDM CLIENT'S COMMUNICATIONS WITH THE SERVER DURING POST-INSTALLATION DEPLOYMENT

The client side of Identity Management (IdM) framework is implemented with two different applications:

- The **ipa** command-line interface (CLI)
- (*optional*) the browser-based Web UI

[CLI post-installation operations](#) shows the operations performed by the CLI during an IdM client post-installation deployment. [Web UI post-installation operations](#) shows the operations performed by the Web UI during an IdM client post-installation deployment.

Table 14.2. CLI post-installation operations

Operation	Protocol used	Purpose
DNS resolution against the DNS resolvers configured on the client system	DNS	To discover the IP addresses of IdM servers
Requests to ports 88 (TCP/TCP6 and UDP/UDP6) and 464 (TCP/TCP6 and UDP/UDP6) on an IdM replica	Kerberos	To obtain a Kerberos ticket; change a Kerberos password; authenticate to the IdM Web UI
JSON-RPC calls to the IdM Apache-based web-service on discovered or configured IdM servers	HTTPS	any ipa utility usage

Table 14.3. Web UI post-installation operations

Operation	Protocol used	Purpose
JSON-RPC calls to the IdM Apache-based web-service on discovered or configured IdM servers	HTTPS	To retrieve the IdM Web UI pages

Additional resources

- [SSSD communication patterns](#) for more information about how the **SSSD** daemon communicates with the services available on the IdM and Active Directory servers.
- [Certmonger communication patterns](#) for more information about how the **certmonger** daemon communicates with the services available on the IdM and Active Directory servers.

14.9. SSSD COMMUNICATION PATTERNS

The System Security Services Daemon (SSSD) is a system service to access remote directories and authentication mechanisms. If configured on an Identity Management IdM client, it connects to the IdM server, which provides authentication, authorization and other identity and policy information. If the IdM server is in a trust relationships with Active Directory (AD), SSSD also connects to AD to perform authentication for AD users using the Kerberos protocol. By default, SSSD uses Kerberos to authenticate any non-local user. In special situations, SSSD might be configured to use the LDAP protocol instead.

The SSSD can be configured to communicate with multiple servers. The tables below show common communication patterns for SSSD in IdM.

Table 14.4. Communication patterns of SSSD on IdM clients when talking to IdM servers

Operation	Protocol used	Purpose
DNS resolution against the DNS resolvers configured on the client system	DNS	To discover the IP addresses of IdM servers
Requests to ports 88 (TCP/TCP6 and UDP/UDP6), 464 (TCP/TCP6 and UDP/UDP6), and 749 (TCP/TCP6) on an Identity Management replica and Active Directory domain controllers	Kerberos	To obtain a Kerberos ticket; to change a Kerberos password
Requests over TCP/TCP6 to ports 389 on IdM servers, using SASL GSSAPI authentication, plain LDAP, or both	LDAP	To obtain information about IdM users and hosts, download HBAC and sudo rules, automount maps, the SELinux user context, public SSH keys, and other information stored in IdM LDAP

Operation	Protocol used	Purpose
(optionally) In case of smart-card authentication, requests to the Online Certificate Status Protocol (OCSP) responder, if it is configured. This often is done via port 80, but it depends on the actual value of the OCSP responder URL in a client certificate.	HTTP	To obtain information about the status of the certificate installed in the smart card

Table 14.5. Communication patterns of SSSD on IdM servers acting as trust agents when talking to Active Directory Domain Controllers

Operation	Protocol used	Purpose
DNS resolution against the DNS resolvers configured on the client system	DNS	To discover the IP addresses of IdM servers
Requests to ports 88 (TCP/TCP6 and UDP/UDP6), 464 (TCP/TCP6 and UDP/UDP6), and 749 (TCP/TCP6) on an Identity Management replica and Active Directory domain controllers	Kerberos	To obtain a Kerberos ticket; change a Kerberos password; administer Kerberos remotely
Requests to ports 389 (TCP/TCP6 and UDP/UDP6) and 3268 (TCP/TCP6)	LDAP	To query Active Directory user and group information; to discover Active Directory domain controllers
(optionally) In case of smart-card authentication, requests to the Online Certificate Status Protocol (OCSP) responder, if it is configured. This often is done via port 80, but it depends on the actual value of the OCSP responder URL in a client certificate.	HTTP	To obtain information about the status of the certificate installed in the smart card

Additional resources

- [IdM client's communications with the server during post-installation deployment](#)

14.10. CERTMONGER COMMUNICATION PATTERNS

Certmonger is a daemon running on Identity Management (IdM) servers and IdM clients to allow a timely renewal of SSL certificates associated with the services on the host. The [Table 14.6, "Certmonger communication patterns"](#) shows the operations performed by the **certmonger** utility on IdM servers.

Table 14.6. Certmonger communication patterns

Operation	Protocol used	Purpose
DNS resolution against the DNS resolvers configured on the client system	DNS	To discover the IP addresses of IdM servers
Requests to ports 88 (TCP/TCP6 and UDP/UDP6) and 464 (TCP/TCP6 and UDP/UDP6) on an IdM replica	Kerberos	To obtain a Kerberos ticket
JSON-RPC calls to the IdM Apache-based web-service on discovered or configured IdM servers	HTTPS	To request new certificates
Access over port 8080 (TCP/TCP6) on the IdM server	HTTP	To obtain an Online Certificate Status Protocol (OCSP) responder and certificate status
(on the first installed server or on the server where certificate tracking has been transferred) Access over port 8443 (TCP/TCP6) on the IdM server	HTTPS	To administer the Certificate Authority on the IdM server (only during IdM server and replica installation). certmonger on the server contacts only its own local server on ports 8080 and 8443 for CA-related certificate renewal.

Additional resources

- [IdM client's communications with the server during post-installation deployment](#)

CHAPTER 15. INSTALLING AN IDM CLIENT WITH KICKSTART

A Kickstart enrollment automatically adds a new system to the Identity Management (IdM) domain at the time Red Hat Enterprise Linux is installed.

15.1. INSTALLING A CLIENT WITH KICKSTART

Follow this procedure to use a Kickstart file to install an Identity Management (IdM) client.

Prerequisites

- Do not start the **sshd** service prior to the kickstart enrollment. Starting **sshd** before enrolling the client generates the SSH keys automatically, but the Kickstart file in [Section 15.2, “Kickstart file for client installation”](#) uses a script for the same purpose, which is the preferred solution.

Procedure

1. Pre-create the host entry on the IdM server, and set a temporary password for the entry:

```
$ ipa host-add client.example.com --password=secret
```

The password is used by Kickstart to authenticate during the client installation and expires after the first authentication attempt. After the client is successfully installed, it authenticates using its keytab.

2. Create a Kickstart file with the contents described in [Section 15.2, “Kickstart file for client installation”](#). Make sure that network is configured properly in the Kickstart file using the **network** command.
3. Use the Kickstart file to install the IdM client.

15.2. KICKSTART FILE FOR CLIENT INSTALLATION

You can use a Kickstart file to install an Identity Management (IdM) client. The contents of the Kickstart file must meet certain requirements as outlined here.

The **ipa-client** package in the list of packages to install

Add the **ipa-client** package to the `%packages` section of the Kickstart file. For example:

```
%packages
...
ipa-client
...
```

Post-installation instructions for the IdM client

The post-installation instructions must include:

- An instruction for ensuring SSH keys are generated before enrollment
- An instruction to run the **ipa-client-install** utility, while specifying:
 - All the required information to access and configure the IdM domain services

- The password which you set when pre-creating the client host on the IdM server. in [Section 15.1, “Installing a client with Kickstart”](#) .

For example, the post-installation instructions for a Kickstart installation that uses a one-time password and retrieves the required options from the command line rather than via DNS can look like this:

```
%post --log=/root/ks-post.log

# Generate SSH keys; ipa-client-install uploads them to the IdM server by default
/usr/libexec/openssh/sshd-keygen rsa

# Run the client install script
/usr/sbin/ipa-client-install --hostname=client.example.com --domain=EXAMPLE.COM --enable-
dns-updates --mkhomedir -w secret --realm=EXAMPLE.COM --server=server.example.com
```

Optionally, you can also include other options in the Kickstart file, such as:

- For a non-interactive installation, add the **--unattended** option to **ipa-client-install**.
- To let the client installation script request a certificate for the machine:
 - Add the **--request-cert** option to **ipa-client-install**.
 - Set the system bus address to **/dev/null** for both the **getcert** and **ipa-client-install** utility in the Kickstart **chroot** environment. To do this, add these lines to the post-installation instructions in the Kickstart file before the **ipa-client-install** instruction:

```
# env DBUS_SYSTEM_BUS_ADDRESS=unix:path=/dev/null getcert list
# env DBUS_SYSTEM_BUS_ADDRESS=unix:path=/dev/null ipa-client-install
```

15.3. TESTING AN IDM CLIENT

The command-line interface informs you that the **ipa-client-install** was successful, but you can also do your own test.

To test that the Identity Management (IdM) client can obtain information about users defined on the server, check that you are able to resolve a user defined on the server. For example, to check the default **admin** user:

```
[user@client ~]$ id admin
uid=1254400000(admin) gid=1254400000(admins) groups=1254400000(admins)
```

To test that authentication works correctly, **su** to a root user from a non-root user:

```
[user@client ~]$ su -
Last login: Thu Oct 18 18:39:11 CEST 2018 from 192.168.122.1 on pts/0
[root@client ~]#
```

CHAPTER 16. TROUBLESHOOTING IDM CLIENT INSTALLATION

The following sections describe how to gather information about a failing IdM client installation, and how to resolve common installation issues.

16.1. REVIEWING IDM CLIENT INSTALLATION ERRORS

When you install an Identity Management (IdM) client, debugging information is appended to `/var/log/ipaclient-install.log`. If a client installation fails, the installer logs the failure and rolls back changes to undo any modifications to the host. The reason for the installation failure may not be at the end of the log file, as the installer also logs the roll back procedure.

To troubleshoot a failing IdM client installation, review lines labeled **ScriptError** in the `/var/log/ipaclient-install.log` file and use this information to resolve any corresponding issues.

Prerequisites

- You must have **root** privileges to display the contents of IdM log files.

Procedure

1. Use the **grep** utility to retrieve any occurrences of the keyword **ScriptError** from the `/var/log/ipaserver-install.log` file.

```
[user@server ~]$ sudo grep ScriptError /var/log/ipaclient-install.log
[sudo] password for user:
2020-05-28T18:24:50Z DEBUG The ipa-client-install command failed, exception:
ScriptError: One of password / principal / keytab is required.
```

2. To review a log file interactively, open the end of the log file using the **less** utility and use the `↑` and `↓` arrow keys to navigate.

```
[user@server ~]$ sudo less -N +G /var/log/ipaclient-install.log
```

Additional resources

- If you are unable to resolve a failing IdM client installation, and you have a Red Hat Technical Support subscription, open a Technical Support case at the [Red Hat Customer Portal](#) and provide an **sosreport** of the client.
- The **sosreport** utility collects configuration details, logs and system information from a RHEL system. For more information about the **sosreport** utility, see [What is an sosreport and how to create one in Red Hat Enterprise Linux?](#)

16.2. RESOLVING ISSUES IF THE CLIENT INSTALLATION FAILS TO UPDATE DNS RECORDS

The IdM client installer issues **nsupdate** commands to create PTR, SSHFP, and additional DNS records. However, the installation process fails if the client is unable to update DNS records after installing and configuring the client software.

To fix this problem, verify the configuration and review DNS errors in `/var/log/client-install.log`.

Prerequisites

- You are using IdM DNS as the DNS solution for your IdM environment

Procedure

1. Ensure that dynamic updates for the DNS zone the client is in are enabled:

```
[user@server ~]$ ipa dnszone-mod idm.example.com. --dynamic-update=TRUE
```

2. Ensure that the IdM server running the DNS service has port 53 opened for both TCP and UDP protocols.

```
[user@server ~]$ sudo firewall-cmd --permanent --add-port=53/tcp --add-port=53/udp
[sudo] password for user:
success
[user@server ~]$ firewall-cmd --runtime-to-permanent
success
```

3. Use the **grep** utility to retrieve the contents of **nsupdate** commands from `/var/log/client-install.log` to see which DNS record updates are failing.

```
[user@server ~]$ sudo grep nsupdate /var/log/ipaclient-install.log
```

Additional resources

- If you are unable to resolve a failing installation, and you have a Red Hat Technical Support subscription, open a Technical Support case at the [Red Hat Customer Portal](#) and provide an **sosreport** of the client.
- The **sosreport** utility collects configuration details, logs and system information from a RHEL system. For more information about the **sosreport** utility, see [What is an sosreport and how to create one in Red Hat Enterprise Linux?](#).

16.3. RESOLVING ISSUES IF THE CLIENT INSTALLATION FAILS TO JOIN THE IDM KERBEROS REALM

The IdM client installation process fails if the client is unable to join the IdM Kerberos realm.

```
Joining realm failed: Failed to add key to the keytab
child exited with 11
```

```
Installation failed. Rolling back changes.
```

This failure can be caused by an empty Kerberos keytab.

Prerequisites

- Removing system files requires **root** privileges.

Procedure

1. Remove **/etc/krb5.keytab**.

```
[user@client ~]$ sudo rm /etc/krb5.keytab
[sudo] password for user:
[user@client ~]$ ls /etc/krb5.keytab
ls: cannot access '/etc/krb5.keytab': No such file or directory
```

2. Retry the IdM client installation.

Additional resources

- If you are unable to resolve a failing installation, and you have a Red Hat Technical Support subscription, open a Technical Support case at the [Red Hat Customer Portal](#) and provide an **sosreport** of the client.
- The **sosreport** utility collects configuration details, logs and system information from a RHEL system. For more information about the **sosreport** utility, see [What is an sosreport and how to create one in Red Hat Enterprise Linux?](#).

16.4. ADDITIONAL RESOURCES

- To troubleshoot installing the first IdM server, see [Troubleshooting IdM server installation](#).
- To troubleshoot installing an IdM replica, see [Troubleshooting IdM replica installation](#).

CHAPTER 17. RE-ENROLLING AN IDM CLIENT

If a client machine has been destroyed and lost connection with the IdM servers, for example due to the client's hardware failure, and you still have its keytab, you can re-enroll the client. In this scenario, you want to get the client back in the IdM environment with the same hostname.

17.1. CLIENT RE-ENROLLMENT IN IDM

If a client machine has been destroyed and lost connection with the IdM servers, for example due to the client's hardware failure, and you still have its keytab, you can re-enroll the client. In this scenario, you want to get the client back in the IdM environment with the same hostname.

During the re-enrollment, the client generates a new Kerberos key and SSH keys, but the identity of the client in the LDAP database remains unchanged. After the re-enrollment, the host has its keys and other information in the same LDAP object with the same **FQDN** as previously, before the machine's loss of connection with the IdM servers.



IMPORTANT

You can only re-enroll clients whose domain entry is still active. If you uninstalled a client (using **ipa-client-install --uninstall**) or disabled its host entry (using **ipa host-disable**), you cannot re-enroll it.

You cannot re-enroll a client after you have renamed it. This is because in IdM, the key attribute of the client's entry in LDAP is the client's hostname, its **FQDN**. As opposed to re-enrolling a client, during which the client's LDAP object remains unchanged, the outcome of renaming a client is that the client has its keys and other information in a different LDAP object with a new **FQDN**. Therefore, the only way to rename a client is to uninstall the host from IdM, change the host's hostname, and install it as an IdM client with a new name. For details on how to rename a client, see [Renaming IdM client systems](#).

What happens during client re-enrollment

During re-enrollment, IdM:

- Revokes the original host certificate
- Creates new SSH keys
- Generates a new keytab

17.2. RE-ENROLLING A CLIENT BY USING USER CREDENTIALS: INTERACTIVE RE-ENROLLMENT

Follow this procedure to re-enroll an Identity Management (IdM) client interactively by using the credentials of an authorized user.

1. Re-create the client machine with the same host name.
2. Run the **ipa-client-install --force-join** command on the client machine:

```
# ipa-client-install --force-join
```

3. The script prompts for a user whose identity will be used to re-enroll the client. This could be, for example, a **hostadmin** user with the Enrollment Administrator role:

User authorized to enroll computers: **hostadmin**
 Password for **hostadmin@EXAMPLE.COM**:

Additional resources

- For a more detailed procedure on enrolling clients by using an authorized user's credentials, see [Installing a client by using user credentials: Interactive installation](#) .

17.3. RE-ENROLLING A CLIENT BY USING THE CLIENT KEYTAB: NON-INTERACTIVE RE-ENROLLMENT

Prerequisites

- Back up the original client keytab file, for example in the **/tmp** or **/root** directory.

Procedure

Follow this procedure to re-enroll an Identity Management (IdM) client non-interactively by using the keytab of the client system. For example, re-enrollment using the client keytab is appropriate for an automated installation.

1. Re-create the client machine with the same host name.
2. Copy the keytab file from the backup location to the **/etc/** directory on the re-created client machine.
3. Use the **ipa-client-install** utility to re-enroll the client, and specify the keytab location with the **-keytab** option:

```
# ipa-client-install --keytab /etc/krb5.keytab
```



NOTE

The keytab specified in the **--keytab** option is only used when authenticating to initiate the enrollment. During the re-enrollment, IdM generates a new keytab for the client.

17.4. TESTING AN IDM CLIENT

The command-line interface informs you that the **ipa-client-install** was successful, but you can also do your own test.

To test that the Identity Management (IdM) client can obtain information about users defined on the server, check that you are able to resolve a user defined on the server. For example, to check the default **admin** user:

```
[user@client ~]$ id admin
uid=1254400000(admin) gid=1254400000(admins) groups=1254400000(admins)
```

To test that authentication works correctly, **su** to a root user from a non-root user:

```
[user@client ~]$ su -  
Last login: Thu Oct 18 18:39:11 CEST 2018 from 192.168.122.1 on pts/0  
[root@client ~]#
```

CHAPTER 18. UNINSTALLING AN IDM CLIENT

As an administrator, you can remove an Identity Management (IdM) client from the environment.

18.1. UNINSTALLING AN IDM CLIENT

Uninstalling a client removes the client from the Identity Management (IdM) domain, along with all of the specific IdM configuration of system services, such as System Security Services Daemon (SSSD). This restores the previous configuration of the client system.

Procedure

1. Enter the **ipa-client-install --uninstall** command:

```
[root@client ~]# ipa-client-install --uninstall
```

2. Optional: Check that you cannot obtain a Kerberos ticket-granting ticket (TGT) for an IdM user:

```
[root@client ~]# kinit admin
kinit: Client 'admin@EXAMPLE.COM' not found in Kerberos database while getting initial
credentials
[root@client ~]#
```

If a Kerberos TGT ticket has been returned successfully, follow the additional uninstallation steps in [Uninstalling an IdM client: additional steps after multiple past installations](#) .

3. On the client, remove old Kerberos principals from each identified keytab other than **/etc/krb5.keytab**:

```
[root@client ~]# ipa-rmkeytab -k /path/to/keytab -r EXAMPLE.COM
```

4. On an IdM server, remove all DNS entries for the client host from IdM:

```
[root@server ~]# ipa dnsrecord-del
Record name: old-client-name
Zone name: idm.example.com
No option to delete specific record provided.
Delete all? Yes/No (default No): yes
-----
Deleted record "old-client-name"
```

5. On the IdM server, remove the client host entry from the IdM LDAP server. This removes all services and revokes all certificates issued for that host:

```
[root@server ~]# ipa host-del client.idm.example.com
```



IMPORTANT

Removing the client host entry from the IdM LDAP server is crucial if you think you might re-enroll the client in the future, with a different IP address or a different hostname.

18.2. UNINSTALLING AN IDM CLIENT: ADDITIONAL STEPS AFTER MULTIPLE PAST INSTALLATIONS

If you install and uninstall a host as an Identity Management (IdM) client multiple times, the uninstallation procedure might not restore the pre-IdM Kerberos configuration.

In this situation, you must manually remove the IdM Kerberos configuration. In extreme cases, you must reinstall the operating system.

Prerequisites

- You have used the **ipa-client-install --uninstall** command to uninstall the IdM client configuration from the host. However, you can still obtain a Kerberos ticket-granting ticket (TGT) for an IdM user from the IdM server.
- You have checked that the **/var/lib/ipa-client/sysrestore** directory is empty and hence you cannot restore the prior-to-IdM-client configuration of the system using the files in the directory.

Procedure

1. Check the **/etc/krb5.conf.ipa** file:

- If the contents of the **/etc/krb5.conf.ipa** file are the same as the contents of the **krb5.conf** file prior to the installation of the IdM client, you can:

i. Remove the **/etc/krb5.conf** file:

```
# rm /etc/krb5.conf
```

ii. Rename the **/etc/krb5.conf.ipa** file into **/etc/krb5.conf**:

```
# mv /etc/krb5.conf.ipa /etc/krb5.conf
```

- If the contents of the **/etc/krb5.conf.ipa** file are not the same as the contents of the **krb5.conf** file prior to the installation of the IdM client, you can at least restore the Kerberos configuration to the state directly after the installation of the operating system:

i. Re-install the **krb5-libs** package:

```
# yum reinstall krb5-libs
```

As a dependency, this command will also re-install the **krb5-workstation** package and the original version of the **/etc/krb5.conf** file.

2. Remove the **var/log/ipaclient-install.log** file if present.

Verification steps

- Try to obtain IdM user credentials. This should fail:

```
[root@r8server ~]# kinit admin
kinit: Client 'admin@EXAMPLE.COM' not found in Kerberos database while getting initial
credentials
```

```
| [root@r8server ~]#
```

The **/etc/krb5.conf** file is now restored to its factory state. As a result, you cannot obtain a Kerberos TGT for an IdM user on the host.

CHAPTER 19. RENAMING IDM CLIENT SYSTEMS

The following sections describe how to change the host name of an Identity Management (IdM) client system.



WARNING

Renaming a client is a manual procedure. Do not perform it unless changing the host name is absolutely required.

Renaming an IdM client involves:

1. Preparing the host. For details, see [Preparing an IdM client for its renaming](#) .
2. Uninstalling the IdM client from the host. For details, see [Uninstalling a client](#).
3. Renaming the host. For details, see [Renaming a client](#).
4. Installing the IdM client on the host with the new name. For details, see [Reinstalling a client](#).
5. Configuring the host after the IdM client installation. For details, see [Re-adding services, re-generating certificates, and re-adding host groups](#).

19.1. PREPARING AN IDM CLIENT FOR ITS RENAMING

Before uninstalling the current client, make note of certain settings for the client. You will apply this configuration after re-enrolling the machine with a new host name.

- Identify which services are running on the machine:
 - Use the **ipa service-find** command, and identify services with certificates in the output:

```
$ ipa service-find old-client-name.example.com
```

- In addition, each host has a default *host service* which does not appear in the **ipa service-find** output. The service principal for the host service, also called a *host principal*, is **host/old-client-name.example.com**.
- For all service principals displayed by **ipa service-find old-client-name.example.com**, determine the location of the corresponding keytabs on the **old-client-name.example.com** system:

```
# find / -name "*.keytab"
```

Each service on the client system has a Kerberos principal in the form *service_name/host_name@REALM*, such as **ldap/old-client-name.example.com@EXAMPLE.COM**.

- Identify all host groups to which the machine belongs.

```
# ipa hostgroup-find old-client-name.example.com
```

19.2. UNINSTALLING AN IDM CLIENT

Uninstalling a client removes the client from the Identity Management (IdM) domain, along with all of the specific IdM configuration of system services, such as System Security Services Daemon (SSSD). This restores the previous configuration of the client system.

Procedure

1. Enter the **ipa-client-install --uninstall** command:

```
[root@client ~]# ipa-client-install --uninstall
```

2. Optional: Check that you cannot obtain a Kerberos ticket-granting ticket (TGT) for an IdM user:

```
[root@client ~]# kinit admin
kinit: Client 'admin@EXAMPLE.COM' not found in Kerberos database while getting initial
credentials
[root@client ~]#
```

If a Kerberos TGT ticket has been returned successfully, follow the additional uninstallation steps in [Uninstalling an IdM client: additional steps after multiple past installations](#) .

3. On the client, remove old Kerberos principals from each identified keytab other than **/etc/krb5.keytab**:

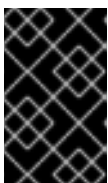
```
[root@client ~]# ipa-rmkeytab -k /path/to/keytab -r EXAMPLE.COM
```

4. On an IdM server, remove all DNS entries for the client host from IdM:

```
[root@server ~]# ipa dnsrecord-del
Record name: old-client-name
Zone name: idm.example.com
No option to delete specific record provided.
Delete all? Yes/No (default No): yes
-----
Deleted record "old-client-name"
```

5. On the IdM server, remove the client host entry from the IdM LDAP server. This removes all services and revokes all certificates issued for that host:

```
[root@server ~]# ipa host-del client.idm.example.com
```



IMPORTANT

Removing the client host entry from the IdM LDAP server is crucial if you think you might re-enroll the client in the future, with a different IP address or a different hostname.

19.3. UNINSTALLING AN IDM CLIENT: ADDITIONAL STEPS AFTER MULTIPLE PAST INSTALLATIONS

If you install and uninstall a host as an Identity Management (IdM) client multiple times, the uninstallation procedure might not restore the pre-IdM Kerberos configuration.

In this situation, you must manually remove the IdM Kerberos configuration. In extreme cases, you must reinstall the operating system.

Prerequisites

- You have used the **ipa-client-install --uninstall** command to uninstall the IdM client configuration from the host. However, you can still obtain a Kerberos ticket-granting ticket (TGT) for an IdM user from the IdM server.
- You have checked that the **/var/lib/ipa-client/sysrestore** directory is empty and hence you cannot restore the prior-to-IdM-client configuration of the system using the files in the directory.

Procedure

1. Check the **/etc/krb5.conf.ipa** file:

- If the contents of the **/etc/krb5.conf.ipa** file are the same as the contents of the **krb5.conf** file prior to the installation of the IdM client, you can:

i. Remove the **/etc/krb5.conf** file:

```
# rm /etc/krb5.conf
```

ii. Rename the **/etc/krb5.conf.ipa** file into **/etc/krb5.conf**:

```
# mv /etc/krb5.conf.ipa /etc/krb5.conf
```

- If the contents of the **/etc/krb5.conf.ipa** file are not the same as the contents of the **krb5.conf** file prior to the installation of the IdM client, you can at least restore the Kerberos configuration to the state directly after the installation of the operating system:

i. Re-install the **krb5-libs** package:

```
# yum reinstall krb5-libs
```

As a dependency, this command will also re-install the **krb5-workstation** package and the original version of the **/etc/krb5.conf** file.

2. Remove the **var/log/ipaclient-install.log** file if present.

Verification steps

- Try to obtain IdM user credentials. This should fail:

```
[root@r8server ~]# kinit admin
kinit: Client 'admin@EXAMPLE.COM' not found in Kerberos database while getting initial
credentials
[root@r8server ~]#
```

■

The `/etc/krb5.conf` file is now restored to its factory state. As a result, you cannot obtain a Kerberos TGT for an IdM user on the host.

19.4. RENAMING THE HOST SYSTEM

Rename the machine as required. For example:

```
# hostnamectl set-hostname new-client-name.example.com
```

You can now re-install the Identity Management (IdM) client to the IdM domain with the new host name.

19.5. RE-INSTALLING AN IDM CLIENT

Install an client on your renamed host following the procedure described in [Installing a client](#).

19.6. RE-ADDING SERVICES, RE-GENERATING CERTIFICATES, AND RE-ADDING HOST GROUPS

Procedure

1. On the Identity Management (IdM) server, add a new keytab for every service identified in the [Preparing an IdM client for its renaming](#).

```
[root@server ~]# ipa service-add service_name/new-client-name
```

2. Generate certificates for services that had a certificate assigned in the [Preparing an IdM client for its renaming](#). You can do this:
 - Using the IdM administration tools
 - Using the `certmonger` utility
3. Re-add the client to the host groups identified in the [Preparing an IdM client for its renaming](#).

CHAPTER 20. PREPARING THE SYSTEM FOR IDM REPLICA INSTALLATION

The following links list the requirements to install an Identity Management (IdM) replica. Before the installation, verify your system meets these requirements.

1. Ensure [the target system meets the general requirements for IdM server installation](#) .
2. Ensure [the target system meets the additional, version requirements for IdM replica installation](#) .
3. Authorize the target system for enrollment into the IdM domain. For more information, see one of the following sections that best fits your needs:
 - [Authorizing the installation of a replica on an IdM client](#)
 - [Authorizing the installation of a replica on a system that is not enrolled into IdM](#)

Additional resources

- [Planning the replica topology](#)

20.1. REPLICA VERSION REQUIREMENTS

Red Hat Enterprise Linux (RHEL) 8 replicas only work with Identity Management (IdM) servers running on RHEL 7.4 and later. Before introducing IdM replicas running on RHEL 8 into an existing deployment, upgrade all IdM servers to RHEL 7.4 or later, and change the domain level to 1.

In addition, the replica must be running the same or later version of IdM. For example:

- You have an IdM server installed on Red Hat Enterprise Linux 8 and it uses IdM 4.x packages.
- You must install the replica also on Red Hat Enterprise Linux 8 or later and use IdM version 4.x or later.

This ensures that configuration can be properly copied from the server to the replica.

For details on how to display the IdM software version, see [Methods for displaying IdM software version](#) .

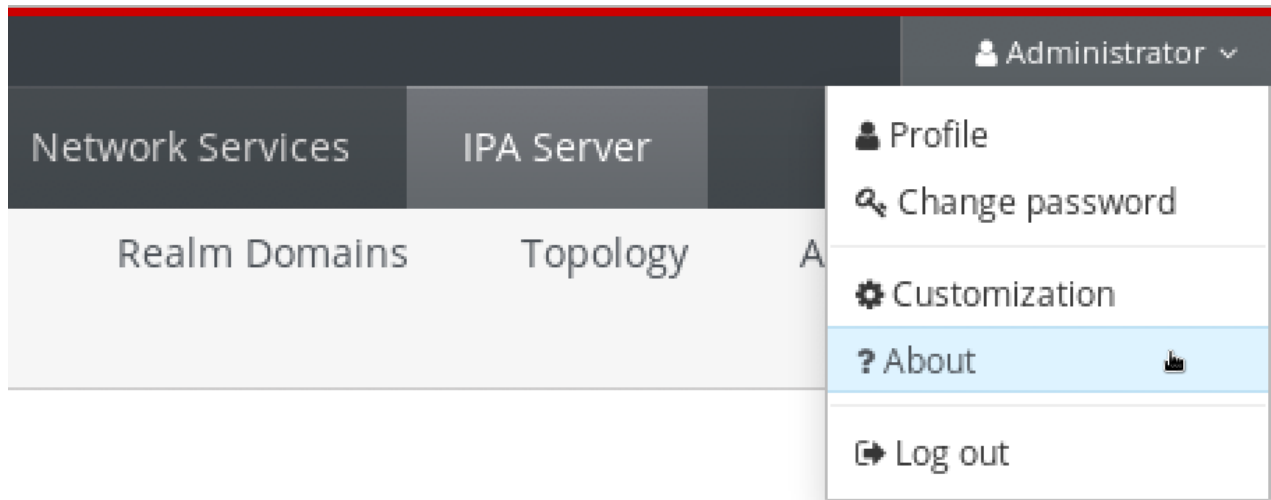
20.2. METHODS FOR DISPLAYING IDM SOFTWARE VERSION

You can display the IdM version number with:

- The IdM WebUI
- **ipa** commands
- **rpm** commands

Displaying version through the WebUI

In the IdM WebUI, the software version can be displayed by choosing **About** from the username menu at the upper-right.



Displaying version with ipa commands

From the command line, use the **ipa --version** command.

```
[root@server ~]# ipa --version
VERSION: 4.8.0, API_VERSION: 2.233
```

Displaying version with rpm commands

If IdM services are not operating properly, you can use the **rpm** utility to determine the version number of the **ipa-server** package that is currently installed.

```
[root@server ~]# rpm -q ipa-server
ipa-server-4.8.0-11.module+el8.1.0+4247+9f3fd721.x86_64
```

20.3. AUTHORIZING THE INSTALLATION OF A REPLICA ON AN IDM CLIENT

When [installing a replica](#) on an existing Identity Management (IdM) client by running the **ipa-replica-install** utility, choose **Method 1** or **Method 2** below to authorize the replica installation. Choose **Method 1** if one of the following applies:

- You want a senior system administrator to perform the initial part of the procedure and a junior administrator to perform the rest.
- You want to automate your replica installation.

