

Pelican User Guide

Version no : 1.0.9.2_84

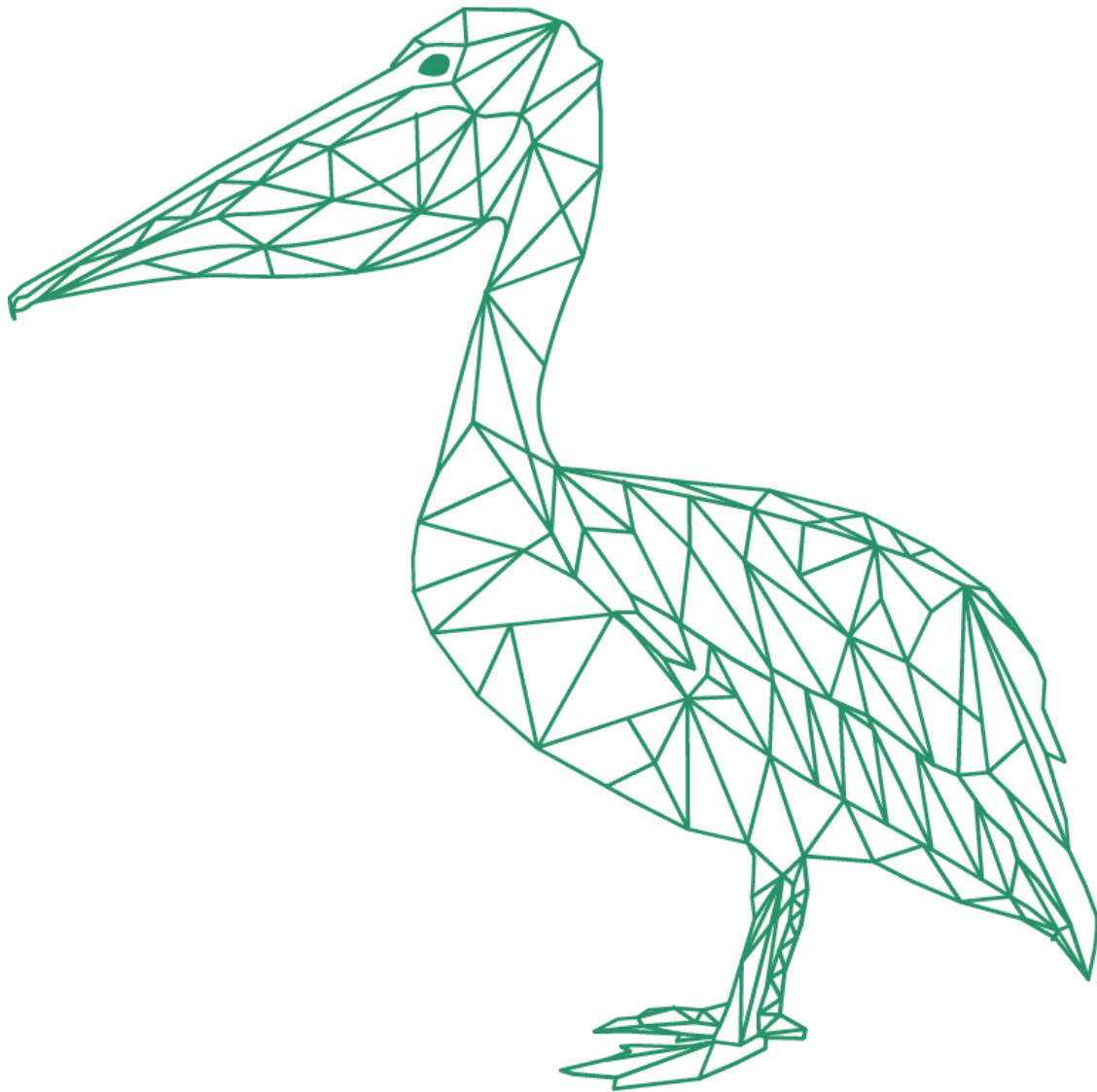


Table of Contents

1. Preface	3
2. Pelican Overview	4
3. Data Store Support	5
4. Architecture	6
5. Dashboard	7
6. User Management	8
6.1. Login	8
6.2. Configure SMTP server	9
6.3. SMTP User creation	10
6.4. Authentication using LDAP/AD	13
6.5. LDAP/AD User Management	15
6.6. Admin Password	17
6.7. Forget Password	17
7. Datastore Configuration	18
7.1. Steps for Adding Database configuration	18
7.1.1. Steps for Teradata Configuration	21
7.1.2. Steps for Netezza Configuration	23
7.1.3. Steps for Oracle Configuration	25
7.1.4. Steps for BigQuery Configuration	26
7.1.5. Steps for HIVE Configuration	27
7.1.6. Steps for DB2 Configuration	29
7.1.7. Steps for Synapse Configuration	31
7.1.8. Steps for Snowflake Configuration	32
7.1.9. Steps for Deltalake Configuration	34
7.1.10. Steps for Greenplum Configuration	36
7.1.11. Steps for Redshift Configuration	38
7.1.12. Steps for MSSQL Server Configuration	39
7.1.13. Steps for Vertica Configuration	42
7.2. Editing an existing Datastore	43
7.3. Deleting existing datastore	44
8. Validation Configuration	44
8.1. Table Mapping	45

8.1.1. Edit the existing configuration	49
8.1.2. Delete Mapping	52
8.1.3. Override unmapped columns	52
8.1.4. Surrogate Key	53
8.1.5. Add Tags	57
9. Custom Source and Target	58
10. Scheduler Configuration	60
10.1. Modes in Scheduler Configuration	60
10.1.1. LITMUS mode	60
10.1.2. Full mode	60
10.2. Steps to configure the scheduler:	61
10.3 Scheduler Report	64
11. Validation Result	65
11.1. Results from Report page	65
11.2. Results from Scheduler page	72
12. License - Buy Your Own license (BYOL)	73
12.1. Adding /updating license	74
13. Lineage	77
13.1. Add Lineage	77
13.2. How to View Lineage?	81
14. API	83
14.1 Export/Import API	83
14.2. Pelican V3 API	95
14.3. API to get Scheduler status	96
15. Automatic Backup	98
16. Frequently Asked Questions	99
17. Glossary	102

1. Preface

The Pelican user guide is written for developers and software engineers who are responsible for comparing and validating the data across various data stores. The Pelican user guide assumes that you have an understanding of your operating systems, relational database concepts, database engines, flat files, and how the mainframe system works in your environment. This guide also assumes that you are familiar with the interface requirements for your supporting applications.

2. Pelican Overview

Pelican is an innovative enterprise application that ensures businesses that the data is migrated from reference to destination data store accurately and reliably by performing validation over the migrated data. Pelican helps the user to compare and validate the data across different data stores quickly.

Pelican helps the user to validate large data sets without any data transfer from source to target. Pelican allows the user to copy a large dataset (historical and real time data) to a destination data store for validation. It uses an innovative approach to compare the data without moving it across data stores. It also displays the data records having differences.

The user can automate the comparison using schedulers and API. Pelican uses the phonetic matching and approximate matching algorithms to search the best possible combination of tables from the target datastore. It also facilitates configuring and executing various Schedulers as per the business requirement. These Schedulers are created to compare and validate table dataset periodically.

You can configure an email notification for schedulers that sends an email automatically to various users when the scheduler executes. Pelican provides lineage support, which enables the user to track the movement of data across various nodes. This information is retrieved from the job history server of various data stores configured in the Pelican. Once the validation is done, the application generates statistics with the following information:

- Count of total rows at source
- Count of total rows at destination
- Count of mismatch rows at destination
- Count of extra rows at destination
- Count of missing rows at destination
- Total mismatch row count
- Validation Status
- Sample of mismatch data

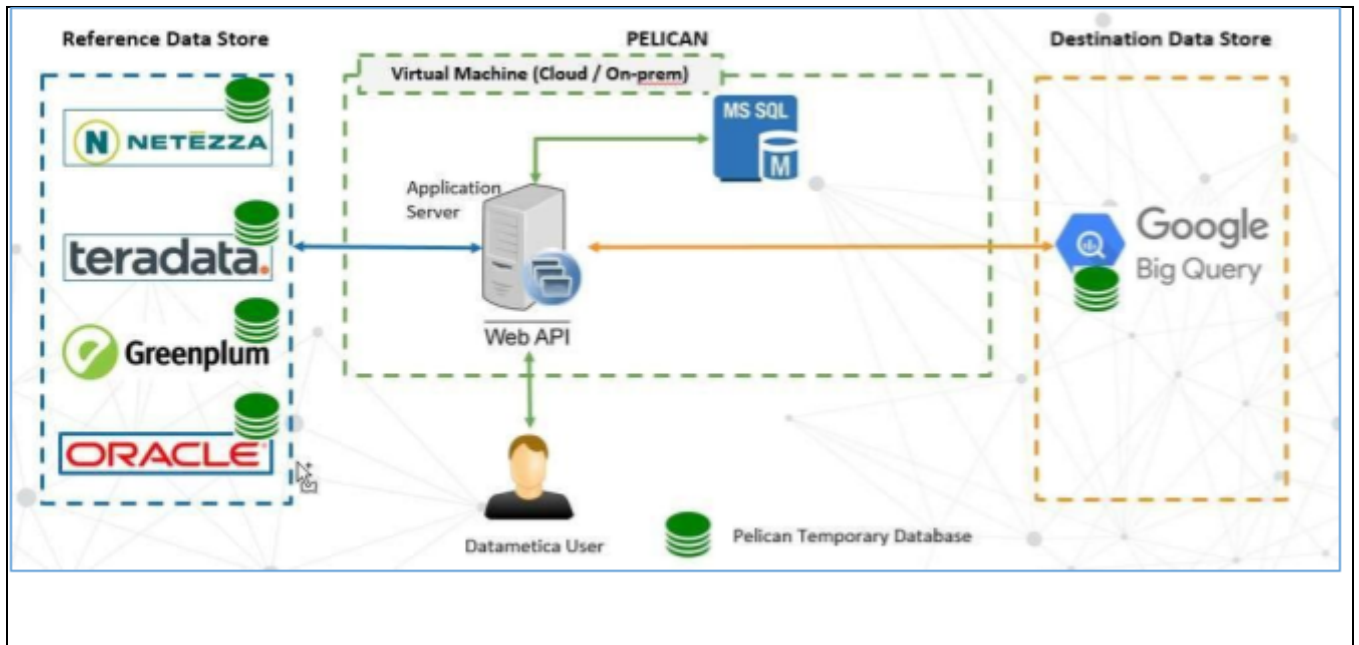
3. Data Store Support

Pelican supports comparison between the following data stores using an innovative approach:

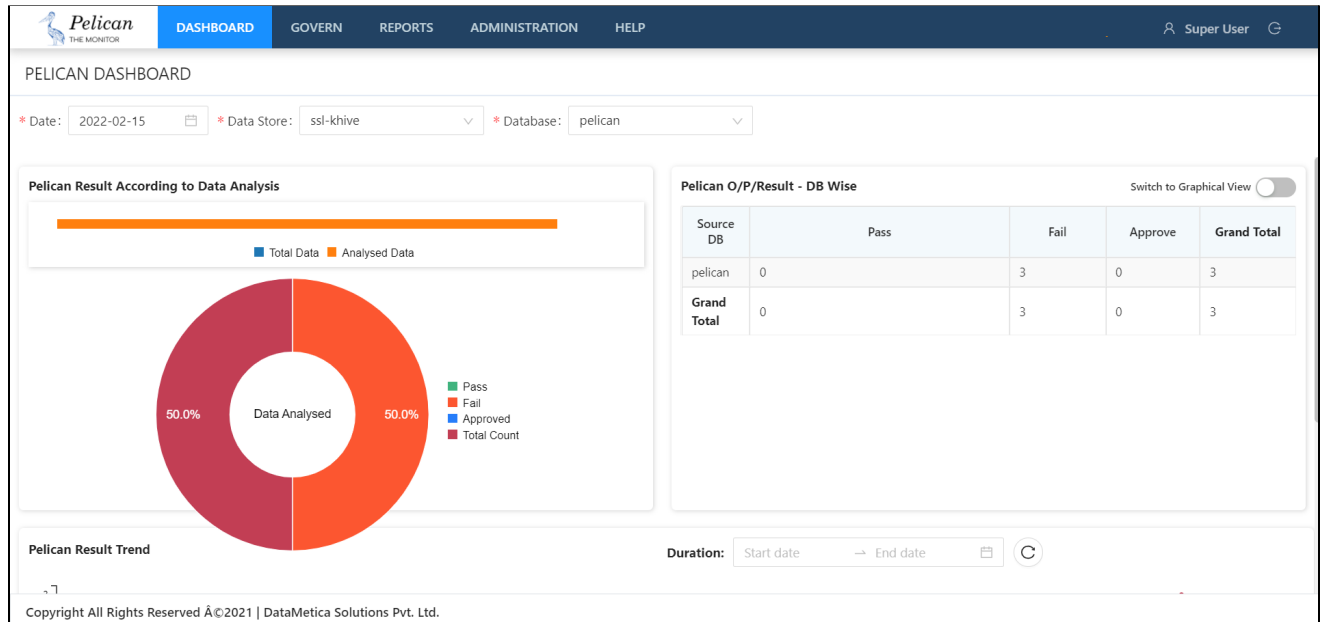
Sr. No	Source Data store	Destination Data store
1	Teradata	Big Query
2	Netezza	Big Query
3	Hive	Big Query
4	Oracle	Big Query
5	BigQuery	Big Query
6	DB2	Big Query
7	Oracle	Hive
8	Teradata	Hive
9	Netezza	Hive
10	Hive	Hive
11	Big Query	Hive
12	Vertica	BQ
13	Teradata	Snowflake
14	Teradata	Synapse
15	Netezza	Snowflake
16	Netezza	Synapse
17	Ms SQL Server	Hive
18	Ms SQL Server	Big Query
19	Greenplum	Redshift
20	Hive	Deltalake
21	Hive	Synapse

4. Architecture

Here is the complete architecture of how Pelican works.



5. Dashboard



Dashboard screen displays result, trends, graph of the scheduled mappings based on datastore, database, tags, and date. When for the first time the dashboard screen is loaded it will display the dashboard screen for the default datastore. User can select datastore and databases to see the corresponding graphs.

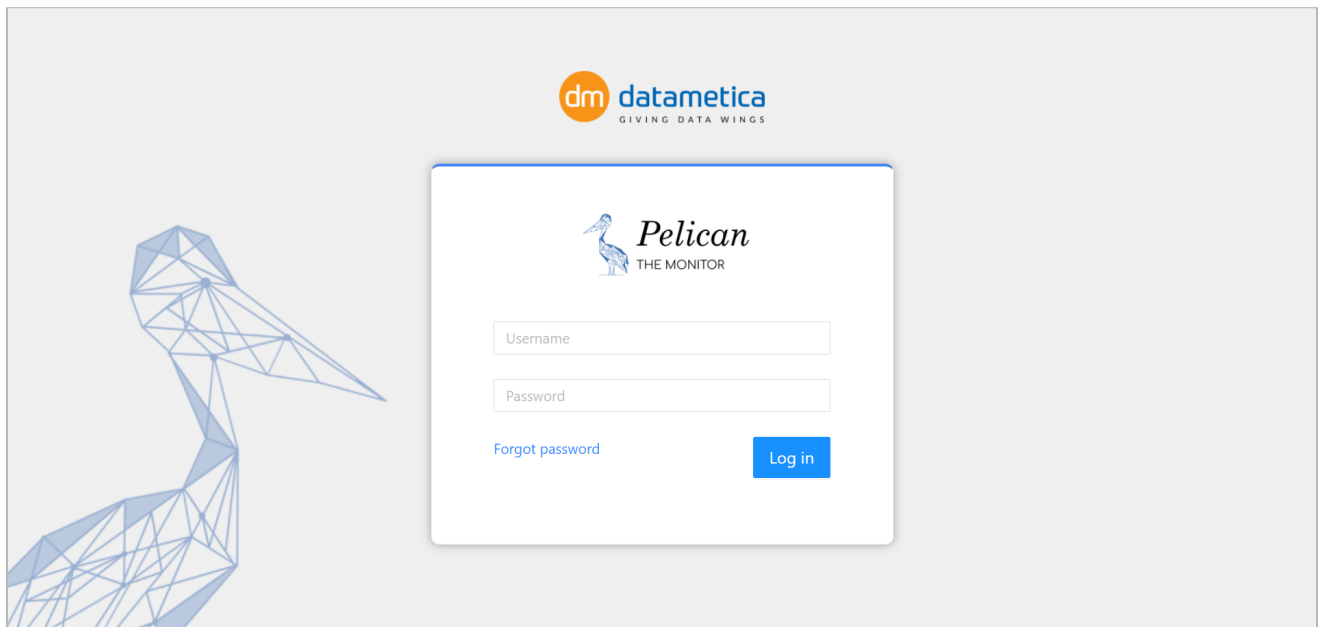
6. User Management

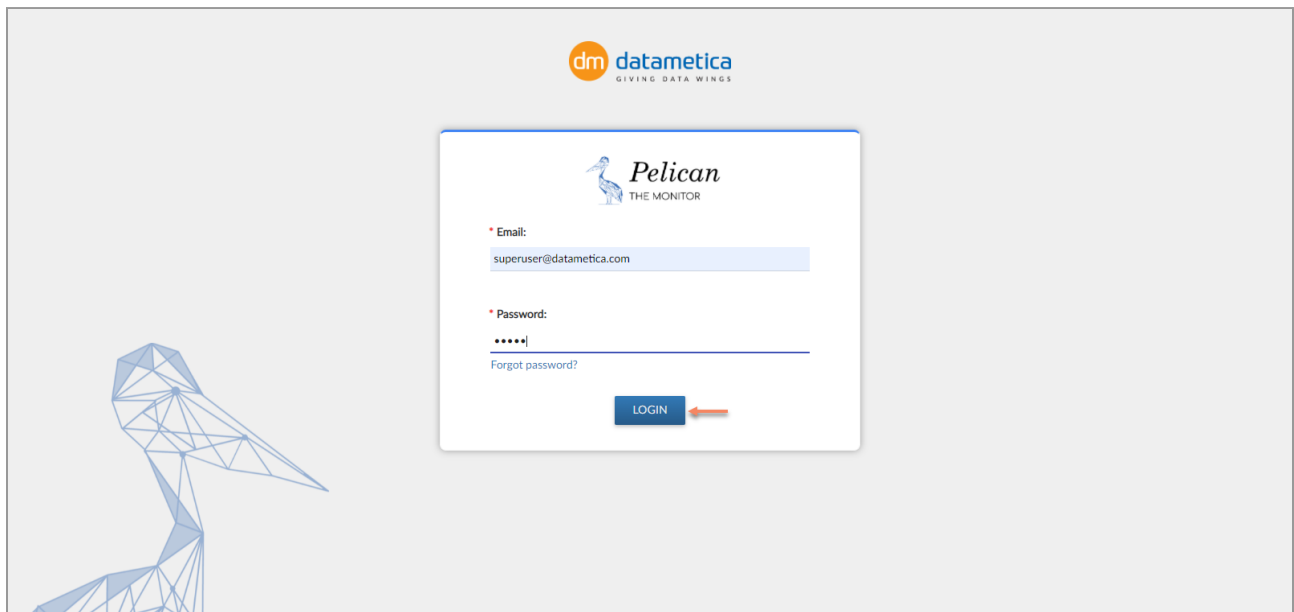
In **User Management** the entire administration of the application is accessed, providing secured authorization to the user. Each user is assigned a unique identity to authenticate the application.

6.1. Login

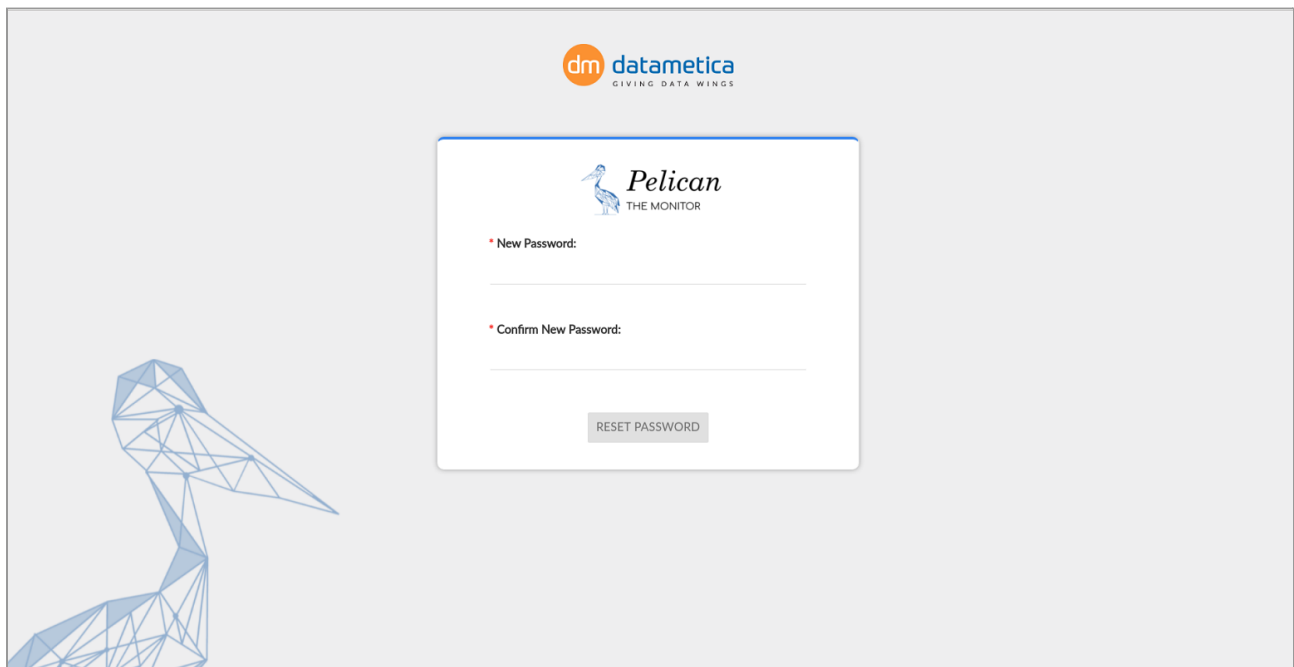
Pelican needs the following details for login:

- User's Email Id
- Password





First time users will be requested to reset the default password.



After completing the steps the user will navigate to the **Validation Result** page.

6.2. Configure SMTP server

The SMTP server configuration functionality allows the Administrator to configure the SMTP server which will be used for sending passwords to the newly added user.

Configure SMTP server and Sender Email Address ;

To configure the SMTP Server and senders email address:

1. Go to the **Administration** → **Configure** → **Email Configuration**.
2. Fill the below details in the Email Configuration window.
 - Enter **Host Name** (http://smtp.googlemail.com/)
 - Enter **Port Name** (587)
 - Enter **User Name** (firstname@datametica.com)
 - Enter **Password** in the provided field.

Email Configuration

Fill email configuration details here

* Host:

* Port:

* Username:

* Password:

Copyright All Rights Reserved ©2021 | DataMetica Solutions Pvt. Ltd.