Method 1: theipaservers host group

1. Log in to any IdM host as IdM admin:

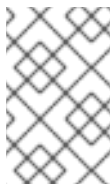
```
$ kinit admin
```

2. Add the client machine to the **ipaservers** host group:

```
$ ipa hostgroup-add-member ipaservers --hosts client.idm.example.com
Host-group: ipaservers
Description: IPA server hosts
```


Member hosts: server.idm.example.com, client.idm.example.com

Number of members added 1



NOTE

Membership in the **ipaservers** group grants the machine elevated privileges similar to the administrator's credentials. Therefore, in the next step, the **ipa-replica-install** utility can be run on the host successfully by a junior system administrator.

Method 2: a privileged user's credentials

Choose one of the following methods to authorize the replica installation by providing a privileged user's credentials:

- Let Identity Management (IdM) prompt you for the credentials interactively after you start the **ipa-replica-install** utility. This is the default behavior.
- Log in to the client as a privileged user immediately before running the **ipa-replica-install** utility. The default privileged user is **admin**:

\$ kinit admin

Additional resources

- To start the installation procedure, see [Installing an IdM replica](#).
- You can use an Ansible playbook to install IdM replicas. For more information, see [Installing an Identity Management replica using an Ansible playbook](#).

20.4. AUTHORIZING THE INSTALLATION OF A REPLICA ON A SYSTEM THAT IS NOT ENROLLED INTO IDM

When [installing a replica](#) on a system that is not enrolled in the Identity Management (IdM) domain, the **ipa-replica-install** utility first enrolls the system as a client and then installs the replica components. For this scenario, choose **Method 1** or **Method 2** below to authorize the replica installation. Choose **Method 1** if one of the following applies:

- You want a senior system administrator to perform the initial part of the procedure and a junior administrator to perform the rest.
- You want to automate your replica installation.

Method 1: a random password generated on an IdM server

Enter the following commands on any server in the domain:

1. Log in as the administrator.

\$ kinit admin

2. Add the external system as an IdM host. Use the **--random** option with the **ipa host-add** command to generate a random one-time password to be used for the subsequent replica installation.

```
$ ipa host-add replica.example.com --random
```

```
-----
Added host "replica.example.com"
-----
Host name: replica.example.com
Random password: W5YpARI=7M.n
Password: True
Keytab: False
Managed by: server.example.com
```

The generated password will become invalid after you use it to enroll the machine into the IdM domain. It will be replaced with a proper host keytab after the enrollment is finished.

3. Add the system to the **ipaservers** host group.

```
$ ipa hostgroup-add-member ipaservers --hosts replica.example.com
```

```
Host-group: ipaservers
Description: IPA server hosts
Member hosts: server.example.com, replica.example.com
```

```
-----
Number of members added 1
-----
```



NOTE

Membership in the **ipaservers** group grants the machine elevated privileges similar to the administrator's credentials. Therefore, in the next step, the **ipa-replica-install** utility can be run on the host successfully by a junior system administrator that provides the generated random password.

Method 2: a privileged user's credentials

Using this method, you authorize the replica installation by providing a privileged user's credentials. The default privileged user is **admin**.

No action is required prior to running the IdM replica installation utility. Add the principal name and password options (**--principal admin --admin-password password**) to the **ipa-replica-install** command directly during the installation.

Additional resources

- To start the installation procedure, see [Installing an IdM replica](#).
- You can use an Ansible playbook to install IdM replicas. For more information, see [Installing an Identity Management replica using an Ansible playbook](#).

CHAPTER 21. INSTALLING AN IDM REPLICA

The following sections describe how to install an Identity Management (IdM) replica interactively, by using the command-line interface (CLI). The replica installation process copies the configuration of the existing server and installs the replica based on that configuration.



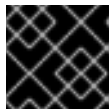
NOTE

Red Hat recommends [using Ansible roles to install replicas](#) . By using Ansible roles, you can consistently install and customize multiple replicas.

Interactive and non-interactive methods that do not use Ansible are useful in topologies where, for example, the replica preparation is delegated to a user or a third party. You can also use these methods in geographically distributed topologies where you do not have access from the Ansible controller node.

Prerequisites

- You are installing one IdM replica at a time. The installation of multiple replicas at the same time is not supported.
- Ensure your system is [prepared for IdM replica installation](#) .



IMPORTANT

If this preparation is not performed, installing an IdM replica will fail.

For the individual types of replica installation procedures, see:

- [Section 21.1, “Installing an IdM replica with integrated DNS and a CA”](#)
- [Section 21.2, “Installing an IdM replica with integrated DNS and no CA”](#)
- [Section 21.3, “Installing an IdM replica without integrated DNS and with a CA”](#)
- [Section 21.4, “Installing an IdM replica without integrated DNS and without a CA”](#)
- [Section 21.5, “Installing an IdM hidden replica”](#)

To troubleshoot the replica installation procedure, see:

- [Chapter 22, *Troubleshooting IdM replica installation*](#)

After the installation, see:

- [Section 21.6, “Testing an IdM replica”](#)
- [Backing Up and Restoring IdM](#)

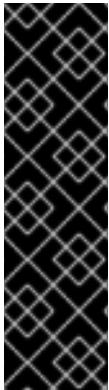
21.1. INSTALLING AN IDM REPLICA WITH INTEGRATED DNS AND A CA

Follow this procedure to install an Identity Management (IdM) replica:

- With integrated DNS

- With a certificate authority (CA)

You can do this to, for example, replicate the CA service for resiliency after installing an IdM server with an integrated CA.



IMPORTANT

When configuring a replica with a CA, the CA configuration of the replica must mirror the CA configuration of the other server.

For example, if the server includes an integrated IdM CA as the root CA, the new replica must also be installed with an integrated CA as the root CA. No other CA configuration is available in this case.

Including the **--setup-ca** option in the **ipa-replica-install** command copies the CA configuration of the initial server.

Prerequisites

- Ensure your system is [prepared for an IdM replica installation](#) .

Procedure

1. Enter **ipa-replica-install** with these options:

- **--setup-dns** to configure the replica as a DNS server
- **--forwarder** to specify a per-server forwarder, or **--no-forwarder** if you do not want to use any per-server forwarders. To specify multiple per-server forwarders for failover reasons, use **--forwarder** multiple times.



NOTE

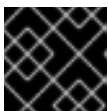
The **ipa-replica-install** utility accepts a number of other options related to DNS settings, such as **--no-reverse** or **--no-host-dns**. For more information about them, see the **ipa-replica-install(1)** man page.

- **--setup-ca** to include a CA on the replica

For example, to set up a replica with an integrated DNS server and a CA that forwards all DNS requests not managed by the IdM servers to the DNS server running on IP 192.0.2.1:

```
# ipa-replica-install --setup-dns --forwarder 192.0.2.1 --setup-ca
```

2. After the installation completes, add a DNS delegation from the parent domain to the IdM DNS domain. For example, if the IdM DNS domain is **idm.example.com**, add a name server (NS) record to the **example.com** parent domain.



IMPORTANT

Repeat this step each time after you install an IdM DNS server.

21.2. INSTALLING AN IDM REPLICA WITH INTEGRATED DNS AND NO CA

Follow this procedure to install an Identity Management (IdM) replica:

- With integrated DNS
- Without a certificate authority (CA) in an IdM environment in which a CA is already installed. The replica will forward all certificate operations to the IdM server with a CA installed.

Prerequisites

- Ensure your system is [prepared for an IdM replica installation](#).

Procedure

1. Enter **ipa-replica-install** with these options:

- **--setup-dns** to configure the replica as a DNS server
- **--forwarder** to specify a per-server forwarder, or **--no-forwarder** if you do not want to use any per-server forwarders. To specify multiple per-server forwarders for failover reasons, use **--forwarder** multiple times.

For example, to set up a replica with an integrated DNS server that forwards all DNS requests not managed by the IdM servers to the DNS server running on IP 192.0.2.1:

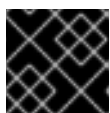
```
# ipa-replica-install --setup-dns --forwarder 192.0.2.1
```



NOTE

The **ipa-replica-install** utility accepts a number of other options related to DNS settings, such as **--no-reverse** or **--no-host-dns**. For more information about them, see the **ipa-replica-install(1)** man page.

2. After the installation completes, add a DNS delegation from the parent domain to the IdM DNS domain. For example, if the IdM DNS domain is **idm.example.com**, add a name server (NS) record to the **example.com** parent domain.



IMPORTANT

Repeat this step each time after you install an IdM DNS server.

21.3. INSTALLING AN IDM REPLICA WITHOUT INTEGRATED DNS AND WITH A CA

Follow this procedure to install an Identity Management (IdM) replica:

- Without integrated DNS
- With a certificate authority (CA)



IMPORTANT

When configuring a replica with a CA, the CA configuration of the replica must mirror the CA configuration of the other server.

For example, if the server includes an integrated IdM CA as the root CA, the new replica must also be installed with an integrated CA as the root CA. No other CA configuration is available in this case.

Including the `--setup-ca` option in the `ipa-replica-install` command copies the CA configuration of the initial server.

Prerequisites

- Ensure your system is [prepared for an IdM replica installation](#) .

Procedure

1. Enter `ipa-replica-install` with the `--setup-ca` option.

```
# ipa-replica-install --setup-ca
```

2. Add the newly created IdM DNS service records to your DNS server:

- a. Export the IdM DNS service records into a file in the `nsupdate` format:

```
$ ipa dns-update-system-records --dry-run --out dns_records_file.nsupdate
```

- b. Submit a DNS update request to your DNS server using the `nsupdate` utility and the `dns_records_file.nsupdate` file. For more information, see [Updating External DNS Records Using nsupdate](#) in RHEL 7 documentation. Alternatively, refer to your DNS server documentation for adding DNS records.

21.4. INSTALLING AN IDM REPLICA WITHOUT INTEGRATED DNS AND WITHOUT A CA

Follow this procedure to install an Identity Management (IdM) replica:

- Without integrated DNS
- Without a certificate authority (CA) by providing the required certificates manually. The assumption here is that the first server was installed without a CA.



IMPORTANT

You cannot install a server or replica using self-signed third-party server certificates because the imported certificate files must contain the full CA certificate chain of the CA that issued the LDAP and Apache server certificates.

Prerequisites

- Ensure your system is [prepared for an IdM replica installation](#) .

Procedure

- Enter **ipa-replica-install**, and provide the required certificate files by adding these options:
 - **--dirsrv-cert-file**
 - **--dirsrv-pin**
 - **--http-cert-file**
 - **--http-pin**

For details about the files that are provided using these options, see [Section 5.1, “Certificates required to install an IdM server without a CA”](#).

For example:

```
# ipa-replica-install \
  --dirsrv-cert-file /tmp/server.crt \
  --dirsrv-cert-file /tmp/server.key \
  --dirsrv-pin secret \
  --http-cert-file /tmp/server.crt \
  --http-cert-file /tmp/server.key \
  --http-pin secret
```



NOTE

Do not add the **--ca-cert-file** option. The **ipa-replica-install** utility takes this part of the certificate information automatically from the first server you installed.

21.5. INSTALLING AN IDM HIDDEN REPLICA

A hidden (unadvertised) replica is an Identity Management (IdM) server that has all services running and available. However, it has no SRV records in DNS, and LDAP server roles are not enabled. Therefore, clients cannot use service discovery to detect these hidden replicas.

For further details about hidden replicas, see [The hidden replica mode](#).

Prerequisites

- Ensure your system is [prepared for an IdM replica installation](#) .

Procedure

- To install a hidden replica, use the following command:

```
ipa-replica-install --hidden-replica
```

Note that the command installs a replica without DNS SRV records and with disabled LDAP server roles.

You can also change the mode of existing replica to hidden. For details, see [Demotion and promotion of hidden replicas](#).

21.6. TESTING AN IDM REPLICA

After creating a replica, check if the replica replicates data as expected. You can use the following procedure.

Procedure

1. Create a user on the new replica:

```
[admin@new_replica ~]$ ipa user-add test_user
```

2. Make sure the user is visible on another replica:

```
[admin@another_replica ~]$ ipa user-show test_user
```

21.7. CONNECTIONS PERFORMED DURING AN IDM REPLICA INSTALLATION

[Requests performed during an IdM replica installation](#) lists the operations performed by **ipa-replica-install**, the Identity Management (IdM) replica installation tool.

Table 21.1. Requests performed during an IdM replica installation

Operation	Protocol used	Purpose
DNS resolution against the DNS resolvers configured on the client system	DNS	To discover the IP addresses of IdM servers
Requests to ports 88 (TCP/TCP6 and UDP/UDP6) on the discovered IdM servers	Kerberos	To obtain a Kerberos ticket
JSON-RPC calls to the IdM Apache-based web-service on the discovered or configured IdM servers	HTTPS	IdM client enrollment; replica keys retrieval and certificate issuance if required
Requests over TCP/TCP6 to port 389 on the IdM server, using SASL GSSAPI authentication, plain LDAP, or both	LDAP	IdM client enrollment; CA certificate chain retrieval; LDAP data replication
Requests over TCP/TCP6 to port 22 on IdM server	SSH	To check if the connection is working
(optionally) Access over port 8443 (TCP/TCP6) on the IdM servers	HTTPS	To administer the Certificate Authority on the IdM server (only during IdM server and replica installation)

CHAPTER 22. TROUBLESHOOTING IDM REPLICA INSTALLATION

The following sections describe the process for gathering information about a failing IdM replica installation, and how to resolve some common installation issues.

22.1. IDM REPLICA INSTALLATION ERROR LOG FILES

When you install an Identity Management (IdM) replica, debugging information is appended to the following log files on the **replica**:

- **`/var/log/ipareplica-install.log`**
- **`/var/log/ipareplica-conncheck.log`**
- **`/var/log/ipaclient-install.log`**
- **`/var/log/httpd/error_log`**
- **`/var/log/dirsrv/slapd-INSTANCE-NAME/access`**
- **`/var/log/dirsrv/slapd-INSTANCE-NAME/errors`**
- **`/var/log/ipaserver-install.log`**

The replica installation process also appends debugging information to the following log files on the IdM **server** the replica is contacting:

- **`/var/log/httpd/error_log`**
- **`/var/log/dirsrv/slapd-INSTANCE-NAME/access`**
- **`/var/log/dirsrv/slapd-INSTANCE-NAME/errors`**

The last line of each log file reports success or failure, and **ERROR** and **DEBUG** entries provide additional context.

Additional resources

- [Reviewing IdM replica installation errors](#)

22.2. REVIEWING IDM REPLICA INSTALLATION ERRORS

To troubleshoot a failing IdM replica installation, review the errors at the end of the installation error log files on the new replica and the server, and use this information to resolve any corresponding issues.

Prerequisites

- You must have **root** privileges to display the contents of IdM log files.

Procedure

1. Use the **tail** command to display the latest errors from the primary log file **`/var/log/ipareplica-install.log`**. The following example displays the last 10 lines.

```
[user@replica ~]$ sudo tail -n 10 /var/log/ipareplica-install.log
[sudo] password for user:
func(installer)
File "/usr/lib/python3.6/site-packages/ipaserver/install/server/replicainstall.py", line 424, in
decorated
func(installer)
File "/usr/lib/python3.6/site-packages/ipaserver/install/server/replicainstall.py", line 785, in
promote_check
ensure_enrolled(installer)
File "/usr/lib/python3.6/site-packages/ipaserver/install/server/replicainstall.py", line 740, in
ensure_enrolled
raise ScriptError("Configuration of client side components failed!")

2020-05-28T18:24:51Z DEBUG The ipa-replica-install command failed, exception:
ScriptError: Configuration of client side components failed!
2020-05-28T18:24:51Z ERROR Configuration of client side components failed!
2020-05-28T18:24:51Z ERROR The ipa-replica-install command failed. See
/var/log/ipareplica-install.log for more information
```

- To review the log file interactively, open the end of the log file using the **less** utility and use the ↑ and ↓ arrow keys to navigate.

```
[user@replica ~]$ sudo less -N +G /var/log/ipareplica-install.log
```

- (Optional) While **/var/log/ipareplica-install.log** is the primary log file for a replica installation, you can gather additional troubleshooting information by repeating this review process with additional files on the replica and the server.

On the replica:

```
[user@replica ~]$ sudo less -N +G /var/log/ipareplica-conncheck.log
[user@replica ~]$ sudo less -N +G /var/log/ipaclient-install.log
[user@replica ~]$ sudo less -N +G /var/log/httpd/error_log
[user@replica ~]$ sudo less -N +G /var/log/dirsrv/slapd-INSTANCE-NAME/access
[user@replica ~]$ sudo less -N +G /var/log/dirsrv/slapd-INSTANCE-NAME/errors
[user@replica ~]$ sudo less -N +G /var/log/ipaserver-install.log
```

On the server:

```
[user@server ~]$ sudo less -N +G /var/log/httpd/error_log
[user@server ~]$ sudo less -N +G /var/log/dirsrv/slapd-INSTANCE-NAME/access
[user@server ~]$ sudo less -N +G /var/log/dirsrv/slapd-INSTANCE-NAME/errors
```

Additional resources

- [IdM replica installation error log files](#)
- If you are unable to resolve a failing replica installation, and you have a Red Hat Technical Support subscription, open a Technical Support case at the [Red Hat Customer Portal](#) and provide an **sosreport** of the replica and an **sosreport** of the server.
- The **sosreport** utility collects configuration details, logs and system information from a RHEL system. For more information about the **sosreport** utility, see [What is an sosreport and how to create one in Red Hat Enterprise Linux?](#)

22.3. IDM CA INSTALLATION ERROR LOG FILES

Installing the Certificate Authority (CA) service on an Identity Management (IdM) replica appends debugging information to several locations on the replica and the IdM server the replica communicates with.

Table 22.1. On the replica (in order of recommended priority):

Location	Description
<code>/var/log/pki/pki-ca-spawn.\$TIME_OF_INSTALLATION.log</code>	High-level issues and Python traces for the pkispawn installation process
<code>journalctl -u pki-tomcatd@pki-tomcat</code> output	Errors from the pki-tomcatd@pki-tomcat service
<code>/var/log/pki/pki-tomcat/ca/debug.\$DATE.log</code>	Large JAVA stacktraces of activity in the core of the Public Key Infrastructure (PKI) product
<code>/var/log/pki/pki-tomcat/ca/signedAudit/ca_audit</code>	Audit log of the PKI product
<ul style="list-style-type: none"> • <code>/var/log/pki/pki-tomcat/ca/system</code> • <code>/var/log/pki/pki-tomcat/ca/transactions</code> • <code>/var/log/pki/pki-tomcat/catalina.\$DATE.log</code> 	Low-level debug data of certificate operations for service principals, hosts, and other entities that use certificates

On the server contacted by the replica:

- **`/var/log/httpd/error_log`** log file

Installing the CA service on an existing IdM replica also writes debugging information to the following log file:

- **`/var/log/ipareplica-ca-install.log`** log file



NOTE

If a full IdM replica installation fails while installing the optional CA component, no details about the CA are logged; a message is logged in the **`/var/log/ipareplica-install.log`** file indicating that the overall installation process failed. Red Hat recommends reviewing the log files listed above for details specific to the CA installation failure.

The only exception to this behavior is when you are installing the CA service and the root CA is an external CA. If there is an issue with the certificate from the external CA, errors are logged in **`/var/log/ipareplica-install.log`**.

Additional resources

- [Reviewing IdM CA installation errors](#)

22.4. REVIEWING IDM CA INSTALLATION ERRORS

To troubleshoot a failing IdM CA installation, review the errors at the end of the CA installation error log files and use this information to resolve any corresponding issues.

Prerequisites

- You must have **root** privileges to display the contents of IdM log files.

Procedure

1. To review a log file interactively, open the end of the log file using the **less** utility and use the ↑ and ↓ arrow keys to navigate, while searching for **ScriptError** entries. The following example opens `/var/log/pki/pki-ca-spawn.$TIME_OF_INSTALLATION.log`.

```
[user@server ~]$ sudo less -N +G /var/log/pki/pki-ca-spawn.20200527185902.log
```

2. Gather additional troubleshooting information by repeating this review process with all the CA installation error log files.

Additional resources

- [IdM CA installation error log files](#)
- If you are unable to resolve a failing IdM server installation, and you have a Red Hat Technical Support subscription, open a Technical Support case at the [Red Hat Customer Portal](#) and provide an **sosreport** of the server.
- The **sosreport** utility collects configuration details, logs and system information from a RHEL system. For more information about the **sosreport** utility, see [What is an sosreport and how to create one in Red Hat Enterprise Linux?](#).

22.5. REMOVING A PARTIAL IDM REPLICA INSTALLATION

If an IdM replica installation fails, some configuration files may be left behind. Additional attempts to install the IdM replica can fail and the installation script reports that IPA is already configured:

Example system with existing partial IdM configuration

```
[root@server ~]# ipa-replica-install
Your system may be partly configured.
Run /usr/sbin/ipa-server-install --uninstall to clean up.

IPA server is already configured on this system.
If you want to reinstall the IPA server, please uninstall it first using 'ipa-server-install --uninstall'.
The ipa-replica-install command failed. See /var/log/ipareplica-install.log for more information
```

To resolve this issue, uninstall IdM software from the replica, remove the replica from the IdM topology, and retry the installation process.

Prerequisites

- You must have **root** privileges.

Procedure

1. Uninstall the IdM server software on the host you are trying to configure as an IdM replica.

```
[root@replica ~]# ipa-server-install --uninstall
```

2. On all other servers in the topology, use the **ipa server-del** command to delete any references to the replica that did not install properly.

```
[root@other-replica ~]# ipa server-del replica.idm.example.com
```

3. Attempt installing the replica.
4. If you continue to experience difficulty installing an IdM replica because of repeated failed installations, reinstall the operating system.
One of the requirements for installing an IdM replica is a clean system without any customization. Failed installations may have compromised the integrity of the host by unexpectedly modifying system files.

Additional resources

- For additional details on uninstalling an IdM replica, see [Uninstalling an IdM replica](#).
- If installation attempts fail after repeated uninstallation attempts, and you have a Red Hat Technical Support subscription, open a Technical Support case at the [Red Hat Customer Portal](#) and provide an **sosreport** of the replica and an **sosreport** of the server.
- The **sosreport** utility collects configuration details, logs and system information from a RHEL system. For more information about the **sosreport** utility, see [What is an sosreport and how to create one in Red Hat Enterprise Linux?](#).

22.6. RESOLVING INVALID CREDENTIAL ERRORS

If an IdM replica installation fails with an **Invalid credentials** error, the system clocks on the hosts may be out of sync with each other:

```
[27/40]: setting up initial replication
Starting replication, please wait until this has completed.
Update in progress, 15 seconds elapsed
[ldap://server.example.com:389] reports: Update failed! Status: [49 - LDAP error: Invalid credentials]
```

```
[error] RuntimeError: Failed to start replication
Your system may be partly configured.
Run /usr/sbin/ipa-server-install --uninstall to clean up.
```

```
ipa.ipapython.install.cli.install_tool(CompatServerReplicaInstall): ERROR Failed to start replication
ipa.ipapython.install.cli.install_tool(CompatServerReplicaInstall): ERROR The ipa-replica-install
command failed. See /var/log/ipareplica-install.log for more information
```

If you use the **--no-ntp** or **-N** options to attempt the replica installation while clocks are out of sync, the installation fails because services are unable to authenticate with Kerberos.

To resolve this issue, synchronize the clocks on both hosts and retry the installation process.

Prerequisites

- You must have **root** privileges to change system time.

Procedure

1. Synchronize the system clocks manually or with **chronyd**.

Synchronizing manually

Display the system time on the server and set the replica's time to match.

```
[user@server ~]$ date  
Thu May 28 21:03:57 EDT 2020
```

```
[user@replica ~]$ sudo timedatectl set-time '2020-05-28 21:04:00'
```

- **Synchronizing with chronyd:**

See [Using the Chrony suite to configure NTP](#) to configure and set system time with **chrony** tools.

2. Attempt the IdM replica installation again.

Additional resources

- If you are unable to resolve a failing replica installation, and you have a Red Hat Technical Support subscription, open a Technical Support case at the [Red Hat Customer Portal](#) and provide an **sosreport** of the replica and an **sosreport** of the server.
- The **sosreport** utility collects configuration details, logs and system information from a RHEL system. For more information about the **sosreport** utility, see [What is an sosreport and how to create one in Red Hat Enterprise Linux?](#).

22.7. ADDITIONAL RESOURCES

- [Troubleshooting the first IdM server installation](#)
- [Troubleshooting IdM client installation](#)
- [Backing up and restoring IdM](#)

CHAPTER 23. UNINSTALLING AN IDM REPLICA

As an IdM administrator, you can remove an Identity Management (IdM) replica from the topology. For more information, see [Uninstalling an IdM server](#).

CHAPTER 24. INSTALLING DNS ON AN EXISTING IDM SERVER

Follow this procedure to install the DNS service on an Identity Management (IdM) server that was originally installed without it.

Prerequisites

- You understand the advantages and limitations of using IdM with integrated DNS as described in [Installing an IdM server: With integrated DNS, with an integrated CA as the root CA](#) .
- You have **root** access to the IdM server.

Procedure

1. [Optional] Verify that DNS is not already installed on the IdM server.

```
[root@r8server ~]# ipa server-role-show r8server.idm.example.com
Role name: DNS server
Server name: r8server.idm.example.com
Role name: DNS server
Role status: absent
```

The output confirms that IdM DNS is not available on the server.

2. Enable the **idm:DL1** stream:

```
[root@r8server ~]# yum module enable idm:DL1
```

3. Download the **ipa-dns-server** package and its dependencies:

```
[root@r8server ~]# yum module install idm:DL1/dns
```

4. Start the script to install DNS on the server:

```
[root@r8server ~]# ipa-dns-install
```

- a. The script prompts for per-server DNS forwarders.

```
Do you want to configure DNS forwarders? [yes]:
```

- To configure per-server DNS forwarders, enter **yes**, and then follow the instructions on the command line. The installation process will add the forwarder IP addresses to the IdM LDAP.
 - For the forwarding policy default settings, see the **--forward-policy** description in the **ipa-dns-install(1)** man page.
- If you do not want to use DNS forwarding, enter **no**.
With no DNS forwarders, hosts in your IdM domain will not be able to resolve names from other, internal, DNS domains in your infrastructure. The hosts will only be left with public DNS servers to resolve their DNS queries.

- b. The script prompts to check if any DNS reverse (PTR) records for the IP addresses associated with the server need to be configured.

Do you want to search for missing reverse zones? [yes]:

If you run the search and missing reverse zones are discovered, the script asks you whether to create the reverse zones along with the PTR records.

Do you want to create reverse zone for IP 192.0.2.1 [yes]:

Please specify the reverse zone name [2.0.192.in-addr.arpa.]:

Using reverse zone(s) 2.0.192.in-addr.arpa.



NOTE

Using IdM to manage reverse zones is optional. You can use an external DNS service for this purpose instead.

Additional resources

- **man ipa-dns-install(1)**

CHAPTER 25. UNINSTALLING THE INTEGRATED IDM DNS SERVICE FROM AN IDM SERVER

If you have more than one server with integrated DNS in an Identity Management (IdM) deployment, you might decide to remove the integrated DNS service from one of the servers. To do this, you must first decommission the IdM server completely before re-installing IdM on it, this time without the integrated DNS.



NOTE

While you can add the DNS role to an IdM server, IdM does not provide a method to remove only the DNS role from an IdM server: the **ipa-dns-install** command does not have an **--uninstall** option.

Prerequisites

- You have integrated DNS installed on an IdM server.
- This is not the last integrated DNS service in your IdM topology.

Procedure

1. Identify the redundant DNS service and follow the procedure in [Uninstalling an IdM server](#) on the IdM replica that hosts this service.
2. On the same host, follow the procedure in either [Without integrated DNS, with an integrated CA as the root CA](#) or [Without integrated DNS, with an external CA as the root CA](#), depending on your use case.

CHAPTER 26. ADDING THE IDM CA SERVICE TO AN IDM SERVER IN A DEPLOYMENT WITHOUT A CA

If you previously installed an Identity Management (IdM) domain without the certificate authority (CA) component, you can add the IdM CA service to the domain by using the **ipa-ca-install** command. Depending on your requirements, you can select one of the following options:

- [Add the IdM Certificate Server CA as the root CA.](#)
- [Add the IdM Certificate Server CA as a subordinate CA, with an external CA as the root CA.](#)



NOTE

For details on the supported CA configurations, see [Planning your CA services](#).

26.1. INSTALLING THE FIRST IDM CA AS THE ROOT CA INTO AN EXISTING IDM DOMAIN

If you previously installed Identity Management (IdM) without the certificate authority (CA) component, you can install the CA on an IdM server subsequently. Follow this procedure to install, on the *idmserver* server, an IdM CA that is not subordinate to any external root CA.

Prerequisites

- You have **root** permissions on *idmserver*.
- The IdM server is installed on *idmserver*.
- Your IdM deployment has no CA installed.
- You know the IdM **Directory Manager** password.

Procedure

1. On *idmserver*, install the IdM Certificate Server CA:

```
[root@idmserver ~] ipa-ca-install
```

2. On each IdM host in the topology, run the **ipa-certupdate** utility to update the host with the information about the new certificate from the IdM LDAP.



IMPORTANT

If you do not run **ipa-certupdate** after generating the IdM CA certificate, the certificate will not be distributed to the other IdM machines.

26.2. INSTALLING THE FIRST IDM CA WITH AN EXTERNAL CA AS THE ROOT CA INTO AN EXISTING IDM DOMAIN

If you previously installed Identity Management (IdM) without the certificate authority (CA) component, you can install the CA on an IdM server subsequently. Follow this procedure to install, on the *idmserver* server, an IdM CA that is subordinate to an external root CA, with zero or several intermediate CAs in

between.

Prerequisites

- You have **root** permissions on *idmserver*.
- The IdM server is installed on *idmserver*.
- Your IdM deployment has no CA installed.
- You know the IdM **Directory Manager** password.

Procedure

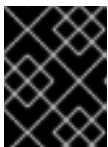
1. Start the installation:

```
[root@idmserver ~] ipa-ca-install --external-ca
```

2. Wait till the command-line interface informs you that a certificate signing request (CSR) has been saved.
3. Submit the CSR to the external CA.
4. Copy the issued certificate to the IdM server.
5. Continue the installation by adding the certificates and full path to the external CA files to **ipa-ca-install**:

```
[root@idmserver ~]# ipa-ca-install --external-cert-file=/root/master.crt --external-cert-file=/root/ca.crt
```

6. On each IdM host in the topology, run the **ipa-certupdate** utility to update the host with the information about the new certificate from the IdM LDAP.



IMPORTANT

Failing to run **ipa-certupdate** after generating the IdM CA certificate means that the certificate will not be distributed to the other IdM machines.

CHAPTER 27. ADDING THE IDM CA SERVICE TO AN IDM SERVER IN A DEPLOYMENT WITH A CA

If your Identity Management (IdM) environment already has the IdM certificate authority (CA) service installed but a particular IdM server, *idmserver*, was installed as an IdM replica without a CA, you can add the CA service to *idmserver* by using the **ipa-ca-install** command.



NOTE

This procedure is identical for both the following scenarios:

- The IdM CA is a root CA.
- The IdM CA is subordinate to an external, root CA.

Prerequisites

- You have **root** permissions on *idmserver*.
- The IdM server is installed on *idmserver*.
- Your IdM deployment has a CA installed on another IdM server.
- You know the IdM **Directory Manager** password.

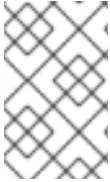
Procedure

- On *idmserver*, install the IdM Certificate Server CA:

```
[root@idmserver ~] ipa-ca-install
```

CHAPTER 28. UNINSTALLING THE IDM CA SERVICE FROM AN IDM SERVER

If you have more than four Identity Management (IdM) replicas with the **CA Role** in your topology and you run into performance problems due to redundant certificate replication, (RH) recommends that you remove redundant CA service instances from IdM replicas. To do this, you must first decommission the affected IdM replicas completely before re-installing IdM on them, this time without the CA service.



NOTE

While you can **add** the CA role to an IdM replica, IdM does not provide a method to **remove** only the CA role from an IdM replica: the **ipa-ca-install** command does not have an **--uninstall** option.