3. Click on **Test Connection**.
4. Click on the **Save** button to save the configuration.

6.3. SMTP User creation

The **User** tab enables creating a new user with the role of Validator or Admin. You can edit or delete an existing user as well. This option will be available if LDAP configuration is not set up. (SMTP)

Steps to add a new SMTP user:

1. Go to **Administration** → **User Management** → **User**.

2. Click **Add New User**.

Users							Add New User
First Name	Last Name	Email Address	Role	Reset Password	Edit	Delete	
aaa	bbb	aaa.bbb@datametica.com	ADMINISTRATOR				
abc	yyy	abc.yyy@datametica.com	ADMINISTRATOR				

< 1 >

3. In **Add / Update User Details** Form, enter First Name, Last Name, Email, and select Role from the drop-down list.

The Roles are Super user and Validator.

Add/Update User Details ✕

* First Name:

* Last Name:

* Email:

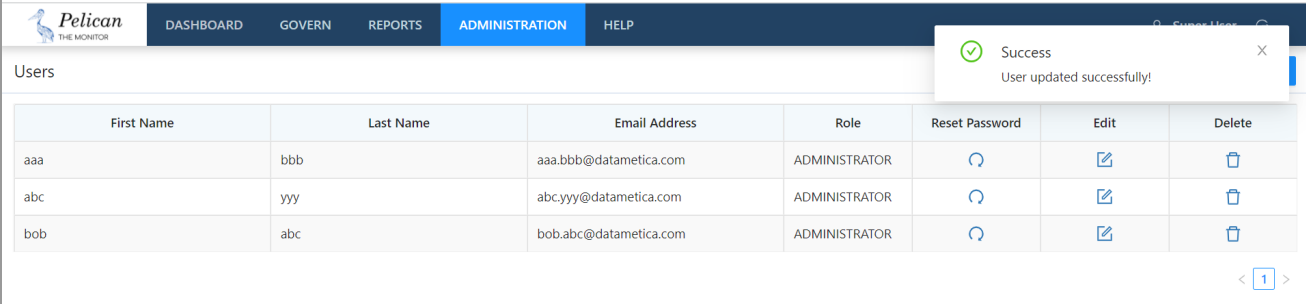
* Role:

Note:

Admin: Users having the role as **admin** have the same rights as Super user. They can create new users having roles as Admin or Validator. They can create data stores for validation and configure email for sending mails.

Validator: The users having the role as **validator** do not have access to create new datastore and can't create a new user, but they can create mappings, run schedulers, and view reports.

4. Click **SAVE**.

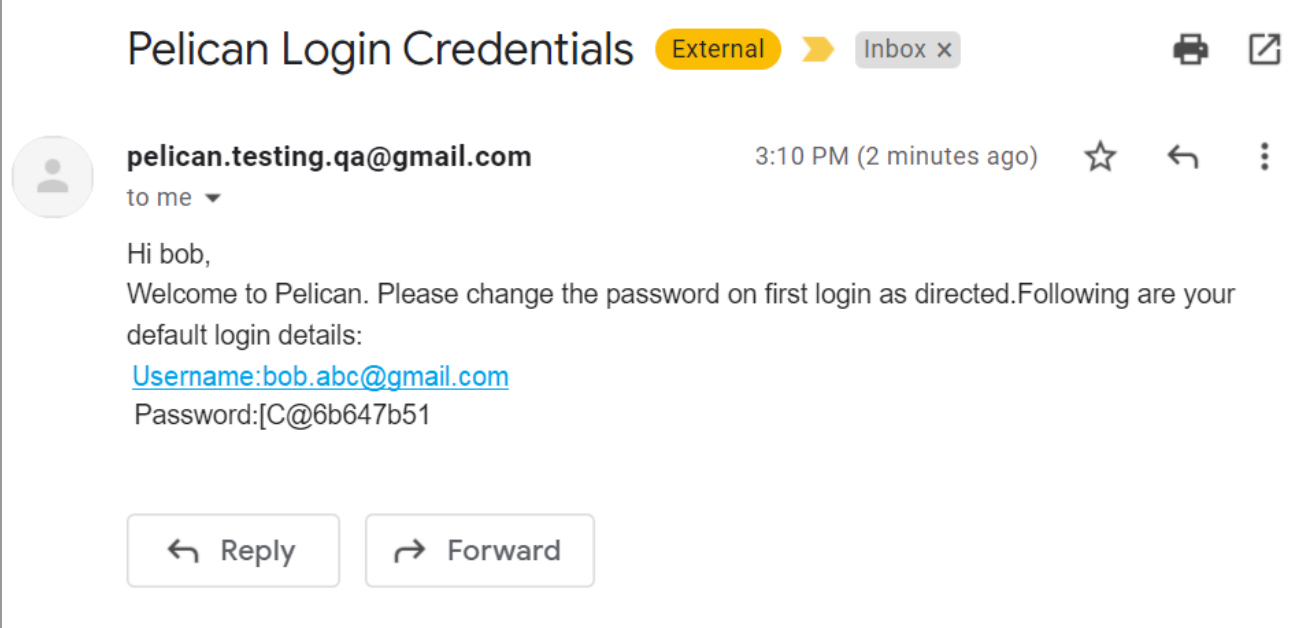


Users

First Name	Last Name	Email Address	Role	Reset Password	Edit	Delete
aaa	bbb	aaa.bbb@datametica.com	ADMINISTRATOR			
abc	yyy	abc.yyy@datametica.com	ADMINISTRATOR			
bob	abc	bob.abc@datametica.com	ADMINISTRATOR			

< 1 >

A system generated Pelical Login Credentials email is generated and sent to the first-time login user which includes username and password. The user should login with those credentials and reset the password.



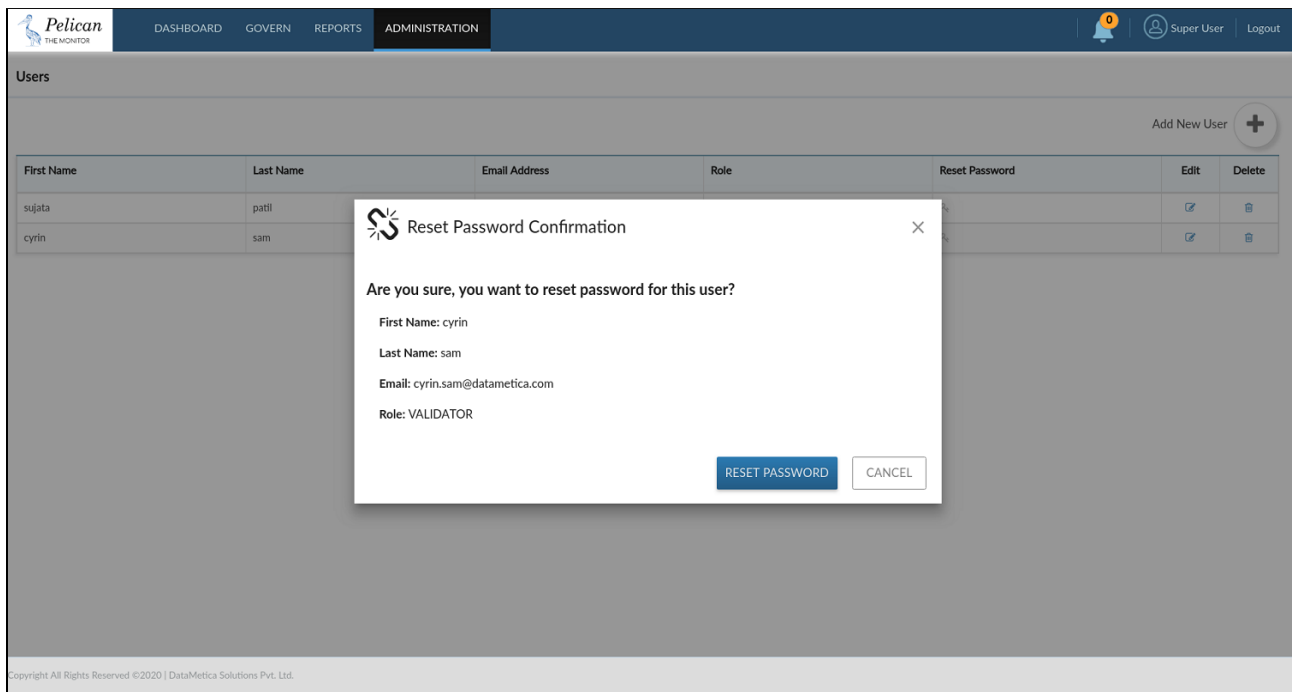
Pelican Login Credentials External Inbox x

pelican.testing.qa@gmail.com 3:10 PM (2 minutes ago)

to me ▾

Hi bob,
 Welcome to Pelican. Please change the password on first login as directed. Following are your default login details:
 Username: bob.abc@gmail.com
 Password: [C@6b647b51]

Reply Forward



The screenshot shows the 'Users' management page in the Datametica application. The page has a dark blue header with navigation tabs: DASHBOARD, GOVERN, REPORTS, and ADMINISTRATION. The ADMINISTRATION tab is active. On the right side of the header, there is a notification bell icon with '0', a user profile icon labeled 'Super User', and a 'Logout' link. Below the header, the page title is 'Users'. There is an 'Add New User' button with a plus sign icon. A table lists users with columns: First Name, Last Name, Email Address, Role, Reset Password, Edit, and Delete. Two users are visible: sujata patil and cyrin sam. A modal dialog box titled 'Reset Password Confirmation' is open in the foreground. The dialog contains the following text: 'Are you sure, you want to reset password for this user?' followed by user details: 'First Name: cyrin', 'Last Name: sam', 'Email: cyrin.sam@datametica.com', and 'Role: VALIDATOR'. At the bottom of the dialog are two buttons: 'RESET PASSWORD' (highlighted in blue) and 'CANCEL'.

First Name	Last Name	Email Address	Role	Reset Password	Edit	Delete
sujata	patil					
cyrin	sam					

Copyright All Rights Reserved ©2020 | DataMetica Solutions Pvt. Ltd.

6.4. Authentication using LDAP/AD

You can add the Authentication details for the LDAP/ Active Directory.

The steps to add authentication details are as follows:

1. Go to **Administration > User Management > Authentication**.
2. Click **Edit**.

Authentication

LDAP / Active Directory Edit

<p>* Host</p> <input style="width: 90%;" type="text"/>	<p>* Port</p> <input style="width: 90%;" type="text"/>
<p>Domain Component</p> <input style="width: 90%;" type="text"/>	<p>Object Class</p> <input style="width: 90%;" type="text"/>
<p>Extra Paramateres Before DC</p> <input style="width: 90%;" type="text"/>	<p>UID or CN</p> <input style="width: 90%;" type="text"/>
<p>Bind DN</p> <input style="width: 90%;" type="text"/>	<p>Bind Password</p> <input style="width: 90%;" type="text"/>

SSL enabled

Enable

3. Enter Host, Port, Domain Component, Object Class, Extra Parameters Before DC, and other fields.
4. Check the checkbox if SSL is enabled.
5. Click **DONE**.

<u>Field Name</u>	<u>Description</u>
Host	Ip or machine name on which Ldap server is running
Port	Port at which connection to the ldap server is to be created
<i>Following fields are used to create the BaseDN, in which user should be searched</i>	
Domain Component	The comma separated values that are put with the attribute "dc"
Object Class	Object class decides the attributes of the entry and is part of the search criteria
Extra Parameters Before DC	Any extra parameters that are present between common name & domain components.
UID or CN	Value of this field will be either uid or cn. This attribute denotes the common name of the user.

Bind DN	This field will hold the qualified value of the user which may be required to connect to ldap
Bind Password	This field will hold the password corresponding to the user mentioned in above field
SSL Enabled	If ldap is secured, check the checkbox. Certificate should also be configured in the pelican jvm

Note:

- Post LDAP/AD successful connection, we will need to restart LDAP so that added users can log in to the application.
- After configuring the **Authentication** page the user has to navigate to the tomcat/bin folder where Pelican is installed. Stop the Pelican application using `./shutdown.sh`. Then navigate to `/webapps/ROOT/WEB-INF/classes/config.properties` file. And add the following properties and corresponding values in it:
 - 1) `authenticationByDistinguishedName = false`
 - 2) `enterpriseldAttribute = sAMAccountName`
- Once you have added the credential super user can only create ldap user and can't create non-ldap user having role as admin or validator.

6.5. LDAP/AD User Management

LDAP/AD user authentication is the process of validating a username and password combination with a directory server. Only the superuser and admin has the right to create LDAP/AD users. In the **User Management** module you can perform the following functions:

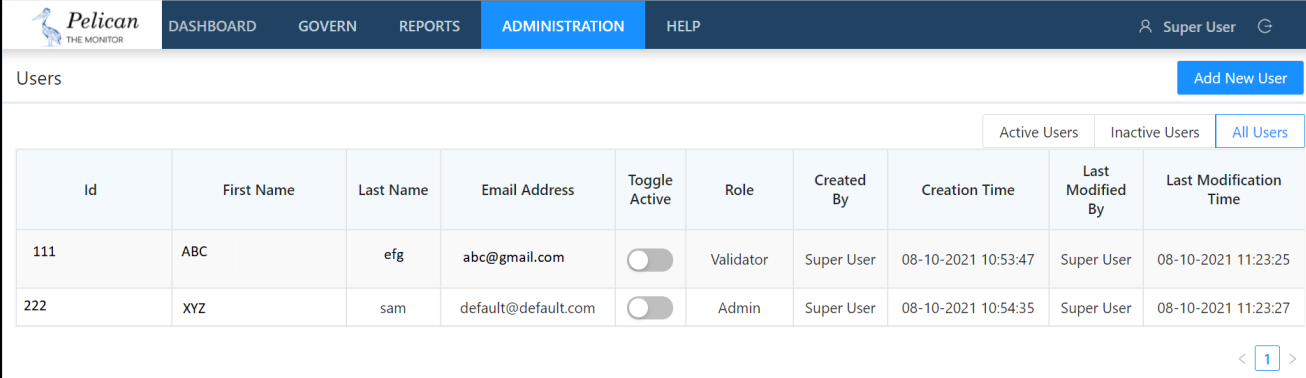
- Setting LDAP/AD credential
- Adding user
- Accessing control
- Assigning roles to the user
- Administrating Pelican

- Adding roles
- Granting permissions
- Assigning user groups

In **User Management** you can add a new LDAP user and view the details on the **User** page.

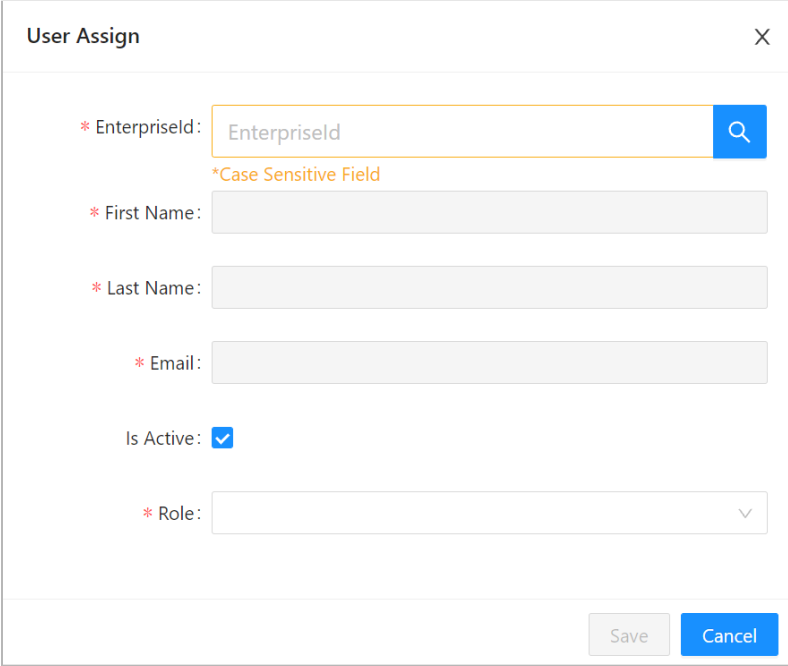
Steps

1. Go to **Administration -> User Management -> user**.
2. Click the **Add New User**.



Id	First Name	Last Name	Email Address	Toggle Active	Role	Created By	Creation Time	Last Modified By	Last Modification Time
111	ABC	efg	abc@gmail.com	<input type="checkbox"/>	Validator	Super User	08-10-2021 10:53:47	Super User	08-10-2021 11:23:25
222	XYZ	sam	default@default.com	<input type="checkbox"/>	Admin	Super User	08-10-2021 10:54:35	Super User	08-10-2021 11:23:27

3. On the **User Assign** form enter EnterpriseID and all other details are visible automatically in the fields.



User Assign [X]

* Enterpriseid: [Search] *Case Sensitive Field

* First Name:

* Last Name:

* Email:

Is Active:

* Role:

[Save] [Cancel]

4. Select the **Is Active** checkbox and click **SUBMIT**.

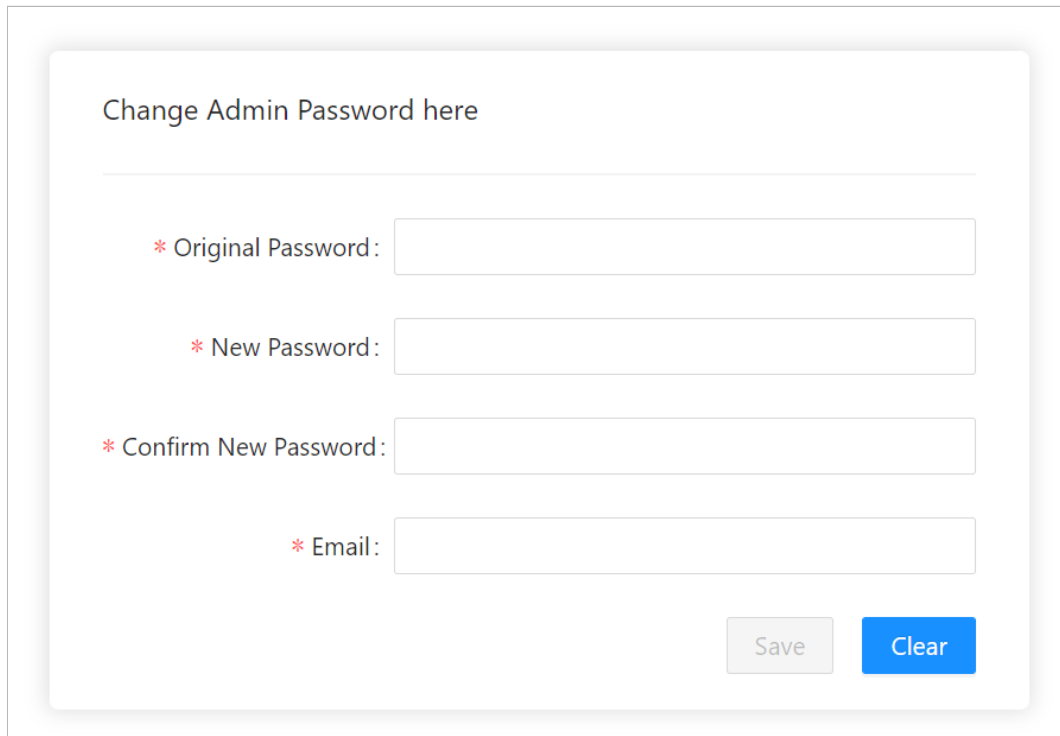
Tips: You can apply filters to view all the active and inactive users by clicking Active or Inactive user.

6.6. Admin Password

In the Admin Password of User Management, the administrator can change the passwords from original to new password.

Steps

1. Go to **Administration > User Management > Admin Password**.



Change Admin Password here

* Original Password:

* New Password:

* Confirm New Password:

* Email:

Save Clear

2. Enter Original Password, New Password, Confirm New Password, Email.
3. Click **SAVE**.

6.7. Forget Password

- The LDAP user will contact their respective Admin to reset the password
- The SMPT the user will contact the Pelican Admin to reset the password.

7. Datastore Configuration

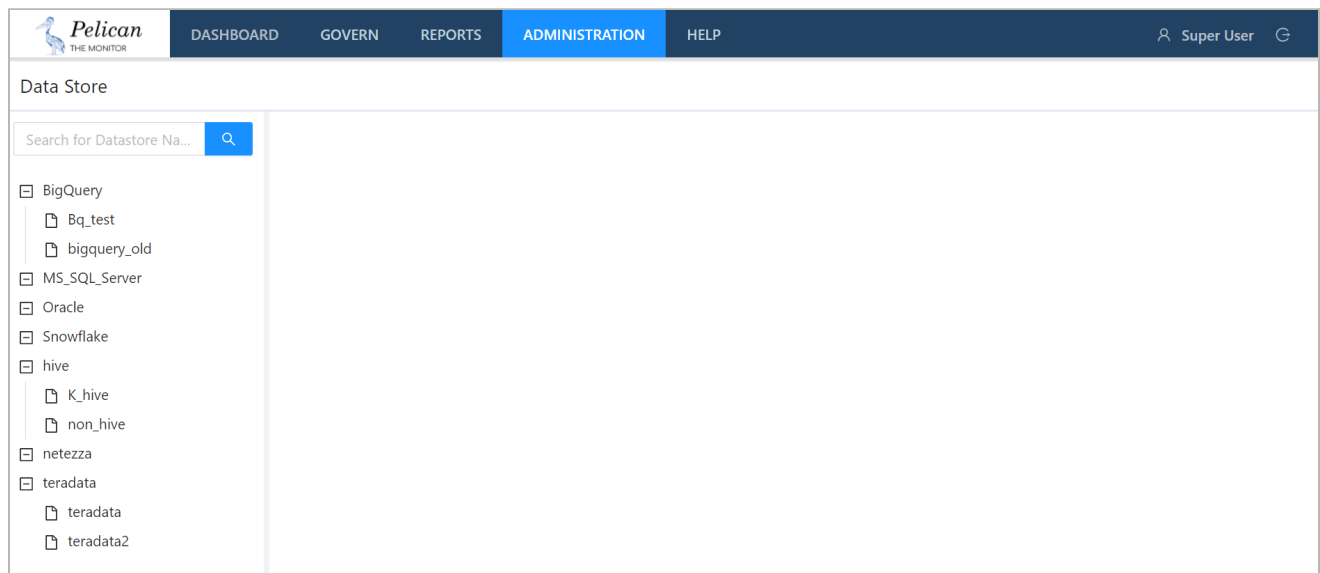
To compare and validate data across various data stores, first, you need to configure source and destination datastores in the Pelican. Pelican provides predefined data store types under which you can configure data stores as per the business requirement.

The datastore configuration is similar for all the data stores except HIVE which has a few more steps.

Let's see how to configure the Datastores.

Go to **Administration** **Configure** **Data Store**.

This displays the **Data Store screen** as shown below.



7.1. Steps for Adding Database configuration

To perform the data store activities, you need to provide the required information in the **Basic Settings** and **Pelican Settings** tabs.