Prerequisites

- You have the IdM CA service installed on more than four IdM servers in your topology.

Procedure

1. Identify the redundant CA service and follow the procedure in [Uninstalling an IdM server](#) on the IdM replica that hosts this service.
2. On the same host, follow the procedure in [Installing an IdM server: With integrated DNS, without a CA](#).

CHAPTER 29. MANAGING REPLICATION TOPOLOGY

This chapter describes how to manage replication between servers in an Identity Management (IdM) domain.

Additional resources

- [Planning the replica topology](#)

29.1. EXPLAINING REPLICATION AGREEMENTS, TOPOLOGY SUFFIXES AND TOPOLOGY SEGMENTS

When you create a replica, Identity Management (IdM) creates a replication agreement between the initial server and the replica. The data that is replicated is then stored in topology suffixes and when two replicas have a replication agreement between their suffixes, the suffixes form a topology segment. These concepts are explained in more detail in the following sections:

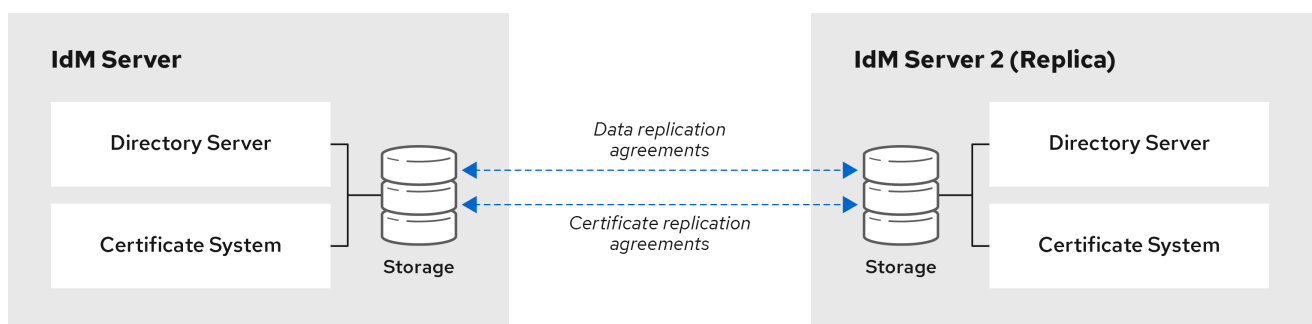
- [Replication agreements](#)
- [Topology suffixes](#)
- [Topology segments](#)

29.1.1. Replication agreements between IdM replicas

When an administrator creates a replica based on an existing server, Identity Management (IdM) creates a *replication agreement* between the initial server and the replica. The replication agreement ensures that the data and configuration is continuously replicated between the two servers.

IdM uses *multiple read/write replica replication*. In this configuration, all replicas joined in a replication agreement receive and provide updates, and are therefore considered suppliers and consumers. Replication agreements are always bilateral.

Figure 29.1. Server and replica agreements



64_RHEL_0120

IdM uses two types of replication agreements:

Domain replication agreements

These agreements replicate the identity information.

Certificate replication agreements

These agreements replicate the certificate information.

Both replication channels are independent. Two servers can have one or both types of replication agreements configured between them. For example, when server A and server B have only domain replication agreement configured, only identity information is replicated between them, not the certificate information.

29.1.2. Topology suffixes

Topology suffixes store the data that is replicated. IdM supports two types of topology suffixes: **domain** and **ca**. Each suffix represents a separate server, a separate replication topology.

When a replication agreement is configured, it joins two topology suffixes of the same type on two different servers.

The domain suffix: `dc=example,dc=com`

The **domain** suffix contains all domain-related data.

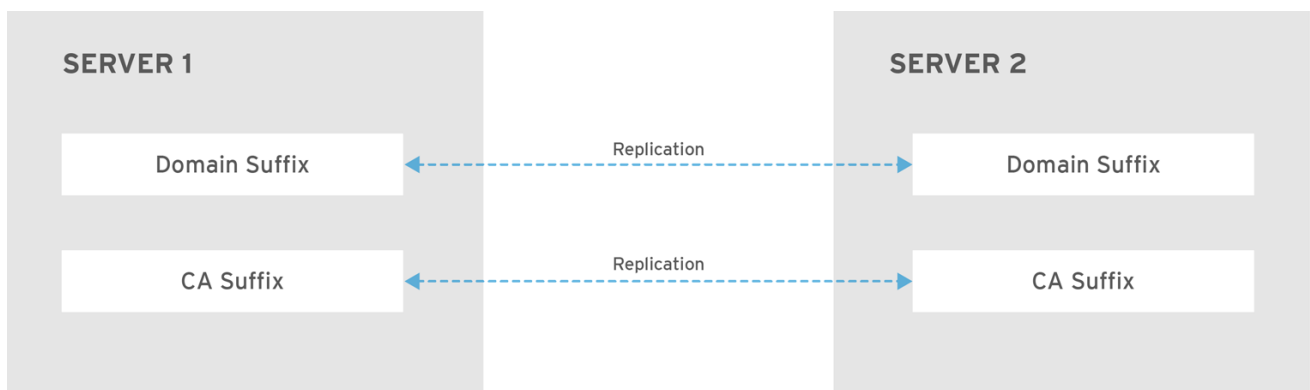
When two replicas have a replication agreement between their **domain** suffixes, they share directory data, such as users, groups, and policies.

The ca suffix: `o=ipaca`

The **ca** suffix contains data for the Certificate System component. It is only present on servers with a certificate authority (CA) installed.

When two replicas have a replication agreement between their **ca** suffixes, they share certificate data.

Figure 29.2. Topology suffixes



RHEL_404973_0916

An initial topology replication agreement is set up between two servers by the **ipa-replica-install** script when installing a new replica.

Example 29.1. Viewing topology suffixes

The **ipa topologysuffix-find** command displays a list of topology suffixes:

```

$ ipa topologysuffix-find
-----
2 topology suffixes matched
-----
Suffix name: ca
Managed LDAP suffix DN: o=ipaca

Suffix name: domain
  
```



```
Managed LDAP suffix DN: dc=example,dc=com
```

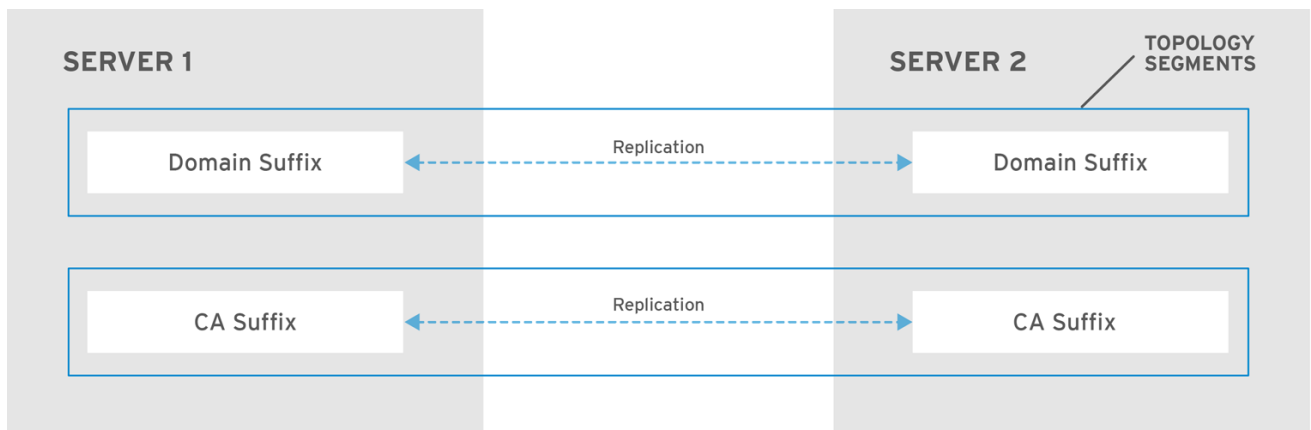
```
-----  
Number of entries returned 2  
-----
```

29.1.3. Topology segments

When two replicas have a replication agreement between their suffixes, the suffixes form a *topology segment*. Each topology segment consists of a *left node* and a *right node*. The nodes represent the servers joined in the replication agreement.

Topology segments in IdM are always bidirectional. Each segment represents two replication agreements: from server A to server B, and from server B to server A. The data is therefore replicated in both directions.

Figure 29.3. Topology segments



RHEL_404973_0916

Example 29.2. Viewing topology segments

The **ipa topologysegment-find** command shows the current topology segments configured for the domain or CA suffixes. For example, for the domain suffix:

```
$ ipa topologysegment-find
Suffix name: domain
-----
1 segment matched
-----
Segment name: server1.example.com-to-server2.example.com
Left node: server1.example.com
Right node: server2.example.com
Connectivity: both
-----
Number of entries returned 1
-----
```

In this example, domain-related data is only replicated between two servers: **server1.example.com** and **server2.example.com**.

To display details for a particular segment only, use the **ipa topologysegment-show** command:

```

$ ipa topologysegment-show
Suffix name: domain
Segment name: server1.example.com-to-server2.example.com
Segment name: server1.example.com-to-server2.example.com
Left node: server1.example.com
Right node: server2.example.com
Connectivity: both

```

29.2. USING THE TOPOLOGY GRAPH TO MANAGE REPLICATION TOPOLOGY

The topology graph in the web UI shows the relationships between the servers in the domain. Using the Web UI, you can manipulate and transform the representation of the topology.

Accessing the topology graph

To access the topology graph:

1. Select **IPA Server** → **Topology** → **Topology Graph**.
2. If you make any changes to the topology that are not immediately reflected in the graph, click **Refresh**.

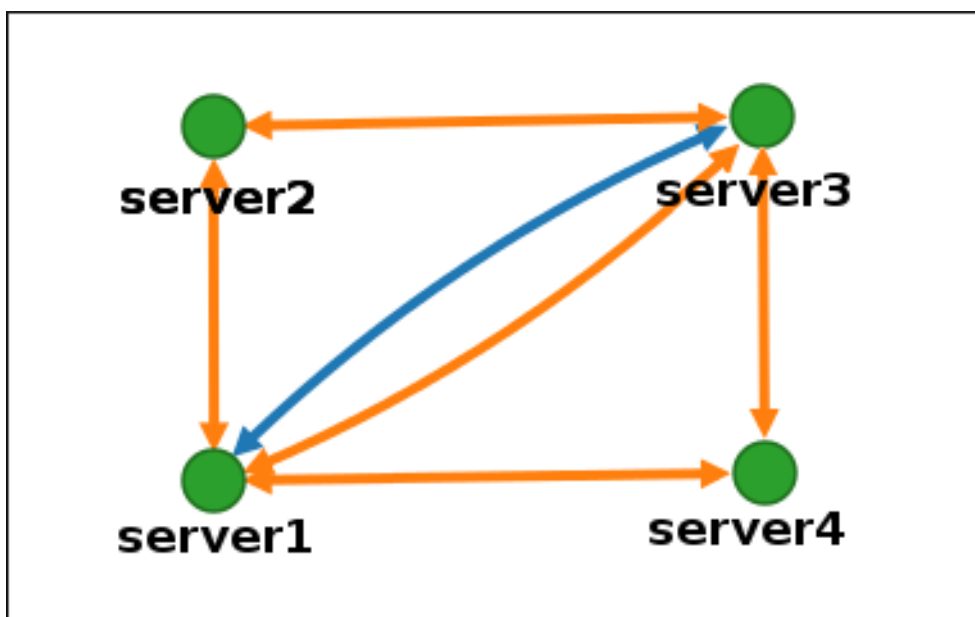
Interpreting the topology graph

Servers joined in a domain replication agreement are connected by an orange arrow. Servers joined in a CA replication agreement are connected by a blue arrow.

Topology graph example: recommended topology

The recommended topology example below shows one of the possible recommended topologies for four servers: each server is connected to at least two other servers, and more than one server is a CA server.

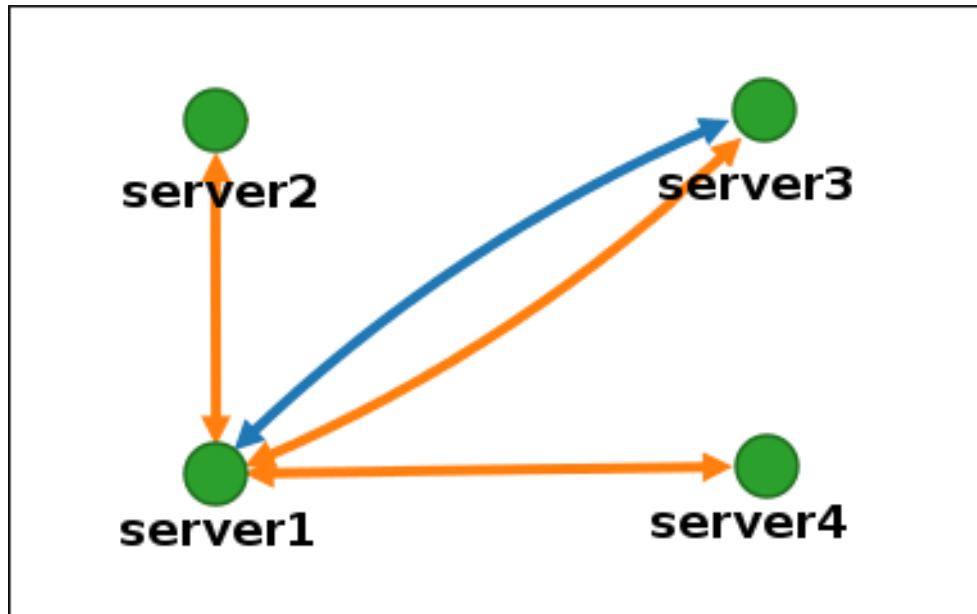
Figure 29.4. Recommended topology example



Topology graph example: discouraged topology

In the discouraged topology example below, **server1** is a single point of failure. All the other servers have replication agreements with this server, but not with any of the other servers. Therefore, if **server1** fails, all the other servers will become isolated. Avoid creating topologies like this.

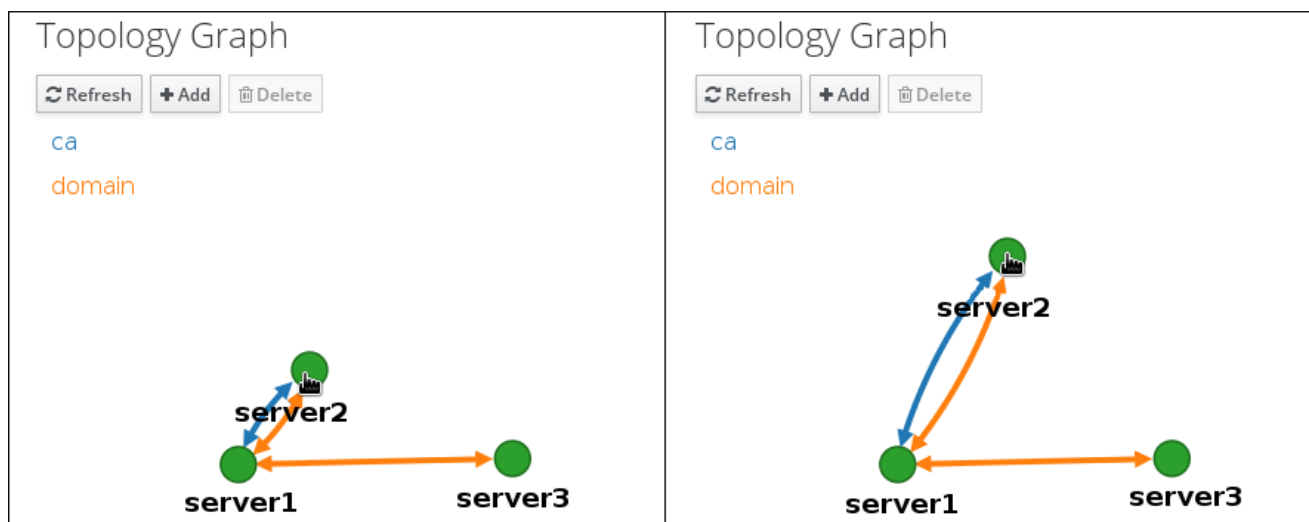
Figure 29.5. Discouraged topology example: Single Point of Failure



Customizing the topology view

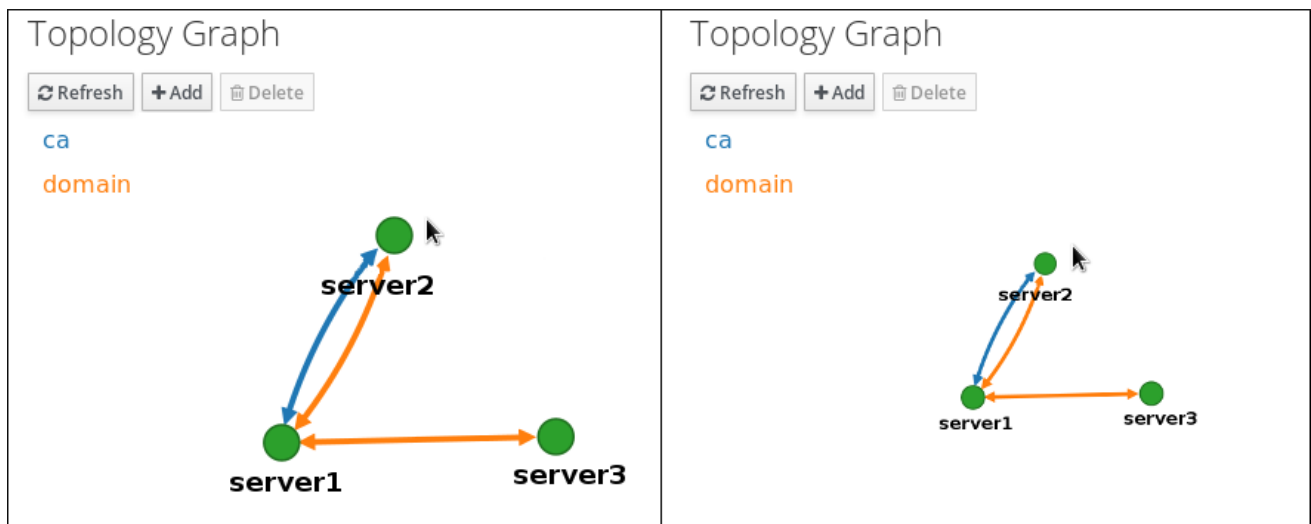
You can move individual topology nodes by dragging the mouse:

Figure 29.6. Moving topology graph nodes



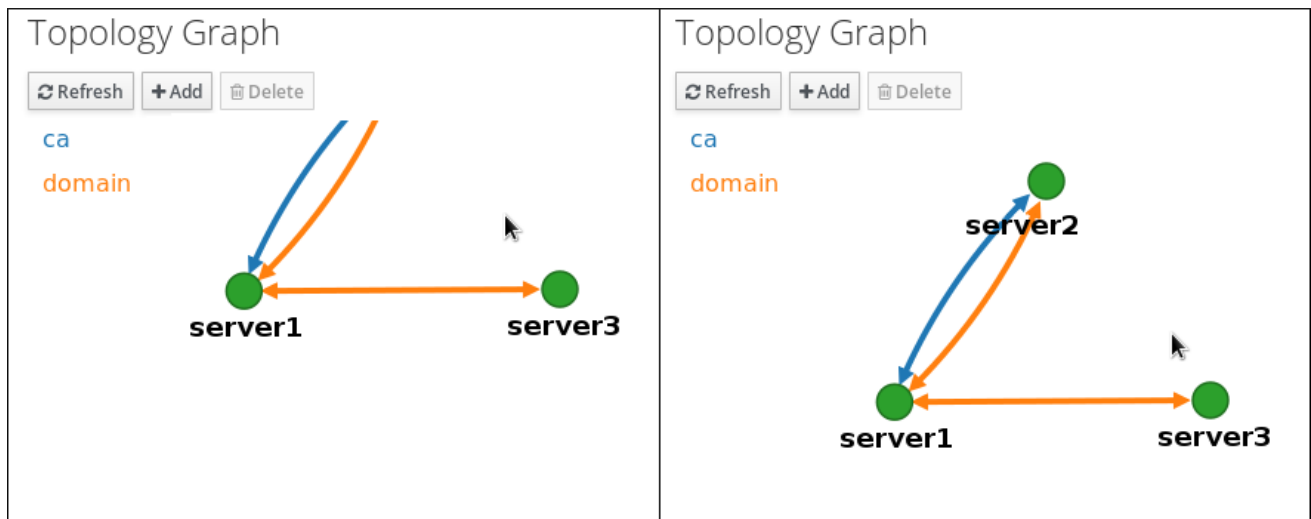
You can zoom in and zoom out the topology graph using the mouse wheel:

Figure 29.7. Zooming the topology graph



You can move the canvas of the topology graph by holding the left mouse button:

Figure 29.8. Moving the topology graph canvas



29.3. SETTING UP REPLICATION BETWEEN TWO SERVERS USING THE WEB UI

Using the Web interface of Identity Management (IdM) you can choose two servers and create new replication agreement between them.

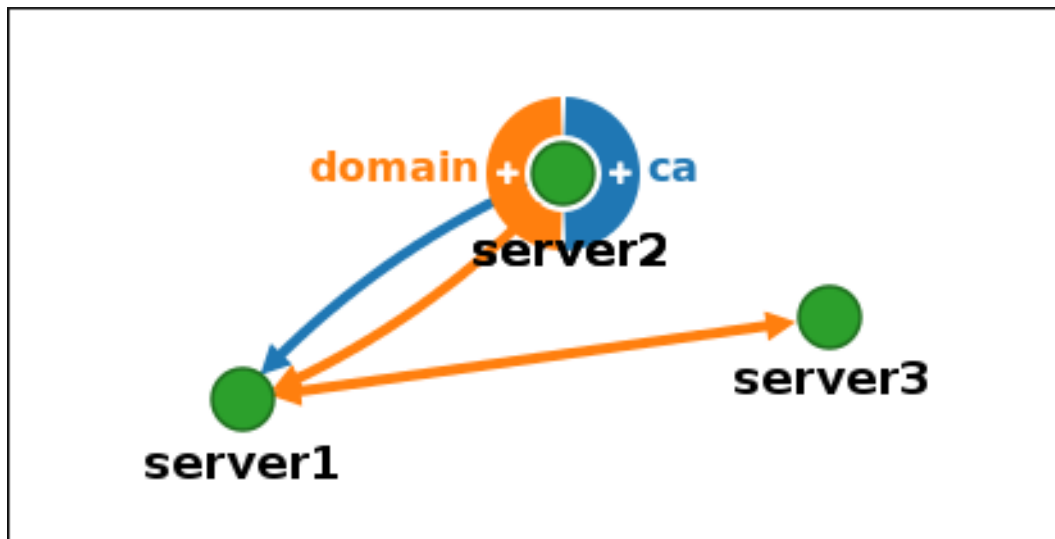
Prerequisites

- You have the IdM administrator credentials.

Procedure

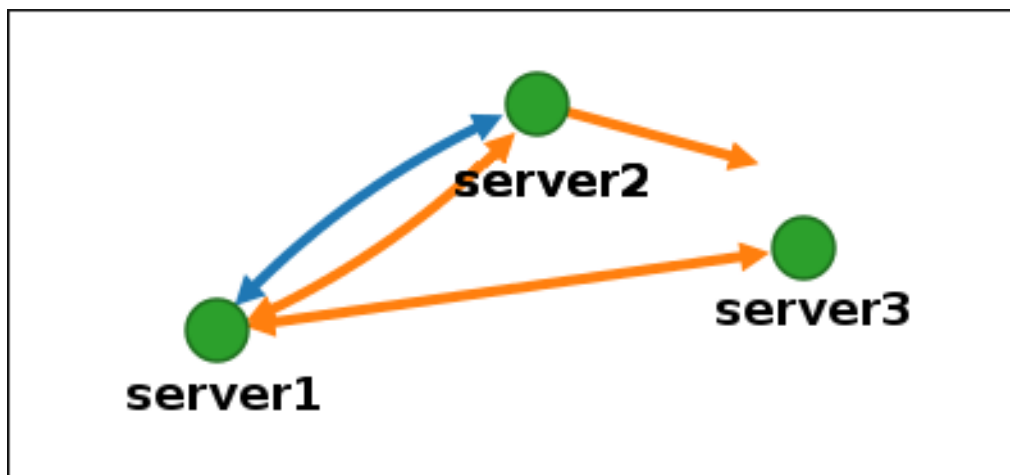
1. In the topology graph, hover your mouse over one of the server nodes.

Figure 29.9. Domain or CA options



2. Click on the **domain** or the **ca** part of the circle depending on what type of topology segment you want to create.
3. A new arrow representing the new replication agreement appears under your mouse pointer. Move your mouse to the other server node, and click on it.

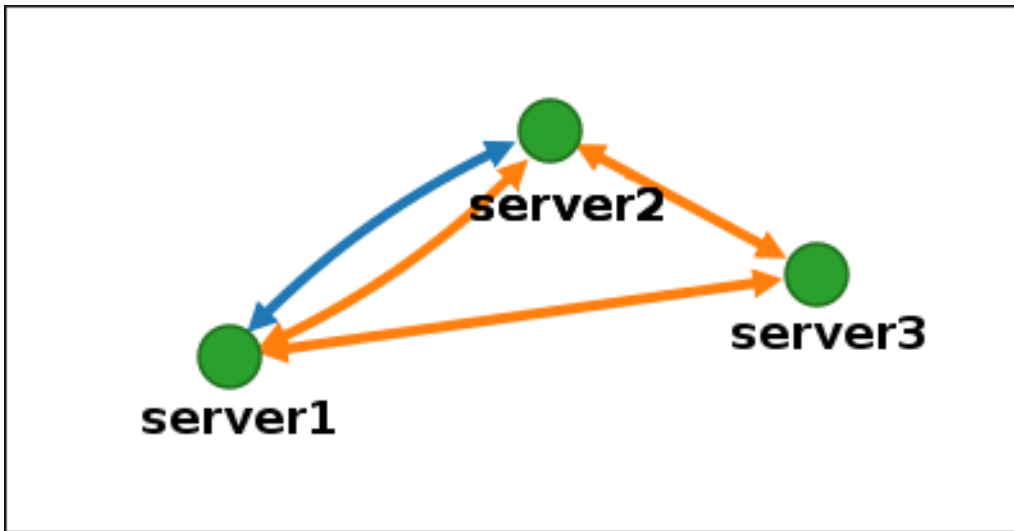
Figure 29.10. Creating a new segment



4. In the **Add topology segment** window, click **Add** to confirm the properties of the new segment.

The new topology segment between the two servers joins them in a replication agreement. The topology graph now shows the updated replication topology:

Figure 29.11. New segment created



29.4. STOPPING REPLICATION BETWEEN TWO SERVERS USING THE WEB UI

Using the web interface of Identity Management (IdM) you can remove a replication agreement from servers.

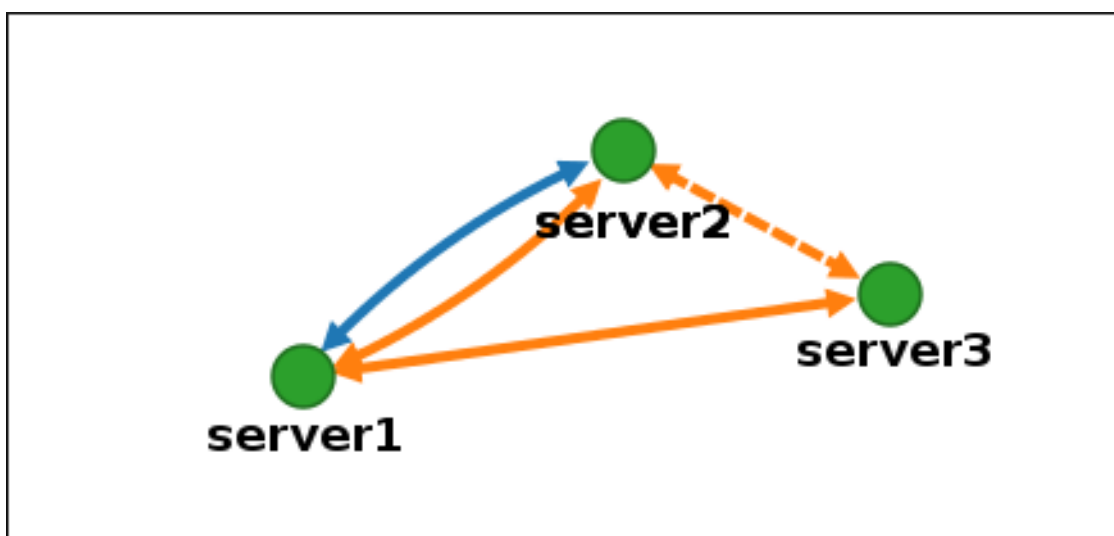
Prerequisites

- You have the IdM administrator credentials.

Procedure

1. Click on an arrow representing the replication agreement you want to remove. This highlights the arrow.

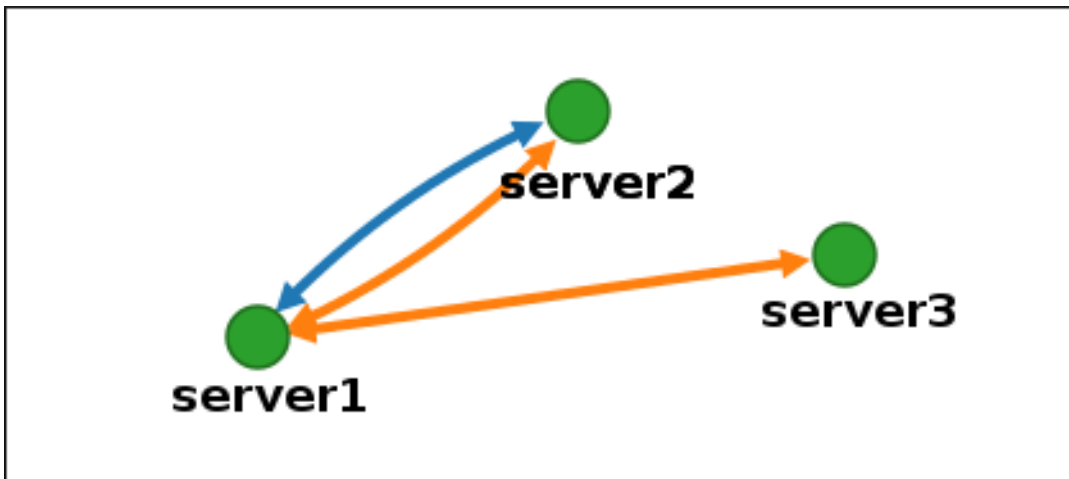
Figure 29.12. Topology segment highlighted



2. Click **Delete**.
3. In the **Confirmation** window, click **OK**.

IdM removes the topology segment between the two servers, which deletes their replication agreement. The topology graph now shows the updated replication topology:

Figure 29.13. Topology segment deleted



29.5. SETTING UP REPLICATION BETWEEN TWO SERVERS USING THE CLI

You can configure replication agreements between two servers using the **ipa topologysegment-add** command.

Prerequisites

- You have the IdM administrator credentials.

Procedure

1. Use the **ipa topologysegment-add** command to create a topology segment for the two servers. When prompted, provide:
 - the required topology suffix: **domain** or **ca**
 - the left node and the right node, representing the two servers
 - optionally, a custom name for the segment
 For example:

```
$ ipa topologysegment-add
Suffix name: domain
Left node: server1.example.com
Right node: server2.example.com
Segment name [server1.example.com-to-server2.example.com]: new_segment
-----
Added segment "new_segment"
-----
Segment name: new_segment
Left node: server1.example.com
Right node: server2.example.com
Connectivity: both
```

Adding the new segment joins the servers in a replication agreement.

2. *Optional.* Use the **ipa topologysegment-show** command to verify that the new segment is configured.

```
$ ipa topologysegment-show
Suffix name: domain
Segment name: new_segment
Segment name: new_segment
Left node: server1.example.com
Right node: server2.example.com
Connectivity: both
```

29.6. STOPPING REPLICATION BETWEEN TWO SERVERS USING THE CLI

You can terminate replication agreements from command line using the **ipa topology segment-del** command.

Prerequisites

- You have the IdM administrator credentials.