1. Click on the datastore you want to create.
2. Go to the **Basic Settings** tab.

Data Store

🔍

- ☐ BigQuery
 - 📄 bq
- ☐ DB2
 - 📄 db2
- ☐ Databricks_Deltalake
 - 📄 deltalake
- ☐ MS_SQL_Server
 - 📄 ms_sql_server
- ☐ Oracle
 - 📄 oracle
- ☐ Snowflake
 - 📄 snowflake
- ☐ Synapse
 - 📄 Synapse_test
- ☐ Vertica
 - 📄 vertica
- ☐ hive

BigQuery

Basic Settings
Pelican Settings

* Data Store Name

* Data Store Host

* Project Id

* File Upload

📄

3. Enter the following information,

Basic setting:

1. Data Store Host: <https://www.googleapis.com/biqquery/v2>
2. Data Store Port: 443
3. Project Id: dm-eagle
4. OAuth Service Account Email: pelican@dmecattest.iam.gserviceaccount.com

Pelican setting:

1. Pipeline Temp Location: gs://dm_pelican/tmp/
2. Pipeline Stage Location: gs://dm_pelican/staging/
3. Cluster Zone: us-west1-b
4. Temporary Database: qastage

4. Add the Data Store Name and Data Store Description.

5. Upload the service_account.json file.

Basic Settings
Pelican Settings

*** Data Store Name**

Data Store Description

*** Data Store Host**

*** Data Store Port**

*** Project Id**

*** OAuth Service Account Email**

*** File Upload**

⬇ Upload service account JSON file

📎 config_json.txt

Once you upload the file **service.account.json** and add all the details in the field, then click on the **TEST CONNECTION**. If everything is correct then it displays the Success message and enables the Save button.

6. Click on the **Pelican settings** button and enter the corresponding details.

🔍

- ☐ BigQuery
 - ☐ bq
- ☐ DB2
 - ☐ db2
- ☐ Databricks_Deltalake
 - ☐ deltalake
- ☐ MS_SQL_Server
 - ☐ ms_sql_server
- ☐ Oracle
 - ☐ oracle
- ☐ Snowflake
 - ☐ snowflake
- ☐ Synapse
 - ☐ Synapse_test
- ☐ Vertica
 - ☐ vertica
- ☐ hive

BigQuery
TEST CONNECTION SAVE

Basic Settings
Pelican Settings

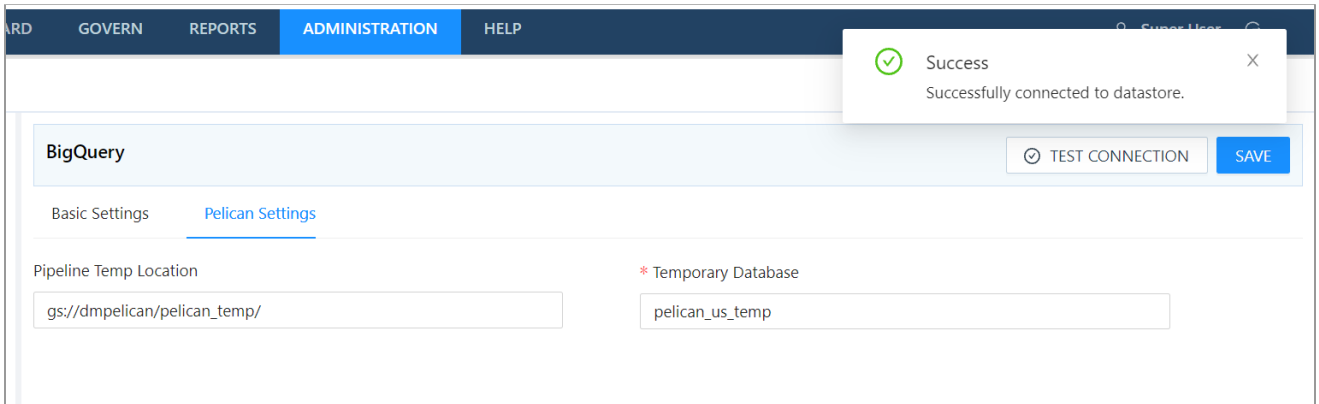
Pipeline Temp Location

*** Temporary Database**

Copyright © 2022 by Datametica Solutions Pvt. Ltd. All Rights Reserved.

20

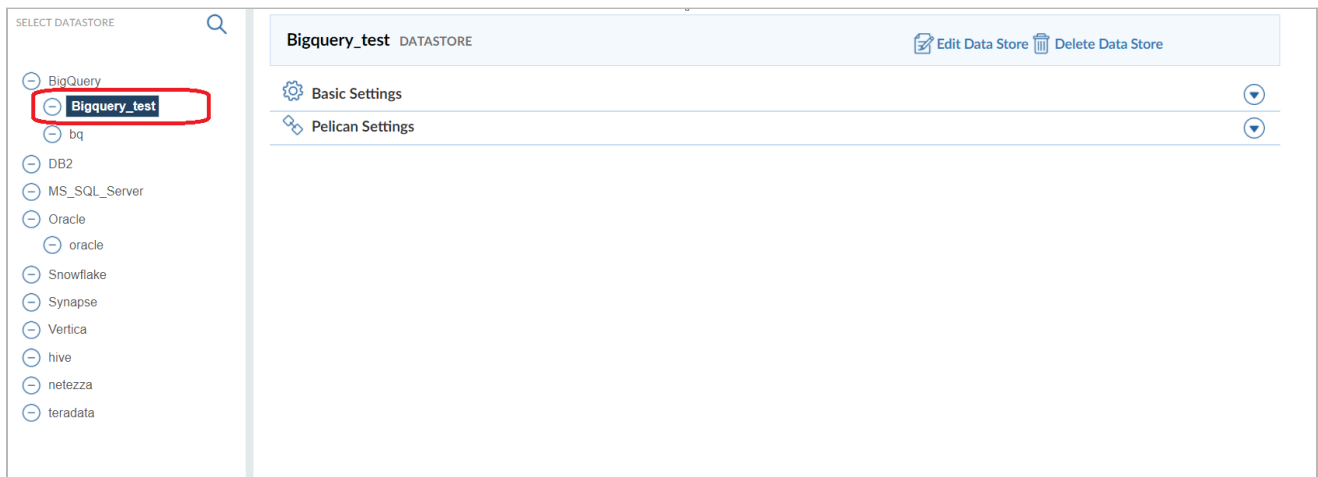
7. Click on the **TEST CONNECTION** button.



If the connection is correct then a message will pop up as **Success** in green color. Once the connection is verified, the **Save** button will be enabled.

8. Click **Save**.

The newly added datastore will be displayed in the datastore list.



7.1.1. Steps for Teradata Configuration

1. Go to **Administration > Configure > Datastore**.

A list of predefined datastores is visible, at the left-hand side pane.

2. Select **Teradata** in the data store field.

A screen to enter the corresponding datastore details will be visible.

Data Store

- ms_sql_server
- Oracle
 - oracle
- Snowflake
 - snowflake
- Synapse
 - Synapse_test
- Vertica
 - vertica
- hive
 - hive
- netezza
 - netezza_user
 - nz
- teradata
 - td

teradata

Basic Settings
Pelican Settings

*** Data Store Name**

*** Data Store Host**

*** JDBC Password**

Data Store Description

*** JDBC Username**

3. Enter all the details in the **Basic Settings** and **Pelican Settings** tab fields.
4. Click **TEST CONNECTION**.
 A success message will be displayed in case of a successful connection.
5. Click **SAVE**.
 The newly created datastore will be displayed in the datastore list.

Basic Settings Descriptions

Field	Description
Data Store Name	The name the user wants given to the data store.
Data Store Description	Description for the data store.
Data Store Host	IP address of the data store host machine.
JDBC Username	JDBC username using which Pelican will connect to the data store.
JDBC Password	JDBC password for the provided JDBC user.

Pelican Settings Description

teradata
TEST CONNECTION
SAVE

Basic Settings Pelican Settings

* Temporary Database

Field	Description
Temporary Database	Database used by Pelican for its functioning.

7.1.2. Steps for Netezza Configuration

1. Go to **Administration > Configure > Datastore**.
A list of predefined datastores is visible, at the left-hand side pane.
2. Select the **Netezza** datastore.
A screen to enter the corresponding datastore details will be visible.

- ms_sql_server
- Oracle
 - oracle
- Snowflake
 - snowflake
- Synapse
 - Synapse_test
- Vertica
 - vertica
- hive
 - hive
- netezza
 - netezza_user
 - nz
- teradata
 - td

netezza
TEST CONNECTION
SAVE

Basic Settings Pelican Settings

* Data Store Name

Data Store Description

* Data Store Host

* Data Store Port

* JDBC Username

* JDBC Password

Copyright © 2022 by Datametica Solutions Pvt. Ltd. All Rights Reserved.

23

3. Enter information in the **Basic Settings** and **Pelican Settings** to configure the data store.
4. Click **TEST CONNECTION**.
A success message will be displayed in case of a successful connection.
5. Click **SAVE**.
The newly created datastore will be displayed in the datastore list.

Basic Settings Field Descriptions

Field	Description
Data Store Name	The name the user wants given to the data store.
Data Store Description	Description for the data store.
Data Store Host	IP address of the data store host machine.
Data Store Port	JDBC port to connect to the data store.
JDBC Username	JDBC username using which Pelican will connect to the data store.
JDBC Password	JDBC password for the provided JDBC user.

Pelican Settings Field Description

netezza

Basic Settings Pelican Settings

* Temporary Database

Supplementary DataStore Location

Field	Description
Temporary Database	Database used by Pelican for its functioning.
Supplementary DataStore Location	Writable Path on the machine where Pelican is installed.

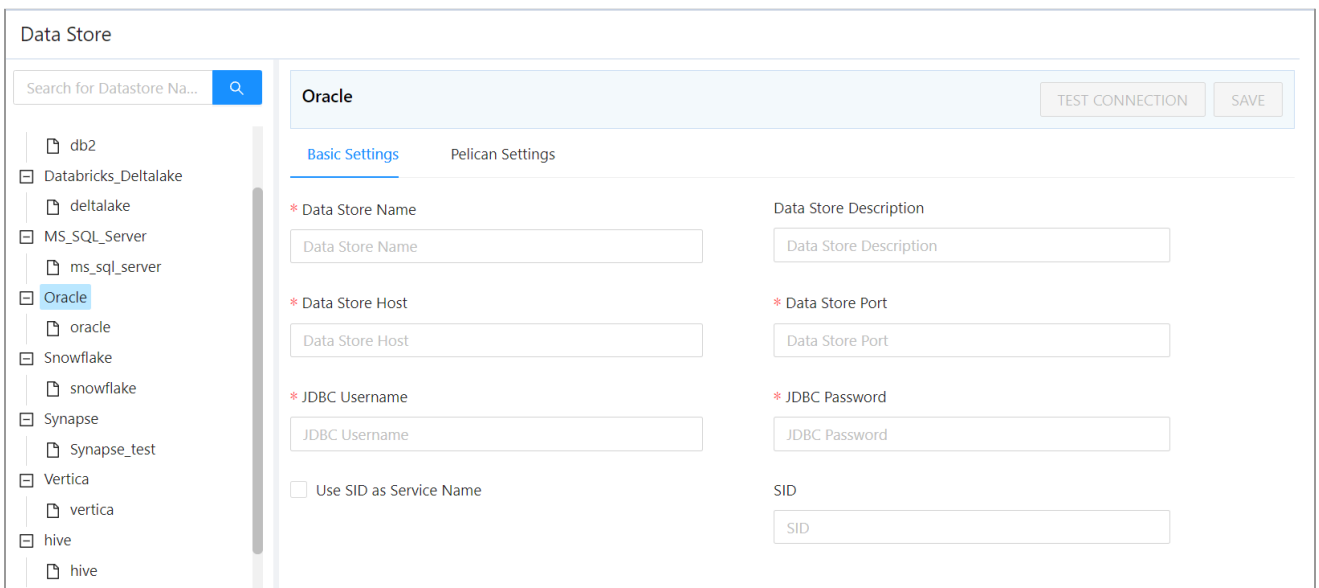
7.1.3. Steps for Oracle Configuration

1. Go to **Administration > Configure > Datastore**.

A list of predefined datastores is visible at the left-hand side pane.

2. Select the **Oracle**.

A screen to enter the corresponding datastore details will be visible.



The screenshot shows the 'Data Store' configuration page for Oracle. On the left, there is a search bar and a list of datastores including db2, Databricks_Deltalake, deltalake, MS_SQL_Server, ms_sql_server, Oracle (selected), oracle, Snowflake, snowflake, Synapse, Synapse_test, Vertica, vertica, hive, and hive. The main area is titled 'Oracle' and has two tabs: 'Basic Settings' (active) and 'Pelican Settings'. Below the tabs are several input fields:

- * Data Store Name: Data Store Name
- Data Store Description: Data Store Description
- * Data Store Host: Data Store Host
- * Data Store Port: Data Store Port
- * JDBC Username: JDBC Username
- * JDBC Password: JDBC Password
- Use SID as Service Name
- SID: SID

 At the top right of the configuration area are 'TEST CONNECTION' and 'SAVE' buttons.

3. Enter all the details in the **Basic Settings** and **Pelican Settings** tab fields.

4. Click **TEST CONNECTION**.

A success message will be displayed in case of a successful connection.

5. Click **SAVE**.

The newly created datastore will be displayed in the datastore list.

Basic Settings Descriptions

Field	Description
Data Store Name	The name the user wants given to the data store.
Data Store Description	Description for the data store.

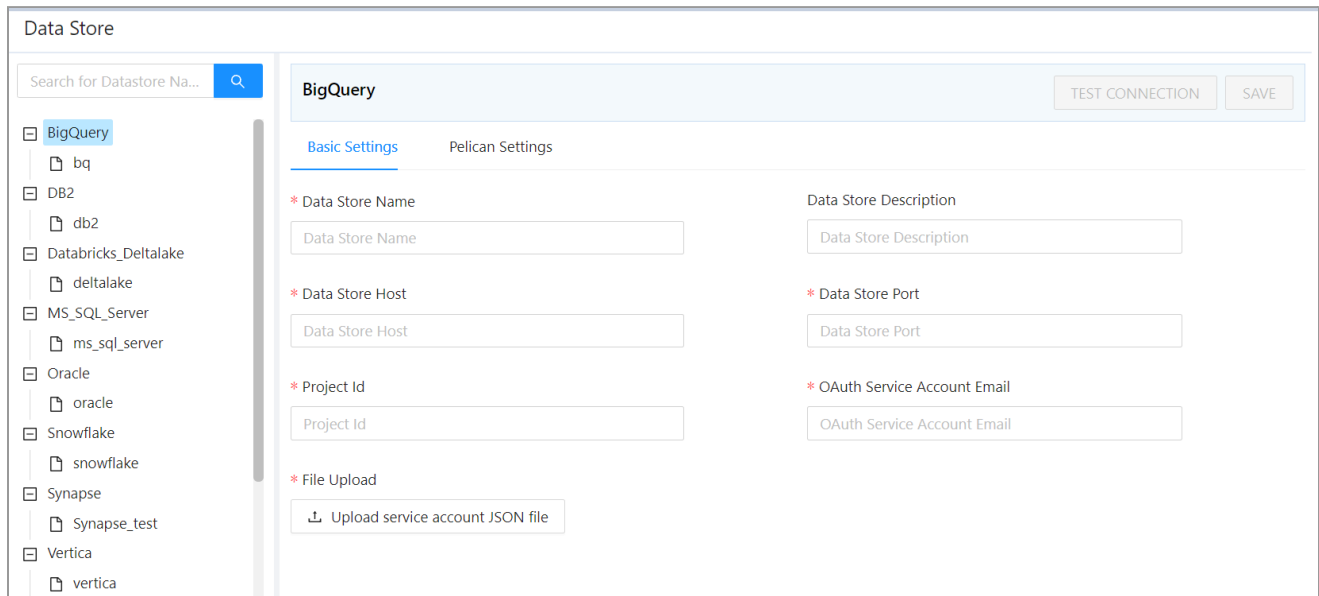
Data Store Host	IP address of the data store host machine.
Data Store Port	JDBC port to connect to the data store.
JDBC Username	JDBC username using which Pelican will connect to the data store.
JDBC Password	JDBC password for the provided JDBC user.
Use SID as Service Name	If the service name is provided for an Oracle instance, then select this checkbox and enter service name value in the SID textbox.
SID	The SID is a site identifier.

Pelican Settings Description

Field	Description
Temporary Database	Database used by Pelican for its functioning.

7.1.4. Steps for BigQuery Configuration

1. Go to **Administration > Configure > Datastore**.
A list of predefined datastores is visible at the left-hand side pane.
2. Select the **BigQuery** datastore.



The screenshot shows the 'Data Store' configuration page for BigQuery. On the left, a sidebar lists various datastores, with 'BigQuery' selected. The main area is titled 'BigQuery' and has two tabs: 'Basic Settings' (active) and 'Pelican Settings'. At the top right of the main area are 'TEST CONNECTION' and 'SAVE' buttons. The configuration fields are as follows:

- Data Store Name:** A text input field with a red asterisk indicating it is required.
- Data Store Description:** A text input field.
- Data Store Host:** A text input field with a red asterisk.
- Data Store Port:** A text input field with a red asterisk.
- Project Id:** A text input field with a red asterisk.
- OAuth Service Account Email:** A text input field with a red asterisk.
- File Upload:** A section with a red asterisk and a button labeled 'Upload service account JSON file'.

3. Enter all the details in the **Basic Settings** and **Pelican Settings** tab fields. Click **Choose File**. And provide the path for the google cloud service account key json file.
4. Click **TEST CONNECTION**.
A success message will be displayed in case of a successful connection.
5. Click **SAVE**.
The newly created datastore will be displayed in the datastore list.

Basic Settings Field Descriptions

Field	Description
Data Store Name	The name the user wants given to the data store.
Data Store Description	Description for the data store.
Data Store Host	IP address of the data store host machine.
Data Store Port	JDBC port to connect to the data store.
Project Id	BigQuery Project to which Pelican will connect,
OAuth Service Account Email	Service account email for authentication.

Pelican Settings Field Description

Field	Description
Pipeline Temp Location	A Cloud Storage path for Dataflow to stage any temporary files.
Temporary Database	Database used by Pelican for its functioning.

7.1.5. Steps for HIVE Configuration

1. Go to **Administration > Configure > Datastore**.
A list of predefined datastores is visible at the left-hand side pane.
2. Select **HIVE** data store field. A screen to enter the corresponding datastore details will be visible.

Data Store

- ms_sql_server
- Oracle
 - oracle
- Snowflake
 - snowflake
- Synapse
 - Synapse_test
- Vertica
 - vertica
- hive
 - hive
- netezza
 - netezza_user
 - nz
- teradata
- td

hive

Basic Settings
Security Settings
Pelican Settings

*** Data Store Name**

Data Store Description

*** JDBC Username**

*** JDBC Password**

*** JDBC URL**

Additional Properties For JDBC URL

3. Enter all the details in the **Basic Settings**, **Security Settings** and **Pelican Settings** fields.
4. Click **TEST CONNECTION**.
A success message will be displayed in case of a successful connection.
5. Click **SAVE**.
The newly created datastore will be displayed in the datastore list.

Basic Settings Field Descriptions

Field	Description
Data Store Name	The name the user wants given to the data store.
Data Store Description	Description for the data store.
JDBC Username	JDBC username using which Pelican will connect to the data store.
JDBC Password	JDBC password for the provided JDBC user.
JDBC URL	Hive host URL for JDBC connection.
Additional Properties For JDBC URL	Additional Properties For JDBC URL

Security Settings Field Description

Field	Description
Kerberos Enabled	Check for kerberized Hive.
Use Sasl	Check box based if sasl is required
User Kerberos Principal	Pattern will be like user@organization.com.
Kerberos Service Principal for JDBC	A Kerberos principal is a unique identity to which Kerberos can assign tickets.
Kerberos Service Principal for Meta Store	HDFS meta store path
User Keytab Location	User keytab file location in case of Kerberos Enable.
SASL QOP Enable	Checkbox to enable SASL QOP
SASL QOP	SASL Mechanisms
Kerberos Enabled for JDBC	Can be checked if Kerberos Enable is true.
Kerberos Enabled for Metastore	Can be checked if Kerberos Enable is true.

Pelican Settings Field Description

Field	Description
Temporary Database	Database used by Pelican for its functioning.
Queue Name	Describes the name of the queue.
Queue Value	Describes the queue value.

7.1.6. Steps for DB2 Configuration

1. Go to **Administration > Configure > Datastore**.

A list of predefined datastores is visible at the left-hand side pane.

2. Select **DB2** data store field. A screen to enter the corresponding datastore details will be visible.

DB2

[Basic Settings](#)

[Pelican Settings](#)

<p>* Data Store Name</p> <input style="width: 95%; border: 1px solid #ccc;" type="text" value="Data Store Name"/>	<p>Data Store Description</p> <input style="width: 95%; border: 1px solid #ccc;" type="text" value="Data Store Description"/>
<p>* Data Store Host</p> <input style="width: 95%; border: 1px solid #ccc;" type="text" value="Data Store Host"/>	<p>* Data Store Port</p> <input style="width: 95%; border: 1px solid #ccc;" type="text" value="Data Store Port"/>
<p>* JDBC Username</p> <input style="width: 95%; border: 1px solid #ccc;" type="text" value="JDBC Username"/>	<p>* JDBC Password</p> <input style="width: 95%; border: 1px solid #ccc;" type="text" value="JDBC Password"/>

3. Enter all the details in the **Basic Settings** and **Pelican Settings** tabs.
4. Click **TEST CONNECTION**.
A success message will be displayed in case of a successful connection.
5. Click **SAVE**.

Basic Settings Description

Field	Description
Data Store Name	The name the user wants given to the data store.
Data Store Description	Description for the data store.
Data Store Host	IP address of the data store host machine.
Data Store Port	JDBC port to connect to the data store.
JDBC Username	JDBC username using which Pelican will connect to the data store.
JDBC Password	JDBC password for the provided JDBC user.

Pelican Settings Description

Field	Description
Temporary Database	Database used by Pelican for its functioning.
Supplementary DataStore Location	Writable Path on the machine where Pelican is installed.

7.1.7. Steps for Synapse Configuration

1. Go to **Administration > Configure > Datastore**.

At the left-hand side pane, a list of predefined datastores is visible.

2. Select **Synapse** datastore.

A screen to enter the corresponding datastore details will be visible.

Synapse

Basic Settings
Pelican Settings

<p>* Data Store Name</p> <input style="width: 90%;" type="text" value="Data Store Name"/>	<p>Data Store Description</p> <input style="width: 90%;" type="text" value="Data Store Description"/>
<p>* Data Store Host</p> <input style="width: 90%;" type="text" value="Data Store Host"/>	<p>* Data Store Port</p> <input style="width: 90%;" type="text" value="Data Store Port"/>
<p>* JDBC Username</p> <input style="width: 90%;" type="text" value="JDBC Username"/>	<p>* JDBC Password</p> <input style="width: 90%;" type="text" value="JDBC Password"/>
<p>* Database</p> <input style="width: 90%;" type="text" value="Database"/>	<p>* Account Name</p> <input style="width: 90%;" type="text" value="Account Name"/>
<p>* Account Key</p> <input style="width: 90%;" type="text" value="Account Key"/>	

3. Enter all the details in the **Basic Settings** and **Pelican Settings** fields.

4. Click **TEST CONNECTION**.

A success message will be displayed in case of a successful connection.

5. Click **SAVE**.

The newly created datastore will be displayed in the datastore list.

Basic Settings Description

Field	Description
Data Store Name	The name the user wants given to the data store.
Data Store Description	Description for the data store.
Data Store Host	IP address of the data store host machine.
Data Store Port	JDBC port to connect to the data store.
JDBC Username	JDBC username using which Pelican will connect to the data store.
JDBC Password	JDBC password for the provided JDBC user.
Database	Database name
Account Name	Name of the account
Account Key	Unique account key

Pelican Settings Field Description

Field	Description
Temporary Database	Database used by Pelican for its functioning.
Supplementary DataStore Location	Writable Path on the machine where Pelican is installed.

7.1.8. Steps for Snowflake Configuration

1. Go to **Administration > Configure > Datastore**.
A list of predefined datastores is visible at the left-hand side pane.
2. Select **Snowflake** datastore field. A screen to enter the corresponding datastore details will be visible.

Snowflake

Basic Settings Pelican Settings

<p>* Data Store Name</p> <input type="text" value="Data Store Name"/>	<p>Data Store Description</p> <input type="text" value="Data Store Description"/>
<p>* Data Store Host</p> <input type="text" value="Data Store Host"/>	<p>* Data Store Port</p> <input type="text" value="Data Store Port"/>
<p>* JDBC Username</p> <input type="text" value="JDBC Username"/>	<p>* JDBC Password</p> <input type="text" value="JDBC Password"/>
<p>* Warehouse Name</p> <input type="text" value="Warehouse Name"/>	<p>* Role</p> <input type="text" value="Role"/>
<p>* OAuth Service Account Email</p> <input type="text" value="OAuth Service Account Email"/>	<p>* Project Id</p> <input type="text" value="Project Id"/>
<p>* File Upload</p> <input type="button" value="Upload service account JSON file"/>	

3. Enter all the details in the **Basic Settings** and **Pelican Settings** fields.
4. Click **TEST CONNECTION**.
 A success message will be displayed in case of a successful connection.
5. Click **SAVE**.
 The newly created datastore will be displayed in the datastore list.

Basic Settings Field Description

Field	Description
Data Store Name	The name the user wants given to the data store.
Data Store Description	Description for the data store.
Data Store Host	IP address of the data store host machine.
Data Store Port	JDBC port to connect to the data store.
JDBC Username	JDBC username using which Pelican will connect to the data store.

JDBC Password	JDBC password for the provided JDBC user.
Warehouse Name	Name of the warehouse
Role	Add the role
OAuth Service Account Email	Add service account email ID
Project Id	Add project ID

Pelican Settings Field Description

Field	Description
Temporary Database	Database used by Pelican for its functioning.
Cluster Zone	Zone of the cluster
Pipeline Temp Location	A Cloud Storage path for Dataflow to stage any temporary files.
Pipeline Stage Location	A Cloud Storage bucket for Dataflow to stage your binary files

7.1.9. Steps for Deltalake Configuration

We are currently supporting the following Deltalake pair: **Hive to Deltalake**.

Steps for Deltalake configuration:

1. Go to **Administration > Configure > Datastore**.
2. At the left-hand side pane a list of predefined datastore lists is visible. Select the Deltalake datastore field.

A screen to enter the corresponding datastore details will be visible.

Databricks_Deltalake

Basic Settings

Pelican Settings

* Data Store Name

Data Store Description

* Data Store Host

* Data Store Port

* User Token

Additional Properties For JDBC URL

3. Enter all the details in the: Basic Settings, Pelican Settings fields.
4. Click **TEST CONNECTION**.
A success message will be displayed in case of a successful connection.
5. Click **SAVE**.
The newly created datastore will be displayed in the datastore list.

Field	Description
Data Store Name	The name the user wants given to the data store.
Data Store Description	Description for the data store.
Data Store Host	IP address of the data store host machine.
Data Store Port	JDBC port to connect to the data store.
User Token	A token allows access to the datastore. Expiry date for the token can be set so that the token can no longer be used by an unauthorised person after its use.
Additional Properties For JDBC URL	Additional Properties For JDBC URL

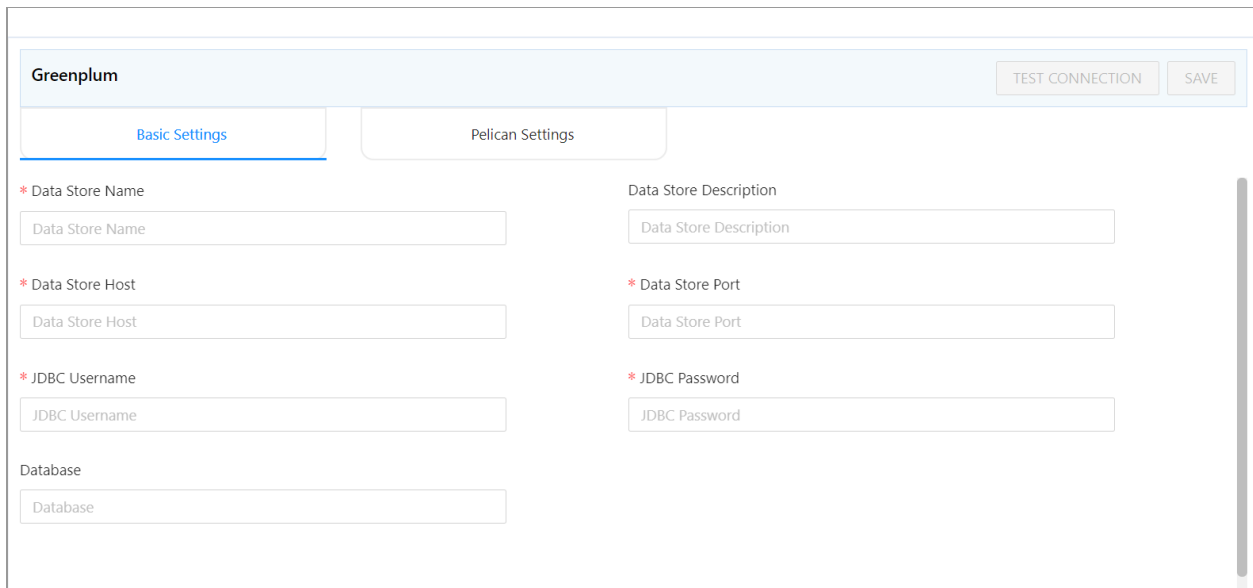
Pelican Settings Field Description

Field	Description
Temporary Database	Database used by Pelican for its functioning.
Supplementary DataStore Location	Writable Path on the machine where Pelican is installed.

7.1.10. Steps for Greenplum Configuration

1. Go to **Administration > Configure > Datastore**.
2. At the left-hand side pane a list of predefined datastore lists is visible. Select the **Greenplum** datastore field.

A screen to enter the corresponding datastore details will be visible.



The screenshot shows the configuration page for a Greenplum datastore. At the top, there are two tabs: "Basic Settings" (selected) and "Pelican Settings". In the top right corner, there are two buttons: "TEST CONNECTION" and "SAVE".

Under the "Basic Settings" tab, the following fields are visible:

- * Data Store Name:
- Data Store Description:
- * Data Store Host:
- * Data Store Port:
- * JDBC Username:
- * JDBC Password:
- Database:

3. Enter all the details in the **Basic Settings**, and **Pelican Settings** tab fields.
4. Click **TEST CONNECTION**.

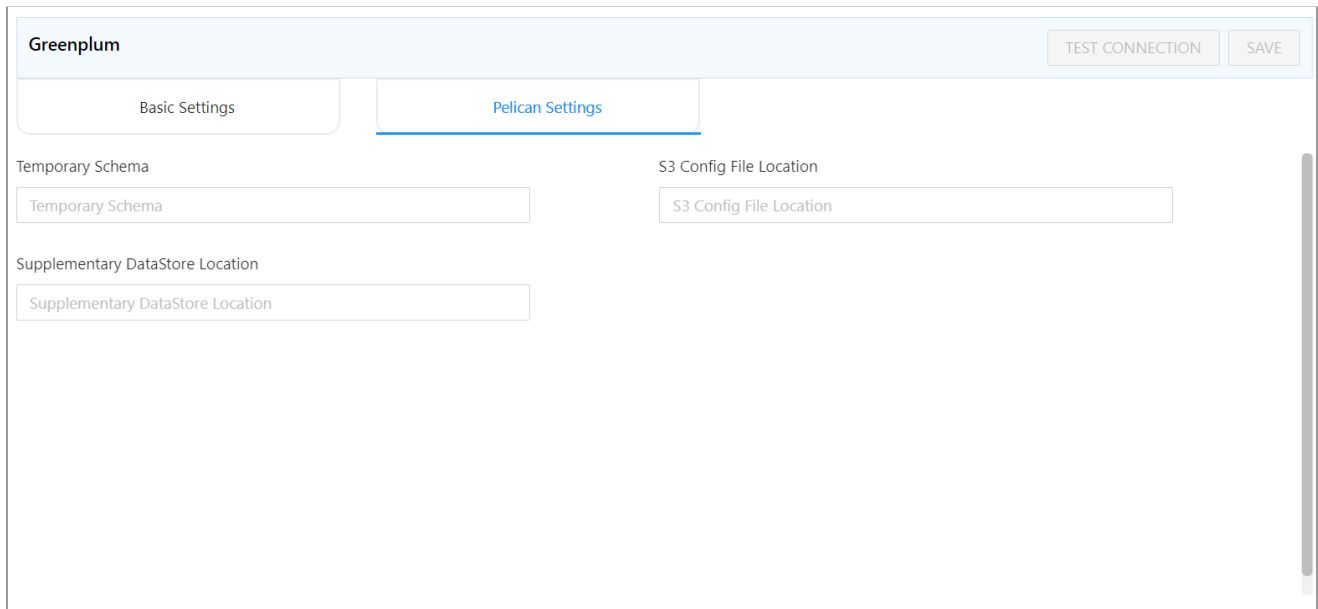
A success message will be displayed in case of a successful connection.

5. Click **SAVE**.

You may view the created data store in the list under the respective data store type.

Field	Description
Data Store Name	The name the user wants given to the data store.
Data Store Description	Description for the data store.
Data Store Host	IP address of the data store host machine.
Data Store Port	JDBC port to connect to the data store.
JDBC Username	JDBC username using which Pelican will connect to the data store.
JDBC Password	JDBC password for the provided JDBC user.
Database	A database the user wants to connect.

Pelican Settings Field Description

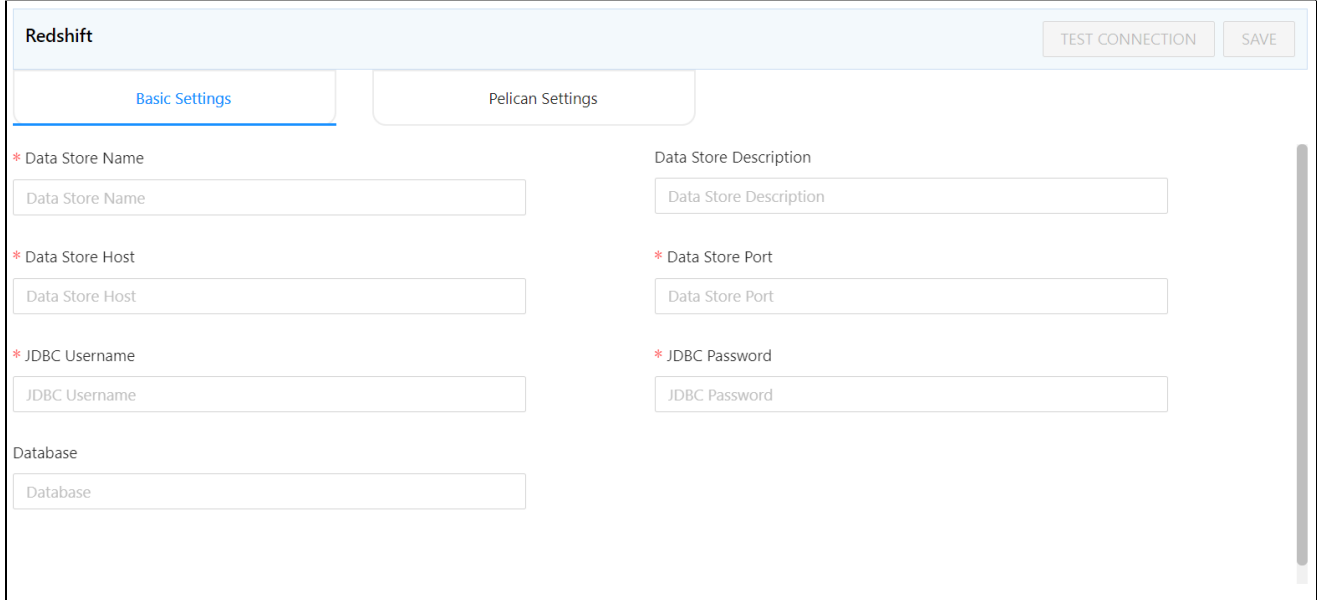


Field	Description
Temporary Schema	Temporary schema used by Pelican for its functioning.
S3 Config File Location	The location of S3 Config Files.

Supplementary DataStore Location	Supplementary DataStore Location
----------------------------------	----------------------------------

7.1.11. Steps for Redshift Configuration

1. Go to **Administration > Configure > Datastore**.
2. At the left-hand side pane a list of predefined datastore lists is visible. Select the **Redshift datastore** field.
3. A screen to enter the corresponding datastore details will be visible.



Redshift TEST CONNECTION SAVE

Basic Settings Pelican Settings

* Data Store Name Data Store Description

Data Store Name Data Store Description

* Data Store Host * Data Store Port

Data Store Host Data Store Port

* JDBC Username * JDBC Password

JDBC Username JDBC Password

Database

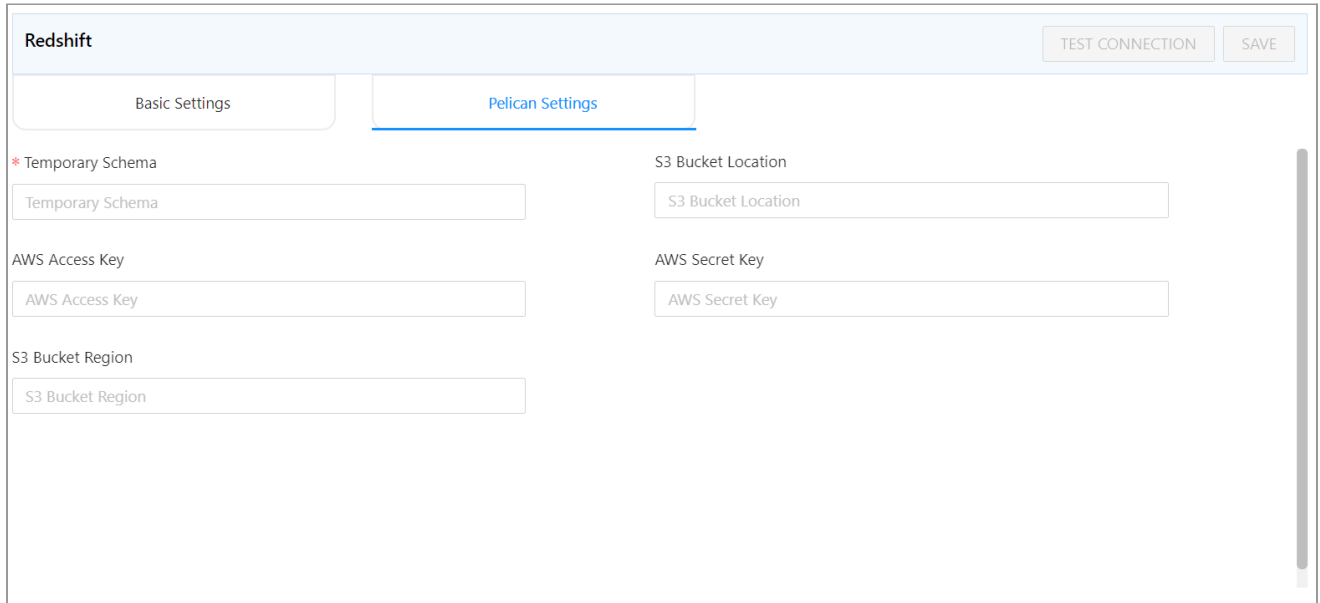
Database

Field	Description
Data Store Name	The name the user wants given to the data store.
Data Store Description	Description for the data store.
Data Store Host	IP address of the data store host machine.
Data Store Port	JDBC port to connect to the data store.
JDBC Username	JDBC username using which Pelican will connect to the data store.
JDBC Password	JDBC password for the provided JDBC user.

Database	The database the user wants to connect.
----------	---

4. Enter all the details in the **Basic Settings**, and **Pelican Settings** tab fields.
5. Click **Test Connection**.

A success message will be displayed in case of a successful connection.



The screenshot shows the 'Redshift' configuration window with the 'Pelican Settings' tab selected. The interface includes a 'TEST CONNECTION' button and a 'SAVE' button. The 'Pelican Settings' tab contains the following fields:

- * Temporary Schema**: A text input field labeled 'Temporary Schema'.
- S3 Bucket Location**: A text input field labeled 'S3 Bucket Location'.
- AWS Access Key**: A text input field labeled 'AWS Access Key'.
- AWS Secret Key**: A text input field labeled 'AWS Secret Key'.
- S3 Bucket Region**: A text input field labeled 'S3 Bucket Region'.

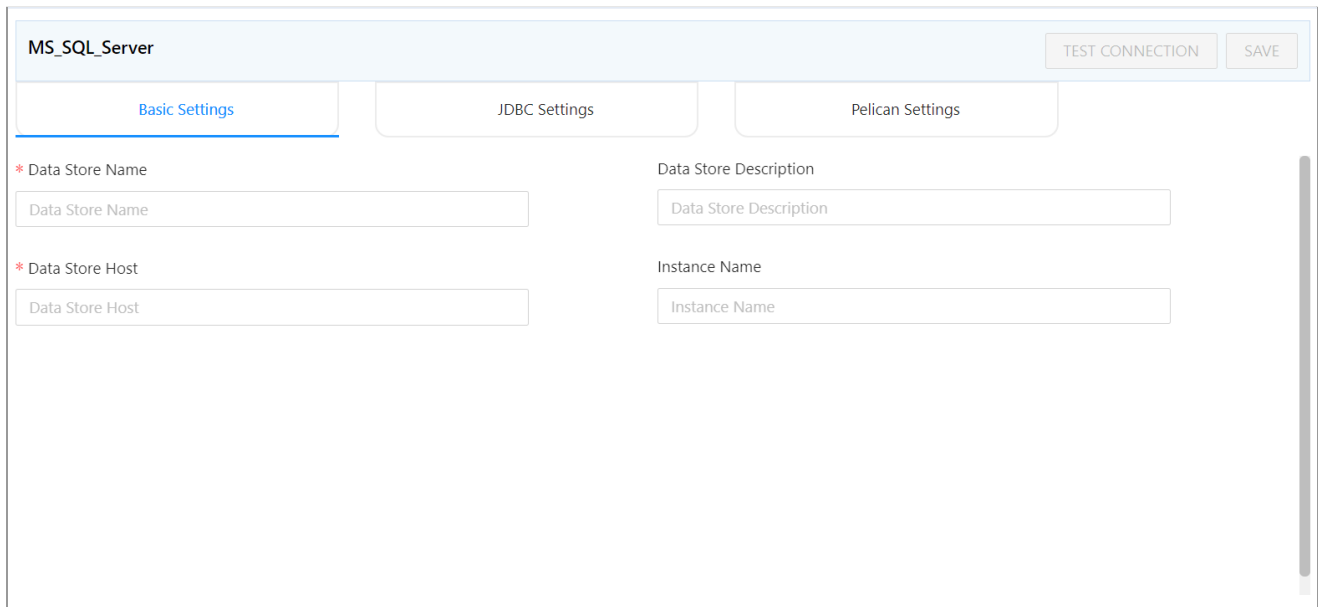
Field	Description
Temporary Schema	Temporary schema used by Pelican for its functioning.
S3 Bucket Location	Amazon S3 creates buckets in a location that you specify
AWS Access Key	Access keys are long-term credentials for an IAM user or the AWS account root user
AWS Secret key	It is a secret key similar to passwords.
S3 Bucket Region	Amazon S3 creates buckets in a Region that you specify.

7.1.12. Steps for MSSQL Server Configuration

1. Go to **Administration > Configure > Datastore**.

- At the left-hand side pane a list of predefined datastore lists is visible. Select the **MS SQL Server** datastore field.

A screen to enter the corresponding datastore details will be visible.



Field	Description
Data Store Name	The name the user wants given to the data store.
Data Store Description	Description for the data store.
Data Store Host	IP address of the data store host machine.
Instance Name	Name of the Instance

- Enter all the details in the **Basic Settings**, **JDBC Settings** and **Pelican Settings** tab fields.

MS_SQL_Server
TEST CONNECTION
SAVE

Basic Settings

JDBC Settings

Pelican Settings

* Data Store Port * JDBC Username

* JDBC Password

Field	Description
Data Store Port	JDBC port to connect to the data store.
JDBC Username	JDBC username using which Pelican will connect to the data store.
JDBC Password	JDBC password for the provided JDBC user.

MS_SQL_Server
TEST CONNECTION
SAVE

Basic Settings

JDBC Settings

Pelican Settings

* Temporary Database Supplementary DataStore Location

Field	Description
Temporary Database	Database used by Pelican for its functioning.
Supplementary DataStore Location	Writable Path on the machine where Pelican is installed.

4. Click **Test Connection**.

A success message will be displayed in case of a successful connection.

7.1.13. Steps for Vertica Configuration

1. Go to **Administration > Configure > Datastore**.
2. At the left-hand side pane a list of predefined datastore lists is visible. Select the **Vertica** datastore field.

A screen to enter the corresponding datastore details will be visible.