Procedure

1. To stop replication, you must delete the corresponding replication segment between the servers. To do that, you need to know the segment name.
If you do not know the name, use the **ipa topologysegment-find** command to display all segments, and locate the required segment in the output. When prompted, provide the required topology suffix: **domain** or **ca**. For example:

```
$ ipa topologysegment-find
Suffix name: domain
-----
8 segments matched
-----
Segment name: new_segment
Left node: server1.example.com
Right node: server2.example.com
Connectivity: both

...

-----
Number of entries returned 8
-----
```

2. Use the **ipa topologysegment-del** command to remove the topology segment joining the two servers.

```
$ ipa topologysegment-del
Suffix name: domain
Segment name: new_segment
```



```
-----
Deleted segment "new_segment"
-----
```

Deleting the segment removes the replication agreement.

3. *Optional.* Use the **ipa topologysegment-find** command to verify that the segment is no longer listed.

```
$ ipa topologysegment-find
Suffix name: domain
-----
7 segments matched
-----
Segment name: server2.example.com-to-server3.example.com
Left node: server2.example.com
Right node: server3.example.com
Connectivity: both
...
-----
Number of entries returned 7
-----
```

29.7. REMOVING SERVER FROM TOPOLOGY USING THE WEB UI

You can use Identity Management (IdM) web interface to remove a server from the topology.

Prerequisites

- You have the IdM administrator credentials.
- The server you want to remove is **not** the only server connecting other servers with the rest of the topology; this would cause the other servers to become isolated, which is not allowed.
- The server you want to remove is **not** your last CA or DNS server.



WARNING

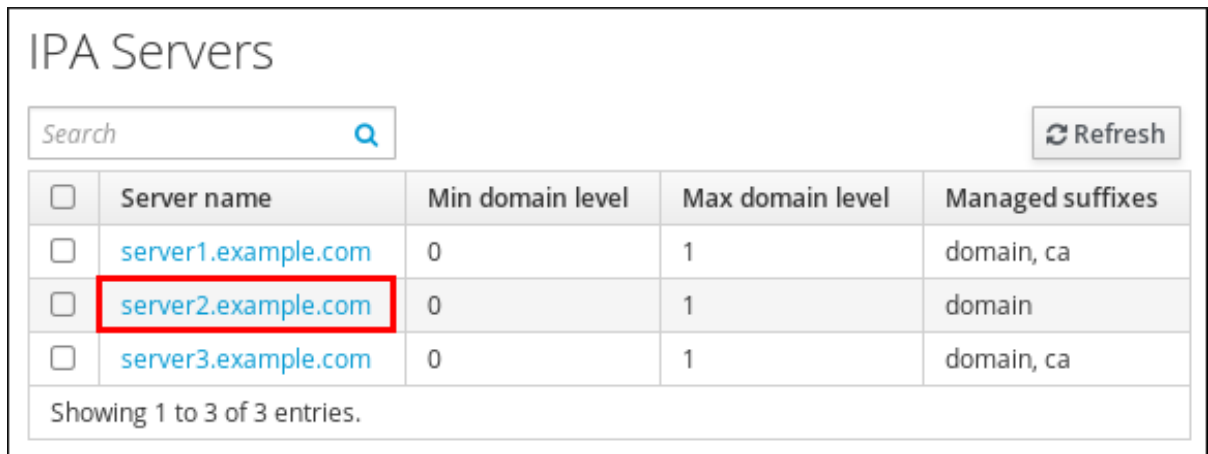
Removing a server is an irreversible action. If you remove a server, the only way to introduce it back into the topology is to install a new replica on the machine.

Procedure

To remove a server from the topology without uninstalling the server components from the machine:

1. Select **IPA Server** → **Topology** → **IPA Servers**.
2. Click on the name of the server you want to delete.

Figure 29.14. Selecting a server



<input type="checkbox"/>	Server name	Min domain level	Max domain level	Managed suffixes
<input type="checkbox"/>	server1.example.com	0	1	domain, ca
<input type="checkbox"/>	server2.example.com	0	1	domain
<input type="checkbox"/>	server3.example.com	0	1	domain, ca

Showing 1 to 3 of 3 entries.

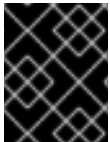
- Click **Delete Server**.

29.8. REMOVING SERVER FROM TOPOLOGY USING THE CLI

You can use the command line interface to remove a server from the topology.

Prerequisites

- You have the IdM administrator credentials.
- The server you want to remove is **not** the only server connecting other servers with the rest of the topology; this would cause the other servers to become isolated, which is not allowed
- The server you want to remove is **not** your last CA or DNS server.



IMPORTANT

Removing a server is an irreversible action. If you remove a server, the only way to introduce it back into the topology is to install a new replica on the machine.

Procedure

To remove **server1.example.com**:

- On another server, run the **ipa server-del** command to remove **server1.example.com**. The command removes all topology segments pointing to the server:

```
[user@server2 ~]$ ipa server-del
Server name: server1.example.com
Removing server1.example.com from replication topology, please wait...
-----
Deleted IPA server "server1.example.com"
-----
```

- Optional:* on **server1.example.com**, run the **ipa server-install --uninstall** command to uninstall the server components from the machine.

```
[root@server1 ~]# ipa server-install --uninstall
```

29.9. VIEWING SERVER ROLES ON AN IDM SERVER USING THE WEB UI

Based on the services installed on an IdM server, it can perform various *server roles*. For example:

- CA server
- DNS server
- Key recovery authority (KRA) server.

For a complete list of the supported server roles, see [IPA Server → Topology → Server Roles](#).



NOTE

- Role status **absent** means that no server in the topology is performing the role.
- Role status **enabled** means that one or more servers in the topology are performing the role.

Figure 29.15. Server roles in the web UI

Role name	Role status
AD trust agent	absent
AD trust controller	absent
CA server	enabled

29.10. VIEWING SERVER ROLES ON AN IDM SERVER USING THE CLI

Based on the services installed on an IdM server, it can perform various *server roles*. For example:

- CA server
- DNS server
- Key recovery authority (KRA) server.

You can view which servers perform which roles in the topology using the following commands.

- The **ipa config-show** command displays all CA servers and the current CA renewal server:

```
$ ipa config-show
...
IPA masters: server1.example.com, server2.example.com, server3.example.com
IPA CA servers: server1.example.com, server2.example.com
IPA CA renewal master: server1.example.com
```

- The **ipa server-show** command displays a list of roles enabled on a particular server. For example, for a list of roles enabled on *server.example.com*:

```
$ ipa server-show
Server name: server.example.com
...
Enabled server roles: CA server, DNS server, KRA server
```

- The **ipa server-find --servrole** searches for all servers with a particular server role enabled. For example, to search for all CA servers:

```
$ ipa server-find --servrole "CA server"
-----
2 IPA servers matched
-----
Server name: server1.example.com
...
Server name: server2.example.com
...
-----
Number of entries returned 2
-----
```

29.11. PROMOTING A REPLICA TO A CA RENEWAL SERVER AND CRL PUBLISHER SERVER

If your IdM deployment uses an embedded certificate authority (CA), one of the IdM CA servers acts as the CA renewal server, a server that manages the renewal of CA subsystem certificates. One of the IdM CA servers also acts as the IdM CRL publisher server, a server that generates certificate revocation lists. By default, the CA renewal server and CRL publisher server roles are installed on the first server on which the system administrator installed the CA role using the **ipa-server-install** or **ipa-ca-install** command.

Prerequisites

- You have the IdM administrator credentials.

Procedure

- [Change the current CA renewal server.](#)
- [Configure replica to generate CRLs.](#)

29.12. DEMOTING OR PROMOTING HIDDEN REPLICAS

After a replica has been installed, you can configure whether the replica is hidden or visible.

For details about hidden replicas, see [The hidden replica mode](#).

If the replica is a CA renewal server, move the service to another replica before making this replica hidden.

For details, see [Changing and resetting IdM CA renewal server](#).

**NOTE**

The hidden replica feature, introduced in RHEL 8.1 as a Technology Preview, is fully supported starting with RHEL 8.2.

Procedure

- To hide the replica, enter:

```
# ipa server-state replica.idm.example.com --state=hidden
```

Alternatively, you can make the replica visible with the following command:

```
# ipa server-state replica.idm.example.com --state=enabled
```

To view a list of all the hidden replicas in your topology, enter:

```
# ipa config-show
```

If all of your replicas are enabled, the command output does not mention hidden replicas

CHAPTER 30. INSTALLING AND RUNNING THE IDM HEALTHCHECK TOOL

Learn more about the IdM Healthcheck tool and how to install and run it.



NOTE

- The Healthcheck tool is only available on RHEL 8.1 or later.

30.1. HEALTHCHECK IN IDM

The Healthcheck tool in Identity Management (IdM) helps find issues that may impact the health of your IdM environment.



NOTE

The Healthcheck tool is a command line tool that can be used without Kerberos authentication.

Modules are Independent

Healthcheck consists of independent modules which test for:

- Replication issues
- Certificate validity
- Certificate Authority infrastructure issues
- IdM and Active Directory trust issues
- Correct file permissions and ownership settings

Two output formats

Healthcheck generates the following outputs, which you can set using the **output-type** option:

- **json**: Machine-readable output in JSON format (default)
- **human**: Human-readable output

You can specify a different file destination with the **--output-file** option.

Results

Each Healthcheck module returns one of the following results:

SUCCESS

configured as expected

WARNING

not an error, but worth keeping an eye on or evaluating

ERROR

not configured as expected

CRITICAL

not configured as expected, with a high possibility for impact

30.2. INSTALLING IDM HEALTHCHECK

Follow this procedure to install the IdM Healthcheck tool.

Procedure

- Install the **ipa-healthcheck** package:

```
[root@server ~]# yum install ipa-healthcheck
```

**NOTE**

On RHEL 8.1 and 8.2 systems, use the **yum install /usr/bin/ipa-healthcheck** command instead.

Verification steps

- Use the **--failures-only** option to have **ipa-healthcheck** only report errors. A fully-functioning IdM installation returns an empty result of [].

```
[root@server ~]# ipa-healthcheck --failures-only  
[]
```

Additional resources

- Use **ipa-healthcheck --help** to see all supported arguments.

30.3. RUNNING IDM HEALTHCHECK

Healthcheck can be run manually or automatically using [log rotation](#).

Prerequisites

- The Healthcheck tool must be installed. See [Installing IdM Healthcheck](#).

Procedure

- To run healthcheck manually, enter the **ipa-healthcheck** command.

```
[root@server ~]# ipa-healthcheck
```

Additional resources

For all options, see the man page: **man ipa-healthcheck**.

30.4. ADDITIONAL RESOURCES

- See the following sections of the [Configuring and managing Identity Management](#) guide for examples of using IdM Healthcheck.
 - [Checking services](#)
 - [Verifying your IdM and AD trust configuration](#)
 - [Verifying certificates](#)
 - [Verifying system certificates](#)
 - [Checking disk space](#)
 - [Verifying permissions of IdM configuration files](#)
 - [Checking replication](#)
- You can also see those chapters organized into a single guide: [Using IdM Healthcheck to monitor your IdM environment](#)

CHAPTER 31. INSTALLING AN IDENTITY MANAGEMENT SERVER USING AN ANSIBLE PLAYBOOK

Learn more about how to configure a system as an IdM server by using [Ansible](#). Configuring a system as an IdM server establishes an IdM domain and enables the system to offer IdM services to IdM clients. You can manage the deployment by using the **ipaserver** Ansible role.

Prerequisites

- You understand [Ansible](#) and IdM concepts:
 - Ansible roles
 - Ansible nodes
 - Ansible inventory
 - Ansible tasks
 - Ansible modules
 - Ansible plays and playbooks

31.1. ANSIBLE AND ITS ADVANTAGES FOR INSTALLING IDM

Ansible is an automation tool used to configure systems, deploy software, and perform rolling updates. Ansible includes support for Identity Management (IdM), and you can use Ansible modules to automate installation tasks such as the setup of an IdM server, replica, client, or an entire IdM topology.

Advantages of using Ansible to install IdM

The following list presents advantages of installing Identity Management using Ansible in contrast to manual installation.

- You do not need to log into the managed node.
- You do not need to configure settings on each host to be deployed individually. Instead, you can have one inventory file to deploy a complete cluster.
- You can reuse an inventory file later for management tasks, for example to add users and hosts. You can reuse an inventory file even for such tasks as are not related to IdM.

Additional resources

- [Automating Red Hat Identity Management installation](#)
- [Planning Identity Management](#)
- [Preparing the system for IdM server installation](#)

31.2. INSTALLING THE ANSIBLE-FREEIPA PACKAGE

Follow this procedure to install the **ansible-freeipa** package that provides Ansible roles and modules for installing and managing Identity Management (IdM).

Prerequisites

- On the **managed node**:
 - Ensure that the managed node is a Red Hat Enterprise Linux 8 system with a static IP address and a working package manager.
- On the **controller**:
 - Ensure that the controller is a Red Hat Enterprise Linux system with a valid subscription. If this is not the case, see the official Ansible documentation [Installation guide](#) for alternative installation instructions.
 - Ensure that you can reach the managed node over the **SSH** protocol from the controller. Check that the managed node is listed in the `/root/.ssh/known_hosts` file of the controller.

Procedure

Use the following procedure on the Ansible controller.

1. If your system is running on RHEL 8.5 and earlier, enable the required repository:

```
# subscription-manager repos --enable ansible-2.8-for-rhel-8-x86_64-rpms
```

2. If your system is running on RHEL 8.5 and earlier, install the **ansible** package:

```
# yum install ansible
```

3. Install the **ansible-freeipa** package:

```
# yum install ansible-freeipa
```

The roles and modules are installed into the `/usr/share/ansible/roles/` and `/usr/share/ansible/plugins/modules` directories.

31.3. ANSIBLE ROLES LOCATION IN THE FILE SYSTEM

By default, the **ansible-freeipa** roles are installed to the `/usr/share/ansible/roles/` directory. The structure of the **ansible-freeipa** package is as follows:

- The `/usr/share/ansible/roles/` directory stores the **ipaserver**, **ipareplica**, and **ipaclient** roles on the Ansible controller. Each role directory stores examples, a basic overview, the license and documentation about the role in a **README.md** Markdown file.

```
[root@server]# ls -l /usr/share/ansible/roles/  
ipaclient  
ipareplica  
ipaserver
```

- The `/usr/share/doc/ansible-freeipa/` directory stores the documentation about individual roles and the topology in **README.md** Markdown files. It also stores the **playbooks/** subdirectory.

```
[root@server]# ls -l /usr/share/doc/ansible-freeipa/  
playbooks  
README-client.md
```

```

README.md
README-replica.md
README-server.md
README-topology.md

```

- The `/usr/share/doc/ansible-freeipa/playbooks/` directory stores the example playbooks:

```

[root@server]# ls -l /usr/share/doc/ansible-freeipa/playbooks/
install-client.yml
install-cluster.yml
install-replica.yml
install-server.yml
uninstall-client.yml
uninstall-cluster.yml
uninstall-replica.yml
uninstall-server.yml

```

31.4. SETTING THE PARAMETERS FOR A DEPLOYMENT WITH AN INTEGRATED DNS AND AN INTEGRATED CA AS THE ROOT CA

Complete this procedure to configure the inventory file for installing an IdM server with an integrated CA as the root CA in an environment that uses the IdM integrated DNS solution.



NOTE

The inventory in this procedure uses the **INI** format. You can, alternatively, use the **YAML** or **JSON** formats.

Procedure

1. Create a `~/MyPlaybooks/` directory:

```
$ mkdir MyPlaybooks
```

2. Create a `~/MyPlaybooks/inventory` file.
3. Open the inventory file for editing. Specify the fully-qualified domain names (**FQDN**) of the host you want to use as an IdM server. Ensure that the **FQDN** meets the following criteria:
 - Only alphanumeric characters and hyphens (-) are allowed. For example, underscores are not allowed and can cause DNS failures.
 - The host name must be all lower-case.
4. Specify the IdM domain and realm information.
5. Specify that you want to use integrated DNS by adding the following option:

```
ipaserver_setup_dns=yes
```

6. Specify the DNS forwarding settings. Choose one of the following options:
 - Use the `ipaserver_auto_forwarders=yes` option if you want the installer to use forwarders from the `/etc/resolv.conf` file. Do not use this option if the nameserver specified in the

/etc/resolv.conf file is the localhost 127.0.0.1 address or if you are on a virtual private network and the DNS servers you are using are normally unreachable from the public internet.

- Use the **ipaserver_forwarders** option to specify your forwarders manually. The installation process adds the forwarder IP addresses to the **/etc/named.conf** file on the installed IdM server.
- Use the **ipaserver_no_forwarders=yes** option to configure root DNS servers to be used instead.

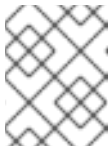


NOTE

With no DNS forwarders, your environment is isolated, and names from other DNS domains in your infrastructure are not resolved.

7. Specify the DNS reverse record and zone settings. Choose from the following options:

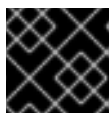
- Use the **ipaserver_allow_zone_overlap=yes** option to allow the creation of a (reverse) zone even if the zone is already resolvable.
- Use the **ipaserver_reverse_zones** option to specify your reverse zones manually.
- Use the **ipaserver_no_reverse=yes** option if you do not want the installer to create a reverse DNS zone.



NOTE

Using IdM to manage reverse zones is optional. You can use an external DNS service for this purpose instead.

- Specify the passwords for **admin** and for the **Directory Manager**. Use the Ansible Vault to store the password, and reference the Vault file from the playbook file. Alternatively and less securely, specify the passwords directly in the inventory file.
- (Optional) Specify a custom **firewalld** zone to be used by the IdM server. If you do not set a custom zone, IdM will add its services to the default **firewalld** zone. The predefined default zone is **public**.



IMPORTANT

The specified **firewalld** zone must exist and be permanent.

Example of an inventory file with the required server information (excluding the passwords)

```
[ipaserver]
server.idm.example.com

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
```

```
ipaserver_setup_dns=yes
ipaserver_auto_forwarders=yes
[...]
```

Example of an inventory file with the required server information (including the passwords)

```
[ipaserver]
server.idm.example.com

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=yes
ipaserver_auto_forwarders=yes
ipaadmin_password=MySecretPassword123
ipadm_password=MySecretPassword234

[...]
```

Example of an inventory file with a custom `firewalld` zone

```
[ipaserver]
server.idm.example.com

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=yes
ipaserver_auto_forwarders=yes
ipaadmin_password=MySecretPassword123
ipadm_password=MySecretPassword234
ipaserver_firewalld_zone=custom zone
```

Example playbook to set up an IdM server using admin and Directory Manager passwords stored in an Ansible Vault file

```
---
- name: Playbook to configure IPA server
  hosts: ipaserver
  become: true
  vars_files:
    - playbook_sensitive_data.yml

  roles:
    - role: ipaserver
      state: present
```

Example playbook to set up an IdM server using admin and Directory Manager passwords from an inventory file

```
---
- name: Playbook to configure IPA server
```

```

hosts: ipaserver
become: true

roles:
- role: ipaserver
  state: present

```

Additional resources

- For the forwarding policy default settings, see the `--forward-policy` description in the `ipa-dns-install(1)` man page.
- For more information about DNS variables used by the `ipaserver` role, see the DNS Variables section in the `README-server.md` file in the `/usr/share/doc/ansible-freeipa` directory.
- For more information on inventory files, see [How to build your inventory](#).

31.5. SETTING THE PARAMETERS FOR A DEPLOYMENT WITH EXTERNAL DNS AND AN INTEGRATED CA AS THE ROOT CA

Complete this procedure to configure the inventory file for installing an IdM server with an integrated CA as the root CA in an environment that uses an external DNS solution.



NOTE

The inventory file in this procedure uses the **INI** format. You can, alternatively, use the **YAML** or **JSON** formats.

Procedure

1. Create a `~/MyPlaybooks/` directory:

```
$ mkdir MyPlaybooks
```

2. Create a `~/MyPlaybooks/inventory` file.
3. Open the inventory file for editing. Specify the fully-qualified domain names (**FQDN**) of the host you want to use as an IdM server. Ensure that the **FQDN** meets the following criteria:
 - Only alphanumeric characters and hyphens (-) are allowed. For example, underscores are not allowed and can cause DNS failures.
 - The host name must be all lower-case.
4. Specify the IdM domain and realm information.
5. Make sure that the `ipaserver_setup_dns` option is set to `no` or that it is absent.
6. Specify the passwords for `admin` and for the **Directory Manager**. Use the Ansible Vault to store the password, and reference the Vault file from the playbook file. Alternatively and less securely, specify the passwords directly in the inventory file.

7. (Optional) Specify a custom **firewalld** zone to be used by the IdM server. If you do not set a custom zone, IdM will add its services to the default **firewalld** zone. The predefined default zone is **public**.



IMPORTANT

The specified **firewalld** zone must exist and be permanent.

Example of an inventory file with the required server information (excluding the passwords)

```
[ipaserver]
server.idm.example.com

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=no
[...]
```

Example of an inventory file with the required server information (including the passwords)

```
[ipaserver]
server.idm.example.com

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=no
ipaadmin_password=MySecretPassword123
ipadm_password=MySecretPassword234

[...]
```

Example of an inventory file with a custom **firewalld** zone

```
[ipaserver]
server.idm.example.com

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=no
ipaadmin_password=MySecretPassword123
ipadm_password=MySecretPassword234
ipaserver_firewalld_zone=custom zone
```

Example playbook to set up an IdM server using admin and Directory Manager passwords stored in an Ansible Vault file

```
---
- name: Playbook to configure IPA server
```

```

hosts: ipaserver
become: true
vars_files:
- playbook_sensitive_data.yml

roles:
- role: ipaserver
  state: present

```

Example playbook to set up an IdM server using admin and Directory Manager passwords from an inventory file

```

---
- name: Playbook to configure IPA server
  hosts: ipaserver
  become: true

  roles:
  - role: ipaserver
    state: present

```

Additional resources

- For more information on inventory files, see [How to build your inventory](#).

31.6. DEPLOYING AN IDM SERVER WITH AN INTEGRATED CA AS THE ROOT CA USING AN ANSIBLE PLAYBOOK

Complete this procedure to deploy an IdM server with an integrated certificate authority (CA) as the root CA using an Ansible playbook.



NOTE

The inventory in this procedure uses the **INI** format. You can, alternatively, use the **YAML** or **JSON** formats.

Prerequisites

- You have set the parameters that correspond to your scenario by choosing one of the following procedures:
 - [Procedure with integrated DNS](#)
 - [Procedure with external DNS](#)
- You have read and understood the variables you can use with the **ipaserver** role as described in the **/usr/share/doc/ansible-freeipa/README-server.md** file.

Procedure

1. Run the **ansible-playbook** command with the name of the playbook file, for example **install-server.yml**. Specify the inventory file with the **-i** option:


```
$ ansible-playbook --vault-password-file=password_file -v -i ~/MyPlaybooks/inventory
~/MyPlaybooks/install-server.yml
```

Specify the level of verbosity by using the **-v**, **-vv**, or **-vvv** option.

You can view the output of the Ansible playbook script on the command-line interface (CLI). The following output shows that the script has run successfully as 0 tasks have failed:

```
PLAY RECAP
server.idm.example.com : ok=18  changed=10  unreachable=0  failed=0  skipped=21
rescued=0  ignored=0
```

2. Choose one of the following options:

- If your IdM deployment uses external DNS: add the DNS resource records contained in the **/tmp/ipa.system.records.UFRPto.db** file to the existing external DNS servers. The process of updating the DNS records varies depending on the particular DNS solution.

```
...
Restarting the KDC
Please add records in this file to your DNS system:
/tmp/ipa.system.records.UFRBto.db
Restarting the web server
...
```



IMPORTANT

The server installation is not complete until you add the DNS records to the existing DNS servers.

- If your IdM deployment uses integrated DNS:
 - Add DNS delegation from the parent domain to the IdM DNS domain. For example, if the IdM DNS domain is **idm.example.com**, add a name server (NS) record to the **example.com** parent domain.



IMPORTANT

Repeat this step each time after an IdM DNS server is installed.

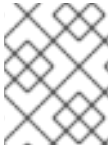
- Add an **_ntp._udp** service (SRV) record for your time server to your IdM DNS. The presence of the SRV record for the time server of the newly-installed IdM server in IdM DNS ensures that future replica and client installations are automatically configured to synchronize with the time server used by this primary IdM server.

Additional resources

- For instruction on how to deploy an IdM server with an **external** CA as the root CA, see [Deploying an IdM server with an external CA as the root CA using an Ansible playbook](#)

31.7. SETTING THE PARAMETERS FOR A DEPLOYMENT WITH AN INTEGRATED DNS AND AN EXTERNAL CA AS THE ROOT CA

Complete this procedure to configure the inventory file for installing an IdM server with an external CA as the root CA in an environment that uses the IdM integrated DNS solution.



NOTE

The inventory file in this procedure uses the **INI** format. You can, alternatively, use the **YAML** or **JSON** formats.

Procedure

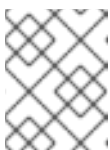
1. Create a `~/MyPlaybooks/` directory:

```
$ mkdir MyPlaybooks
```

2. Create a `~/MyPlaybooks/inventory` file.
3. Open the inventory file for editing. Specify the fully-qualified domain names (**FQDN**) of the host you want to use as an IdM server. Ensure that the **FQDN** meets the following criteria:
 - Only alphanumeric characters and hyphens (-) are allowed. For example, underscores are not allowed and can cause DNS failures.
 - The host name must be all lower-case.
4. Specify the IdM domain and realm information.
5. Specify that you want to use integrated DNS by adding the following option:

```
ipaserver_setup_dns=yes
```

6. Specify the DNS forwarding settings. Choose one of the following options:
 - Use the **ipaserver_auto_forwarders=yes** option if you want the installation process to use forwarders from the `/etc/resolv.conf` file. This option is not recommended if the nameserver specified in the `/etc/resolv.conf` file is the localhost 127.0.0.1 address or if you are on a virtual private network and the DNS servers you are using are normally unreachable from the public internet.
 - Use the **ipaserver_forwarders** option to specify your forwarders manually. The installation process adds the forwarder IP addresses to the `/etc/named.conf` file on the installed IdM server.
 - Use the **ipaserver_no_forwarders=yes** option to configure root DNS servers to be used instead.



NOTE

With no DNS forwarders, your environment is isolated, and names from other DNS domains in your infrastructure are not resolved.

7. Specify the DNS reverse record and zone settings. Choose from the following options:
 - Use the **ipaserver_allow_zone_overlap=yes** option to allow the creation of a (reverse) zone even if the zone is already resolvable.

- Use the **ipaserver_reverse_zones** option to specify your reverse zones manually.
- Use the **ipaserver_no_reverse=yes** option if you do not want the installation process to create a reverse DNS zone.



NOTE

Using IdM to manage reverse zones is optional. You can use an external DNS service for this purpose instead.

8. Specify the passwords for **admin** and for the **Directory Manager**. Use the Ansible Vault to store the password, and reference the Vault file from the playbook file. Alternatively and less securely, specify the passwords directly in the inventory file.
9. (Optional) Specify a custom **firewalld** zone to be used by the IdM server. If you do not set a custom zone, IdM adds its services to the default **firewalld** zone. The predefined default zone is **public**.



IMPORTANT

The specified **firewalld** zone must exist and be permanent.

Example of an inventory file with the required server information (excluding the passwords)

```
[ipaserver]
server.idm.example.com

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=yes
ipaserver_auto_forwarders=yes
[...]
```

Example of an inventory file with the required server information (including the passwords)

```
[ipaserver]
server.idm.example.com

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=yes
ipaserver_auto_forwarders=yes
ipaadmin_password=MySecretPassword123
ipadm_password=MySecretPassword234

[...]
```

Example of an inventory file with a custom **firewalld** zone

```
[ipaserver]
server.idm.example.com

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=yes
ipaserver_auto_forwarders=yes
ipaadmin_password=MySecretPassword123
ipadm_password=MySecretPassword234
ipaserver_firewalld_zone=custom zone

[...]
```

10. Create a playbook for the first step of the installation. Enter instructions for generating the certificate signing request (CSR) and copying it from the controller to the managed node.

```
---
- name: Playbook to configure IPA server Step 1
  hosts: ipaserver
  become: true
  vars_files:
  - playbook_sensitive_data.yml
  vars:
    ipaserver_external_ca: yes

  roles:
  - role: ipaserver
    state: present

  post_tasks:
  - name: Copy CSR /root/ipa.csr from node to "{{ groups.ipaserver[0] + '-ipa.csr' }}"
    fetch:
      src: /root/ipa.csr
      dest: "{{ groups.ipaserver[0] + '-ipa.csr' }}"
      flat: yes
```

11. Create another playbook for the final step of the installation.

```
---
- name: Playbook to configure IPA server Step 2
  hosts: ipaserver
  become: true
  vars_files:
  - playbook_sensitive_data.yml
  vars:
    ipaserver_external_cert_files:
      - "/root/servercert20240601.pem"
      - "/root/cacert.pem"

  pre_tasks:
  - name: Copy "{{ groups.ipaserver[0] }}-{{ item }}" to "/root/{{ item }}" on node
    ansible.builtin.copy:
      src: "{{ groups.ipaserver[0] }}-{{ item }}"
      dest: "/root/{{ item }}"
```

```

force: yes
with_items:
- servercert20240601.pem
- cacert.pem

roles:
- role: ipaserver
state: present

```

Additional resources

- For the forwarding policy default settings, see the **--forward-policy** description in the **ipa-dns-install(1)** man page.
- For more information about DNS variables used by the **ipaserver** role, see the DNS Variables section in the **README-server.md** file in the **/usr/share/doc/ansible-freeipa** directory.
- For more information on inventory files, see [How to build your inventory](#) .

31.8. SETTING THE PARAMETERS FOR A DEPLOYMENT WITH EXTERNAL DNS AND AN EXTERNAL CA AS THE ROOT CA

Complete this procedure to configure the inventory file for installing an IdM server with an external CA as the root CA in an environment that uses an external DNS solution.



NOTE

The inventory file in this procedure uses the **INI** format. You can, alternatively, use the **YAML** or **JSON** formats.

Procedure

1. Create a **~/MyPlaybooks/** directory:

```
$ mkdir MyPlaybooks
```

2. Create a **~/MyPlaybooks/inventory** file.
3. Open the inventory file for editing. Specify the fully-qualified domain names (**FQDN**) of the host you want to use as an IdM server. Ensure that the **FQDN** meets the following criteria:
 - Only alphanumeric characters and hyphens (-) are allowed. For example, underscores are not allowed and can cause DNS failures.
 - The host name must be all lower-case.
4. Specify the IdM domain and realm information.
5. Make sure that the **ipaserver_setup_dns** option is set to **no** or that it is absent.
6. Specify the passwords for **admin** and for the **Directory Manager**. Use the Ansible Vault to store the password, and reference the Vault file from the playbook file. Alternatively and less securely, specify the passwords directly in the inventory file.

- (Optional) Specify a custom **firewalld** zone to be used by the IdM server. If you do not set a custom zone, IdM will add its services to the default **firewalld** zone. The predefined default zone is **public**.



IMPORTANT

The specified **firewalld** zone must exist and be permanent.

Example of an inventory file with the required server information (excluding the passwords)

```
[ipaserver]
server.idm.example.com

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=no
[...]
```

Example of an inventory file with the required server information (including the passwords)

```
[ipaserver]
server.idm.example.com

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=no
ipaadmin_password=MySecretPassword123
ipadm_password=MySecretPassword234

[...]
```

Example of an inventory file with a custom **firewalld** zone

```
[ipaserver]
server.idm.example.com

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=no
ipaadmin_password=MySecretPassword123
ipadm_password=MySecretPassword234
ipaserver_firewalld_zone=custom zone

[...]
```

- Create a playbook for the first step of the installation. Enter instructions for generating the certificate signing request (CSR) and copying it from the controller to the managed node.

```

---
- name: Playbook to configure IPA server Step 1
  hosts: ipaserver
  become: true
  vars_files:
  - playbook_sensitive_data.yml
  vars:
    ipaserver_external_ca: yes

  roles:
  - role: ipaserver
    state: present

  post_tasks:
  - name: Copy CSR /root/ipa.csr from node to "{{ groups.ipaserver[0] + '-ipa.csr' }}"
    fetch:
      src: /root/ipa.csr
      dest: "{{ groups.ipaserver[0] + '-ipa.csr' }}"
      flat: yes

```

9. Create another playbook for the final step of the installation.

```

---
- name: Playbook to configure IPA server Step 2
  hosts: ipaserver
  become: true
  vars_files:
  - playbook_sensitive_data.yml
  vars:
    ipaserver_external_cert_files:
      - "/root/servercert20240601.pem"
      - "/root/cacert.pem"

  pre_tasks:
  - name: Copy "{{ groups.ipaserver[0] }}-{{ item }}" to "/root/{{ item }}" on node
    ansible.builtin.copy:
      src: "{{ groups.ipaserver[0] }}-{{ item }}"
      dest: "/root/{{ item }}"
      force: yes
    with_items:
      - servercert20240601.pem
      - cacert.pem

  roles:
  - role: ipaserver
    state: present

```

Additional resources

- For details on the options available to you when installing an IdM server with external DNS and an externally signed CA, see [Installing an IdM server: Without integrated DNS, with an external CA as the root CA](#).
- For more information on inventory files, see [How to build your inventory](#).

31.9. DEPLOYING AN IDM SERVER WITH AN EXTERNAL CA AS THE ROOT CA USING AN ANSIBLE PLAYBOOK

Complete this procedure to deploy an IdM server with an external certificate authority (CA) as the root CA using an Ansible playbook.



NOTE

The inventory file in this procedure uses the **INI** format. You can, alternatively, use the **YAML** or **JSON** formats.

Prerequisites

- You have set the parameters that correspond to your scenario by choosing one of the following procedures:
 - [Procedure with integrated DNS](#)
 - [Procedure with external DNS](#)
- You have read and understood the variables you can use with the **ipaserver** role as described in the `/usr/share/doc/ansible-freeipa/README-server.md` file.

Procedure

1. Run the **ansible-playbook** command with the name of the playbook file that contains instructions for the first step of the installation, for example **install-server-step1.yml**. Specify the inventory file with the **-i** option:

```
$ ansible-playbook --vault-password-file=password_file -v -i ~/MyPlaybooks/inventory
~/MyPlaybooks/install-server-step1.yml
```

Specify the level of verbosity by using the **-v**, **-vv** or **-vvv** option.

You can view the output of the Ansible playbook script on the command-line interface (CLI). The following output shows that the script has run successfully as 0 tasks have failed:

```
PLAY RECAP
server.idm.example.com : ok=18  changed=10  unreachable=0  failed=0  skipped=21
rescued=0  ignored=0
```

2. Locate the **ipa.csr** certificate signing request file on the controller and submit it to the external CA.
3. Place the IdM CA certificate signed by the external CA in the controller file system so that the playbook in the next step can find it.
4. Run the **ansible-playbook** command with the name of the playbook file that contains instructions for the final step of the installation, for example **install-server-step2.yml**. Specify the inventory file with the **-i** option:

```
$ ansible-playbook -v -i ~/MyPlaybooks/inventory ~/MyPlaybooks/install-server-
step2.yml
```


5. Choose one of the following options:

- If your IdM deployment uses external DNS: add the DNS resource records contained in the `/tmp/ipa.system.records.UFRPto.db` file to the existing external DNS servers. The process of updating the DNS records varies depending on the particular DNS solution.

```
...
Restarting the KDC
Please add records in this file to your DNS system:
/tmp/ipa.system.records.UFRBto.db
Restarting the web server
...
```



IMPORTANT

The server installation is not complete until you add the DNS records to the existing DNS servers.

- If your IdM deployment uses integrated DNS:
 - Add DNS delegation from the parent domain to the IdM DNS domain. For example, if the IdM DNS domain is ***idm.example.com***, add a name server (NS) record to the ***example.com*** parent domain.



IMPORTANT

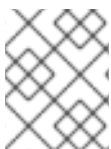
Repeat this step each time after an IdM DNS server is installed.

- Add an ***_ntp._udp*** service (SRV) record for your time server to your IdM DNS. The presence of the SRV record for the time server of the newly-installed IdM server in IdM DNS ensures that future replica and client installations are automatically configured to synchronize with the time server used by this primary IdM server.

Additional resources

For instruction on how to deploy an IdM server with an **integrated** CA as the root CA, see [Deploying an IdM server with an integrated CA as the root CA using an Ansible playbook](#)

31.10. UNINSTALLING AN IDM SERVER USING AN ANSIBLE PLAYBOOK



NOTE

In an existing Identity Management (IdM) deployment, **replica** and **server** are interchangeable terms.

Complete this procedure to uninstall an IdM replica using an Ansible playbook. In this example:

- IdM configuration is uninstalled from ***server123.idm.example.com***.
- ***server123.idm.example.com*** and the associated host entry are removed from the IdM topology.

Prerequisites

- On the control node:
 - You are using Ansible version 2.14 or later.
 - You have installed the [ansible-freeipa](#) package.
 - You have created an [Ansible inventory file](#) with the fully-qualified domain name (FQDN) of the IdM server in the `~/MyPlaybooks/` directory. In this example, the FQDN is `server123.idm.example.com`.
 - You have stored your `ipadmin_password` in the `secret.yml` Ansible vault.
 - For the `ipaserver_remove_from_topology` option to work, the system must be running on RHEL 8.9 or later.
- On the managed node:
 - The system is running on RHEL 7.6 or later.

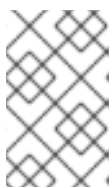
Procedure

1. Create your Ansible playbook file `uninstall-server.yml` with the following content:

```
---
- name: Playbook to uninstall an IdM replica
  hosts: ipaserver
  become: true

  roles:
  - role: ipaserver
    ipaserver_remove_from_domain: true
    state: absent
```

The `ipaserver_remove_from_domain` option unenrolls the host from the IdM topology.



NOTE

If the removal of `server123.idm.example.com` should lead to a disconnected topology, the removal will be aborted. For more information, see [Using an Ansible playbook to uninstall an IdM server even if this leads to a disconnected topology](#).

2. Uninstall the replica:

```
$ ansible-playbook --vault-password-file=password_file -v -i
<path_to_inventory_directory>/hosts <path_to_playbooks_directory>/uninstall-
server.yml
```

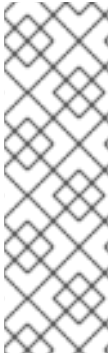
3. Ensure that all name server (NS) DNS records pointing to `server123.idm.example.com` are deleted from your DNS zones. This applies regardless of whether you use integrated DNS managed by IdM or external DNS. For more information on how to delete DNS records from IdM, see [Deleting DNS records in the IdM CLI](#).

31.11. USING AN ANSIBLE PLAYBOOK TO UNINSTALL AN IDM SERVER EVEN IF THIS LEADS TO A DISCONNECTED TOPOLOGY

**NOTE**

In an existing Identity Management (IdM) deployment, **replica** and **server** are interchangeable terms.

Complete this procedure to uninstall an IdM replica using an Ansible playbook even if this results in a disconnected IdM topology. In the example, **server456.idm.example.com** is used to remove the replica and the associated host entry with the FQDN of **server123.idm.example.com** from the topology, leaving certain replicas disconnected from **server456.idm.example.com** and the rest of the topology.

**NOTE**

If removing a replica from the topology using only the **remove_server_from_domain** does not result in a disconnected topology, no other options are required. If the result is a disconnected topology, you must specify which part of the domain you want to preserve. In that case, you must do the following:

- Specify the **ipaserver_remove_on_server** value.
- Set **ipaserver_ignore_topology_disconnect** to True.

Prerequisites

- On the control node:
 - You are using Ansible version 2.14 or later.
 - The system is running on RHEL 8.9 or later.
 - You have installed the [ansible-freeipa](#) package.
 - You have created an [Ansible inventory file](#) with the fully-qualified domain name (FQDN) of the IdM server in the `~/MyPlaybooks/` directory. In this example, the FQDN is **server123.idm.example.com**.
 - You have stored your **ipadmin_password** in the **secret.yml** Ansible vault.
- On the managed node:
 - The system is running on RHEL 7.6 or later.

Procedure

1. Create your Ansible playbook file **uninstall-server.yml** with the following content:

```
---
- name: Playbook to uninstall an IdM replica
  hosts: ipaserver
  become: true

  roles:
  - role: ipaserver
    ipaserver_remove_from_domain: true
    ipaserver_remove_on_server: server456.idm.example.com
    ipaserver_ignore_topology_disconnect: true
    state: absent
```

**NOTE**

Under normal circumstances, if the removal of `server123` does not result in a disconnected topology: if the value for `ipaserver_remove_on_server` is not set, the replica on which `server123` is removed is automatically determined using the replication agreements of `server123`.

2. Uninstall the replica:

```
$ ansible-playbook --vault-password-file=password_file -v -i  
<path_to_inventory_directory>/hosts <path_to_playbooks_directory>/uninstall-  
server.yml
```

3. Ensure that all name server (NS) DNS records pointing to `server123.idm.example.com` are deleted from your DNS zones. This applies regardless of whether you use integrated DNS managed by IdM or external DNS. For more information on how to delete DNS records from IdM, see [Deleting DNS records in the IdM CLI](#).

31.12. ADDITIONAL RESOURCES

- [Planning the replica topology](#)
- [Backing up and restoring IdM servers using Ansible playbooks](#)
- [Inventory basics: formats, hosts, and groups](#)
- You can see sample Ansible playbooks for installing an IdM server and a list of possible variables in the [ansible-freeipa upstream documentation](#).

CHAPTER 32. INSTALLING AN IDENTITY MANAGEMENT REPLICA USING AN ANSIBLE PLAYBOOK

Configuring a system as an IdM replica by using [Ansible](#) enrolls it into an IdM domain and enables the system to use IdM services on IdM servers in the domain.

The deployment is managed by the **ipareplica** Ansible role. The role can use the autodiscovery mode for identifying the IdM servers, domain and other settings. However, if you deploy multiple replicas in a tier-like model, with different groups of replicas being deployed at different times, you must define specific servers or replicas for each group.

Prerequisites

- You have installed the [ansible-freeipa](#) package on the Ansible control node.
- You understand [Ansible](#) and IdM concepts:
 - Ansible roles
 - Ansible nodes
 - Ansible inventory
 - Ansible tasks
 - Ansible modules
 - Ansible plays and playbooks

Additional resources

- [Planning the replica topology](#)

32.1. SPECIFYING THE BASE, SERVER AND CLIENT VARIABLES FOR INSTALLING THE IDM REPLICA

Complete this procedure to configure the inventory file for installing an IdM replica.

Prerequisites

- You have configured your Ansible control node to meet the following requirements:
 - You are using Ansible version 2.14 or later.
 - You have installed the [ansible-freeipa](#) package on the Ansible controller.

Procedure

1. Open the inventory file for editing. Specify the fully-qualified domain names (FQDN) of the hosts to become IdM replicas. The FQDNs must be valid DNS names:
 - Only numbers, alphabetic characters, and hyphens (-) are allowed. For example, underscores are not allowed and can cause DNS failures.
 - The host name must be all lower-case.

Example of a simple inventory hosts file with only the replicas' FQDN defined

```
[ipareplicas]
replica1.idm.example.com
replica2.idm.example.com
replica3.idm.example.com
[...]
```

If the IdM server is already deployed and the SRV records are set properly in the IdM DNS zone, the script automatically discovers all the other required values.

2. [Optional] Provide additional information in the inventory file based on how you have designed your topology:

Scenario 1

If you want to avoid autodiscovery and have all replicas listed in the **[ipareplicas]** section use a specific IdM server, set the server in the **[ipaservers]** section of the inventory file.

Example inventory hosts file with the FQDN of the IdM server and replicas defined

```
[ipaservers]
server.idm.example.com

[ipareplicas]
replica1.idm.example.com
replica2.idm.example.com
replica3.idm.example.com
[...]
```

Scenario 2

Alternatively, if you want to avoid autodiscovery but want to deploy specific replicas with specific servers, set the servers for specific replicas individually in the **[ipareplicas]** section in the inventory file.

Example inventory file with a specific IdM server defined for a specific replica

```
[ipaservers]
server.idm.example.com
replica1.idm.example.com

[ipareplicas]
replica2.idm.example.com
replica3.idm.example.com ipareplica_servers=replica1.idm.example.com
```

In the example above, **replica3.idm.example.com** uses the already deployed **replica1.idm.example.com** as its replication source.

Scenario 3

If you are deploying several replicas in one batch and time is a concern to you, multitier replica deployment can be useful for you. Define specific groups of replicas in the inventory file, for example **[ipareplicas_tier1]** and **[ipareplicas_tier2]**, and design separate plays for each group in the **install-replica.yml** playbook.

Example inventory file with replica tiers defined

```
[ipaservers]
server.idm.example.com

[ipareplicas_tier1]
replica1.idm.example.com

[ipareplicas_tier2]
replica2.idm.example.com \
ipareplica_servers=replica1.idm.example.com,server.idm.example.com
```

The first entry in **ipareplica_servers** will be used. The second entry will be used as a fallback option. When using multiple tiers for deploying IdM replicas, you must have separate tasks in the playbook to first deploy replicas from tier1 and then replicas from tier2:

Example of a playbook file with different plays for different replica groups

```
---
- name: Playbook to configure IPA replicas (tier1)
  hosts: ipareplicas_tier1
  become: true

  roles:
  - role: ipareplica
    state: present

- name: Playbook to configure IPA replicas (tier2)
  hosts: ipareplicas_tier2
  become: true

  roles:
  - role: ipareplica
    state: present
```

3. [Optional] Provide additional information regarding **firewalld** and DNS:

Scenario 1

If you want the replica to use a specified **firewalld** zone instead of the default one, you can specify it in the inventory file. This can be useful, for example, when you want to use an internal **firewalld** zone for your IdM installation instead of a public zone that is set as default. If you do not set a custom zone, IdM will add its services to the default **firewalld** zone. The predefined default zone is **public**.



IMPORTANT

The specified **firewalld** zone must exist and be permanent.

Example of a simple inventory hosts file with a custom **firewalld** zone

```
[ipaservers]
server.idm.example.com
```

```
[ipareplicas]
replica1.idm.example.com
replica2.idm.example.com
replica3.idm.example.com
[...]

[ipareplicas:vars]
ipareplica_firewalld_zone=custom zone
```

Scenario 2

If you want the replica to host the IdM DNS service, add the **ipareplica_setup_dns=yes** line to the **[ipareplicas:vars]** section. Additionally, specify if you want to use per-server DNS forwarders:

- To configure per-server forwarders, add the **ipareplica_forwarders** variable and a list of strings to the **[ipareplicas:vars]** section, for example:
ipareplica_forwarders=192.0.2.1,192.0.2.2
- To configure no per-server forwarders, add the following line to the **[ipareplicas:vars]** section: **ipareplica_no_forwarders=yes**.
- To configure per-server forwarders based on the forwarders listed in the **/etc/resolv.conf** file of the replica, add the **ipareplica_auto_forwarders** variable to the **[ipareplicas:vars]** section.

Example inventory file with instructions to set up DNS and per-server forwarders on the replicas

```
[ipaservers]
server.idm.example.com

[ipareplicas]
replica1.idm.example.com
replica2.idm.example.com
replica3.idm.example.com
[...]

[ipareplicas:vars]
ipareplica_setup_dns=yes
ipareplica_forwarders=192.0.2.1,192.0.2.2
```

Scenario 3

Specify the DNS resolver using the **ipaclient_configure_dns_resolve** and **ipaclient_dns_servers** options (if available) to simplify cluster deployments. This is especially useful if your IdM deployment is using integrated DNS:

An inventory file snippet specifying a DNS resolver:

```
[...]
[ipaclient:vars]
ipaclient_configure_dns_resolver=true
ipaclient_dns_servers=192.168.100.1
```


**NOTE**

The **ipaclient_dns_servers** list must contain only IP addresses. Host names are not allowed.

Additional resources

- For more information about the **ipareplica** variables, see the </usr/share/ansible/roles/ipareplica/README.md> Markdown file.

32.2. SPECIFYING THE CREDENTIALS FOR INSTALLING THE IDM REPLICA USING AN ANSIBLE PLAYBOOK

Complete this procedure to configure the authorization for installing the IdM replica.

Prerequisites

- You have configured your Ansible control node to meet the following requirements:
 - You are using Ansible version 2.14 or later.
 - You have installed the [ansible-freeipa](#) package on the Ansible controller.

Procedure

1. Specify the **password of a user authorized to deploy replicas** for example the IdM **admin**.
 - Red Hat recommends using the Ansible Vault to store the password, and referencing the Vault file from the playbook file, for example **install-replica.yml**:

Example playbook file using principal from inventory file and password from an Ansible Vault file

```
- name: Playbook to configure IPA replicas
  hosts: ipareplicas
  become: true
  vars_files:
    - playbook_sensitive_data.yml

  roles:
    - role: ipareplica
      state: present
```

For details how to use Ansible Vault, see the official [Ansible Vault](#) documentation.

- Less securely, provide the credentials of **admin** directly in the inventory file. Use the **ipaadmin_password** option in the **[ipareplicas:vars]** section of the inventory file. The inventory file and the **install-replica.yml** playbook file can then look as follows:

Example inventory hosts.replica file

```
[...]
[ipareplicas:vars]
ipaadmin_password=Secret123
```

Example playbook using principal and password from inventory file

```
- name: Playbook to configure IPA replicas
  hosts: ipareplicas
  become: true

  roles:
  - role: ipareplica
    state: present
```

- Alternatively but also less securely, provide the credentials of another user authorized to deploy a replica directly in the inventory file. To specify a different authorized user, use the **ipaadmin_principal** option for the user name, and the **ipaadmin_password** option for the password. The inventory file and the **install-replica.yml** playbook file can then look as follows:

Example inventory hosts.replica file

```
[...]
[ipareplicas:vars]
ipaadmin_principal=my_admin
ipaadmin_password=my_admin_secret123
```

Example playbook using principal and password from inventory file

```
- name: Playbook to configure IPA replicas
  hosts: ipareplicas
  become: true

  roles:
  - role: ipareplica
    state: present
```

Additional resources

- For details on the options accepted by the **ipareplica** Ansible role, see the [/usr/share/ansible/roles/ipareplica/README.md](#) Markdown file.

32.3. DEPLOYING AN IDM REPLICA USING AN ANSIBLE PLAYBOOK

Complete this procedure to use an Ansible playbook to deploy an IdM replica.

Prerequisites

- You have configured [the inventory file for installing an IdM replica](#) .
- You have configured [the authorization for installing the IdM replica](#) .

Procedure

- To install an IdM replica using an Ansible playbook, use the **ansible-playbook** command with the name of the playbook file, for example **install-replica.yml**. Specify the inventory file with the **-i** option:

```
$ ansible-playbook --vault-password-file=password_file -v -i
<path_to_inventory_directory>/hosts.replica <path_to_playbooks_directory>/install-
replica.yml
```

Specify the level of verbosity by using the **-v**, **-vv** or **-vvv** option.

Ansible informs you about the execution of the Ansible playbook script. The following output shows that the script has run successfully as 0 tasks have failed:

```
PLAY RECAP
replica.idm.example.com : ok=18  changed=10  unreachable=0  failed=0  skipped=21
rescued=0  ignored=0
```

You have now installed an IdM replica.

32.4. UNINSTALLING AN IDM REPLICA USING AN ANSIBLE PLAYBOOK



NOTE

In an existing Identity Management (IdM) deployment, **replica** and **server** are interchangeable terms. For information on how to uninstall an IdM server, see [Uninstalling an IdM server using an Ansible playbook](#) or [Using an Ansible playbook to uninstall an IdM server even if this leads to a disconnected topology](#).

Additional resources

- [Introduction to IdM servers and clients](#)

CHAPTER 33. INSTALLING AN IDENTITY MANAGEMENT CLIENT USING AN ANSIBLE PLAYBOOK

Learn more about how to configure a system as an Identity Management (IdM) client by using [Ansible](#). Configuring a system as an IdM client enrolls it into an IdM domain and enables the system to use IdM services on IdM servers in the domain.

The deployment is managed by the **ipaclient** Ansible role. By default, the role uses the autodiscovery mode for identifying the IdM servers, domain and other settings. The role can be modified to have the Ansible playbook use the settings specified, for example in the inventory file.

Prerequisites

- You have installed the [ansible-freeipa](#) package on the Ansible control node.
- You understand [Ansible](#) and IdM concepts:
 - Ansible roles
 - Ansible nodes
 - Ansible inventory
 - Ansible tasks
 - Ansible modules
 - Ansible plays and playbooks

33.1. SETTING THE PARAMETERS OF THE INVENTORY FILE FOR THE AUTODISCOVERY CLIENT INSTALLATION MODE

To install an Identity Management client using an Ansible playbook, configure the target host parameters in an inventory file, for example **inventory/hosts**:

- The information about the host
- The authorization for the task

The inventory file can be in one of many formats, depending on the inventory plugins you have. The **INI-like** format is one of Ansible's defaults and is used in the examples below.



NOTE

To use smart cards with the graphical user interface in RHEL, ensure that you include the **ipaclient_mkhome** variable in your Ansible playbook.

Prerequisites

- You have checked the deployment instructions on the control node, see [Checking the parameters in the install-client.yml file](#).

Procedure

1. Specify the fully-qualified hostname (FQDN) of the host to become an IdM client. The fully qualified domain name must be a valid DNS name:
 - Only numbers, alphabetic characters, and hyphens (-) are allowed. For example, underscores are not allowed and can cause DNS failures.
 - The host name must be all lower-case. No capital letters are allowed.

If the SRV records are set properly in the IdM DNS zone, the script automatically discovers all the other required values.

Example of a simple inventory hosts file with only the client FQDN defined

```
[ipaclients]
client.idm.example.com
[...]
```

2. Specify the credentials for enrolling the client. The following authentication methods are available:
 - The **password of a user authorized to enroll clients** This is the default option.
 - Red Hat recommends using the Ansible Vault to store the password, and referencing the Vault file from the playbook file, for example **install-client.yml**, directly:

Example playbook file using principal from inventory file and password from an Ansible Vault file

```
- name: Playbook to configure IPA clients with username/password
  hosts: ipaclients
  become: true
  vars_files:
  - playbook_sensitive_data.yml

  roles:
  - role: ipacient
    state: present
```

- Less securely, provide the credentials of **admin** using the **ipadmin_password** option in the **[ipaclients:vars]** section of the **inventory/hosts** file. Alternatively, to specify a different authorized user, use the **ipadmin_principal** option for the user name, and the **ipadmin_password** option for the password. The **inventory/hosts** inventory file and the **install-client.yml** playbook file can then look as follows:

Example inventory hosts file

```
[...]
[ipaclients:vars]
ipadmin_principal=my_admin
ipadmin_password=Secret123
```

Example Playbook using principal and password from inventory file

```
- name: Playbook to unconfigure IPA clients
  hosts: ipaclients
```

```

become: true

roles:
- role: ipaclient
state: true

```

- The **client keytab** from the previous enrollment if it is still available. This option is available if the system was previously enrolled as an Identity Management client. To use this authentication method, uncomment the **#ipaclient_keytab** option, specifying the path to the file storing the keytab, for example in the **[ipaclient:vars]** section of **inventory/hosts**.
 - A **random, one-time password** (OTP) to be generated during the enrollment. To use this authentication method, use the **ipaclient_use_otp=yes** option in your inventory file. For example, you can uncomment the **ipaclient_use_otp=yes** option in the **[ipaclients:vars]** section of the **inventory/hosts** file. Note that with OTP you must also specify one of the following options:
 - The **password of a user authorized to enroll clients**, for example by providing a value for **ipaadmin_password** in the **[ipaclients:vars]** section of the **inventory/hosts** file.
 - The **admin keytab**, for example by providing a value for **ipaadmin_keytab** in the **[ipaclients:vars]** section of **inventory/hosts**.
3. [Optional] Specify the DNS resolver using the **ipaclient_configure_dns_resolve** and **ipaclient_dns_servers** options (if available) to simplify cluster deployments. This is especially useful if your IdM deployment is using integrated DNS:

An inventory file snippet specifying a DNS resolver:

```

[...]
[ipaclients:vars]
ipaadmin_password: "{{ ipaadmin_password }}"
ipaclient_domain=idm.example.com
ipaclient_configure_dns_resolver=true
ipaclient_dns_servers=192.168.100.1

```



NOTE

The **ipaclient_dns_servers** list must contain only IP addresses. Host names are not allowed.

4. Starting with RHEL 8.9, you can also specify the **ipaclient_subid: true** option to have subid ranges configured for IdM users on the IdM level.

Additional resources

- [/usr/share/ansible/roles/ipaclient/README.md](#)
- [Managing subID ranges manually](#)

33.2. SETTING THE PARAMETERS OF THE INVENTORY FILE WHEN AUTODISCOVERY IS NOT POSSIBLE DURING CLIENT INSTALLATION

To install an Identity Management client using an Ansible playbook, configure the target host parameters in an inventory file, for example **inventory/hosts**:

- The information about the host, the IdM server and the IdM domain or the IdM realm
- The authorization for the task

The inventory file can be in one of many formats, depending on the inventory plugins you have. The **INI-like** format is one of Ansible's defaults and is used in the examples below.



NOTE

To use smart cards with the graphical user interface in RHEL, ensure that you include the **ipaclient_mkhome** variable in your Ansible playbook.

Prerequisites

- You have checked the deployment instructions on the control node, see [Checking the parameters in the install-client.yml file](#).

Procedure

1. Specify the fully-qualified hostname (FQDN) of the host to become an IdM client. The fully qualified domain name must be a valid DNS name:
 - Only numbers, alphabetic characters, and hyphens (-) are allowed. For example, underscores are not allowed and can cause DNS failures.
 - The host name must be all lower-case. No capital letters are allowed.
2. Specify other options in the relevant sections of the **inventory/hosts** file:
 - The FQDN of the servers in the **[ipaservers]** section to indicate which IdM server the client will be enrolled with
 - One of the two following options:
 - The **ipaclient_domain** option in the **[ipaclients:vars]** section to indicate the DNS domain name of the IdM server the client will be enrolled with
 - The **ipaclient_realm** option in the **[ipaclients:vars]** section to indicate the name of the Kerberos realm controlled by the IdM server

Example of an inventory hosts file with the client FQDN, the server FQDN and the domain defined

```
[ipaclients]
client.idm.example.com

[ipaservers]
server.idm.example.com

[ipaclients:vars]
ipaclient_domain=idm.example.com
[...]
```

- Specify the credentials for enrolling the client. The following authentication methods are available:

- The **password of a user authorized to enroll clients** This is the default option.
 - Red Hat recommends using the Ansible Vault to store the password, and referencing the Vault file from the playbook file, for example **install-client.yml**, directly:

Example playbook file using principal from inventory file and password from an Ansible Vault file

```
- name: Playbook to configure IPA clients with username/password
  hosts: ipaclients
  become: true
  vars_files:
    - playbook_sensitive_data.yml

  roles:
    - role: ipaclient
      state: present
```

- Less securely, the credentials of **admin** to be provided using the **ipaadmin_password** option in the **[ipaclients:vars]** section of the **inventory/hosts** file. Alternatively, to specify a different authorized user, use the **ipaadmin_principal** option for the user name, and the **ipaadmin_password** option for the password. The **install-client.yml** playbook file can then look as follows:

Example inventory hosts file

```
[...]
[ipaclients:vars]
ipaadmin_principal=my_admin
ipaadmin_password=Secret123
```

Example Playbook using principal and password from inventory file

```
- name: Playbook to unconfigure IPA clients
  hosts: ipaclients
  become: true

  roles:
    - role: ipaclient
      state: true
```

- The **client keytab** from the previous enrollment if it is still available:

This option is available if the system was previously enrolled as an Identity Management client. To use this authentication method, uncomment the **ipaclient_keytab** option, specifying the path to the file storing the keytab, for example in the **[ipaclient:vars]** section of **inventory/hosts**.
- A **random, one-time password**(OTP) to be generated during the enrollment. To use this authentication method, use the **ipaclient_use_otp=yes** option in your inventory file. For example, you can uncomment the **#ipaclient_use_otp=yes** option in the **[ipaclients:vars]** section of the **inventory/hosts** file. Note that with OTP you must also specify one of the following options:

- The **password of a user authorized to enroll clients** for example by providing a value for **ipaadmin_password** in the **[ipaclients:vars]** section of the **inventory/hosts** file.
 - The **admin keytab**, for example by providing a value for **ipaadmin_keytab** in the **[ipaclients:vars]** section of **inventory/hosts**.
4. Starting with RHEL 8.9, you can also specify the **ipaclient_subid: true** option to have subid ranges configured for IdM users on the IdM level.

Additional resources

- For details on the options accepted by the **ipaclient** Ansible role, see the **/usr/share/ansible/roles/ipaclient/README.md** file.
- [Managing subID ranges manually](#)

33.3. CHECKING THE PARAMETERS IN THE INSTALL-CLIENT.YML FILE

The **install-client.yml** playbook file contains instructions for the IdM client deployment.

Procedure

- Open the file and check if the instructions in the playbook correspond to what you are planning for your deployment. The contents typically look like this:

```
---
- name: Playbook to configure IPA clients with username/password
  hosts: ipaclients
  become: true

  roles:
  - role: ipaclient
    state: present
```

This is what the individual entries mean:

- The **hosts** entry specifies the section of the **inventory/hosts** file where the ansible script searches the **FQDNs** of the hosts on which the **ipa-client-install** script shall be run.
- The **become: true** entry specifies that root's credentials will be invoked during the execution of the **ipa-client-install** script.
- The **role: ipaclient** entry specifies the role that will be installed on the host: in this case, it is the ipa client role.
- The **state: present** entry specifies that the client should be installed rather than uninstalled (**absent**).

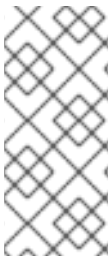
33.4. AUTHORIZATION OPTIONS FOR IDM CLIENT ENROLLMENT USING AN ANSIBLE PLAYBOOK

The individual authorization options for IdM client enrollment with examples of inventory and playbook files are as follows:

Table 33.1. Authorization options for IdM client enrollment using Ansible

Authorization option	Note	Example inventory file	Example <code>install-client.yml</code> playbook file
Password of a user authorized to enroll a client: Option 1	Password stored in Ansible vault	<pre>[ipaclients:vars] [...]</pre>	<pre>- name: Playbook to configure IPA clients with username/password hosts: ipaclients become: true vars_files: - playbook_sensitive_data.yml roles: - role: ipaclient state: present</pre>
Password of a user authorized to enroll a client: Option 2	Password stored in inventory file	<pre>[ipaclients:vars] ipaadmin_password=Secret123</pre>	<pre>- name: Playbook to configure IPA clients hosts: ipaclients become: true roles: - role: ipaclient state: true</pre>
A random, one-time password (OTP): Option 1	OTP + administrative password	<pre>[ipaclients:vars] ipaadmin_password=Secret123 ipaclient_use_otp=true</pre>	<pre>- name: Playbook to configure IPA clients hosts: ipaclients become: true roles: - role: ipaclient state: true</pre>
A random, one-time password (OTP): Option 2	OTP + an administrative keytab	<pre>[ipaclients:vars] ipaadmin_keytab=/root/admin.keytab ipaclient_use_otp=true</pre>	<pre>- name: Playbook to configure IPA clients hosts: ipaclients become: true roles: - role: ipaclient state: true</pre>

Authorization option	Note	Example inventory file	Example <code>install-client.yml</code> playbook file
The client keytab from the previous enrollment		<pre>[ipaclients:vars] ipaclient_keytab=/root/krb5.keytab</pre>	<pre>- name: Playbook to configure IPA clients hosts: ipaclients become: true roles: - role: ipaclient state: true</pre>



NOTE

As of RHEL 8.8, in the two OTP authorization scenarios described above, the requesting of the administrator's TGT by using the **kinit** command occurs on the first specified or discovered IdM server. Therefore, no additional modification of the Ansible control node is required. Prior to RHEL 8.8, the **krb5-workstation** package was required on the control node.

33.5. DEPLOYING AN IDM CLIENT USING AN ANSIBLE PLAYBOOK

Complete this procedure to use an Ansible playbook to deploy an IdM client in your IdM environment.

Prerequisites

- You have set the parameters of the IdM client deployment to correspond to your deployment scenario:
 - [Setting the parameters of the inventory file for the autodiscovery client installation mode](#)
 - [Setting the parameters of the inventory file when autodiscovery is not possible during client installation](#)
- You have checked [the parameters in `install-client.yml`](#).