Vertica
TEST CONNECTION
SAVE

Basic Settings
Pelican Settings

* Data Store Name

Data Store Description

* Data Store Host

* Data Store Port

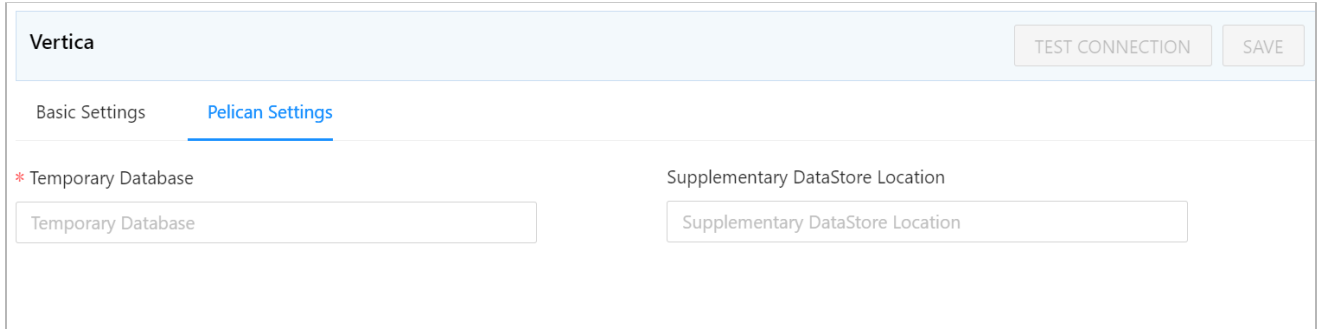
* JDBC Username

* JDBC Password

Database

Field	Description
Data Store Name	The name the user wants given to the data store.
Data Store Description	Description for the data store.
Data Store Host	IP address of the data store host machine.
Datastore Port	JDBC port to connect to the data store.
JDBC Username	JDBC username using which Pelican will connect to the data store.
JDBC Password	JDBC password for the provided JDBC user.
Database	Name of the Database

5. Enter all the details in the **Basic Settings** and **Pelican Settings** tab fields.



Vertica TEST CONNECTION SAVE

Basic Settings Pelican Settings

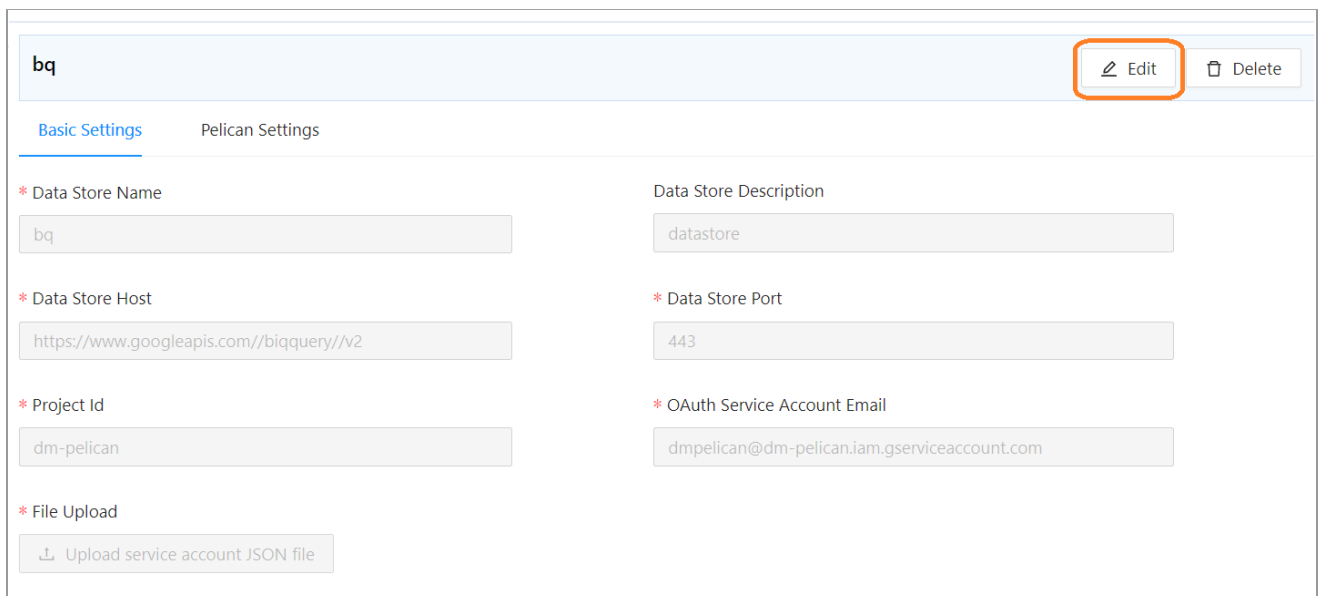
* Temporary Database Supplementary DataStore Location

Temporary Database Supplementary DataStore Location

Field	Description
Temporary Database	Database used by Pelican for its functioning.
Supplementary Datastore Location	Writable Path on the machine where Pelican is installed.

7.2. Editing an existing Datastore

1. Go to **Administration** → **Configure** → **Datastores**.
2. Select the Datastore you want to edit.
3. Click on the **Edit** icon.



bq Edit Delete

Basic Settings Pelican Settings

* Data Store Name Data Store Description

bq datastore

* Data Store Host * Data Store Port

https://www.googleapis.com//biqquery/v2 443

* Project Id * OAuth Service Account Email

dm-pelican dmpelican@dm-pelican.iam.gserviceaccount.com

* File Upload

Upload service account JSON file

-
4. Edit the information of the desired field, upload the service_account.json file.
 5. Click on the **Test Connection button**.
If the connection is correct then the message will pop up as **Success** in green color.
Once the connection is verified, the **Save** button will be enabled.
 6. Click **Save**.

7.3. Deleting existing datastore

Select the data store you want to delete and click the Delete Datastore icon. The system displays a confirmation dialog box to ensure the deletion of the data store so click DELETE.

8. Validation Configuration

After the source and destination data store is created, the next step is validation configuration. Validation configuration allows you to select tables (source and destination) from the respective data stores and map them.

Validation configuration searches the table with similar column name and data type as in the source Datastore. The application provides various filters to search the desired data store, reference database, destination database, reference schema and destination schema. Additionally, Pelican facilitates the following approximate mapping methods which allows you to map the tables based on their names and patterns:

- **Phonetic Matching:** A phonetic matching is an algorithm for **matching words by their pronunciation**.

For example, the words Principal and Principle are phonetically matching words. Which means, table names (source and destination) Principal and Principle will be considered for mapping.

- **Approximate Matching:** Approximate matching is based on Levenshtein distance. It is a metric for measuring the difference between two words.

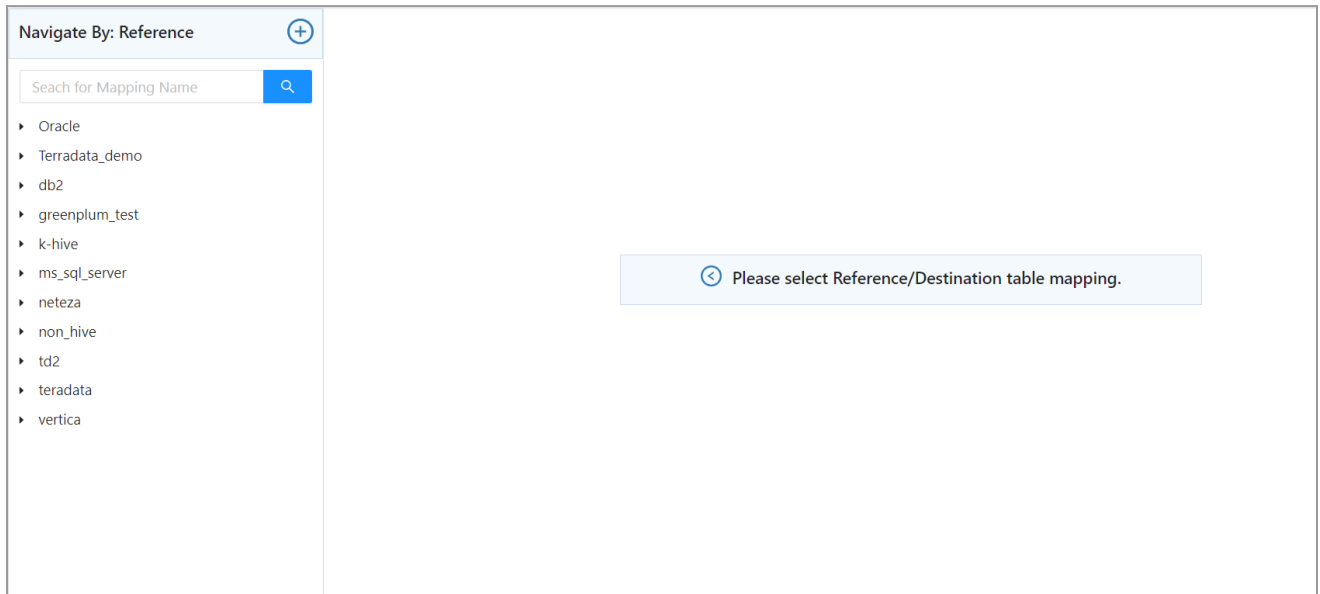
For Example. In the word 'Employee', insertion of "e" at the end → distance is 1, while in 'Employee' (insertion of "ee" at the end) distance is 2.

The above-mentioned table pair (source and destination) is considered for the mapping as the Levenshtein distance between these table pairs is either 1 or 2.

8.1. Table Mapping

To validate tables between source and destination data store.

1. Go to **Govern Validation Configuration.**



2. Click on the create new mapping icon.



Please select Unique Key's for optimal performance.

* Datastore Pair: Approximation Threshold: Enable Phonetic Match:

* Reference Database: * Destination Database:

* Reference Schema: * Destination Schema:

* Tables:

3. Select the following information:

- a **Datastore Pair:** Displays all the combinations of source and target data stores configured in the system.
- b **Reference Database:** Displays all the databases of the corresponding source data store.
- c **Reference Schema:** This field will be enabled only if the source datastore supports three level hierarchy. It displays all the schemas present in the database selected in the **Reference Database**.
- d **Destination Database:** Displays all the databases of the corresponding destination data store.
- e **Destination Schema:** This field will be enabled only if the target datastore supports a three level hierarchy*. Displays all the schemas present in the database selected in the **Destination Database**.
- f Select tables in the table section.

This section allows the user to select a table that exists in the selected reference database and schema. While mapping, the application searches the same table name in the destination database and schema. Further it also

searches similar column names and data types as in source datastore. You can use the approximation matching parameters to search tables, columns, and data types of similar patterns as in the source data store.

g Click **Map**.

After you click on **Map**, the application displays the mapping result in four sections as shown below.

Please select Unique Key's for optimal performance.

Mapped Tables <i>(1 Tables Found)</i>		Partially Mapped Tables <i>(No Data Found)</i>		Unmapped Tables <i>(No Data Found)</i>		Previously Mapped Tables <i>(No Data Found)</i>		
TEST_DATA_1000 (Reference Table)		Test_DATA_1000 (Destination Table)		Configuration			Result	
Expression	Expression Datatype	Expression	Expression Datatype	Unique Key Column	Timestamp Column	Is Matching	Remove	Reset
GENDER	varchar	gender	STRING	<input type="checkbox"/>	<input type="radio"/>	✓		
IP_ADDRESS	varchar	ip_address	STRING	<input type="checkbox"/>	<input type="radio"/>	✓		
ID	integer	id	INT64	<input type="checkbox"/>	<input type="radio"/>	✓		
LAST_NAME	varchar	last_name	STRING	<input type="checkbox"/>	<input type="radio"/>	✓		
EMAIL	varchar	email	STRING	<input type="checkbox"/>	<input type="radio"/>	✓		
FIRST NAME	varchar	first name	STRING	<input type="checkbox"/>	<input type="radio"/>	✓		

Back Save

- a **Mapped Tables:** This tab contains the table where the table name and each column of the source table matches with that of the target table.
- b **Partially Mapped Tables:** This tab contains the table/s where few columns of a source table/s match with the target table/s.
- c **Unmapped Tables:** This tab contains the table/s where none of the columns of the source table matches with the target table/s.
- d **Previously Mapped Tables:** This tab shows the previously mapped tables.

4. Click the respective tab (section) to see the results.

5. The user can perform following operations on the result page:

- Show Column
- Edit Configuration
 - Edit Column
 - Edit Expression
 - Edit Data Type
 - Add New Columns
- Delete table and column
- Override Unmapped Columns

8.1.1. Edit the existing configuration

This functionality allows you to edit existing columns, expression, data type, and add new columns as per the business requirement.

Steps to edit the existing configuration:



Step 1: Click the **Edit Configuration** icon on the screen.

The application displays the **Edit Configuration** pop-up window as shown below.

Edit Configuration +
×

Alias Name	Source Original Column Name	test_500 (Source Table)		Target Original Column Name	test_500 (Target Table)		Action
		Expression	Expression Datatype		Expression	Expression Datatype	
<input type="text" value="city"/>	city	<input type="text" value="city"/>	character varying	city	<input type="text" value="city"/>	character varying	
<input type="text" value="id"/>	id	<input type="text" value="id"/>	integer	id	<input type="text" value="id"/>	integer	
<input type="text" value="age"/>	age	<input type="text" value="age"/>	integer	age	<input type="text" value="age"/>	integer	
<input type="text" value="name"/>	name	<input type="text" value="name"/>	character varying	name	<input type="text" value="name"/>	character varying	

Showing 1 to 4 entries of 4

In the **Edit Configuration** window, you can edit existing columns, existing expressions, existing data type and add new columns.

- **Edit existing column name**

Pelican allows the user to edit the existing column name (Alias Name) to make it uniform on both sides (Reference and Destination).

For example:

Suppose, two columns, namely, **order_name** and **ordername** exist in the reference and destination tables respectively. Both the columns display the order name hence logically they are the same. So, instead of having two different column names you can edit it to either **order_name** or **ordername** as an alias name.

To edit existing column name:

Enter the desired column name in the **Alias Name** field of the respective column and click **OK**.

The changes made in the **Edit Configuration** screen will be saved only in Pelican UI and not in the original database.

- **Edit existing expression**

After mapping, the expressions are automatically created. The expressions are created based on the datatype of the respective column. However, the user can edit these existing expressions as per the requirement.

For example,

Suppose the **date** column exists on both the sides (Reference and Destination tables), the data formats of these columns are **DD:MM: YYYY HH:MM: SS** and **DD:MM: YYYY** respectively. the **HH:MM: SS** is missing in the destination table. So, using the edit expression, you can match the mismatched expression.

To edit the existing expression:

Click the desired expression row, edit the expression and click **OK**.

Edit existing data type

The user can edit the existing data type of any column in the reference and destination table. Suppose there is a column that exists on both the sides (Reference and Destination tables), but the data type of these columns is **Varchar** and **String** respectively. Logically these two data types are similar in nature. Hence, instead of two data types, the user can change it to either **String** or **Varchar**.

To edit the existing data type:

Select the new data type from the **Datatype** drop-down list and click **OK**.

Add new column

Pelican allows the user to add new columns as per the business requirement, the user can add any number of columns.

Edit Configuration +
×

Alias Name	Source Original Column Name	test_500 (Source Table)		Target Original Column Name	test_500 (Target Table)		Action
		Expression	Expression Datatype		Expression	Expression Datatype	
<input type="text" value="city"/>	city	<input type="text" value="city"/>	character varying	city	<input type="text" value="city"/>	character varying	
<input type="text" value="id"/>	id	<input type="text" value="id"/>	integer	id	<input type="text" value="id"/>	integer	
<input type="text" value="age"/>	age	<input type="text" value="age"/>	integer	age	<input type="text" value="age"/>	integer	
<input type="text" value="name"/>	name	<input type="text" value="name"/>	character varying	name	<input type="text" value="name"/>	character varying	

Showing 1 to 4 entries of 4

2. The application adds the new column as shown below.

Edit Configuration ✕

Alias Name	Source Original Column Name	test_500 (Source Table)		Target Original Column Name	test_500 (Target Table)		Action
		Expression	Expression Datatype		Expression	Expression Datatype	
test_500_4							
city	city	city	character varying	city	city	character varying	
id	id	id	integer	id	id	integer	
age	age	age	integer	age	age	integer	
name	name	name	character varying	name	name	character varying	

Showing 1 to 5 entries of 5 < 1 > 10 / page ▾

3. Enter column name (Alias Name).

4. Enter expression, data type on both the sides (Reference and Destination tables) and click Save to save the newly added column.

8.1.2. Delete Mapping

The user can delete mappings from the respective sections if they are not required. However, the deleted mappings will be removed only from the Pelican UI and not from the original database.

To delete a table:

1. Click Delete Table  icon as shown below.

Edit Configuration ✕

Alias Name	Source Original Column Name	test_500 (Source Table)		Target Original Column Name	test_500 (Target Table)		Action
		Expression	Expression Datatype		Expression	Expression Datatype	
test_500_4							
city	city	city	character varying	city	city	character varying	
id	id	id	integer	id	id	integer	
age	age	age	integer	age	age	integer	
name	name	name	character varying	name	name	character varying	

Showing 1 to 5 entries of 5 < 1 > 10 / page ▾

Delete columns

The user can delete the existing columns of the reference and destination table if they are not required. Users can delete columns from the mapped, partially mapped, unmapped and previously mapped table sections as well. However, the deleted columns will be removed only from the Pelican UI and not from the original database.


8.1.3. Override unmapped columns




Pelican allows the user to override the unmapped column.

For example,

Suppose column ID_P and id_p exist on both the sides (Reference and Destination tables), logically these two columns are identical as both represent student identification numbers in the respective databases. However, while mapping the application won't map ID_P and id_p due to variation in name and mark then under the unmapped columns section. In such cases, the user can override the unmapped columns into mapped columns.

To override unmapped columns:

1. Locate unmapped column row on the screen and click the  symbol as shown below.

Please select Unique Key's for optimal performance.								
Mapped Tables <i>(No Data Found)</i>		Partially Mapped Tables <i>(No Data Found)</i>		Unmapped Tables <i>(1 Tables Found)</i>		Previously Mapped Tables <i>(No Data Found)</i>		
hive_table (Reference Table)		(Destination Table)		Configuration			Result	
Expression	Expression Datatype	Expression	Expression Datatype	Unique Key Column	Timestamp Column	Is Matching	Remove	Reset
bar	string			<input type="checkbox"/>	<input type="radio"/>	✗		
foo	int			<input type="checkbox"/>	<input type="radio"/>	✗		
ds	string			<input type="checkbox"/>	<input type="radio"/>	✗		

< 1 >

2. Click the **Save Mapping** button to save the changes.

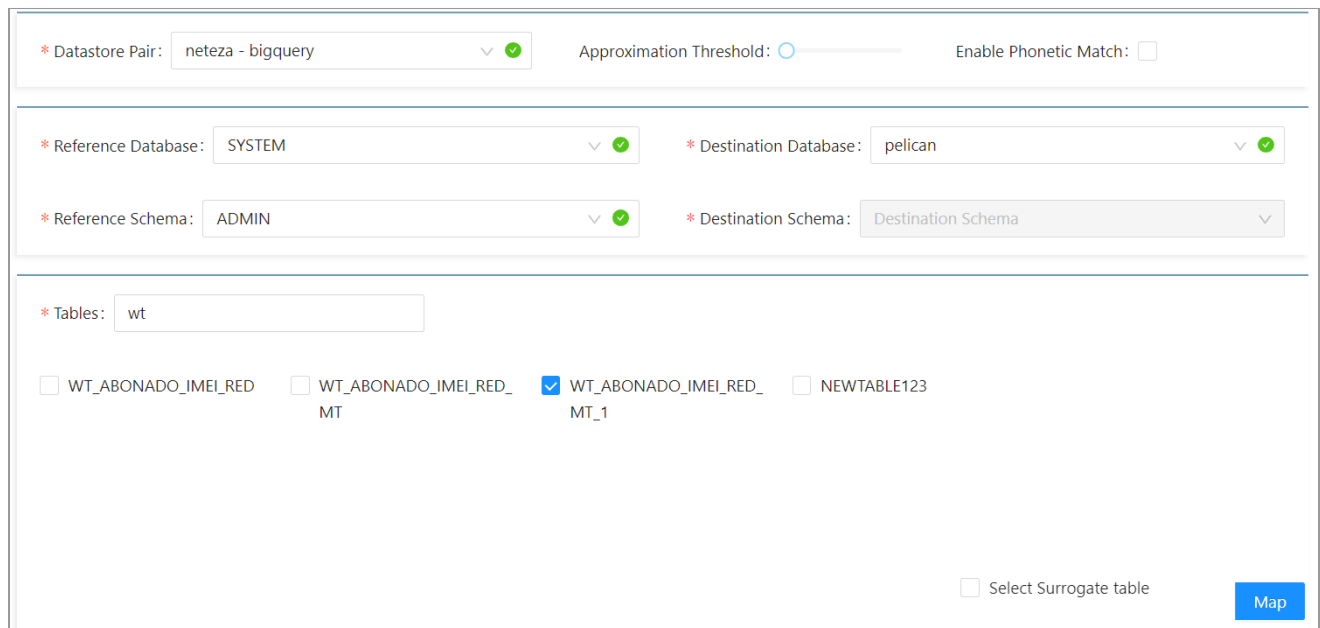
8.1.4. Surrogate Key

Pelican has the capability to validate any such tables where the unique key is an auto-generated value and corresponding business key exists in another table that can be used as unique (i.e, Surrogate Key). Therefore we are providing a way in Pelican to map a join of two tables on source to a join of two tables on target.

The steps to map tables which have surrogate key in another table:

1. Select **Validation Configuration** from the **Govern** tab.
2. Click on the '+' sign.
- This opens a page to **Map tables from source to target**.
3. Select a **Datastore** pair.
4. Select **Reference Database**.

This page allows you to select the actual table which is to be mapped, i.e. the table which is to be validated.



* Datastore Pair: neteza - bigquery ✓ Approximation Threshold: 0 Enable Phonetic Match:

* Reference Database: SYSTEM ✓ * Destination Database: pelican ✓

* Reference Schema: ADMIN ✓ * Destination Schema: Destination Schema

* Tables: wt

WT_ABONADO_IMEI_RED WT_ABONADO_IMEI_RED_MT WT_ABONADO_IMEI_RED_MT_1 NEWTABLE123

Select Surrogate table **Map**

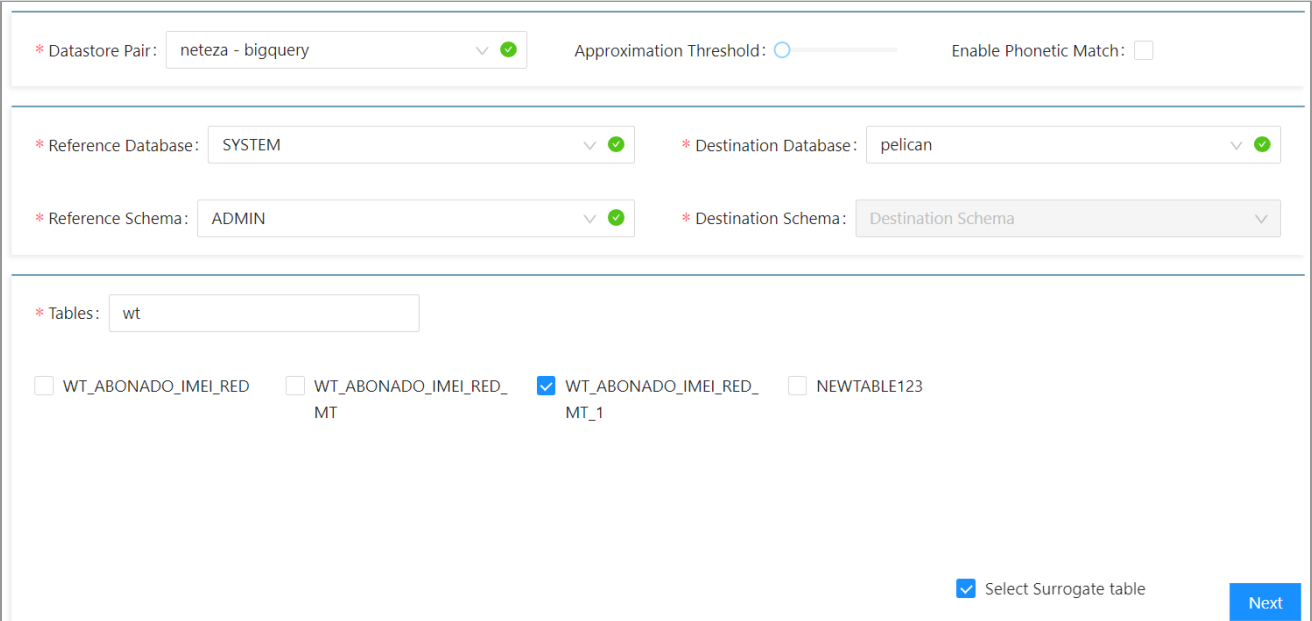
The tables associated with the database will be displayed on the screen.

Select the table to be validated. The table may or may not have a unique reference key.

On Selecting the table, the '**Select Surrogate Table**' checkbox will be displayed.

Note:

- You can select only one table at a time.
 - On selecting multiple tables, an error message will be displayed.
 - If no table is selected the '**Select Surrogate Table**' checkbox will not be visible.
5. Select Destination Database for Actual table, click on the checkbox - "Select surrogate table", then click **Next**.



* Datastore Pair: neteza - bigquery ✓ Approximation Threshold: 0 Enable Phonetic Match:

* Reference Database: SYSTEM ✓ * Destination Database: pelican ✓

* Reference Schema: ADMIN ✓ * Destination Schema: Destination Schema

* Tables: wt

WT_ABONADO_IMEI_RED WT_ABONADO_IMEI_RED_MT WT_ABONADO_IMEI_RED_MT_1 NEWTABLE123

Select Surrogate table Next

The below page will be displayed. This page is for selecting the table which has the surrogate key for the actual table.

6. Select **Reference Database**.

This page is for selecting the table which has the surrogate key for the actual table.

All the tables related to the database will be displayed.

7. Select the table which has the surrogate key.

8. Select the Destination database for surrogate table

9. Click **Map**.

* Datastore Pair: Approximation Threshold: Enable Phonetic Match:

Add Surrogate Table (*Surrogate table should be present on both source and target side)

* Reference Database: * Destination Database:

* Reference Schema: * Destination Schema:

* Tables:

WT_ABONADO_IMEI_RED
 WT_ABONADO_IMEI_RED_MT
 WT_ABONADO_IMEI_RED_MT_1
 NEWTABLE123

Back Map

The mapped tables page will be displayed.

We will see 2 mapped tables, First one is the Actual table (which is to be validated) and second one is the Surrogate table (which has the surrogate key)

Please select Unique Key's for optimal performance.								
Mapped Tables <i>(No Data Found)</i>	Partially Mapped Tables <i>(2 Tables Found)</i>	Unmapped Tables <i>(No Data Found)</i>	Previously Mapped Tables <i>(No Data Found)</i>					
WT_ABONADO_IMEI_RED_MT_1 <small>(Reference Table)</small>	WT_ABONADO_IMEI_RED_MT_1 <small>(Destination Table)</small>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th colspan="2">Configuration</th> <th rowspan="2">Result</th> </tr> <tr> <td style="text-align: center;"></td> <td style="text-align: center;"></td> </tr> </table>		Configuration		Result		
Configuration		Result						
*Surrogate Table								
WT_ABONADO_IMEI_RED_MT_1 <small>(Reference Table)</small>	WT_ABONADO_IMEI_RED_MT_1 <small>(Destination Table)</small>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th colspan="2">Configuration</th> <th rowspan="2">Result</th> </tr> <tr> <td style="text-align: center;"></td> <td style="text-align: center;"></td> </tr> </table>		Configuration		Result		
Configuration		Result						

Note:

Click on the corresponding **Configuration** button if you want to change or edit the configuration of any table.

8.1.5. Add Tags

Tagging is an optional feature associated with the table mappings. It can be used to have additional information for the mappings. user can set one or more tags on a table mapping.

Same tags can be used with multiple mappings. user can change tags any number of times from the Edit Mapping page.

Steps to Add tags

1. A user can add a tag to an existing mapping or a new mapping.



2. Add new Tags from the **Mapping page** by clicking the Tag icon

Please select Unique Key's for optimal performance.

Mapped Tables <i>(No Data Found)</i>	Partially Mapped Tables <i>(1 Tables Found)</i>	Unmapped Tables <i>(No Data Found)</i>
WT_ABONADO (Reference)		
Expression		Unique Column
W_ID		<input type="checkbox"/>
_DWH_DIA		<input type="checkbox"/>
_DWH_MES		<input type="checkbox"/>
_DWH_IME_RED		<input type="checkbox"/>
NUMERO_ABONADO	BIGINT	NUMERO_ABONADO INT64 <input type="checkbox"/>
FECHA_INSCRIPCION	VARCHAR	FECHA_INSCRIPCION STRING <input type="checkbox"/>

Tags
X

Table Name: WT_ABONADO_IMEI_RED_MT_1

Add/Edit Tags:

Please enter comma seperated tags!

3. Click **Ok**.

4. Click **Save**.

Note: Existing tags can be edited by following similar steps.

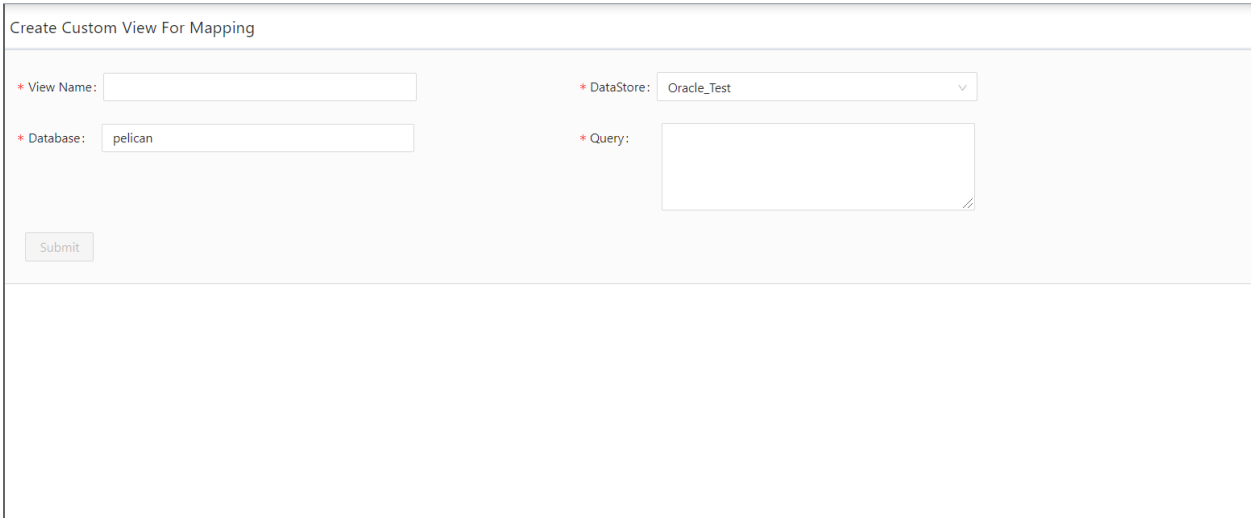
9. Custom Source and Target

This functionality provides the capability to create views from the User Interface, i.e. Pelican provides an option from the UI to generate a 'Create view' query in which multiple tables to be viewed can be specified. The View Creation feature is only visible to Admin/Super user.

Steps to create custom Views for mapping:

1. Select **View Creation** from the **Govern** menu.

The **Create Custom Views For Mapping** page will be displayed.



Create Custom View For Mapping

* View Name:

* Database:

* DataStore:

* Query:

2. Enter a unique **View Name**.
3. Select the **DataStore**.
4. Select **Database**.
5. Enter an appropriate **Query**.

6. Click **Submit**.

Create Custom View For Mapping

* View Name:	<input type="text" value="samplView_0001"/>	* DataStore:	<input type="text" value="Oracle_Test"/>
* Database:	<input type="text" value="pelican"/>	* Query:	<pre>select a.* b.subject_id,b.subject_name from pelican.courses a left join pelican.subjects b on a.course_id=b.course_id</pre>

The corresponding view will be generated in the database.

This newly created view can be used for mapping and validation. It can also be used as an alternative to the surrogate feature.

Note:

Rules and Guidelines for SQL Views:

Certain rules and guidelines apply when you create an SQL view.

- All tables in the SQL view must exist in the same database.
- The SQL statement must use valid SQL syntax based on the database of the tables included in the SQL view. Data Validation Option does not validate the SQL statement syntax.
- The number and order of the columns must match in the column definitions and the SQL statement of the SQL view.
- The scale must be zero for datetime, integer, and string data types.
- If you create a column in the SQL view, verify that the datatype, precision, and scale match the PowerCenter data type, precision, and scale shown in the Column Definitions area.
- You can add database functions, any join type, subqueries, and stored procedures in the SQL statement of an SQL view.
- If the SQL view calls a stored procedure, the connection assigned to SQL view must have Execute permission on the stored procedure.

-
- If you include the SQL view in a table pair, you can create a WHERE clause for the table pair based on a field in the SQL view.
 - When you use SQL view as the data source, you can edit the connection in the table pairs, single tables, aggregate views, and join views.

10. Scheduler Configuration

In Pelican, the user can create a scheduler for a saved mapping, so that, after a specific time period the scheduler executes the process and validates the source table with the destination table. Once the user completes the table mapping process the respective schedulers can be configured.

10.1. Modes in Scheduler Configuration

The Scheduler supports two modes of validation - Litmus and Full.

10.1.1. LITMUS mode

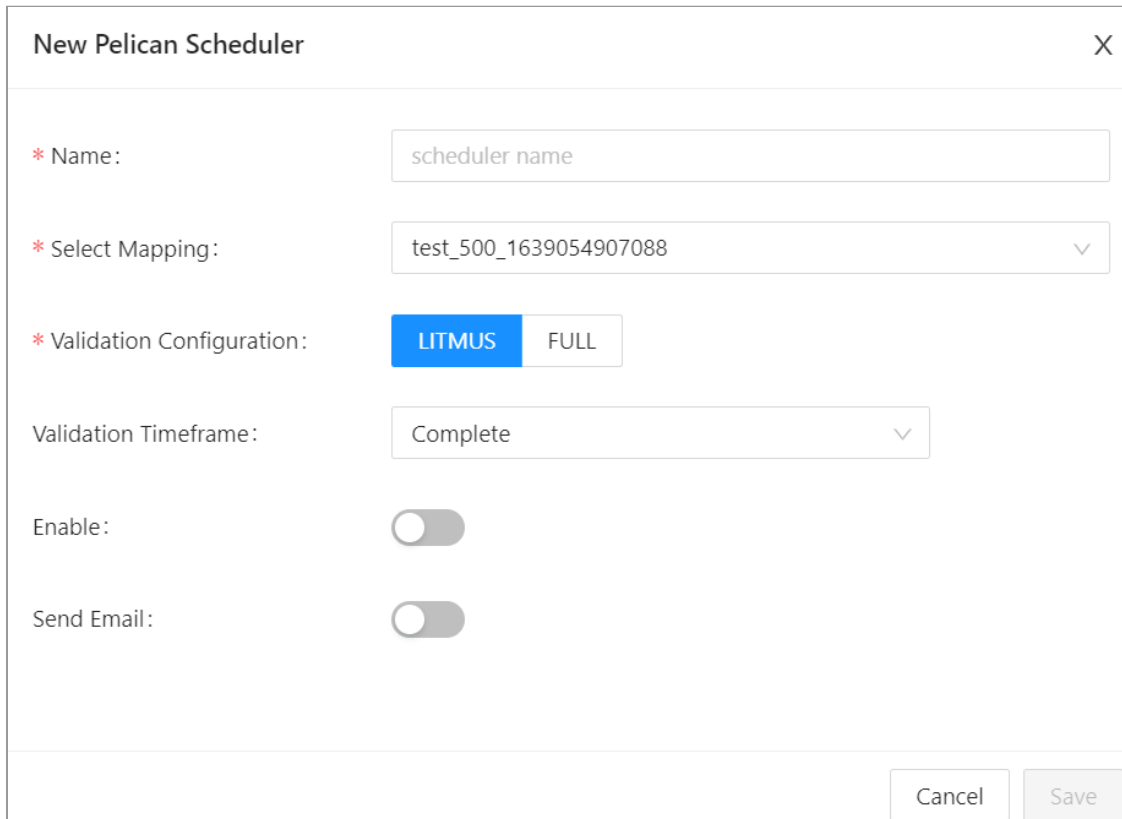
The Litmus mode analyzes whether the tables at source and target are matching or not. It does not support cell level differences or samples.

10.1.2. Full mode

Along with data validation as in Litmus mode, the Full mode displays the cell level differences through the sample rows fetched from both source and target tables.

10.2. Steps to configure the scheduler:

1. Navigate **Govern** -> **Validation Configuration**.
(Create **Mappings**-> then go to **Schedulers** -> **Create New scheduler**)



The screenshot shows a dialog box titled "New Pelican Scheduler" with a close button (X) in the top right corner. The dialog contains the following fields and controls:

- * Name:** A text input field containing "scheduler name".
- * Select Mapping:** A dropdown menu showing "test_500_1639054907088".
- * Validation Configuration:** Two radio buttons, "LITMUS" (selected) and "FULL".
- Validation Timeframe:** A dropdown menu showing "Complete".
- Enable:** A toggle switch that is currently turned off.
- Send Email:** A toggle switch that is currently turned off.

At the bottom right of the dialog, there are two buttons: "Cancel" and "Save".

2. Select Table Mapping from the hierarchy and click **View Scheduler** icon on the toolbar.
3. Click the **Add Pelican Scheduler** icon.
4. Enter Scheduler Name in the field provided and select table.
5. For validation the user can select either **LITMUS** or **FULL** mode as per requirement.

New Pelican Scheduler ✕

* Name:

* Select Mapping:

* Validation Configuration: LITMUS FULL

Validation Timeframe:

Enable:

Send Email:

6. Select **Validation Time Frame** from the drop-down and then click the **Enable** toggle button to set recurrence Pattern.
7. Select **Email** toggle button and enter email address if user wants reports to be sent for specific scheduler.

New Pelican Scheduler ✕

* Name :

* Select Mapping :

* Validation Configuration : LITMUS FULL

Validation Timeframe :

Enable :

Send Email :

Email :

Cancel
Save

8. Click **SAVE**.

9. Click **Execute Now** for on-demand execution of any of the schedulers.

The newly created scheduler is added to the **Schedulers** screen as shown below.

Sr.No	Scheduler Name	Scheduler Running Time	Scheduling Status	Status	Start Time / End Time	Result	Configure	Exec ute Now	Delet e
1	Demo		Disabled	Completed	Start Time :- End Time :-	▼	⏻ Configure	▶	🗑


Showing 1 to 1 entries of 1

1
>
10 / page

Note:

The following scheduler report email will be sent to the email address mentioned while creating the scheduler:

From: <pelican_testing@datametica.in>
 Date: Wed, Jan 19, 2022 at 2:28 PM
 Subject: Pelican Scheduler Execution Report nz_quotes
 To: <test.mail@gmail.com>



Sr. No	Scheduler Name	Table Mapping Name	Source DataStore	Source Name	Target DataStore	Target Name	Mismatch Count	MissingRowsCount	Mismatch Rows Count	Extra Rows Count	Execution Time	Status
1	nz_quotes	WOMCASTISSUE_1641905565072	Netezza_keshav	"TESTDB"."DMUSER".WOMCASTISSUE	BQ	pelican.womcastissue	2	0	1	1	2022-01-19 14:26:28.617	false

10.3 Scheduler Report

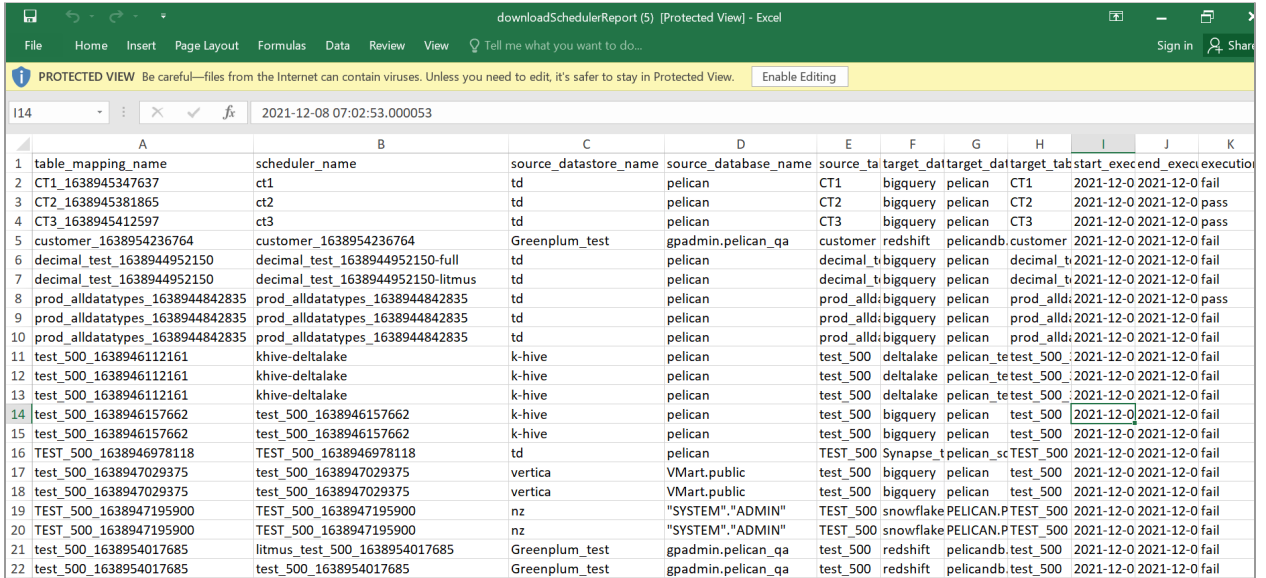
- The Scheduler Status Report functionality enables the user to view all the iterations of all the schedulers of the given mapping in the Pelican application.
- To View the Scheduler report click on the **Scheduler Status Report** hyperlink located on the top of the Schedulers page. An .xls file will be downloaded at the bottom of the screen.

Schedulers [Scheduler Status Report](#) Search for Scheduler Na...

Sr.No.	Scheduler Name	Scheduler Running Time	Scheduling Status	Status	Start Time / End Time	Configure	Execute Now	Delete
1	prod_alldatatypes_1638944842835		Disabled	Completed	Start Time :- 08-12-2021 12:53:16 End Time :- 08-12-2021 12:53:45	<input type="button" value="Configure"/>	<input type="button" value="Execute Now"/>	<input type="button" value="Delete"/>
2	decimal_test_1638944952150-fuII		Disabled	Completed	Start Time :- 08-12-2021 11:59:28 End Time :- 08-12-2021 11:59:52	<input type="button" value="Configure"/>	<input type="button" value="Execute Now"/>	<input type="button" value="Delete"/>
3	decimal_test_1638944952150-litmus		Disabled	Completed	Start Time :- 08-12-2021 12:05:09 End Time :- 08-12-2021 12:05:12	<input type="button" value="Configure"/>	<input type="button" value="Execute Now"/>	<input type="button" value="Delete"/>
4	TEST_DATA_1000_1638945130505		Disabled	Completed	Start Time :- 08-12-2021 12:54:23 End Time :- 08-12-2021 12:54:57	<input type="button" value="Configure"/>	<input type="button" value="Execute Now"/>	<input type="button" value="Delete"/>

Showing 1 to 10 entries of 20

- This scheduler report file contains the scheduler and mapping details in columns such as mapping name, scheduler name, source and destination details, etc.



1	A	B	C	D	E	F	G	H	I	J	K
1	table_mapping_name	scheduler_name	source_datastore_name	source_database_name	source_target	target_data	target_database	target_table	start_exec	end_exec	execution
2	CT1_1638945347637	ct1	td	pelican	CT1	bigquery	pelican	CT1	2021-12-0	2021-12-0	fail
3	CT2_1638945381865	ct2	td	pelican	CT2	bigquery	pelican	CT2	2021-12-0	2021-12-0	pass
4	CT3_1638945412597	ct3	td	pelican	CT3	bigquery	pelican	CT3	2021-12-0	2021-12-0	pass
5	customer_1638954236764	customer_1638954236764	Greenplum_test	gpadm.pelican_qa	customer	redshift	pelican	customer	2021-12-0	2021-12-0	fail
6	decimal_test_1638944952150	decimal_test_1638944952150-full	td	pelican	decimal_t	bigquery	pelican	decimal_t	2021-12-0	2021-12-0	fail
7	decimal_test_1638944952150	decimal_test_1638944952150-litmus	td	pelican	decimal_t	bigquery	pelican	decimal_t	2021-12-0	2021-12-0	fail
8	prod_alldatatypes_1638944842835	prod_alldatatypes_1638944842835	td	pelican	prod_all	bigquery	pelican	prod_all	2021-12-0	2021-12-0	pass
9	prod_alldatatypes_1638944842835	prod_alldatatypes_1638944842835	td	pelican	prod_all	bigquery	pelican	prod_all	2021-12-0	2021-12-0	fail
10	prod_alldatatypes_1638944842835	prod_alldatatypes_1638944842835	td	pelican	prod_all	bigquery	pelican	prod_all	2021-12-0	2021-12-0	fail
11	test_500_1638946112161	khive-deltalake	k-hive	pelican	test_500	deltalake	pelican	test_500	2021-12-0	2021-12-0	fail
12	test_500_1638946112161	khive-deltalake	k-hive	pelican	test_500	deltalake	pelican	test_500	2021-12-0	2021-12-0	fail
13	test_500_1638946112161	khive-deltalake	k-hive	pelican	test_500	deltalake	pelican	test_500	2021-12-0	2021-12-0	fail
14	test_500_1638946157662	test_500_1638946157662	k-hive	pelican	test_500	bigquery	pelican	test_500	2021-12-0	2021-12-0	fail
15	test_500_1638946157662	test_500_1638946157662	k-hive	pelican	test_500	bigquery	pelican	test_500	2021-12-0	2021-12-0	fail
16	TEST_500_1638946978118	TEST_500_1638946978118	td	pelican	TEST_500	Synapse_t	pelican_sc	TEST_500	2021-12-0	2021-12-0	fail
17	test_500_1638947029375	test_500_1638947029375	vertica	VMart.public	test_500	bigquery	pelican	test_500	2021-12-0	2021-12-0	fail
18	test_500_1638947029375	test_500_1638947029375	vertica	VMart.public	test_500	bigquery	pelican	test_500	2021-12-0	2021-12-0	fail
19	TEST_500_1638947195900	TEST_500_1638947195900	nz	"SYSTEM"."ADMIN"	TEST_500	snowflake	PELICAN.P	TEST_500	2021-12-0	2021-12-0	fail
20	TEST_500_1638947195900	TEST_500_1638947195900	nz	"SYSTEM"."ADMIN"	TEST_500	snowflake	PELICAN.P	TEST_500	2021-12-0	2021-12-0	fail
21	test_500_1638954017685	litmus_test_500_1638954017685	Greenplum_test	gpadm.pelican_qa	test_500	redshift	pelican	test_500	2021-12-0	2021-12-0	fail
22	test_500_1638954017685	test_500_1638954017685	Greenplum_test	gpadm.pelican_qa	test_500	redshift	pelican	test_500	2021-12-0	2021-12-0	fail

11. Validation Result

After scheduler execution, the application automatically generates the validation result in the form of a report.

11.1. Results from Report page

Steps to view the validation result:

1. Go to **Reports**.
2. Click on **Validation Result**.

The application displays the Validation Result screen as shown below.