Procedure

- To install an IdM client using an Ansible playbook, use the **ansible-playbook** command with the name of the playbook file, for example **install-client.yml**. Specify the inventory file with the **-i** option:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory/hosts install-client.yml
```

Specify the level of verbosity by using the **-v**, **-vv** or **-vvv** option.

Ansible informs you about the execution of the Ansible playbook script. The following output shows that the script has run successfully as no tasks have failed:

PLAY RECAP

```
client1.idm.example.com : ok=18 changed=10 unreachable=0 failed=0 skipped=21
rescued=0 ignored=0
```

**NOTE**

Ansible uses different colors to provide different types of information about the running process. You can modify the default colors in the **[colors]** section of the `/etc/ansible/ansible.cfg` file:

```
[colors]
[...]
#error = red
#debug = dark gray
#deprecate = purple
#skip = cyan
#unreachable = red
#ok = green
#changed = yellow
[...]
```

You have now installed an IdM client on your host using an Ansible playbook.

33.6. TESTING AN IDENTITY MANAGEMENT CLIENT AFTER ANSIBLE INSTALLATION

The command-line interface (CLI) informs you that the **ansible-playbook** command was successful, but you can also do your own test.

To test that the Identity Management client can obtain information about users defined on the server, check that you are able to resolve a user defined on the server. For example, to check the default **admin** user:

```
[user@client1 ~]$ id admin
uid=1254400000(admin) gid=1254400000(admins) groups=1254400000(admins)
```

To test that authentication works correctly, **su -** as another already existing IdM user:

```
[user@client1 ~]$ su - idm_user
Last login: Thu Oct 18 18:39:11 CEST 2018 from 192.168.122.1 on pts/0
[idm_user@client1 ~]$
```

33.7. UNINSTALLING AN IDM CLIENT USING AN ANSIBLE PLAYBOOK

Complete this procedure to use an Ansible playbook to uninstall your host as an IdM client.

Prerequisites

- IdM administrator credentials.

Procedure

- To uninstall the IdM client, use the **ansible-playbook** command with the name of the playbook file, for example **uninstall-client.yml**. Specify the inventory file with the **-i** option and, optionally, specify the level of verbosity by using the **-v**, **-vv** or **-vvv** options:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory/hosts uninstall-client.yml
```

IMPORTANT

The uninstallation of the client only removes the basic IdM configuration from the host but leaves the configuration files on the host in case you decide to re-install the client. In addition, the uninstallation has the following limitations:

- It does not remove the client host entry from the IdM LDAP server. The uninstallation only unenrolls the host.
- It does not remove any services residing on the client from IdM.
- It does not remove the DNS entries for the client from the IdM server.
- It does not remove the old principals for keytabs other than **/etc/krb5.keytab**.

Note that the uninstallation does remove all certificates that were issued for the host by the IdM CA.

Additional resources

- See [Uninstalling an IdM client](#).

PART II. INTEGRATING IDM AND AD

CHAPTER 34. INSTALLING TRUST BETWEEN IDM AND AD

Learn more about how to create a trust between the Identity Management IdM server and Active Directory (AD), where both servers are located in the same forest.

NOTE

In RHEL 7, *synchronization* and *trust* were two possible approaches to indirect integration of RHEL systems to Active Directory (AD). In RHEL 8, synchronization is deprecated. To integrate IdM and AD, use the trust approach instead. To migrate from synchronization to trust, see [Migrating an existing environment from synchronization to trust in the context of integrating a Linux domain with an Active Directory domain](#).

Prerequisites

- First, read the [Planning a cross-forest trust between Identity Management and Active Directory](#) document.
- AD is installed with a domain controller on it.
- The IdM server is installed and running.
 - For details, see [Installing Identity Management](#).
- Both the AD server and the IdM server must have their clocks in sync because Kerberos requires max 5 mins delay in communication.
- Unique NetBIOS names for each of the servers placed in the trust because the NetBIOS names are critical for identifying the Active Directory domain.
 - The NetBIOS name of an Active Directory or IdM domain is usually the first part of the corresponding DNS domain. If the DNS domain is **ad.example.com**, the NetBIOS name is typically **AD**. However, it is not required. Important is that the NetBIOS name is just one word without periods. The maximum length of a NetBIOS name is 15 characters.
- The IdM system must have the IPv6 protocol enabled in the kernel.
 - If IPv6 is disabled, then the CLDAP plug-in used by the IdM services fails to initialize.

34.1. SUPPORTED VERSIONS OF WINDOWS SERVER

You can establish a trust relationship with Active Directory (AD) forests that use the following forest and domain functional levels:

- Forest functional level range: Windows Server 2012 – Windows Server 2016
- Domain functional level range: Windows Server 2012 – Windows Server 2016

Identity Management (IdM) supports establishing a trust with Active Directory domain controllers running the following operating systems:

- Windows Server 2022 (RHEL 8.7 and later)
- Windows Server 2019
- Windows Server 2016

- Windows Server 2012 R2
- Windows Server 2012

**NOTE**

In RHEL 8.4, Identity Management (IdM) does not support establishing trust to Active Directory with Active Directory domain controllers running Windows Server 2008 R2 or earlier versions. RHEL IdM now requires SMB encryption when establishing the trust relationship, which is only supported in Windows Server 2012 or later.

34.2. HOW THE TRUST WORKS

The trust between Identity Management IdM and Active Directory (AD) is established on the Cross-realm Kerberos trust. This solution uses the Kerberos capability to establish trusts between different identity sources. Therefore, all AD users can:

- Log in to access Linux systems and resources.
- Use single sign-on (SSO).

All IdM objects are managed in IdM in the trust.

All AD objects are managed in AD in the trust.

In complex environments, a single IdM forest can be connected to multiple AD forests. This setup enables better separation of duties for different functions in the organization. AD administrators can focus on users and policies related to users while Linux administrators have full control over the Linux infrastructure. In such a case, the Linux realm controlled by IdM is analogous to an AD resource domain or realm but with Linux systems in it.

From the perspective of AD, Identity Management represents a separate AD forest with a single AD domain. When cross-forest trust between an AD forest root domain and an IdM domain is established, users from the AD forest domains can interact with Linux machines and services from the IdM domain.

**NOTE**

In trust environments, IdM enables you to use ID views to configure POSIX attributes for AD users on the IdM server.

34.3. AD ADMINISTRATION RIGHTS

When you want to build a trust between AD (Active Directory) and IdM (Identity Management), you will need to use an AD administrator account with appropriate AD privileges.

Such an AD administrator must be a member of one of the following groups:

- Enterprise Admin group in the AD forest
- Domain Admins group in the forest root domain for your AD forest

Additional resources

- For details about Enterprise Admins, see [Enterprise Admins](#).

- For details about Domain Admins, see [Domain Admins](#).
- For details about AD trust, see [How Domain and Forest Trusts Work](#).

34.4. ENSURING SUPPORT FOR COMMON ENCRYPTION TYPES IN AD AND RHEL

By default, Identity Management establishes a cross-realm trust with support for RC4, AES-128, and AES-256 Kerberos encryption types. Additionally, by default SSSD and Samba Winbind support RC4, AES-128, and AES-256 Kerberos encryption types.

RC4 encryption has been deprecated and disabled by default, as it is considered less secure than the newer AES-128 and AES-256 encryption types. In contrast, Active Directory (AD) user credentials and trusts between AD domains support RC4 encryption and they might not support all AES encryption types.

Without any common encryption types, communication between RHEL hosts and AD domains might not work, or some AD accounts might not be able to authenticate. To address this situation, perform one of the configurations outlined in the following sections.



IMPORTANT

If IdM is in FIPS mode, the IdM-AD integration does not work due to AD only supporting the use of RC4 or AES HMAC-SHA1 encryptions, while RHEL 9 in FIPS mode allows only AES HMAC-SHA2 by default. To enable the use of AES HMAC-SHA1 in RHEL 9, enter **# update-crypto-policies --set FIPS:AD-SUPPORT**.

IdM does not support the more restrictive **FIPS:OSPP** crypto policy, which should only be used on Common Criteria evaluated systems.

34.4.1. Enabling AES encryption in AD (recommended)

To ensure trusts between Active Directory (AD) domains in an AD forest support strong AES encryption types, see the following Microsoft article: [AD DS: Security: Kerberos "Unsupported etype" error when accessing a resource in a trusted domain](#)

34.4.2. Enabling the AES encryption type in Active Directory using a GPO

This section describes how to enable the AES encryption type in Active Directory (AD) using a group policy object (GPO). Certain features on RHEL, such as running a Samba server on an IdM client, require this encryption type.

Note that RHEL no longer supports the weak DES and RC4 encryption types.

Prerequisites

- You are logged into AD as a user who can edit group policies.
- The **Group Policy Management Console** is installed on the computer.

Procedure

1. Open the **Group Policy Management Console**.

2. Right-click **Default Domain Policy**, and select **Edit**. The **Group Policy Management Editor** opens.
3. Navigate to **Computer Configuration** → **Policies** → **Windows Settings** → **Security Settings** → **Local Policies** → **Security Options**.
4. Double-click the **Network security: Configure encryption types allowed for Kerberos** policy.
5. Select **AES256_HMAC_SHA1** and, optionally, **Future encryption types**.
6. Click **OK**.
7. Close the **Group Policy Management Editor**.
8. Repeat the steps for the **Default Domain Controller Policy**.
9. Wait until the Windows domain controllers (DC) applied the group policy automatically. Alternatively, to apply the GPO manually on a DC, enter the following command using an account that has administrator permissions:

```
C:\> gpupdate /force /target:computer
```

34.4.3. Enabling RC4 support in RHEL

On every RHEL host where authentication against AD Domain Controllers takes place, complete the steps outlined below.

Procedure

1. Use the **update-crypto-policies** command to enable the **AD-SUPPORT** cryptographic subpolicy in addition to the **DEFAULT** cryptographic policy.

```
[root@host ~]# update-crypto-policies --set DEFAULT:AD-SUPPORT
Setting system policy to DEFAULT:AD-SUPPORT
Note: System-wide crypto policies are applied on application start-up.
It is recommended to restart the system for the change of policies
to fully take place.
```

2. Restart the host.



IMPORTANT

The **AD-SUPPORT** cryptographic subpolicy is only available on RHEL 8.3 and newer.

- To enable support for RC4 in RHEL 8.2, create and enable a custom cryptographic module policy with **cipher = RC4-128+**. For more details, see [Customizing system-wide cryptographic policies with subpolicies](#).
- To enable support for RC4 in RHEL 8.0 and RHEL 8.1, add **+rc4** to the **permitted_encetypes** option in the `/etc/crypto-policies/back-ends/krb5.config` file:

```
[libdefaults]
permitted_encetypes = aes256-cts-hmac-sha1-96 aes256-cts-hmac-sha384-
192 camellia256-cts-cmac aes128-cts-hmac-sha1-96 aes128-cts-hmac-
sha256-128 camellia128-cts-cmac +rc4
```

34.4.4. Additional resources

- See [Using system-wide cryptographic policies](#).
- See [Trust controllers and trust agents](#).

34.5. PORTS REQUIRED FOR COMMUNICATION BETWEEN IDM AND AD

To enable communication between your Active Directory (AD) and Identity Management (IdM) environments, open the following ports on the firewalls of your AD Domain Controllers and IdM servers.

Table 34.1. Ports required for an AD trust

Service	Port	Protocol
Endpoint resolution portmapper	135	TCP
NetBIOS-DGM	138	TCP and UDP
NetBIOS-SSN	139	TCP and UDP
Microsoft-DS	445	TCP and UDP
Dynamic RPC	49152-65535	TCP
AD Global Catalog	3268	TCP
LDAP	389	TCP and UDP



NOTE

The TCP port 389 is not required to be open on IdM servers for trust, but it is necessary for clients communicating with the IdM server.

The TCP port 135 is required for the DCE RPC end-point mapper to work and is used during the IdM-AD trust creation.

To open ports, you can use the following methods:

- **firewalld** service – you can enable the particular ports or enable the following services which includes the ports:
 - FreeIPA trust setup
 - FreeIPA with LDAP
 - Kerberos
 - DNS

For details, see the **firewall-cmd** man page.



NOTE

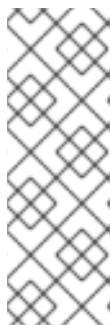
If you are using RHEL 8.2 and earlier, the **freeipa-trust** firewalld service includes an RPC port range of **1024-1300**, which is incorrect. On RHEL 8.2 and earlier, you must manually open the TCP port range **49152-65535** in addition to enabling the **freeipa-trust** firewalld service.

This issue has been fixed for RHEL 8.3 and later in [Bug 1850418 - update freeipa-trust.xml definition to include correct dynamic RPC range](#).

- The RHEL web console, which is a UI with firewall settings based on the **firewalld** service.

Service	TCP	UDP
Cockpit	9090	
DHCPv6 Client		546
DNS	53	53
FreeIPA trust setup	135, 138-139, 389, 445, 1024-1300, 3268	138-139, 389, 445
FreeIPA with LDAP	80, 443, 88, 464, 389	88, 464, 123
FreeIPA with LDAPS	80, 443, 88, 464, 636	88, 464, 123
Kerberos	88	88

For details about firewall configuration through the web console, see [Enabling services on the firewall using the web console](#)

**NOTE**

If you are using RHEL 8.2 and earlier, the **FreeIPA Trust Setup** service includes an RPC port range of **1024-1300**, which is incorrect. On RHEL 8.2 and earlier, you must manually open the TCP port range **49152-65535** in addition to enabling the **FreeIPA Trust Setup** service in the RHEL web console.

This issue has been fixed for RHEL 8.3 and later in [Bug 1850418 - update freeipa-trust.xml definition to include correct dynamic RPC range](#).

Table 34.2. Ports required by IdM servers in a trust

Service	Port	Protocol
Kerberos	88, 464	TCP and UDP
LDAP	389	TCP
DNS	53	TCP and UDP

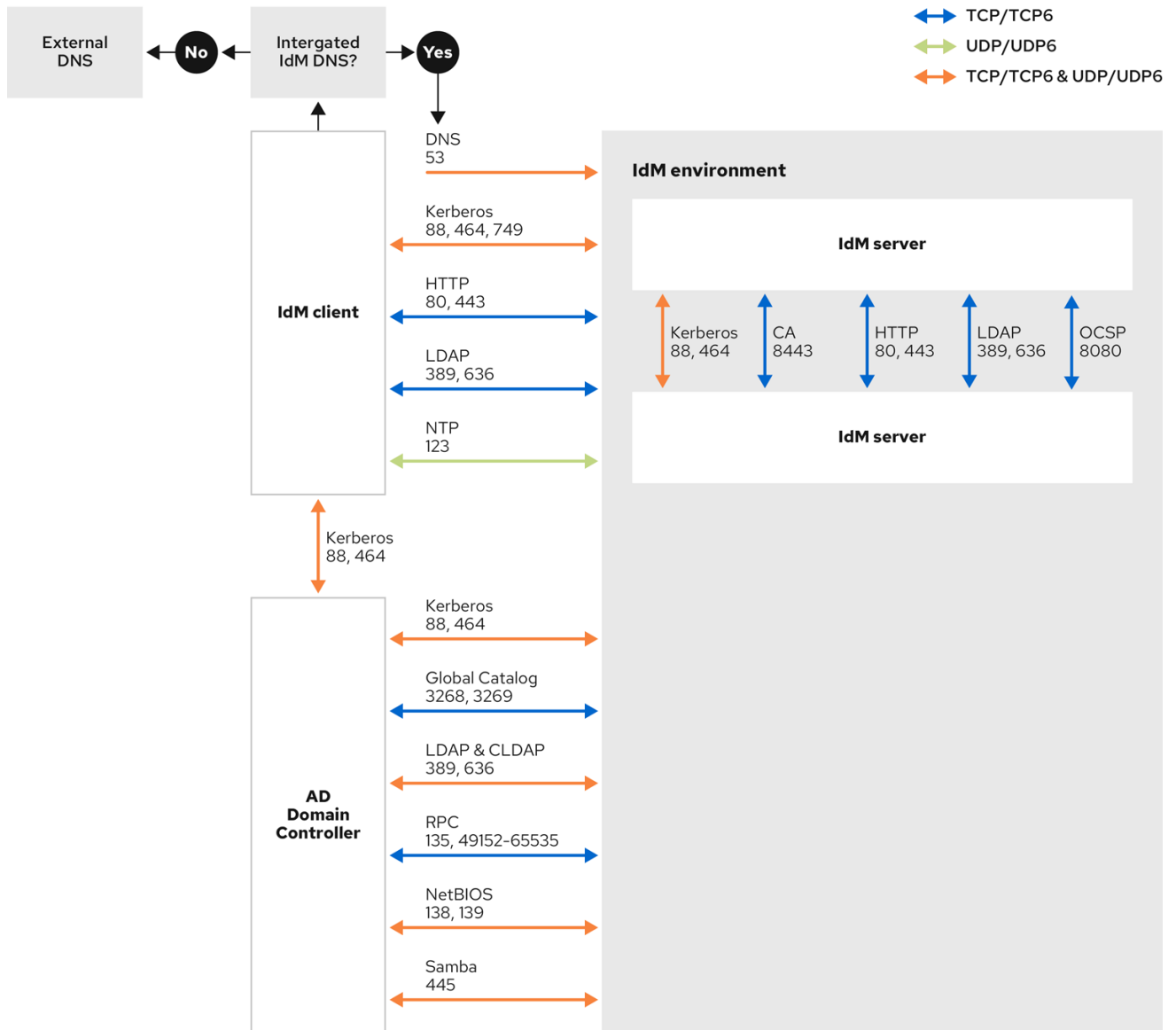
Table 34.3. Ports required by IdM clients in an AD trust

Service	Port	Protocol
Kerberos	88	UDP and TCP

**NOTE**

The **libkrb5** library uses UDP and falls back to the TCP protocol if the data sent from the Key Distribution Center (KDC) is too large. Active Directory attaches a Privilege Attribute Certificate (PAC) to the Kerberos ticket, which increases the size and requires to use the TCP protocol. To avoid the fall-back and resending the request, by default, SSSD in Red Hat Enterprise Linux 7.4 and later uses TCP for user authentication. If you want to configure the size before **libkrb5** uses TCP, set the **udp_preference_limit** in the **/etc/krb5.conf** file. For details, see the **krb5.conf(5)** man page.

The following diagram shows communication sent by IdM clients, and received and responded to by IdM servers and AD Domain Controllers. To set the incoming and outgoing ports and protocols on your firewall, Red Hat recommends using the **firewalld** service, which already has definitions for FreeIPA services.



231_RHEL_0422

Additional resources

- For more information about the Dynamic RPC port range in Windows Server 2008 and later, see [The default dynamic port range for TCP/IP has changed since Windows Vista and in Windows Server 2008](#).

34.6. CONFIGURING DNS AND REALM SETTINGS FOR A TRUST

Before you connect Identity Management (IdM) and Active Directory (AD) in a trust, you need to ensure that servers see each other and resolve domain names correctly. To configure DNS to allow using domain names between:

- One primary IdM server using integrated DNS server and Certification Authority.
- One AD Domain Controller.

DNS settings require:

- Configuring DNS zones in the IdM server

- Configuring conditional DNS forwarding in AD
- Verifying correctness of the DNS configuration

34.6.1. Unique primary DNS domains

In Windows, every domain is a Kerberos realm and a DNS domain at the same time. Every domain managed by the domain controller needs to have its own dedicated DNS zone. The same applies when Identity Management (IdM) is trusted by Active Directory (AD) as a forest. AD expects IdM to have its own DNS domain. For the trust setup to work, the DNS domain needs to be dedicated to the Linux environment.

Each system must have its own unique primary DNS domain configured. For example:

- ***ad.example.com*** for AD and ***idm.example.com*** for IdM
- ***example.com*** for AD and ***idm.example.com*** for IdM
- ***ad.example.com*** for AD and ***example.com*** for IdM

The most convenient management solution is an environment where each DNS domain is managed by integrated DNS servers, but it is possible to use any other standard-compliant DNS server as well.

Kerberos realm names as upper-case versions of primary DNS domain names

Kerberos realm names must be the same as the primary DNS domain names, with all letters uppercase. For example, if the domain names are ***ad.example.com*** for AD and ***idm.example.com*** for IdM, the Kerberos realm names are required to be ***AD.EXAMPLE.COM*** and ***IDM.EXAMPLE.COM***.

DNS records resolvable from all DNS domains in the trust

All machines must be able to resolve DNS records from all DNS domains involved in the trust relationship.

IdM and AD DNS Domains

Systems joined to IdM can be distributed over multiple DNS domains. Red Hat recommends that you deploy IdM clients in a DNS zone different to the ones owned by Active Directory. The primary IdM DNS domain must have proper SRV records to support AD trusts.



NOTE

In some environments with trusts between IdM and Active Directory, you can install an IdM client on a host that is part of the Active Directory DNS domain. The host can then benefit from the Linux-focused features of IdM. This is not a recommended configuration and has some limitations. See [Configuring IdM clients in an Active Directory DNS domain](#) for more details.

You can acquire a list of the required SRV records specific to your system setup by running the following command:

```
$ ipa dns-update-system-records --dry-run
```

The generated list can look for example like this:

```
IPA DNS records:
  _kerberos-master._tcp.idm.example.com. 86400 IN SRV 0 100 88 server.idm.example.com.
```

```

_kerberos-master._udp.idm.example.com. 86400 IN SRV 0 100 88 server.idm.example.com.
_kerberos._tcp.idm.example.com. 86400 IN SRV 0 100 88 server.idm.example.com.
_kerberos._tcp.idm.example.com. 86400 IN SRV 0 100 88 server.idm.example.com.
_kerberos.idm.example.com. 86400 IN TXT "IDM.EXAMPLE.COM"
_kpasswd._tcp.idm.example.com. 86400 IN SRV 0 100 464 server.idm.example.com.
_kpasswd._udp.idm.example.com. 86400 IN SRV 0 100 464 server.idm.example.com.
_ldap._tcp.idm.example.com. 86400 IN SRV 0 100 389 server.idm.example.com.
_ipa-ca.idm.example.com. 86400 IN A 192.168.122.2

```

For other DNS domains that are part of the same IdM realm, it is not required for the SRV records to be configured when the trust to AD is configured. The reason is that AD domain controllers do not use SRV records to discover KDCs but rather base the KDC discovery on name suffix routing information for the trust.

34.6.2. Configuring a DNS forward zone in the IdM Web UI

Follow this procedure to add a DNS forward zone to the Identity Management (IdM) server by using the IdM Web UI.

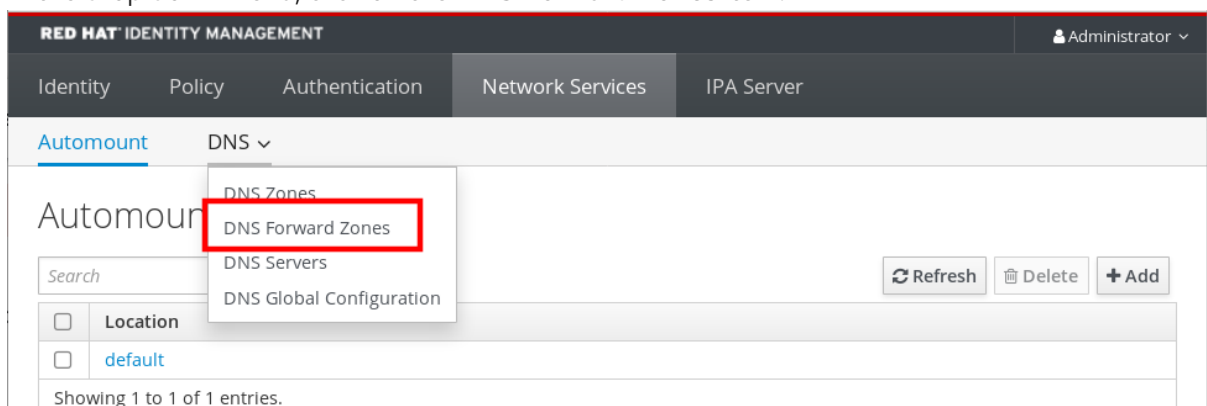
With DNS forward zones, you can forward DNS queries for a specific zone to a different DNS server. For example, you can forward DNS queries for the Active Directory (AD) domain to an AD DNS server.

Prerequisites

- Access to the IdM Web UI with a user account that has administrator rights.
- Correctly configured DNS server.

Procedure

1. Log in to the IdM Web UI with administrator privileges. For details, see [Accessing the IdM Web UI in a web browser](#).
2. Click on the **Network Services** tab.
3. Click on the **DNS** tab.
4. In the drop down menu, click on the **DNS Forward Zones** item.



5. Click on the **Add** button.
6. In the **Add DNS forward zone** dialog box, add a zone name.
7. In the **Zone forwarders** item, click on the **Add** button.

- In the **Zone forwarders** field, add the IP address of the server for which you want to create the forward zone.
- Click on the **Add** button.

Add DNS forward zone ✕

Zone name *

Reverse zone
IP network

Zone forwarders *

Forward policy **Forward first** **Forward only** **Forwarding disabled**

Skip overlap check ⓘ

* Required field

The forwarded zone has been added to the DNS settings and you can verify it in the DNS Forward Zones settings. The Web UI informs you about success with the following pop-up message: **DNS Forward Zone successfully added.**

NOTE

The Web UI might display a warning about a DNSSEC validation failure after adding a forward zone to the configuration.

The screenshot shows the Red Hat Identity Management web interface. At the top, there is a navigation bar with tabs for Identity, Policy, Authentication, and Network Services. A green notification banner at the top right states "DNS Forward Zone successfully added". Below this, a warning banner with an orange triangle icon indicates a "DNSSEC validation failed: record 'ad.example.com. SOA' failed DNSSEC validation on server 192.168.122.2. Please verify your DNSSEC configuration or disable DNSSEC validation on all IPA servers." The main content area is titled "DNS Forward Zones" and contains a search bar and a table with one entry:

<input type="checkbox"/>	Zone name	Status	Zone forwarders
<input type="checkbox"/>	ad.example.com.	✓ Enabled	192.168.122.3

Showing 1 to 1 of 1 entries.

DNSSEC (Domain Name System Security Extensions) secures DNS data with a digital signature to protect DNS from attacks. This service is enabled by default in the IdM server. The warning appears because the remote DNS server does not use DNSSEC. Red Hat recommends that you enable DNSSEC on the remote DNS server.

If you cannot enable DNSSEC validation on the remote server, you can disable DNSSEC in the IdM server:

1. Choose the appropriate configuration file to edit:
 - If your IdM server is using RHEL 8.0 or RHEL 8.1, open the **/etc/named.conf** file.
 - If your IdM server is using RHEL 8.2 or later, open the **/etc/named/ipa-options-ext.conf** file.

2. Add the following DNSSEC parameters:

```
dnssec-enable no;
dnssec-validation no;
```

3. Save and close the configuration file.
4. Restart the DNS service:

```
# systemctl restart named-pkcs11
```

Verification steps

- Use the **nslookup** command with the name of the remote DNS server:

```
$ nslookup ad.example.com
Server:      192.168.122.2
Address:    192.168.122.2#53
```

```
No-authoritative answer:
Name:      ad.example.com
Address:   192.168.122.3
```

If you configured the domain forwarding correctly, the IP address of the remote DNS server is displayed.

34.6.3. Configuring a DNS forward zone in the CLI

Follow this procedure to add a new DNS forward zone to the Identity Management (IdM) server using the command line interface (CLI).

With DNS forward zones, you can forward DNS queries for a specific zone to a different DNS server. For example, you can forward DNS queries for the Active Directory (AD) domain to an AD DNS server.

Prerequisites

- Access to the CLI with a user account that has administrator rights.
- Correctly configured DNS server.

Procedure

- Create a DNS forward zone for the AD domain, and specify the IP address of the remote DNS server with the **--forwarder** option:

```
# ipa dnsforwardzone-add ad.example.com --forwarder=192.168.122.3 --forward-policy=first
```

NOTE

You might see a warning about a DNSSEC validation failure in the `/var/log/messages` system logs after adding a new forward zone to the configuration:

```
named-pkcs11[2572]: no valid DS resolving 'host.ad.example.com/A/IN':
192.168.100.25#53
```

DNSSEC (Domain Name System Security Extensions) secures DNS data with a digital signature to protect DNS from attacks. This service is enabled by default in the IdM server. The warning appears because the remote DNS server does not use DNSSEC. Red Hat recommends that you enable DNSSEC on the remote DNS server.

If you cannot enable DNSSEC validation on the remote server, you can disable DNSSEC in the IdM server:

1. Choose the appropriate configuration file to edit:
 - If your IdM server is using RHEL 8.0 or RHEL 8.1, open the `/etc/named.conf` file.
 - If your IdM server is using RHEL 8.2 or later, open the `/etc/named/ipa-options-ext.conf` file.

2. Add the following DNSSEC parameters:

```
dnssec-enable no;
dnssec-validation no;
```

3. Save and close the configuration file.
4. Restart the DNS service:

```
# systemctl restart named-pkcs11
```

Verification steps

- Use the `nslookup` command with the name of the remote DNS server:

```
$ nslookup ad.example.com
Server:    192.168.122.2
Address:   192.168.122.2#53

No-authoritative answer:
Name:     ad.example.com
Address:  192.168.122.3
```

If the domain forwarding is configured correctly, the `nslookup` request displays an IP address of the remote DNS server.

34.6.4. Configuring DNS forwarding in AD

Follow this procedure to set up a DNS forwarding in Active Directory (AD) for the Identity Management (IdM) server.

Prerequisites

- Windows Server with AD installed.
- DNS port open on both servers.

Procedure

1. Log in to the Windows Server.
2. Open **Server Manager**.
3. Open **DNS Manager**.
4. In **Conditional Forwarders**, add a new conditional forwarder with:
 - The IdM server IP address
 - A fully qualified domain name, for example, **server.idm.example.com**
5. Save the settings.

34.6.5. Verifying the DNS configuration

Before configuring trust, verify that the Identity Management (IdM) and Active Directory (AD) servers can resolve themselves and each other.

Prerequisites

- You need to be logged in with sudo permissions.

Procedure

1. Run a DNS query for the Kerberos over UDP and LDAP over TCP service records.

```
[admin@server ~]# dig +short -t SRV _kerberos._udp.idm.example.com.
0 100 88 server.idm.example.com.
```

```
[admin@server ~]# dig +short -t SRV _ldap._tcp.idm.example.com.
0 100 389 server.idm.example.com.
```

The commands are expected to list all IdM servers.

2. Run a DNS query for the TXT record with the IdM Kerberos realm name. The obtained value is expected to match the Kerberos realm you specified when installing IdM.

```
[admin@server ~]# dig +short -t TXT _kerberos.idm.example.com.
"IDM.EXAMPLE.COM"
```

If the previous steps did not return all the expected records, update the DNS configuration with the missing records:

- If your IdM environment uses an integrated DNS server, enter the **ipa dns-update-system-records** command without any options to update your system records:

```
[admin@server ~]$ ipa dns-update-system-records
```

- If your IdM environment does not use an integrated DNS server:
 1. On the IdM server, export the IdM DNS records into a file:

```
[admin@server ~]$ ipa dns-update-system-records --dry-run --out
dns_records_file.nsupdate
```

The command creates a file named **dns_records_file.nsupdate** with the relevant IdM DNS records.

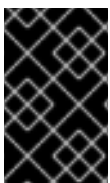
2. Submit a DNS update request to your DNS server using the **nsupdate** utility and the **dns_records_file.nsupdate** file. For more information, see [Updating External DNS Records Using nsupdate](#) in RHEL 7 documentation. Alternatively, refer to your DNS server documentation for adding DNS records.
3. Verify that IdM is able to resolve service records for AD with a command that runs a DNS query for Kerberos and LDAP over TCP service records:

```
[admin@server ~]# dig +short -t SRV _kerberos._tcp.dc._msdcs.ad.example.com.
0 100 88 addc1.ad.example.com.
```

```
[admin@server ~]# dig +short -t SRV _ldap._tcp.dc._msdcs.ad.example.com.
0 100 389 addc1.ad.example.com.
```

34.7. CONFIGURING IDM CLIENTS IN AN ACTIVE DIRECTORY DNS DOMAIN

If you have client systems in a DNS domain controlled by Active Directory and you require those clients to be able to join the IdM Server to benefit from its RHEL features, you can configure users to access a client using a host name from the Active Directory DNS domain.



IMPORTANT

This is not a recommended configuration and has some limitations. Red Hat recommends to always deploy IdM clients in a DNS zone different from the ones owned by Active Directory and access IdM clients through their IdM host names.

Your IdM client configuration depends on whether you require single sign-on with Kerberos.

34.7.1. Configuring an IdM client without Kerberos single sign-on

Password authentication is the only authentication method that is available for users to access resources on IdM clients if the IdM clients are in an Active Directory DNS domain. Follow this procedure to configure your client without Kerberos single sign-on.

Procedure

1. Install the IdM client with the **--domain=IPA_DNS_Domain** option to ensure the System Security Services Daemon (SSSD) can communicate with the IdM servers:

```
[root@idm-client.ad.example.com ~]# ipa-client-install --domain=idm.example.com
```

- This option disables the SRV record auto-detection for the Active Directory DNS domain.
- 2. Open the `/etc/krb5.conf` configuration file and locate the existing mapping for the Active Directory domain in the `[domain_realm]` section.

```
.ad.example.com = IDM.EXAMPLE.COM
ad.example.com = IDM.EXAMPLE.COM
```

- 3. Replace both lines with an entry mapping the fully qualified domain name (FQDN) of the Linux clients in the Active Directory DNS zone to the IdM realm:

```
idm-client.ad.example.com = IDM.EXAMPLE.COM
```

By replacing the default mapping, you prevent Kerberos from sending its requests for the Active Directory domain to the IdM Kerberos Distribution Center (KDC). Instead Kerberos uses auto-discovery through SRV DNS records to locate the KDC.

34.7.2. Requesting SSL certificates without single sign-on

SSL-based services require a certificate with **dnsName** extension records that cover all system host names, because both original (A/AAAA) and CNAME records must be in the certificate. Currently, IdM only issues certificates to host objects in the IdM database.

In the described setup without single sign-on available, IdM already has a host object for the FQDN in the database, and **certmonger** can request a certificate using this name.

Prerequisites

- Installed and configured the IdM client by following the procedure in [Configuring an IdM client without Kerberos single sign-on](#).

Procedure

- Use **certmonger** to request a certificate using the FQDN:

```
[root@idm-client.ad.example.com ~]# ipa-getcert request -r \
-f /etc/httpd/alias/server.crt \
-k /etc/httpd/alias/server.key \
-N CN=ipa-client.ad.example.com \
-D ipa-client.ad.example.com \
-K host/idm-client.ad.example.com@IDM.EXAMPLE.COM \
-U id-kp-serverAuth
```

The **certmonger** service uses the default host key stored in the `/etc/krb5.keytab` file to authenticate to the IdM Certificate Authority (CA).

34.7.3. Configuring an IdM client with Kerberos single sign-on

If you require Kerberos single sign-on to access resources on the IdM client, the client must be within the IdM DNS domain, for example **idm-client.idm.example.com**. You must create a CNAME record **idm-client.ad.example.com** in the Active Directory DNS domain pointing to the A/AAAA record of the IdM client.

For Kerberos-based application servers, MIT Kerberos supports a method to allow the acceptance of any host-based principal available in the application's keytab.

Procedure

- On the IdM client, disable the strict checks on what Kerberos principal is used to target the Kerberos server by setting the following option in the **[libdefaults]** section of the **/etc/krb5.conf** configuration file:

```
ignore_acceptor_hostname = true
```

34.7.4. Requesting SSL certificates with single sign-on

SSL-based services require a certificate with **dnsName** extension records that cover all system host names, because both original (A/AAAA) and CNAME records must be in the certificate. Currently, IdM only issues certificates to host objects in the IdM database.

Follow this procedure to create a host object for **ipa-client.example.com** in IdM and make sure the real IdM machine's host object is able to manage this host.

Prerequisites

- You have disabled the strict checks on what Kerberos principal is used to target the Kerberos server as outlined in [Configuring an IdM client with Kerberos single sign-on](#).

Procedure

1. Create a new host object on the IdM server:

```
[root@idm-server.idm.example.com ~]# ipa host-add idm-client.ad.example.com --force
```

Use the **--force** option, because the host name is a CNAME and not an A/AAAA record.

2. On the IdM server, allow the IdM DNS host name to manage the Active Directory host entry in the IdM database:

```
[root@idm-server.idm.example.com ~]# ipa host-add-managedby idm-client.ad.example.com \
--hosts=idm-client.idm.example.com
```

3. You can now request an SSL certificate for your IdM client with the **dnsName** extension record for its host name within the Active Directory DNS domain:

```
[root@idm-client.idm.example.com ~]# ipa-getcert request -r \
-f /etc/httpd/alias/server.crt \
-k /etc/httpd/alias/server.key \
-N CN=`hostname --fqdn` \
-D `hostname --fqdn` \
-D idm-client.ad.example.com \
-K host/idm-client.idm.example.com@IDM.EXAMPLE.COM \
-U id-kp-serverAuth
```

34.8. SETTING UP A TRUST

This section describes how to configure the Identity Management (IdM)/Active Directory (AD) trust on the IdM side using the command line.

Prerequisites

- DNS is correctly configured. Both IdM and AD servers must be able to resolve each other names. For details, see [Configuring DNS and realm settings for a trust](#).
- Supported versions of AD and IdM are deployed. For details, see [Supported versions of Windows Server](#).
- You have obtained a Kerberos ticket. For details, see [Using kinit to log in to IdM manually](#).

34.8.1. Preparing the IdM server for the trust

Before you can establish a trust with AD, you must prepare the IdM domain using the **ipa-adtrust-install** utility on an IdM server.



NOTE

Any system where you run the **ipa-adtrust-install** command automatically becomes an AD trust controller. However, you must run **ipa-adtrust-install** only once on an IdM server.

Prerequisites

- IdM server is installed.
- You need root privileges to install packages and restart IdM services.

Procedure

1. Install the required packages:

```
[root@ipaserver ~]# yum install ipa-server-trust-ad samba-client
```

2. Authenticate as the IdM administrative user:

```
[root@ipaserver ~]# kinit admin
```

3. Run the **ipa-adtrust-install** utility:

```
[root@ipaserver ~]# ipa-adtrust-install
```

The DNS service records are created automatically if IdM was installed with an integrated DNS server.

If you installed IdM without an integrated DNS server, **ipa-adtrust-install** prints a list of service records that you must manually add to DNS before you can continue.

4. The script prompts you that the **/etc/samba/smb.conf** already exists and will be rewritten:

```
WARNING: The smb.conf already exists. Running ipa-adtrust-install will break your existing Samba configuration.
```

```
Do you wish to continue? [no]: yes
```

- The script prompts you to configure the **slapi-nis** plug-in, a compatibility plug-in that allows older Linux clients to work with trusted users:

```
Do you want to enable support for trusted domains in Schema Compatibility plugin?  
This will allow clients older than SSSD 1.9 and non-Linux clients to work with trusted users.
```

```
Enable trusted domains support in slapi-nis? [no]: yes
```

- When prompted, enter the NetBIOS name for the IdM domain or press **Enter** to accept the name suggested:

```
Trust is configured but no NetBIOS domain name found, setting it now.  
Enter the NetBIOS name for the IPA domain.  
Only up to 15 uppercase ASCII letters, digits and dashes are allowed.  
Example: EXAMPLE.
```

```
NetBIOS domain name [IDM]:
```

- You are prompted to run the SID generation task to create a SID for any existing users:

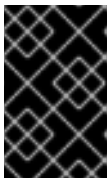
```
Do you want to run the ipa-sidgen task? [no]: yes
```

This is a resource-intensive task, so if you have a high number of users, you can run this at another time.

- (Optional)** By default, the Dynamic RPC port range is defined as **49152-65535** for Windows Server 2008 and later. If you need to define a different Dynamic RPC port range for your environment, configure Samba to use different ports and open those ports in your firewall settings. The following example sets the port range to **55000-65000**.

```
[root@ipaserver ~]# net conf setparm global 'rpc server dynamic port range' 55000-65000  
[root@ipaserver ~]# firewall-cmd --add-port=55000-65000/tcp  
[root@ipaserver ~]# firewall-cmd --runtime-to-permanent
```

- Make sure that DNS is properly configured, as described in [Verifying the DNS configuration for a trust](#).



IMPORTANT

Red Hat strongly recommends you verify the DNS configuration as described in [Verifying the DNS configuration for a trust](#) every time after running **ipa-adtrust-install**, especially if IdM or AD do not use integrated DNS servers.

- Restart the **ipa** service:

```
[root@ipaserver ~]# ipactl restart
```

- Use the **smbclient** utility to verify that Samba responds to Kerberos authentication from the IdM side:

```
[root@ipaserver ~]# smbclient -L ipaserver.idm.example.com -U user_name --use-
kerberos=required
lp_load_ex: changing to config backend registry
  Sharename      Type      Comment
  -----      -
  IPC$           IPC       IPC Service (Samba 4.15.2)
  ...
```

34.8.2. Setting up a trust agreement using the command line

Follow this procedure to set up the trust agreement using the command line. The Identity Management (IdM) server allows you to configure three types of trust agreements:

- **One-way trust** – default option. One-way trust enables Active Directory (AD) users and groups to access resources in IdM, but not the other way around. The IdM domain trusts the AD forest, but the AD forest does not trust the IdM domain.
- **Two-way trust** – Two-way trust enables AD users and groups to access resources in IdM. You must configure a two-way trust for solutions such as Microsoft SQL Server that expect the **S4U2Self** and **S4U2Proxy** Microsoft extensions to the Kerberos protocol to work over a trust boundary. An application on a RHEL IdM host might request **S4U2Self** or **S4U2Proxy** information from an Active Directory domain controller about an AD user, and a two-way trust provides this feature.

Note that this two-way trust functionality does not allow IdM users to login to Windows systems, and the two-way trust in IdM does not give the users any additional rights compared to the one-way trust solution in AD.

- To create the two-way trust, add the following option to the command: **--two-way=true**
- **External trust** – a trust relationship between IdM and an AD domain in different forests. While a forest trust always requires establishing a trust between IdM and the root domain of an Active Directory forest, an external trust can be established from IdM to a domain within a forest. This is only recommended if it is not possible to establish a forest trust between forest root domains due to administrative or organizational reasons.
 - To create the external trust, add the following option to the command: **--external=true**

The steps below show you how to create a one-way trust agreement.

Prerequisites

- User name and password of a Windows administrator.
- You have [prepared the IdM server for the trust](#).

Procedure

- Create a trust agreement for the AD domain and the IdM domain by using the **ipa trust-add** command:
 - To have SSSD automatically generate UIDs and GIDs for AD users based on their SID, create a trust agreement with the **Active Directory domain** ID range type. This is the most common configuration.

```
[root@server ~]# ipa trust-add --type=ad ad.example.com --admin
<ad_admin_username> --password --range-type=ipa-ad-trust
```

- If you have configured POSIX attributes for your users in Active Directory (such as **uidNumber** and **gidNumber**) and you want SSSD to process this information, create a trust agreement with the **Active Directory domain with POSIX attributes** ID range type:

```
[root@server ~]# ipa trust-add --type=ad ad.example.com --admin
<ad_admin_username> --password --range-type=ipa-ad-trust-posix
```



WARNING

If you do not specify an ID Range type when creating a trust, IdM attempts to automatically select the appropriate range type by requesting details from AD domain controllers in the forest root domain. If IdM does not detect any POSIX attributes, the trust installation script selects the **Active Directory domain** ID range.

If IdM detects any POSIX attributes in the forest root domain, the trust installation script selects the **Active Directory domain with POSIX attributes** ID range and assumes that UIDs and GIDs are correctly defined in AD. If POSIX attributes are not correctly set in AD, you will not be able to resolve AD users.

For example, if the users and groups that need access to IdM systems are not part of the forest root domain, but instead are located in a child domain of the forest domain, the installation script may not detect the POSIX attributes defined in the child AD domain. In this case, Red Hat recommends that you explicitly choose the POSIX ID range type when establishing the trust.

34.8.3. Setting up a trust agreement in the IdM Web UI

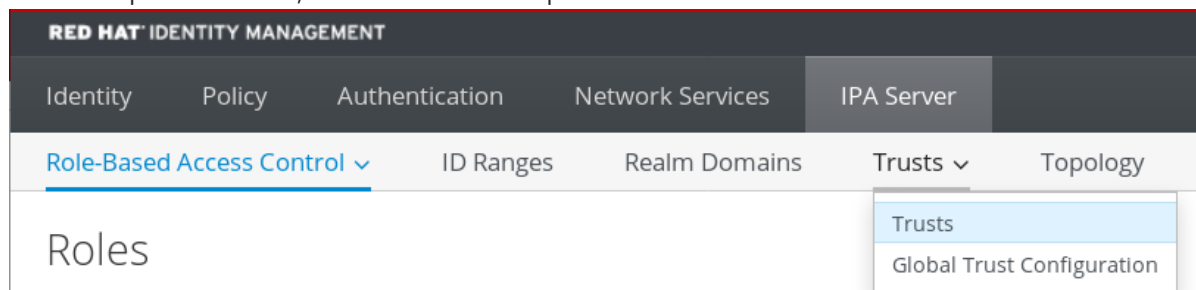
Follow this procedure to configure the Identity Management (IdM)/Active Directory (AD) trust agreement on the IdM side using the IdM Web UI.

Prerequisites

- DNS is correctly configured. Both IdM and AD servers must be able to resolve each other names.
- Supported versions of AD and IdM are deployed.
- You have obtained a Kerberos ticket.
- Before creating a trust in the Web UI, prepare the IdM server for the trust as described in: [Preparing the IdM server for the trust](#).
- You need to be logged in as an IdM administrator.

Procedure

1. Log in to the IdM Web UI with administrator privileges. For details, see [Accessing the IdM Web UI in a web browser](#).
2. In the IdM Web UI, click the **IPA Server** tab.
3. In the **IPA Server** tab, click the **Trusts** tab.
4. In the drop down menu, select the **Trusts** option.



5. Click the **Add** button.
6. In the **Add Trust** dialog box, enter the name of the Active Directory domain.
7. In the **Account** and **Password** fields, add the administrator credentials of the Active Directory administrator.

8. (Optional) Select **Two-way trust**, if you want to enable AD users and groups to access resources in IdM. However, the two-way trust in IdM does not give the users any additional rights compared to the one-way trust solution in AD. Both solutions are considered equally secure because of default cross-forest trust SID filtering settings.
9. (Optional) Select **External trust** if you are configuring a trust with an AD domain that is not the root domain of an AD forest. While a forest trust always requires establishing a trust between IdM and the root domain of an Active Directory forest, you can establish an external trust from IdM to any domain within an AD forest.
10. (Optional) By default, the trust installation script tries to detect the appropriate ID range type. You can also explicitly set the ID range type by choosing one of the following options:

- To have SSSD automatically generate UID and GID for AD users based on their SID

- a. To have SSSD automatically generate UIDs and GIDs for AD users based on their SID, select the **Active Directory domain** ID range type. This is the most common configuration.
- b. If you have configured POSIX attributes for your users in Active Directory (such as **uidNumber** and **gidNumber**) and you want SSSD to process this information, select the **Active Directory domain with POSIX attributes** ID range type.

<p>Range type</p> <p><input checked="" type="radio"/> Detect</p> <p><input type="radio"/> Active Directory domain</p> <p><input type="radio"/> Active Directory domain with POSIX attributes</p>
--



WARNING

If you leave the **Range type** setting on the default **Detect** option, IdM attempts to automatically select the appropriate range type by requesting details from AD domain controllers in the forest root domain. If IdM does not detect any POSIX attributes, the trust installation script selects the **Active Directory domain** ID range.

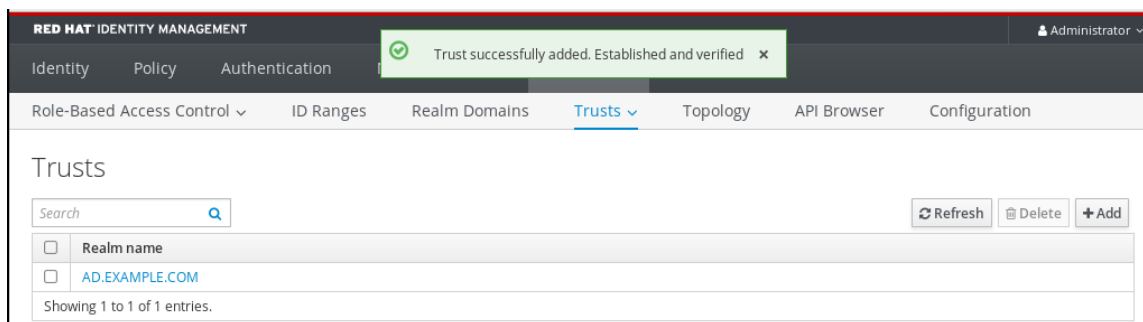
If IdM detects any POSIX attributes in the forest root domain, the trust installation script selects the **Active Directory domain with POSIX attributes** ID range and assumes that UIDs and GIDs are correctly defined in AD. If POSIX attributes are not correctly set in AD, you will not be able to resolve AD users.

For example, if the users and groups that need access to IdM systems are not part of the forest root domain, but instead are located in a child domain of the forest domain, the installation script may not detect the POSIX attributes defined in the child AD domain. In this case, Red Hat recommends that you explicitly choose the POSIX ID range type when establishing the trust.

11. Click **Add**.

Verification steps

- If the trust has been successfully added to the IdM server, you can see the green pop-up window in the IdM Web UI. It means that the:
 - Domain name exists
 - User name and password of the Windows Server has been added correctly.



Now you can continue to test the trust connection and Kerberos authentication.

34.8.4. Setting up a trust agreement using Ansible

Follow this procedure to set up a one-way trust agreement between Identity Management (IdM) and Active Directory (AD) by using an Ansible playbook. You can configure three types of trust agreements:

- **One-way trust** – default option. One-way trust enables Active Directory (AD) users and groups to access resources in IdM, but not the other way around. The IdM domain trusts the AD forest, but the AD forest does not trust the IdM domain.
- **Two-way trust** – Two-way trust enables AD users and groups to access resources in IdM. You must configure a two-way trust for solutions such as Microsoft SQL Server that expect the **S4U2Self** and **S4U2Proxy** Microsoft extensions to the Kerberos protocol to work over a trust boundary. An application on a RHEL IdM host might request **S4U2Self** or **S4U2Proxy** information from an Active Directory domain controller about an AD user, and a two-way trust provides this feature.

Note that this two-way trust functionality does not allow IdM users to login to Windows systems, and the two-way trust in IdM does not give the users any additional rights compared to the one-way trust solution in AD.

- To create the two-way trust, add the following variable to the playbook task below:
 - two_way: true**
- **External trust** – a trust relationship between IdM and an AD domain in different forests. While a forest trust always requires establishing a trust between IdM and the root domain of an Active Directory forest, an external trust can be established from IdM to a domain within a forest. This is only recommended if it is not possible to establish a forest trust between forest root domains due to administrative or organizational reasons.
 - To create the external trust, add the following variable to the playbook task below: **external: true**

Prerequisites

- User name and password of a Windows administrator.
- The IdM **admin** password.
- You have [prepared the IdM server for the trust](#).
- You are using the 4.8.7 version of IdM or later. To view the version of IdM you have installed on your server, run **ipa --version**.
- You have configured your Ansible control node to meet the following requirements:

- You are using Ansible version 2.14 or later.
- You have installed the **ansible-freeipa** package on the Ansible controller.
- The example assumes that in the `~/MyPlaybooks/` directory, you have created an [Ansible inventory file](#) with the fully-qualified domain name (FQDN) of the IdM server.
- The example assumes that the **secret.yml** Ansible vault stores your **ipadmin_password**.
- The target node, that is the node on which the **ansible-freeipa** module is executed, is part of the IdM domain as an IdM client, server or replica.

Procedure

1. Navigate to your `~/MyPlaybooks/` directory:

```
$ cd ~/MyPlaybooks/
```

2. Select one of the following scenarios based on your use case:

- To create an ID mapping trust agreement, in which SSSD automatically generates UIDs and GIDs for AD users and groups based on their SIDs, create an **add-trust.yml** playbook with the following content:

```
---
- name: Playbook to create a trust
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: ensure the trust is present
    ipatrust:
      ipadmin_password: "{{ ipadmin_password }}"
      realm: ad.example.com
      admin: Administrator
      password: secret_password
      range_type: ipa-ad-trust
      state: present
```

In the example:

- **realm** defines the AD realm name string.
- **admin** defines the AD domain administrator string.
- **password** defines the AD domain administrator's password string.
- To create a POSIX trust agreement, in which SSSD processes POSIX attributes stored in AD, such as **uidNumber** and **gidNumber**, create an **add-trust.yml** playbook with the following content:

```
---
- name: Playbook to create a trust
  hosts: ipaserver
```



```

vars_files:
- /home/user_name/MyPlaybooks/secret.yml
tasks:
- name: ensure the trust is present
  ipatrust:
    ipaadmin_password: "{{ ipaadmin_password }}"
    realm: ad.example.com
    admin: Administrator
    password: secret_password
    range_type: ipa-ad-trust-posix
    state: present

```

- To create a trust agreement in which IdM attempts to automatically select the appropriate range type, **ipa-ad-trust** or **ipa-ad-trust-posix**, by requesting details from AD domain controllers in the forest root domain, create an **add-trust.yml** playbook with the following content:

```

---
- name: Playbook to create a trust
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: ensure the trust is present
    ipatrust:
      ipaadmin_password: "{{ ipaadmin_password }}"
      realm: ad.example.com
      admin: Administrator
      password: secret_password
      state: present

```



WARNING

If you do not specify an ID range type when creating a trust, and if IdM does not detect any POSIX attributes in the AD forest root domain, the trust installation script selects the **Active Directory domain** ID range.

If IdM detects any POSIX attributes in the forest root domain, the trust installation script selects the **Active Directory domain with POSIX attributes** ID range and assumes that UIDs and GIDs are correctly defined in AD.

However, if POSIX attributes are not correctly set in AD, you will not be able to resolve AD users. For example, if the users and groups that need access to IdM systems are not part of the forest root domain, but instead are located in a child domain of the forest domain, the installation script may not detect the POSIX attributes defined in the child AD domain. In this case, Red Hat recommends that you explicitly choose the POSIX ID range type when establishing the trust.

3. Save the file.
4. Run the Ansible playbook. Specify the playbook file, the file storing the password protecting the **secret.yml** file, and the inventory file:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory add-trust.yml
```

Additional resources

- /usr/share/doc/ansible-freeipa/README-trust.md
- /usr/share/doc/ansible-freeipa/playbooks/trust

34.8.5. Verifying the Kerberos configuration

To verify the Kerberos configuration, test if it is possible to obtain a ticket for an Identity Management (IdM) user and if the IdM user can request service tickets.

Procedure

1. Request a ticket for an Active Directory (AD) user:

```
[root@ipaserver ~]# kinit user@AD.EXAMPLE.COM
```

2. Request service tickets for a service within the IdM domain:

```
[root@server ~]# kvno -S host server.idm.example.com
```

If the AD service ticket is successfully granted, there is a cross-realm ticket-granting ticket (TGT) listed with all of the other requested tickets. The TGT is named `krbtgt/IPA.DOMAIN@AD.DOMAIN`.

```
[root@server ~]# klist
Ticket cache: KEYRING:persistent:0:krb_ccache_hRtox00
Default principal: user@AD.EXAMPLE.COM
```

```
Valid starting    Expires          Service principal
03.05.2016 18:31:06 04.05.2016 04:31:01 host/server.idm.example.com@IDM.EXAMPLE.COM
renew until 04.05.2016 18:31:00
03.05.2016 18:31:06 04.05.2016 04:31:01 krbtgt/IDM.EXAMPLE.COM@AD.EXAMPLE.COM
renew until 04.05.2016 18:31:00
03.05.2016 18:31:01 04.05.2016 04:31:01 krbtgt/AD.EXAMPLE.COM@AD.EXAMPLE.COM
renew until 04.05.2016 18:31:00
```

The **localauth** plug-in maps Kerberos principals to local System Security Services Daemon (SSSD) user names. This allows AD users to use Kerberos authentication and access Linux services, which support GSSAPI authentication directly.

34.8.6. Verifying the trust configuration on IdM

Before configuring trust, verify that the Identity Management (IdM) and Active Directory (AD) servers can resolve themselves and each other.

Prerequisites

- You need to be logged in with administrator privileges.

Procedure

1. Run a DNS query for the MS DC Kerberos over UDP and LDAP over TCP service records.

```
[root@server ~]# dig +short -t SRV _kerberos._udp.dc._msdcs.idm.example.com.
0 100 88 server.idm.example.com.
```

```
[root@server ~]# dig +short -t SRV _ldap._tcp.dc._msdcs.idm.example.com.
0 100 389 server.idm.example.com.
```

These commands list all IdM servers on which **ipa-adtrust-install** has been executed. The output is empty if **ipa-adtrust-install** has not been executed on any IdM server, which is typically before establishing the first trust relationship.

2. Run a DNS query for the Kerberos and LDAP over TCP service records to verify that IdM is able to resolve service records for AD:

```
[root@server ~]# dig +short -t SRV _kerberos._tcp.dc._msdcs.ad.example.com.
0 100 88 addc1.ad.example.com.
```

```
[root@ipaserver ~]# dig +short -t SRV _ldap._tcp.dc._msdcs.ad.example.com.
0 100 389 addc1.ad.example.com.
```

34.8.7. Verifying the trust configuration on AD

After configuring the trust, verify that:

- The Identity Management (IdM)-hosted services are resolvable from the Active Directory (AD) server.
- AD services are resolvable from the AD server.

Prerequisites

- You need to be logged in with administrator privileges.

Procedure

1. On the AD server, set the **nslookup.exe** utility to look up service records.

```
C:\>nslookup.exe
> set type=SRV
```

2. Enter the domain name for the Kerberos over UDP and LDAP over TCP service records.

```
> _kerberos._udp.idm.example.com.
_kerberos._udp.idm.example.com.    SRV service location:
priority          = 0
```

```

weight          = 100
port            = 88
svr hostname    = server.idm.example.com
> _ldap._tcp.idm.example.com
_ldap._tcp.idm.example.com  SRV service location:
priority        = 0
weight          = 100
port            = 389
svr hostname    = server.idm.example.com

```

3. Change the service type to TXT and run a DNS query for the TXT record with the IdM Kerberos realm name.

```

C:\>nslookup.exe
> set type=TXT
> _kerberos.idm.example.com.
_kerberos.idm.example.com.  text =

"IDM.EXAMPLE.COM"

```

4. Run a DNS query for the MS DC Kerberos over UDP and LDAP over TCP service records.

```

C:\>nslookup.exe
> set type=SRV
> _kerberos._udp.dc._msdcs.idm.example.com.
_kerberos._udp.dc._msdcs.idm.example.com.  SRV service location:
priority = 0
weight = 100
port = 88
svr hostname = server.idm.example.com
> _ldap._tcp.dc._msdcs.idm.example.com.
_ldap._tcp.dc._msdcs.idm.example.com.  SRV service location:
priority = 0
weight = 100
port = 389
svr hostname = server.idm.example.com

```

Active Directory only expects to discover domain controllers that can respond to AD-specific protocol requests, such as other AD domain controllers and IdM trust controllers. Use the **ipa-adtrust-install** tool to promote an IdM server to a trust controller, and you can verify which servers are trust controllers with the **ipa server-role-find --role 'AD trust controller'** command.

5. Verify that AD services are resolvable from the AD server.

```

C:\>nslookup.exe
> set type=SRV

```

6. Enter the domain name for the Kerberos over UDP and LDAP over TCP service records.

```

> _kerberos._udp.dc._msdcs.ad.example.com.
_kerberos._udp.dc._msdcs.ad.example.com.  SRV service location:
priority = 0
weight = 100
port = 88
svr hostname = addc1.ad.example.com

```

```
> _ldap._tcp.dc._msdcs.ad.example.com.
_ldap._tcp.dc._msdcs.ad.example.com. SRV service location:
  priority = 0
  weight = 100
  port = 389
  svr hostname = addc1.ad.example.com
```

34.8.8. Creating a trust agent

A trust agent is an IdM server that can perform identity lookups against AD domain controllers.

For example, if you are creating a replica of an IdM server that has a trust with Active Directory, you can set up the replica as a trust agent. A replica does not automatically have the AD trust agent role installed.

Prerequisites

- IdM is installed with an Active Directory trust.
- The **sssd-tools** package is installed.

Procedure

1. On an existing trust controller, run the **ipa-adtrust-install --add-agents** command:

```
[root@existing_trust_controller]# ipa-adtrust-install --add-agents
```

The command starts an interactive configuration session and prompts you for the information required to set up the agent.

2. Restart the IdM service on the trust agent.

```
[root@new_trust_agent]# ipactl restart
```

3. Remove all entries from the SSSD cache on the trust agent:

```
[root@new_trust_agent]# sssctl cache-remove
```

4. Verify that the replica has the AD trust agent role installed:

```
[root@existing_trust_controller]# ipa server-show new_replica.idm.example.com
...
Enabled server roles: CA server, NTP server, AD trust agent
```

Additional resources

- For further information about the **--add-agents** option, see the **ipa-adtrust-install(1)** man page.
- For more information about trust agents, see [Trust controllers and trust agents](#) in the Planning Identity Management guide.

34.8.9. Enabling automatic private group mapping for a POSIX ID range on the CLI

By default, SSSD does not map private groups for Active Directory (AD) users if you have established a POSIX trust that relies on POSIX data stored in AD. If any AD users do not have primary groups configured, IdM is not be able to resolve them.

This procedure explains how to enable automatic private group mapping for an ID range by setting the **hybrid** option for the **auto_private_groups** SSSD parameter on the command line. As a result, IdM is able to resolve AD users that do not have primary groups configured in AD.

Prerequisites

- You have successfully established a POSIX cross-forest trust between your IdM and AD environments.

Procedure

1. Display all ID ranges and make note of the AD ID range you want to modify.

```
[root@server ~]# ipa idrange-find
-----
2 ranges matched
-----
Range name: IDM.EXAMPLE.COM_id_range
First Posix ID of the range: 882200000
Number of IDs in the range: 200000
Range type: local domain range

Range name: AD.EXAMPLE.COM_id_range
First Posix ID of the range: 1337000000
Number of IDs in the range: 200000
Domain SID of the trusted domain: S-1-5-21-4123312420-990666102-3578675309
Range type: Active Directory trust range with POSIX attributes
-----
Number of entries returned 2
-----
```

2. Adjust the automatic private group behavior for the AD ID range with the **ipa idrange-mod** command.

```
[root@server ~]# ipa idrange-mod --auto-private-groups=hybrid
AD.EXAMPLE.COM_id_range
```

3. Reset the SSSD cache to enable the new setting.