Validation Result								
Sr.No.	Source Table Name	Mapping Name	Statistics	Schedulers	Table History Results(Last 10 Result)	Results	Sample	Lineage
1	SAMPLE.DB2ADMIN . TEST_DATA_1000	TEST_DATA_1000_1638447832895			● ● ● ● ● ● ● ● ● ●	✗ ⓘ		
2	gpadmin.pelican_qa . test_504	test_504_1638856756080			● ● ● ●	✗ ⓘ	N/A	
3	gpadmin.pelican_qa . test_500	Greenplum-Redshift_test_500			● ● ● ● ● ● ● ● ● ●	✗ ⓘ	N/A	
4	gpadmin.pelican_qa . view_test	view_test_1638857124916			● ●	✗ ⓘ	N/A	
5	gpadmin.pelican_qa . view_sample_test	view_sample_test_1638857676517			● ●	✗ ⓘ	N/A	
6	gpadmin.pelican_qa . test_503	test_503_1638856709265			●	✗ ⓘ		
7	gpadmin.pelican_qa . test_502	test_502_1638856659370			●	✗ ⓘ		
8	gpadmin.pelican_qa . test_501	test_501_1638856554144			●	✗ ⓘ		

Showing 1 to 34 entries of 34

The Validation Result screen displays the following information.

- Source Table Name** : This column displays the list of source tables for which mapping has been executed in the system.
- Mapping Name**: This column displays the mapping which has been executed for the particular table.
- Statistics**: This column displays the scheduler history of the respective table mapping.


Click the **View Mapping Historical Results**  icon to view detailed history. This displays the detailed history of the respective scheduler as shown below.

Table Name: pelican . demo_customers2 , Mapping Name: demo_customers2_1638445949214													
Sr. No	Scheduler Name	Result	Sample	Start Time	End Time	Source Rows Count	Target Rows Count	Cell Mismatch Count	Extra Rows	Missing Rows	Mismatched Rows	Approve	Exceptions
1	demo_customers2_full	✗		07-12-2021 10:56:5	07-12-2021 10:56:40	320	319	1	0	1	2		
2	demo_customers2_full	✗		07-12-2021 10:53:8	07-12-2021 10:54:4	319	319	1	0	0	1		
3	demo_customers2_full	✓		02-12-2021 17:53:0	02-12-2021 17:54:13	319	319	N/A	N/A	N/A	0		

The scheduler history page displays the detailed information of the scheduler, such as:

- Scheduler name
- Result
- Sample
- Start time
- End time
- Source Rows Count
- Target Rows Count
- Cell Mismatch Count
- Extra Rows Count
- Missing Rows Count
- Total Mismatch RowsCount
- Approve
- Exceptions

Among these following are the some of the major features: -

Approve

- It is a feature to approve a mapping if a mapping fails.
- This feature has been added for the cases where the cause of failure is a known issue and can be considered as an exceptional case.
- As shown below we can approve a mapping by clicking on the icon in the approve column and click save to approve a mapping.

Table Name: pelican . demo_customers2 , Mapping Name: demo_customers2_1638445949214 < Back

Sr No	Scheduler Name	Result	Sample	Start Time	End Time	Source Rows Count	Target Rows Count	Cell Mismatch Count	Extra Rows	Missing Rows	Mismatched Rows	Approve	Exceptions
1	demo_customers2_full	✗	☰	07-12-2025					0	1	2	✍	☰
2	demo_customers2_full	✗	☰	07-12-2028					0	0	1	✍	☰
3	demo_customers2_full	✓	☰	02-12-2020					N/A	N/A	0	✍	✍

demo_customers2_1638445949214 ✕

* Approval Reason:

Save

- If the result is either approved or passed, the icon to open approve a mapping modal will be disabled.


Exceptions



- If a scheduler failed due to an exception, then log will display its exception stack trace. Click on the logs icon to view exception logs.

Mismatched Rows	Approve	Exceptions
2	✍	☰
1	✍	☰
0	✍	✍





- Once we click the icon on the logs column, the new pop-up window will open for the logs.
- If Scheduler runs without Exception, then the Log option will be disabled.

4. **Schedulers:** Click on the  icon to view scheduler details.

 DASHBOARD GOVERN **REPORTS** ADMINISTRATION

 0
  Super User
 [Logout](#)

Validation Result

Sr. No.	Source Table Name	Mapping Name	Statistics	Schedulers	Table Historical Results (Last 10 Result)	Results	Sample
1	SAMPLE.DB2ADMIN . TEST_10K				● ● ● ● ● ● ● ● ● ●	✓	N/A
2	qauser . Test_500				● ● ● ● ● ● ● ● ● ●	✗ ⓘ	
3	SAMPLE.DB2ADMIN . TEST_500				● ● ● ● ● ● ● ● ● ●	✗ ⓘ	
4	qauser . test_rmg				● ● ● ● ● ● ● ● ● ●	✗ ⓘ	
5	qauser . test_rmg2				● ● ● ● ● ● ● ● ● ●	✗ ⓘ	
6	SAMPLE.DB2ADMIN . TEST_1K				● ● ● ● ● ● ● ● ● ●	✓	N/A
7	SAMPLE.PELICAN . ALLMATCHES_1K				● ● ● ● ● ● ● ● ● ●	✗ ⓘ	N/A
8	SCOTT . ALL_MATCHES_1K				● ● ● ● ● ● ● ● ● ●	✗ ⓘ	N/A

Showing 1 to 10 entries of 19

[Previous](#)
[1](#)
[2](#)
[Next](#)




Copyright All Rights Reserved ©2020 | DataMetica Solutions Pvt. Ltd.

 Google Hangouts is sharing a Chrome tab and audio with meet.google.com.
 [Stop sharing](#)
[Hide](#)

The screen will pop up and will show information and properties of the scheduler such as scheduler name, scheduler running time, on schedule, starting and ending time of last execution of scheduler, execution option of scheduler using Execute Now button and status of the execution.


5. **Table Historical Result:** This column displays the execution history of the schedulers of the respective mapping. This information is represented by **Green**, **Red** and **Blue** circles. The **Green** circle indicates the **Success**, the **Red** circle indicates the **Failure**, the **Blue** circle indicates the **Approve**. The user can hover mouse over these circles to view execution start date and end date of the respective scheduler as shown below.

6. **Results:** This column displays the data validation result. The Result is represented by two flags, namely:

- Success 
- Failure  

Hover mouse over icon to know the reason for failure.

7. **Sample:** This column displays the sample of mismatch data in case the total

mismatch count is greater than zero. Click the  symbol to view samples of mismatched data. This displays the **Sample Data** screen as shown below:

Sample Data
< Back

● Row missing
 ● Cell mismatch
 ● Cell match
 ● Row duplicate

Show Match Column: Split Horizontal:

Sample from Source Table.

id <small>UID</small>	OREPLACE(CAST(...	CAST(CAST(join_...	CAST(CAST(join_...	
109	66000.120	2012-03-08	2012-11-21 09:36:13.000	LC


Sample from Target Table.

id <small>UID</small>	FORMAT("%.*f",3...)	FORMAT_DATE("...	FORMAT_TIMEST...	
109	55000.120	2013-03-08	2013-11-21 09:36:13.000	JC

The Sample Data report displays sample mismatch data from source and target table with different colours. This report also provides the following filters which helps the user to view the desired information:

- Hide/Show Match Column
- Split Horizontal

Hide/Show Match Column: The Hide/Show Match Column is a toggle button which shows two States ON and OFF.



When the **Hide/Show Match Column** toggle button is set to **OFF**, then the application displays both matched and unmatched columns as shown below.

Copyright © 2022 by Datametica Solutions Pvt. Ltd. All Rights Reserved.

69

Sample Data [← Back](#)

● Row missing
 ● Cell mismatch
 ● Cell match
 ● Row duplicate

Show Match Column:
 Split Horizontal:

Sample from Source Table.

id <small>UID</small>	zip
76	"99212"
94	"99212"
206	"99212"
226	"99212"
238	"99212"
374	"99212"
257	"99212"
307	"99212"

Sample from Target Table.

id <small>UID</small>	zip
76	99212
94	99212
206	99212
226	99212
238	99212
374	99212
257	99212
307	99212

When the **Hide/Show Match Column** toggle button is set to **ON**, then the application displays only unmatched columns as shown below.

Sample Data [← Back](#)

● Row missing
 ● Cell mismatch
 ● Cell match
 ● Row duplicate

Show Match Column:
 Split Horizontal:

Sample from Source Table.

id <small>UID</small>	zip	
76	"99212"	Kanpur
94	"99212"	Kanpur
206	"99212"	Kanpur
226	"99212"	Kanpur
238	"99212"	Kanpur
374	"99212"	Kanpur
257	"99212"	Kanpur
307	"99212"	Kanpur

Sample from Target Table.

id <small>UID</small>	zip	
76	99212	Los Angeles
94	99212	Los Angeles
206	99212	Los Angeles
226	99212	Los Angeles
238	99212	Los Angeles
374	99212	Los Angeles
257	99212	Los Angeles
307	99212	Los Angeles

Split Horizontal: Set **Split Horizontal** toggle button to **ON** to view mismatched data in the horizontal format, for more information, refer below image.

Sample Data [Back](#)

● Row missing
 ● Cell mismatch
 ● Cell match
 ● Row duplicate

Show Match Column: Split Horizontal:

Sample from Source Table.			
id <small>UID</small>	zip	city	name
76	"99212"	Kanpur	Divyanshu
94	"99212"	Kanpur	Divyanshu
206	"99212"	Kanpur	Divyanshu
226	"99212"	Kanpur	Divyanshu
238	"99212"	Kanpur	Divyanshu
374	"99212"	Kanpur	Divyanshu
257	"99212"	Kanpur	Divyanshu
307	"99212"	Kanpur	Divyanshu
317	"99212"	Kanpur	Divyanshu


Sample from Target Table

11.2. Results from Scheduler page

Here, the user can view validation results for the saved mapping.

To view validation result:

1. Go to the **Validation Configuration** page and select the saved mapping for which


you want to see the validation result and then click on the view scheduler icon 







.The following screen will be displayed.



Sr.No.	Scheduler Name	Scheduler Running Time	Scheduling Status	Status	Start Time / End Time	Result	Configure	Execute Now	Delete
1	TEST_DATA_1000_1638946631934		Disabled	Completed	Start Time :- 08-12-2021 12:53:45 End Time :- 08-12-2021 12:54:20		Configure	Execute Now	Delete

Showing 1 to 1 entries of 1



2. Click on the icon  to see the statistics of the validation result page for the mapping.

TEST_DATA_1000_1638946631934													< Back	
Sr. No.	Scheduler Name	Result	Sample	Start Time	End Time	Source Rows Count	Target Rows Count	Cell Mismatch Count	Extra Rows	Missing Rows	Mismatched Rows	Approve	Exceptions	
1	TEST_DATA_1000_1638946631934	✘		08-12-2021 12:53:45	08-12-2021 12:54:20	1001	1003	2	3	1	6			
2	TEST_DATA_1000_1638946631934	✘		08-12-2021 12:27:25	08-12-2021 12:27:55	1001	1003	2	3	1	6			


1


The fields (columns of the validation result page are the same as explained above).

12. License - Buy Your Own license (BYOL)

- There are two types of licenses by which you can access Pelican:
 1. **License for Production environments** - Allows validation of all the records in a table.
 2. **License for Non production environments:** This license will be used for all the lower level environments. It will support all the Pelican functionalities but with limited capacity. Allows validation of tables having upto 1000 records. Tables with more than 1000 records will not be validated.
- Buy Your Own license (BYOL) notifies you when your license will expire. Pelican has introduced a new feature, i.e Add licence. This new feature notifies you, the date of the renewal.
- To get a new License contact : pelican.support@datametica.atlassian.net.

12.1. Adding /updating license

Once you received the pelican license file, i.e. licence.pel follow the following steps:

1. Log in to **Pelican**.
2. Administration -> **Add License**.


Add License

Upload License document here

↓ Choose license to upload

Email Preferences

✉ Enter email id/s to add

Email	Remove
 No Data	

Uploaded Licenses

Datstore Type	Instance Limit	Expiration
db2	10	Expires in 347 days

3. Click on the **Choose license to upload** button and select the license from the file system.

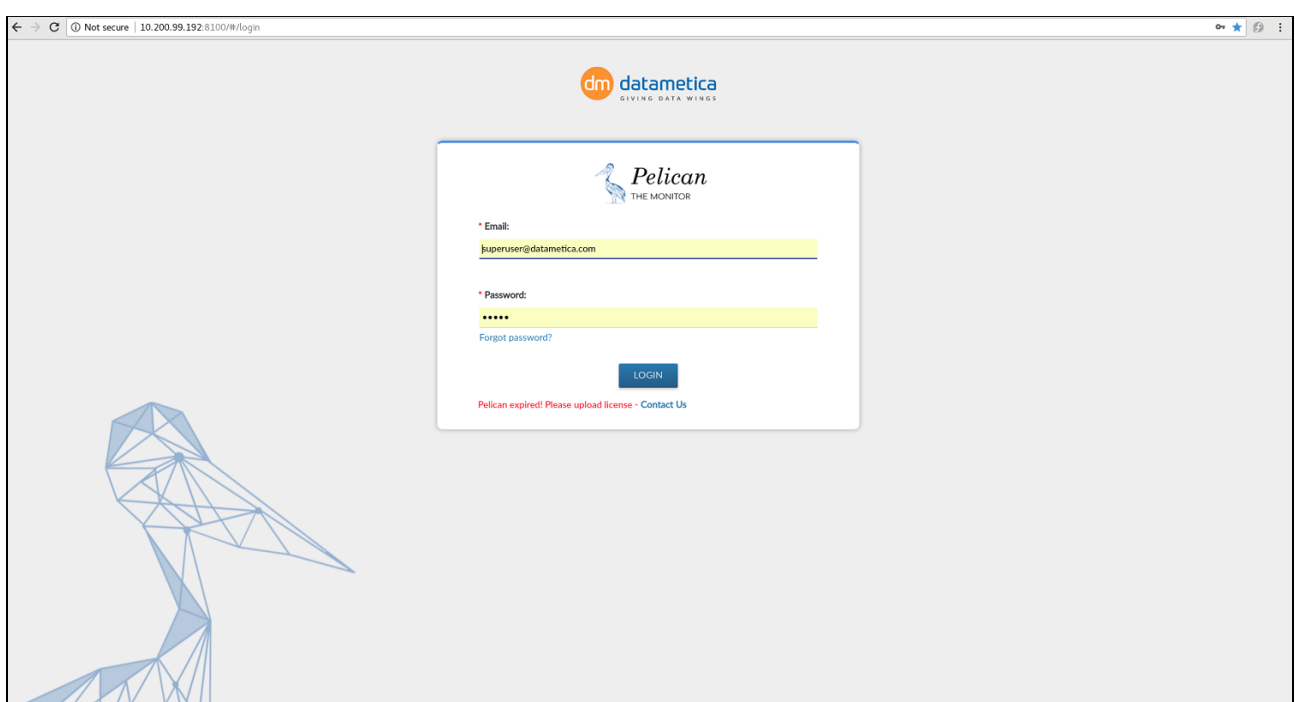
A license Preview pop-up will be displayed.

License Preview X		
Datastore Type	Instance Limit	Expiration(mm/dd/yyyy)
teradata	10	11/20/2022
netezza	10	11/20/2022
oracle	10	11/20/2022
db2	10	11/20/2022
ms_sql_server	10	11/20/2022
hive	10	11/20/2022
vertica	10	11/20/2022

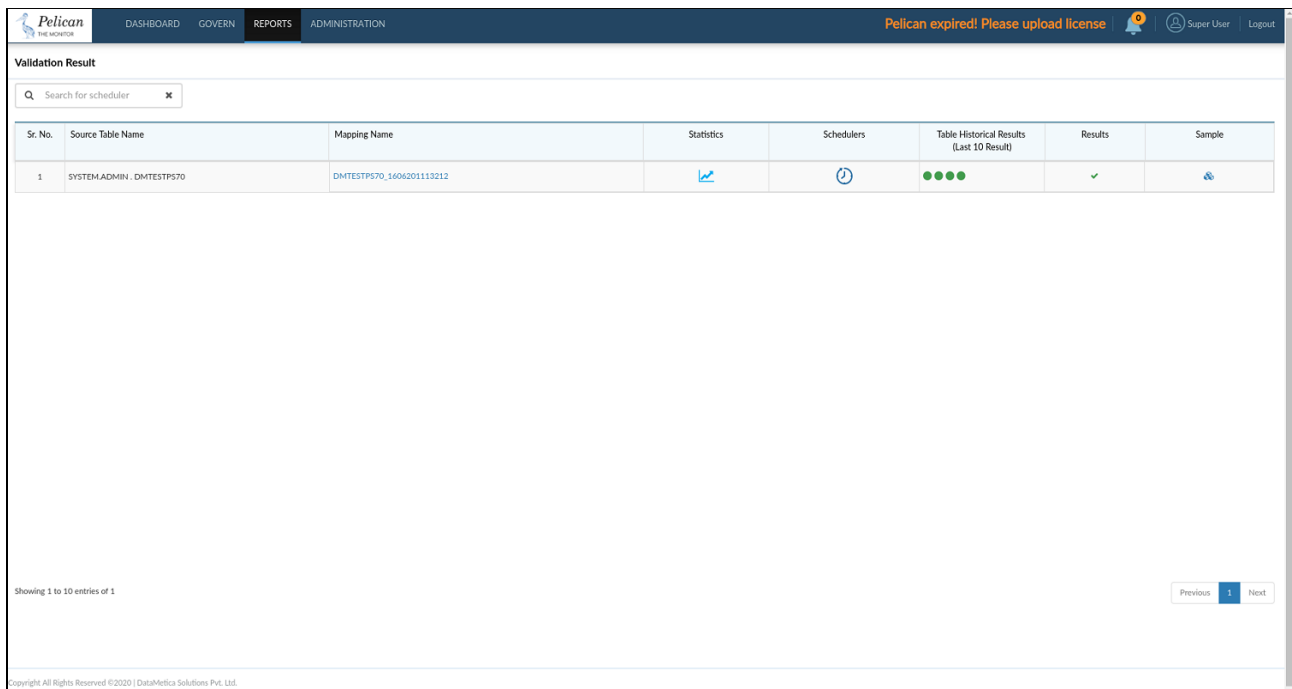
- You will be taken back to the Login Page.
Login back to continue the usage.

Note: You will be notified in the following three ways regarding the license renewal:

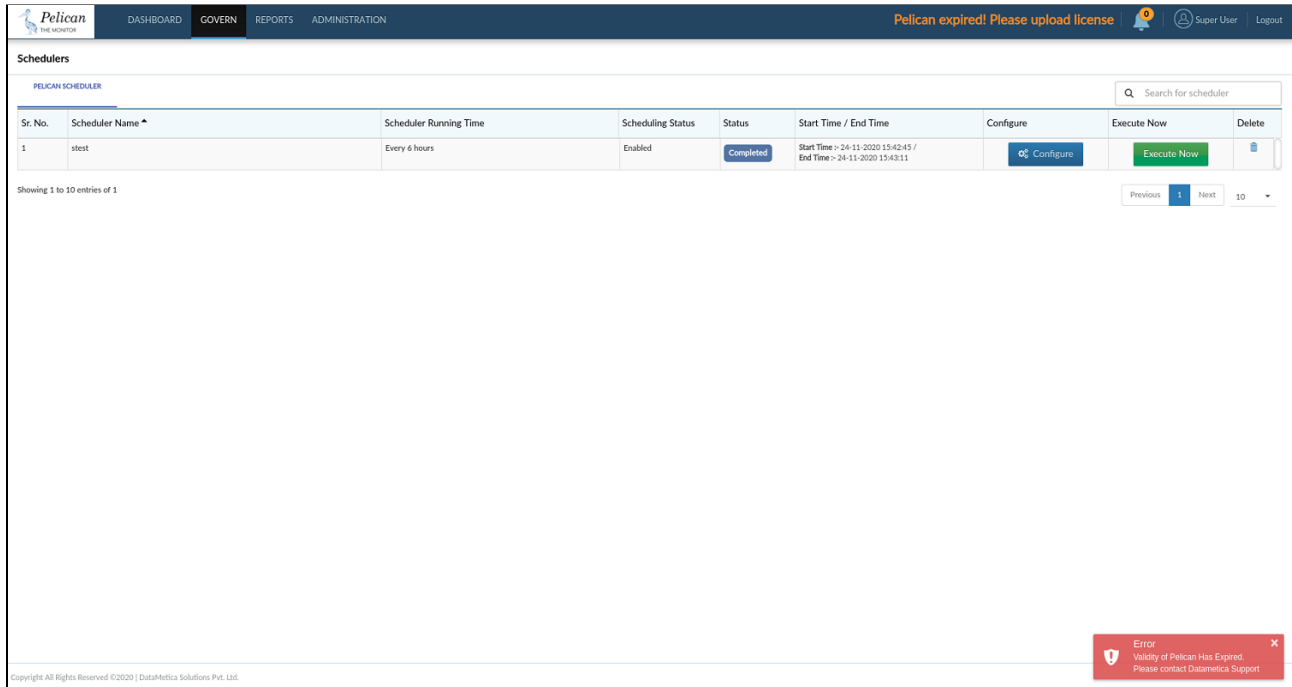
- Detailed email is sent to your registered email address prior to 15 days of renewal , mentioning your product ID and date of renewal. All the registered email addresses will receive the email of notification. The supported mails (SMTP providers) are Gmail, Microsoft(Office 365) and Yahoo.
- A message is displayed on the login screen, below the login tab,
Eg. Pelican validity has expired. Contact Datametica support, Contact Us.



- On the main screen, on the menu bar,
Eg. Pelican validity has expired. Contact Datametica support, Contact Us.



Once your license expires, you are able to create the mappings, but cannot run the schedulers. To continue, you need to contact Datametica.



The screenshot shows the 'SCHEDULERS' page in the Pelican interface. The top navigation bar includes 'DASHBOARD', 'GOVERN', 'REPORTS', and 'ADMINISTRATION'. A notification at the top right states 'Pelican expired! Please upload license'. The main content area features a table with the following data:

Sr. No.	Scheduler Name	Scheduler Running Time	Scheduling Status	Status	Start Time / End Time	Configure	Execute Now	Delete
1	stest	Every 6 hours	Enabled	Completed	Start Time -> 24-11-2020 15:42:45 / End Time -> 24-11-2020 15:43:11	Configure	Execute Now	

Below the table, it says 'Showing 1 to 10 entries of 1'. At the bottom right, there is a red error message: 'Error: Validity of Pelican Has Expired. Please contact Datametica Support'.

Once you renew the licence, all the functionalities will be resumed.

Now, you can run schedulers.

13. Lineage

The Lineage feature enables the user to view the last executions of source tables and display them graphically. Therefore, it becomes easy for the tester / developer to backtrack and identify the table where the issue started. This results in quicker root cause analysis.

The benefits of this functionality is as follows:

1. Finding the root-cause of the error.
2. Provides metadata information, assisting to fixing issues if any gap is identified in the data processing pipeline.
3. Graphic visualisation of the data flow.