```
[root@server ~]# sss_cache -E
```

Additional resources

- [Options for automatically mapping private groups for AD users](#)

34.8.10. Enabling automatic private group mapping for a POSIX ID range in the IdM WebUI

By default, SSSD does not map private groups for Active Directory (AD) users if you have established a POSIX trust that relies on POSIX data stored in AD. If any AD users do not have primary groups configured, IdM is not be able to resolve them.

This procedure explains how to enable automatic private group mapping for an ID range by setting the **hybrid** option for the **auto_private_groups** SSSD parameter in the Identity Management (IdM) WebUI. As a result, IdM is able to resolve AD users that do not have primary groups configured in AD.

Prerequisites

- You have successfully established a POSIX cross-forest trust between your IdM and AD environments.

Procedure

1. Log into the IdM Web UI with your user name and password.
2. Open the **IPA Server** → **ID Ranges** tab.
3. Select the ID range you want to modify, such as **AD.EXAMPLE.COM_id_range**.
4. From the **Auto private groups** drop down menu, select the **hybrid** option.

The screenshot displays the IdM WebUI interface for configuring an ID range. The navigation tabs at the top include Identity, Policy, Authentication, Network Services, IPA Server, and Topology. The current view is for the ID Range: AD.EXAMPLE.COM_id_range. Below the title, there are buttons for Settings, Refresh, Revert, and Save. The Range Settings section shows the following configuration:

Range name	AD.EXAMPLE.COM_id_range
Range type	Active Directory trust range with POSIX attributes
Base ID *	1045000000
Range size *	200000
Domain SID	S-1-5-21-4029230055-4155305145-370140224
Auto private groups	hybrid

5. Click the **Save** button to save your changes.

Additional resources

- [Options for automatically mapping private groups for AD users](#)

34.9. TROUBLESHOOTING SETTING UP A CROSS-FOREST TRUST

Learn more about troubleshooting the process of configuring a cross-forest trust between your Identity Management (IdM) environment and an Active Directory (AD) forest.

34.9.1. Sequence of events when establishing a cross-forest trust with AD

When you use the **ipa trust-add** command to establish a cross-forest trust with an Active Directory (AD) Domain Controller (DC), the command operates on behalf of the user who ran the command and performs the following actions on the IdM server. If you have trouble establishing a cross-forest trust, you can use this list to help narrow down and troubleshoot your issue.

Part 1: The command verifies settings and inputs

1. Verify that the IdM server has the **Trust Controller** role.
2. Validate the options passed to the **ipa trust-add** command.
3. Validate the ID range associated with a trusted forest root domain. If you did not specify the ID range type and properties as options to the **ipa trust-add** command, they are discovered from Active Directory.

Part 2: The command attempts to establish a trust to an Active Directory domain

4. Create a separate trust object for each trust direction. Each of the objects get created on both sides (IdM and AD). If you are establishing a one-way trust, only one object is created on each side.
5. The IdM server uses the Samba suite to handle domain controller capabilities for Active Directory and creates a trust object on the target AD PDC:
 - a. The IdM server establishes a secure connection to the **IPC\$** share on the target DC. Since RHEL 8.4, the connection requires at least the SMB3 protocol with Windows Server 2012 and above to ensure the connection is sufficiently secure with AES-based encryption used for the session.
 - b. The IdM server queries for the presence of the trusted domain object (TDO) using an **LSA QueryTrustedDomainInfoByName** call.
 - c. If the TDO is already present, remove it with an **LSA DeleteTrustedDomain** call.



NOTE

This call fails if the AD user account used to establish the trust does not have full **Enterprise Admin (EA)** or **Domain Admin (DA)** privileges for the forest root, such as members of the **Incoming Forest Trust Builders** group. If the old TDO is not automatically removed, an AD Administrator must manually remove it from AD.

- d. The IdM server creates a new TDO with an **LSA CreateTrustedDomainEx2** call. The TDO credentials are randomly generated using a Samba-provided password generator with 128 random characters.
 - e. The new TDO is then modified with an **LSA SetInformationTrustedDomain** call to make sure encryption types supported by the trust are set properly:
 - i. The **RC4_HMAC_MD5** encryption type is enabled, even if there are no RC4 keys in use, due to how Active Directory is designed.
 - ii. **AES128_CTS_HMAC_SHA1_96** and **AES256_CTS_HMAC_SHA1_96** encryption types are enabled.
6. For a forest trust, verify that in-forest domains can be reached transitively with an **LSA SetInformationTrustedDomain** call.
 7. Add trust topology information about the other forest (IdM in the case of communicating with AD, AD in the case of communicating with IdM) using an **LSA RSetForestTrustInformation** call.

NOTE

This step might cause a conflict for one of three reasons:

1. A SID namespace conflict, reported as an **LSA_SID_DISABLED_CONFLICT** error. This conflict cannot be resolved.
2. A NetBIOS namespace conflict, reported as an **LSA_NB_DISABLED_CONFLICT** error. This conflict cannot be resolved.
3. A DNS namespace conflict with a top level name (TLN), reported as an **LSA_TLN_DISABLED_CONFLICT** error. The IdM server can automatically resolve a TLN conflict if it is caused by another forest.

To resolve a TLN conflict, the IdM server performs the following steps:

1. Retrieve forest trust information for the conflicting forest.
2. Add an exclusion entry for the IdM DNS namespace to the AD forest.
3. Set forest trust information for the forest we conflict on.
4. Re-try establishing the trust to the original forest.

The IdM server can only resolve these conflicts if you authenticated the **ipa trust-add** command with the privileges of an AD administrator that can change forest trusts. If you do not have access to those privileges, the administrator of the original forest must manually perform the steps above in the **Active Directory Domains and Trusts** section of the Windows UI.

8. If it does not exist, create the ID range for the trusted domain.
9. For a forest trust, query Active Directory domain controllers from the forest root for details about the forest topology. The IdM server uses this information to create additional ID ranges for any additional domains from the trusted forest.

Additional resources

- [Trust controllers and trust agents](#)
- [Overview Documents](#) (Microsoft)
- [Technical Documents](#) (Microsoft)
- [Privileged Accounts and Groups in Active Directory](#) (Microsoft)

34.9.2. Checklist of prerequisites for establishing an AD trust

You can use the following checklist to review the prerequisites for creating a trust with an AD domain.

Table 34.4. Table

Component	Configuration	Additional details
Product versions	Your Active Directory domain is using a supported version of Windows Server.	Supported versions of Windows Server
AD Administrator privileges	The Active Directory administration account must be a member of one of the following groups: <ul style="list-style-type: none"> • Enterprise Admin (EA) group in the AD forest • Domain Admins (DA) group in the forest root domain for your AD forest 	
Networking	IPv6 support is enabled in the Linux kernel for all IdM servers.	IPv6 requirements in IdM
Date and time	Verify the date and time settings on both servers match.	Time service requirements for IdM

Component	Configuration	Additional details
Encryption types	<p>The following AD accounts have AES encryption keys:</p> <ul style="list-style-type: none"> ● AD Administrator ● AD user accounts ● AD services <p>If you have recently enabled AES encryption in AD, generate new AES keys with the following steps:</p> <ol style="list-style-type: none"> 1. Re-establish trust relationships between any AD domains in your forest. 2. Change the passwords for the AD Administrator, user accounts, and services. 	<ul style="list-style-type: none"> ● Support for encryption types in IdM ● Enabling the AES encryption type in Active Directory using a GPO
Firewall	<p>You have opened all necessary ports on IdM servers and AD Domain Controllers for bidirectional communication.</p>	<p>Ports required for communication between IdM and AD</p>
DNS	<ul style="list-style-type: none"> ● IdM and AD each have unique primary DNS domains. ● IdM and AD DNS domains do not overlap. ● Proper DNS service (SRV) records for LDAP and Kerberos services. ● You can resolve DNS records from all DNS domains in the trust. ● Kerberos realm names are the upper-case versions of primary DNS domain names. For example, DNS domain example.com has a corresponding Kerberos realm EXAMPLE.COM 	<p>Configuring DNS and realm settings for a trust</p>

Component	Configuration	Additional details
Topology	Ensure you are attempting to establish a trust with an IdM server you have configured as a trust controller.	Trust controllers and trust agents

34.9.3. Gathering debug logs of an attempt to establish an AD trust

If you are experiencing issues with establishing a trust between an IdM environment and AD domain, use the following steps to enable detailed error logging so you can gather logs of an attempt to establish a trust. You can review these logs to help with your troubleshooting efforts, or you can provide them in a Red Hat Technical Support case.

Prerequisites

- You need root permissions to restart IdM services.

Procedure

- To enable debugging for the IdM server, create the file `/etc/ipa/server.conf` with the following contents.

```
[global]
debug=True
```

- Restart the **httpd** service to load the debugging configuration.

```
[root@trust_controller ~]# systemctl restart httpd
```

- Stop the **smb** and **winbind** services.

```
[root@trust_controller ~]# systemctl stop smb winbind
```

- Set the debugging log level for the **smb** and **winbind** services.

```
[root@trust_controller ~]# net conf setparm global 'log level' 100
```

- To enable debug logging for Samba client code used by the IdM framework, edit the `/usr/share/ipa/smb.conf.empty` configuration file to have the following contents.

```
[global]
log level = 100
```

- Remove previous Samba logs.

```
[root@trust_controller ~]# rm /var/log/samba/log.*
```

- Start the **smb** and **winbind** services.

```
[root@trust_controller ~]# systemctl start smb winbind
```

8. Print a timestamp as you attempt to establish a trust with verbose mode enabled.

```
[root@trust_controller ~]# date; ipa -vvv trust-add --type=ad ad.example.com
```

9. Review the following error log files for information about the failed request:

- a. `/var/log/httpd/error_log`

- b. `/var/log/samba/log.*`

10. Disable debugging.

```
[root@trust_controller ~]# mv /etc/ipa/server.conf /etc/ipa/server.conf.backup
[root@trust_controller ~]# systemctl restart httpd
[root@trust_controller ~]# systemctl stop smb winbind
[root@trust_controller ~]# net conf setparm global 'log level' 0
[root@trust_controller ~]# mv /usr/share/ipa/smb.conf.empty
/usr/share/ipa/smb.conf.empty.backup
[root@trust_controller ~]# systemctl start smb winbind
```

11. (Optional) If you are unable to determine the cause of the authentication issue:

- a. Collect and archive the log files you recently generated.

```
[root@trust_controller ~]# tar -cvf debugging-trust.tar /var/log/httpd/error_log
/var/log/samba/log.*
```

- b. Open a Red Hat Technical Support case and provide the timestamp and debug logs from the attempt.

Additional resources

- [IPA - AD Trust Troubleshooting](#)

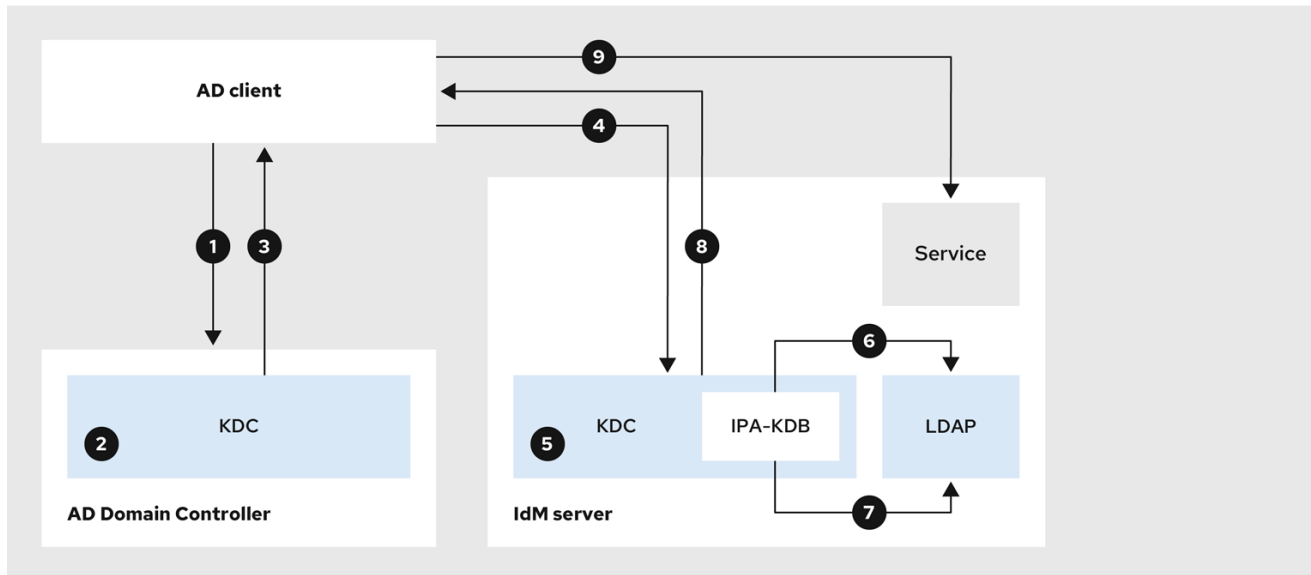
34.10. TROUBLESHOOTING CLIENT ACCESS TO SERVICES IN THE OTHER FOREST

After configuring a trust between your Identity Management (IdM) and Active Directory (AD) environments, you might experience issues where a client in one domain is not able to access a service in the other domain. Use the following diagrams to troubleshoot the issue.

34.10.1. Flow of information when a host in the AD forest root domain requests services from an IdM server

The following diagram explains the flow of information when an Active Directory (AD) client requests a service in the Identity Management (IdM) domain.

If you have trouble accessing IdM services from AD clients, you can use this information to narrow your troubleshooting efforts and identify the source of the issue.



231_RHEL_0422

1. The AD client contacts the AD Kerberos Distribution Center (KDC) to perform a TGS Request for the service in the IdM domain.
2. The AD KDC recognizes that the service belongs to the trusted IdM domain.
3. The AD KDC sends the client a cross-realm ticket-granting ticket (TGT), along with a referral to the trusted IdM KDC.
4. The AD client uses the cross-realm TGT to request a ticket to the IdM KDC.
5. The IdM KDC validates the Privileged Attribute Certificate (MS-PAC) that is transmitted with the cross-realm TGT.
6. The IPA-KDB plugin might check the LDAP directory to see if foreign principals are allowed to get tickets for the requested service.
7. The IPA-KDB plugin decodes the MS-PAC, verifies, and filters the data. It performs lookups in the LDAP server to check if it needs to augment the MS-PAC with additional information, such as local groups.
8. The IPA-KDB plugin then encodes the PAC, signs it, attaches it to the service ticket, and sends it to the AD client.
9. The AD client can now contact the IdM service using the service ticket issued by IdM KDC.

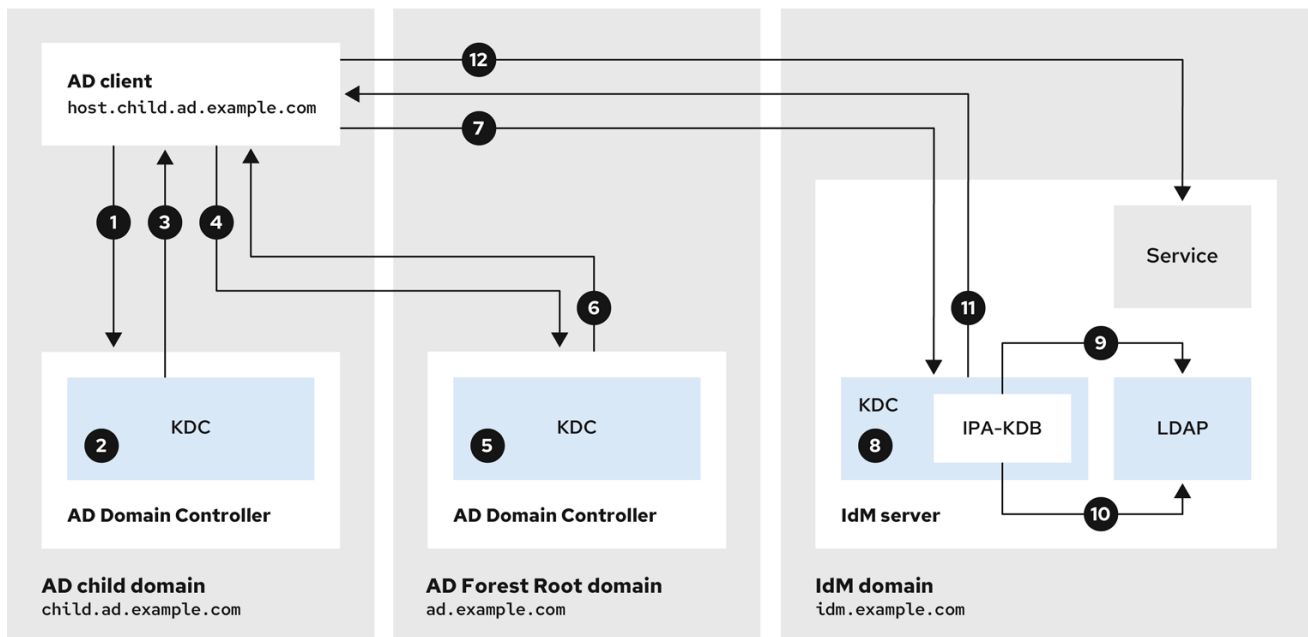
Additional resources

- [Flow of information when a host in an AD child domain requests services from an IdM server](#)

34.10.2. Flow of information when a host in an AD child domain requests services from an IdM server

The following diagram explains the flow of information when an Active Directory (AD) host in a child domain requests a service in the Identity Management (IdM) domain. In this scenario, the AD client contacts the Kerberos Distribution Center (KDC) in the child domain, then contacts the KDC in the AD forest root, and finally contacts the IdM KDC to request access to the IdM service.

If you have trouble accessing IdM services from AD clients, and your AD client belongs to a domain that is a child domain of an AD forest root, you can use this information to narrow your troubleshooting efforts and identify the source of the issue.



231_RHEL_0422

1. The AD client contacts the AD Kerberos Distribution Center (KDC) in its own domain to perform a TGS Request for the service in the IdM domain.
2. The AD KDC in **child.ad.example.com**, the child domain, recognizes that the service belongs to the trusted IdM domain.
3. The AD KDC in the child domain sends the client a referral ticket for the AD forest root domain **ad.example.com**.
4. The AD client contacts the KDC in the AD forest root domain for the service in the IdM domain.
5. The KDC in the forest root domain recognizes that the service belongs to the trusted IdM domain.
6. The AD KDC sends the client a cross-realm ticket-granting ticket (TGT), along with a referral to the trusted IdM KDC.
7. The AD client uses the cross-realm TGT to request a ticket to the IdM KDC.
8. The IdM KDC validates the Privileged Attribute Certificate (MS-PAC) that is transmitted with the cross-realm TGT.
9. The IPA-KDB plugin might check the LDAP directory to see if foreign principals are allowed to get tickets for the requested service.
10. The IPA-KDB plugin decodes the MS-PAC, verifies, and filters the data. It performs lookups in the LDAP server to check if it needs to augment the MS-PAC with additional information, such as local groups.
11. The IPA-KDB plugin then encodes the PAC, signs it, attaches it to the service ticket, and sends it to the AD client.

- The AD client can now contact the IdM service using the service ticket issued by IdM KDC.

Additional resources

- [Flow of information when a host in the AD forest root domain requests services from an IdM server](#)

34.10.3. Flow of information when an IdM client requests services from an AD server

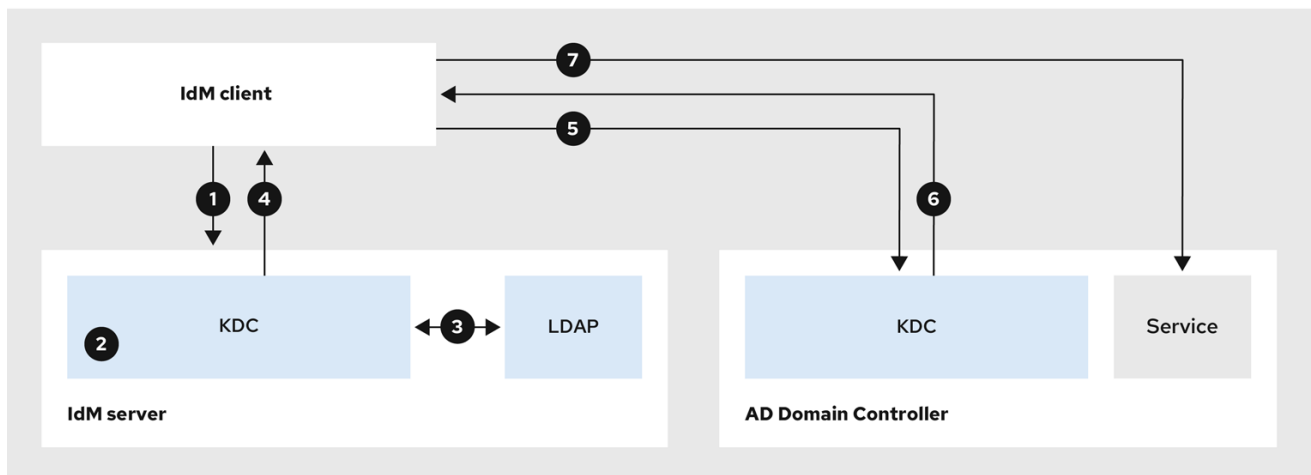
The following diagram explains the flow of information when an Identity Management (IdM) client requests a service in the Active Directory (AD) domain when you have configured a two-way trust between IdM and AD.

If you have trouble accessing AD services from IdM clients, you can use this information to narrow your troubleshooting efforts and identify the source of the issue.



NOTE

By default, IdM establishes a one-way trust to AD, which means it is not possible to issue cross-realm ticket-granting ticket (TGT) for resources in an AD forest. To be able to request tickets to services from trusted AD domains, configure a two-way trust.



231_RHEL_0422

- The IdM client requests a ticket-granting ticket (TGT) from the IdM Kerberos Distribution Center (KDC) for the AD service it wants to contact.
- The IdM KDC recognizes that the service belongs to the AD realm, verifies that the realm is known and trusted, and that the client is allowed to request services from that realm.
- Using information from the IdM Directory Server about the user principal, the IdM KDC creates a cross-realm TGT with a Privileged Attribute Certificate (MS-PAC) record about the user principal.
- The IdM KDC sends back a cross-realm TGT to the IdM client.
- The IdM client contacts the AD KDC to request a ticket for the AD service, presenting the cross-realm TGT that contains the MS-PAC provided by the IdM KDC.
- The AD server validates and filters the PAC, and returns a ticket for the AD service.

- The IPA client can now contact the AD service.

Additional resources

- [One-way trusts and two-way trusts](#)

34.11. REMOVING THE TRUST USING THE COMMAND LINE

Follow this procedure to remove the Identity Management (IdM)/Active Directory (AD) trust on the IdM side using the command line interface.

Prerequisites

- You have obtained a Kerberos ticket as an IdM administrator. For details, see [Logging in to IdM in the Web UI: Using a Kerberos ticket](#).

Procedure

- Use the **ipa trust-del** command to remove the trust configuration from IdM.

```
[root@server ~]# ipa trust-del ad_domain_name
-----
Deleted trust "ad_domain_name"
-----
```

- Remove the trust object from your Active Directory configuration.

NOTE

Removing the trust configuration does not automatically remove the ID range IdM has created for AD users. This way, if you add the trust again, the existing ID range is re-used. Also, if AD users have created files on an IdM client, their POSIX IDs are preserved in the file metadata.

To remove all information related to an AD trust, remove the AD user ID range after removing the trust configuration and trust object:

```
# ipa idrange-del AD.EXAMPLE.COM_id_range
# systemctl restart sssd
```

Verification steps

- Use the **ipa trust-show** command to confirm that the trust has been removed.

```
[root@server ~]# ipa trust-show ad.example.com
ipa: ERROR: ad.example.com: trust not found
```

Additional resources

- [Removing an ID range after removing a trust to AD](#)

34.12. REMOVING THE TRUST USING THE IDM WEB UI

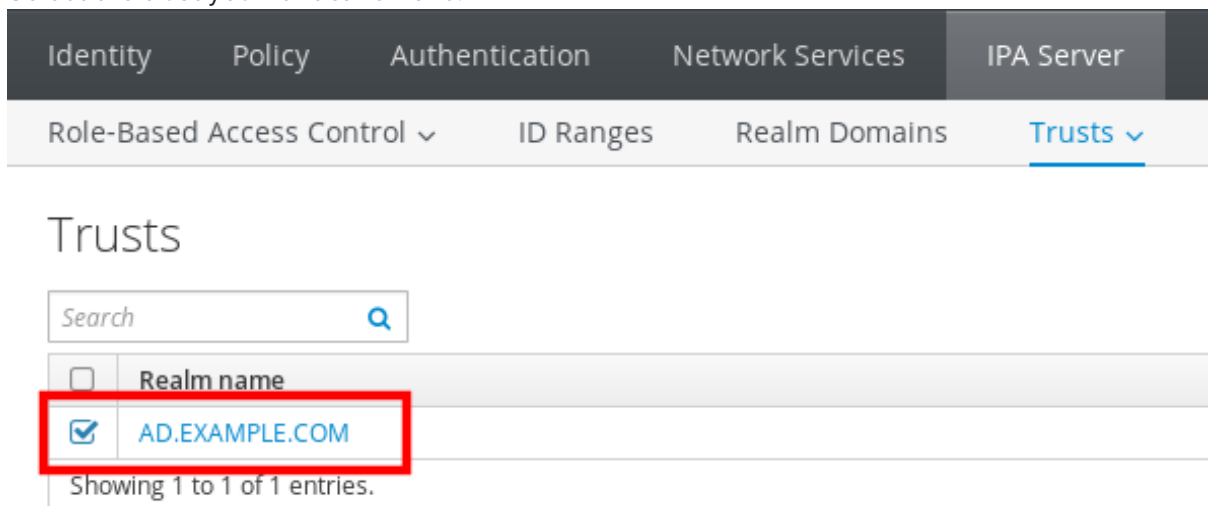
Follow this procedure to remove the Identity Management (IdM)/Active Directory (AD) trust using the IdM Web UI.

Prerequisites

- You have obtained a Kerberos ticket. For details, see [Logging in to IdM in the Web UI: Using a Kerberos ticket](#).

Procedure

1. Log in to the IdM Web UI with administrator privileges. For details, see [Accessing the IdM Web UI in a web browser](#).
2. In the IdM Web UI, click the **IPA Server** tab.
3. In the **IPA Server** tab, click the **Trusts** tab.
4. Select the trust you want to remove.

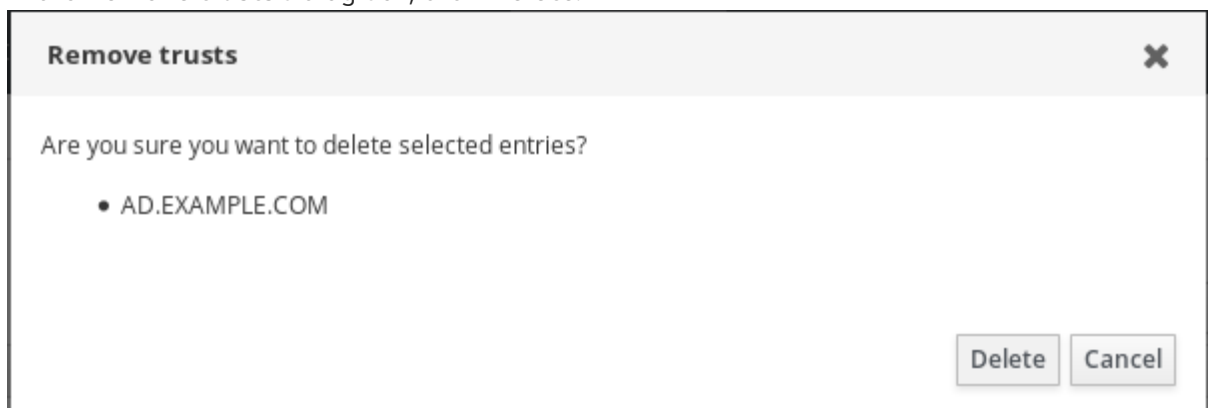


The screenshot shows the IdM Web UI navigation menu with the following tabs: Identity, Policy, Authentication, Network Services, and IPA Server. Under the IPA Server tab, the following sub-tabs are visible: Role-Based Access Control, ID Ranges, Realm Domains, and Trusts. The Trusts tab is selected. Below the navigation menu, the page title is "Trusts". There is a search input field with the placeholder text "Search" and a magnifying glass icon. Below the search field is a table with the following structure:

<input type="checkbox"/>	Realm name
<input checked="" type="checkbox"/>	AD.EXAMPLE.COM

Below the table, it says "Showing 1 to 1 of 1 entries." The checkbox for the selected trust is highlighted with a red box.

5. Click the **Delete** button.
6. In the **Remove trusts** dialog box, click **Delete**.



The screenshot shows a dialog box titled "Remove trusts" with a close button (X) in the top right corner. The dialog contains the text "Are you sure you want to delete selected entries?" and a list of selected entries: "• AD.EXAMPLE.COM". At the bottom right of the dialog, there are two buttons: "Delete" and "Cancel".

7. Remove the trust object from your Active Directory configuration.



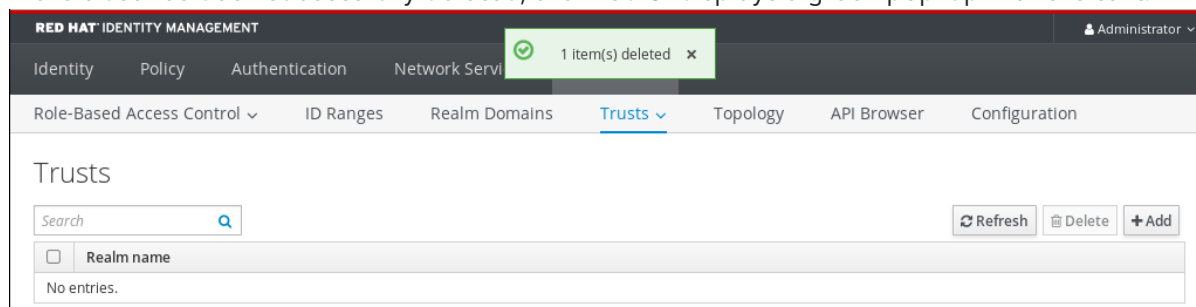
NOTE

Removing the trust configuration does not automatically remove the ID range IdM has created for AD users. This way, if you add the trust again, the existing ID range is re-used. Also, if AD users have created files on an IdM client, their POSIX IDs are preserved in the file metadata.

To remove all information related to an AD trust, remove the AD user ID range in the **ID Ranges** tab after removing the trust configuration and trust object.

Verification steps

- If the trust has been successfully deleted, the Web UI displays a green pop-up with the text:



Additional resources

- [Removing an ID range after removing a trust to AD](#)

34.13. REMOVING THE TRUST USING ANSIBLE

Follow this procedure to remove the Identity Management (IdM)/Active Directory (AD) trust on the IdM side by using an Ansible playbook.

Prerequisites

- You have obtained a Kerberos ticket as an IdM administrator. For details, see [Logging in to IdM in the Web UI: Using a Kerberos ticket](#).
- You have configured your Ansible control node to meet the following requirements:
 - You are using Ansible version 2.14 or later.
 - You have installed the [ansible-freeipa](#) package on the Ansible controller.
 - The example assumes that in the `~/MyPlaybooks/` directory, you have created an [Ansible inventory file](#) with the fully-qualified domain name (FQDN) of the IdM server.
 - The example assumes that the `secret.yml` Ansible vault stores your `ipadmin_password`.
- The target node, that is the node on which the `ansible-freeipa` module is executed, is part of the IdM domain as an IdM client, server or replica.

Procedure

1. Navigate to your `~/MyPlaybooks/` directory:

```
$ cd ~/MyPlaybooks/
```

2. Create an **del-trust.yml** playbook with the following content:

```
---
- name: Playbook to delete trust
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: ensure the trust is absent
    ipatrust:
      ipaadmin_password: "{{ ipaadmin_password }}"
      realm: ad.example.com
      state: absent
```

In the example, **realm** defines the AD realm name string.

3. Save the file.
4. Run the Ansible playbook. Specify the playbook file, the file storing the password protecting the **secret.yml** file, and the inventory file:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory del-trust.yml
```

NOTE

Removing the trust configuration does not automatically remove the ID range IdM has created for AD users. This way, if you add the trust again, the existing ID range is re-used. Also, if AD users have created files on an IdM client, their POSIX IDs are preserved in the file metadata.

To remove all information related to an AD trust, remove the AD user ID range after removing the trust configuration and trust object:

```
# ipa idrange-del AD.EXAMPLE.COM_id_range
# systemctl restart sssd
```

Verification steps

- Use the **ipa trust-show** command to confirm that the trust has been removed.

```
[root@server ~]# ipa trust-show ad.example.com
ipa: ERROR: ad.example.com: trust not found
```

Additional resources

- [/usr/share/doc/ansible-freeipa/README-trust.md](#)
- [/usr/share/doc/ansible-freeipa/playbooks/trust](#)
- [Removing an ID range after removing a trust to AD](#)

34.14. REMOVING AN ID RANGE AFTER REMOVING A TRUST TO AD

If you have removed a trust between your IdM and Active Directory (AD) environments, you might want to remove the ID range associated with it.



WARNING

IDs allocated to ID ranges associated with trusted domains might still be used for ownership of files and directories on systems enrolled into IdM.

If you remove the ID range that corresponds to an AD trust that you have removed, you will not be able to resolve the ownership of any files and directories owned by AD users.

Prerequisites

- You have removed a trust to an AD environment.

Procedure

1. Display all the ID ranges that are currently in use:

```
[root@server ~]# ipa idrange-find
```

2. Identify the name of the ID range associated with the trust you have removed. The first part of the name of the ID range is the name of the trust, for example **AD.EXAMPLE.COM_id_range**.
3. Remove the range:

```
[root@server ~]# ipa idrange-del AD.EXAMPLE.COM_id_range
```

4. Restart the SSSD service to remove references to the ID range you have removed.

```
[root@server ~]# systemctl restart sssd
```

Additional resources

- See [Removing the trust using the command line](#).
- See [Removing the trust using the IdM Web UI](#).