13.1. Add Lineage

Add Lineage is the process of adding a CSV file, which contains two columns (FromTable, ToTable) with their qualified names in the format.

database_name.table_name or database_name.schema_name.table_name

Example:

CSV file contain following details:

FromTable ToTable

Pelican.customer pelican.customer_salesman_order_details

Pelican.salesman pelican.customer_salesman_order_details

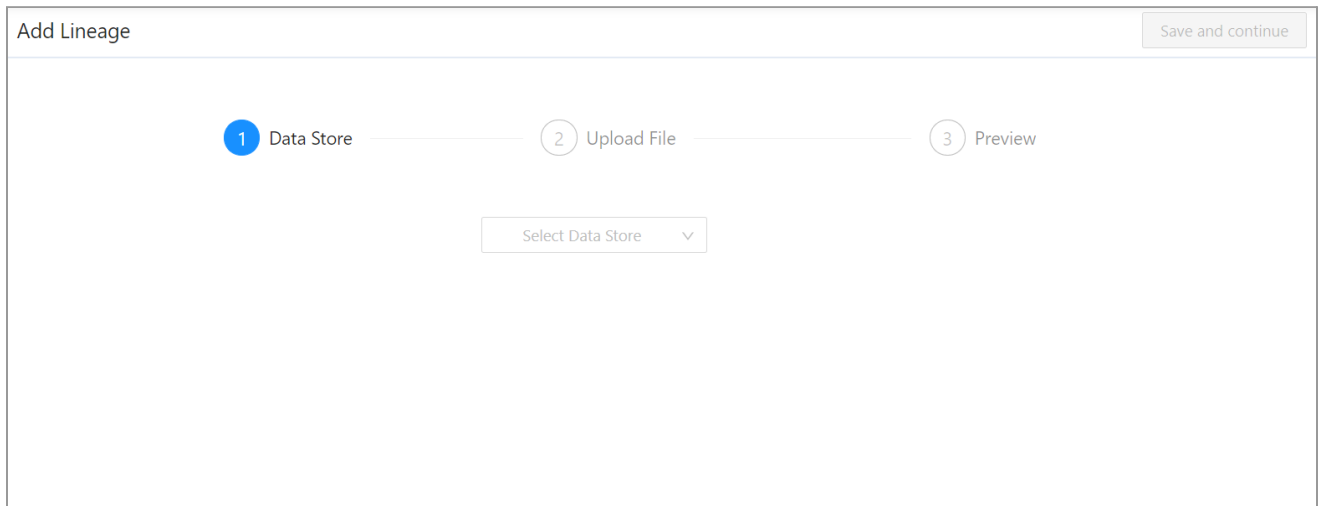
Pelican.orders pelican.customer_salesman_order_details

How to Add lineage?

Only Admin or superuser has access to the add Lineage Or upload CSV file.

To steps to add Lineage are as follows:

1. Go to **Administration** → **Add Lineage**.



2. Select the **Datastore** and click on **Preview Uploaded File**.

1 bigquery ————— 2 Upload File ————— 3 Preview

bigquery ▼

3. The previously uploaded table will be displayed on the screen.

Add Lineage Preview uploaded file

1 td ————— 2 Upload File ————— 3 Preview

td ▼

Add Lineage Previous

✓ td —————
 ✓ Upload File —————
 3 Preview

Please run the lineage scheduler


From Table	To Table
pelican.CT1	pelican.CT2
pelican.CT2	pelican.CT3

Verify and click on the **Preview** button if you want to add a table to the existing tables.

You can **drag and drop** a CSV file or click on **Upload file** link.

Add Lineage Previous Preview uploaded file

1 td — 2 Upload File — 3 Preview


Click or drag file to this area to upload
Upload file for td.

From Table	To Table
pelican.CT1	pelican.CT2
pelican.CT2	pelican.CT3

4. Select a lineage file with .csv extension and click **open**.

- The add lineage file preview will be displayed. If you want to remove the file click on **delete icon** and repeat the process with the new file. And click on **Upload**.
- When the file is successfully uploaded, you can see the preview.

13.2. How to View Lineage?

To view the lineage status, follow the steps mentioned below

- Go to the **Reports** → **Validation Result** → **Lineage** (can be viewed on each mapping result)
- Click on the **Lineage** record.

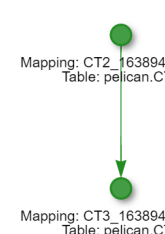
Validation Result								
Search for Source Table Name <input type="text"/>								
Sr.No.	Source Table Name	Mapping Name	Statistics	Schedulers	Table History Results(Last 10 Result)	Results	Sample	Lineage
1	gpadmin.pelican_qa . customer	customer_1638954236764			●	✗		
2	gpadmin.pelican_qa . test_500	test_500_1638954017685			● ●	✗	N/A	
3	pelican . test_500	test_500_1638946157662			● ●	✗		
		test_500_1638946112161			● ● ●	✗		
4	test_db.dbo . Test_DATA_1000	Test_DATA_1000_1638946322135			● ●	✗		

- Lineage results will be displayed in the pictorial format.

Table Name: pelican . CT3 , Mapping Name: CT3_1638945412597 [Back](#)

Showing Lineage

● Single click on node to see the result history
● Double click on node to see it's parent lineage
Fail: ● | Pass: ●



```




graph TD
    A["Mapping: CT2_1638945381885  
Table: pelican.CT2"] --> B["Mapping: CT3_1638945412597  
Table: pelican.CT3"]
            
```

Copyright All Rights Reserved ©2021 | DataMetica Solutions Pvt. Ltd.

Note: Single click on the node to see the result history, while double clicking on the node will take you to it's parent lineage.

Single Click Result

Table Name: pelican.CT2 , Mapping Name: Mapping: CT2_1638945381865 Table: pelican.CT2 [Back](#)

Sr. No.	Scheduler Name	Result	Sample	Start Time	End Time	Source Rows Count	Target Rows Count	Cell Mismatch Count	Extra Rows	Missing Rows	Mismatched Rows	Approve	Exceptions
1	ct2	✓		08-12-2021 12:06:30	08-12-2021 12:06:52	1	1	N/A	N/A	N/A	0		

< 1 >

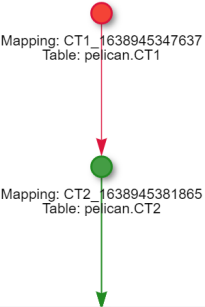
Double click Result

Table Name: pelican . CT3 , Mapping Name: CT3_1638945412597 [Back](#)

Showing Lineage

- Single click on node to see the result history
- Double click on node to see it's parent lineage

Fail: ● | Pass: ●



```

graph TD
    A["Mapping: CT1_1638945347637  
Table: pelican.CT1"] -- red --> B["Mapping: CT2_1638945381865  
Table: pelican.CT2"]
    B -- green --> C[ ]
    style C fill:none,stroke:none
  
```

Copyright All Rights Reserved ©2021 | DataMetica Solutions Pvt. Ltd.

14. API

14.1 Export/Import API

Step 1

Url=<http://10.200.104.115:8080/oauth/token>

Method=POST

Header

Content-Type application/x-www-form-urlencoded

Body

username superuser@datametica.com

password admin

grant_type password

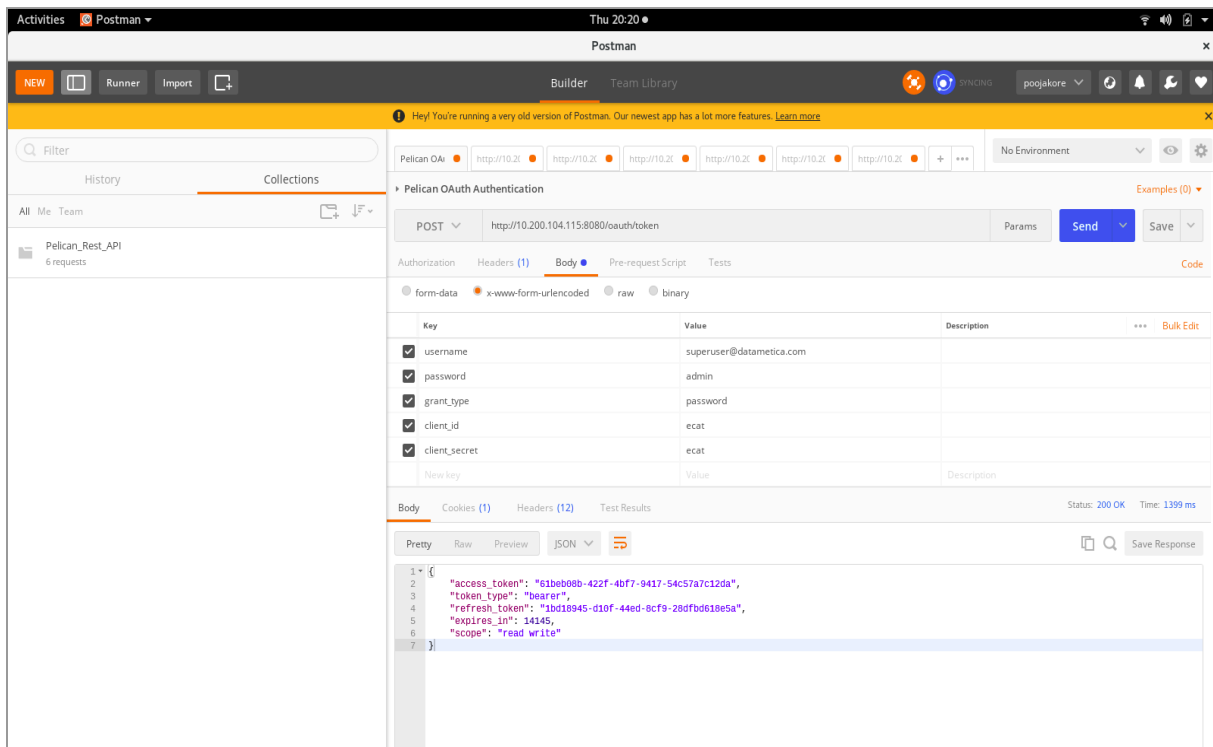
client_id ecat

client_secret ecat

Expected Result [Example]

```
{
  "access_token": "61beb08b-422f-4bf7-9417-54c57a7c12da",
  "token_type": "bearer",
  "refresh_token": "1bd18945-d10f-44ed-8cf9-28dfbd618e5a",
  "expires_in": 14145,
  "scope": "read write"
}
```

Screenshot for reference



Step 2

Uri=<http://10.200.104.115:8080/app/rest/pelican/tablemappings>

Method=GET

Header

Authorization bearer {access_token generated in step 1 }

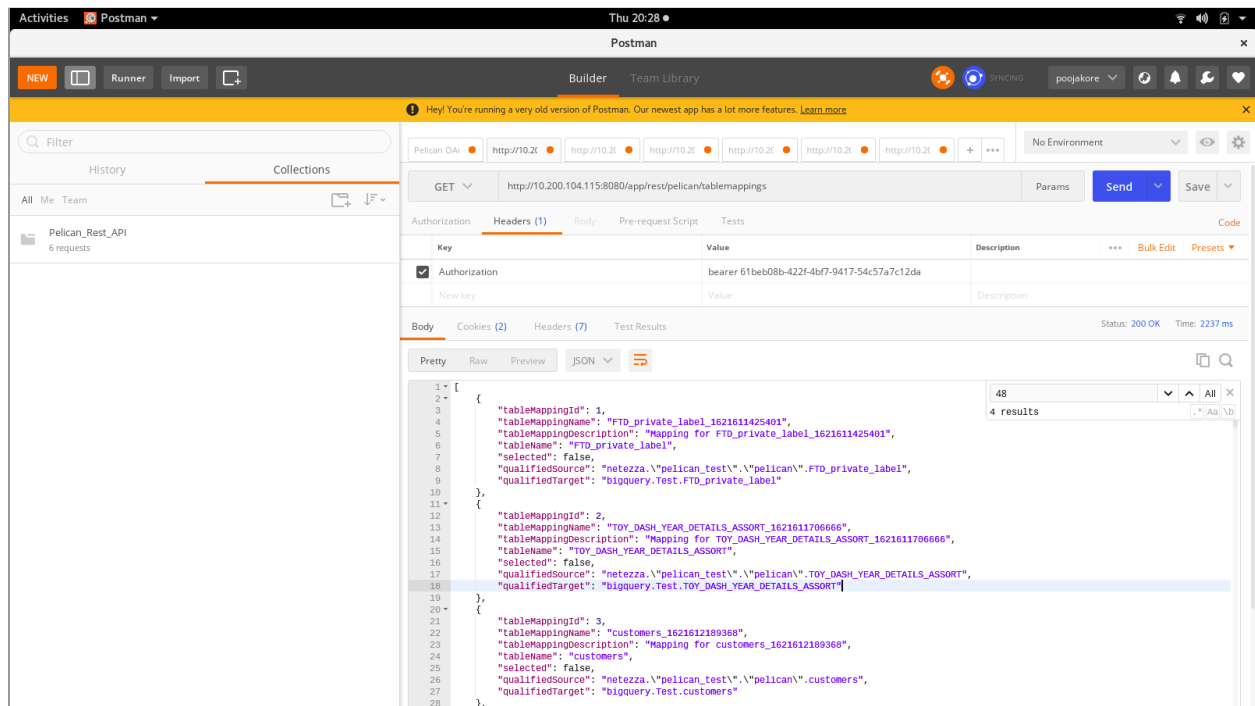
Expected Result [Example]

```

[
{
  "tableMappingId": 62,
  "tableMappingName": "BLOOMIES_OFR_SALES_BY_STR_SUMM_T_1622096947844",
  "tableMappingDescription": "Mapping for
BLOOMIES_OFR_SALES_BY_STR_SUMM_T_1622096947844",
  "tableName": "BLOOMIES_OFR_SALES_BY_STR_SUMM_T",
  "selected": false,
  "qualifiedSource": "Teradata_Copy.pelican.BLOOMIES_OFR_SALES_BY_STR_SUMM_T",
  "qualifiedTarget": "bigquery.Test.BLOOMIES_OFR_SALES_BY_STR_SUMM_T"
}
]

```

Screenshot for reference



Step 3

Url=<http://10.200.104.115:8080/app/rest/pelican/tablemapping/{tableMappingId}> generated in step 1 which need to be exported Ex-42}

Method=GET

Header

Authorization bearer {access_token generated in step 1 }

Expected Result [Example]

```

{
  "tableMappingId": 42,
  "mappingConfigId": 5,
  "targetTablename": "TableauExecutiveDashboard",
  "sourceDataStoreId": 9,
  "sourceDbname": "test_db.dbo",
  "sourceTablename": "TableauExecutiveDashboard",
  "comparisionDescription": null,
  "tableMatching": false,
  "approximateMatch": false,
  "sourceTableFilter": null,
  "targetTableFilter": null,

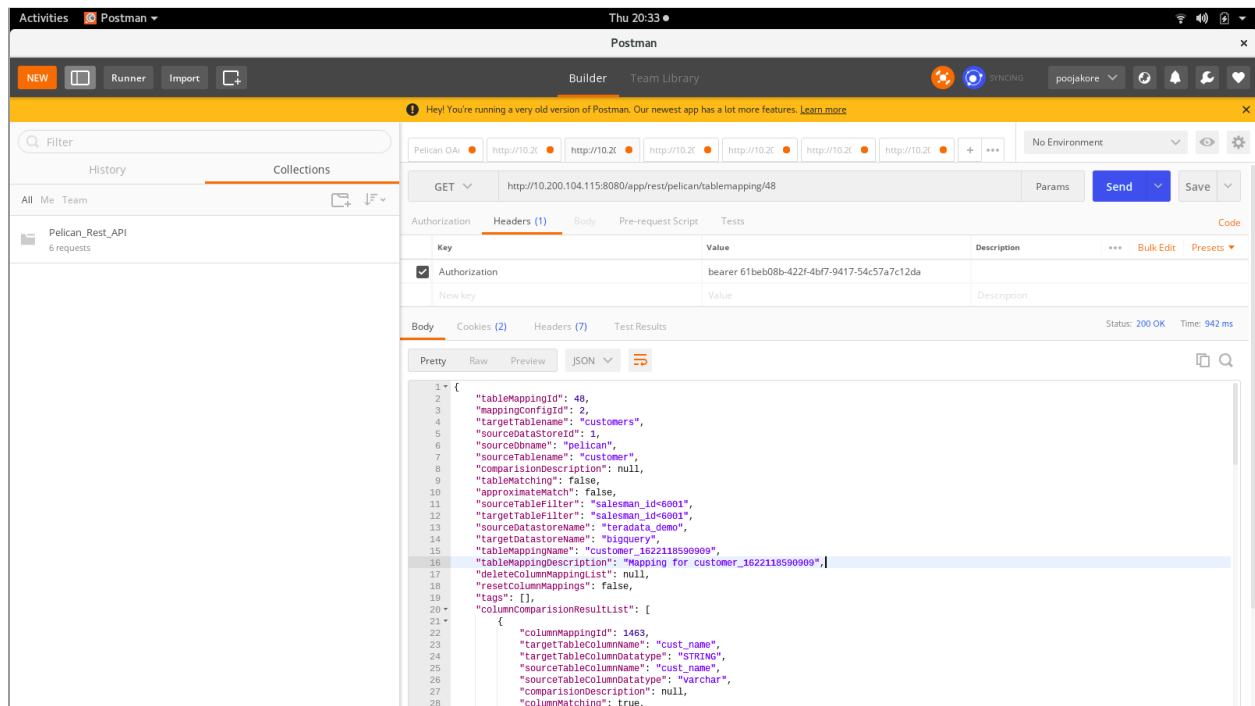
```

```

"sourceDatastoreName": "ms_sql_server_test",
"targetDatastoreName": "bigquery",
"tableMappingName": "TableauExecutiveDashboard_1622022651390_clone",
"tableMappingDescription": "Mapping for TableauExecutiveDashboard_1622022651390",
"deleteColumnMappingList": null,
"resetColumnMappings": false,
"tags": [],
"columnComparisionResultList": [
  {
    "columnMappingId": 1163,
    "targetTableColumnName": "FORMAT(\"%.*f\",0,CYBilledRentanchor)",
    "targetTableColumnDatatype": "STRING",
    "sourceTableColumnName": "LTRIM(STR(CYBilledRentanchor, (CASE
WHEN(CYBilledRentanchor < 0) THEN 21 ELSE 20 END) , 4))",
    "sourceTableColumnDatatype": "VARCHAR",
    "comparisionDescription": null,
    "columnMatching": true,
    "ignore": false,
    "uniqueKeyColumn": false,
    "createTimestampField": false,
    "modifyTimestampField": false,
    "columnFilter": null,
    "sourceColumnLogicalDatatype": "Decimal Number",
    "targetColumnLogicalDatatype": "Decimal Number",
    "expressionName": "CYBilledRentanchor",
    "sourceColumnLength": 400,
    "sourcePrecision": null,
    "targetColumnLength": 40,
    "targetPrecision": null,
    "initialSourceColumnname": "CYBilledRentanchor",
    "initialTargetColumnname": "CYBilledRentanchor",
    "initialSourceColumnDatatype": "money",
    "initialTargetColumnDatatype": "NUMERIC",
    "initialSourceColumnLength": null,
    "initialTargetColumnLength": null,
    "initialSourceColumnPrecision": null,
    "initialTargetColumnPrecision": null,
    "defaultSourceExpression": false,
    "defaultTargetExpression": false,
    "columnMappingStatus": 1,
    "incompatibleNumericDatatype": false,
    "approximateMatch": false
  }
]

```

Screenshot for reference



Import Mapping API

Step 1

Url=<http://10.200.104.100:8080/oauth/token>

Method=POST

Header

Content-Type application/x-www-form-urlencoded

For more information please refer to the Step 1 of Export API

Step 2

API to get the datastore ID.

Url=<http://10.200.104.100:8080/app/rest/datastoreforui>

Method=GET

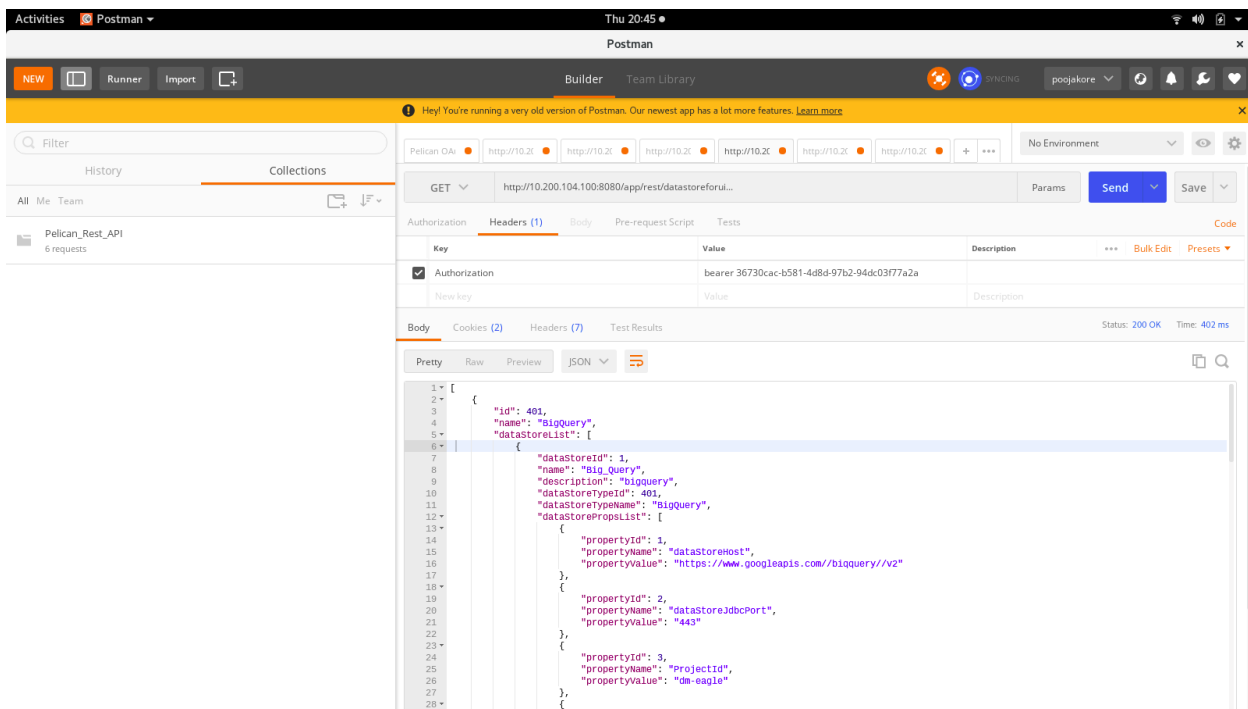
Header

Authorization bearer {access_token generated in step 1 }

Expected Result [Example]

```
{
  "id": 401,
  "name": "BigQuery",
  "dataStoreList": [
    {
      "dataStoreId": 1,
      "name": "bq_ftd",
      "description": "data store",
      "dataStoreTypeId": 401,
      "dataStoreTypeName": "BigQuery",
      "dataStorePropsList": [
        {
          "propertyId": 1,
          "propertyName": "dataStoreHost",
          "propertyValue": "https://www.googleapis.com//biquery/v2"
        },
        {
          "propertyId": 2,
          "propertyName": "dataStoreJdbcPort",
          "propertyValue": "443"
        }
      ]
    }
  ]
}
```

Screenshot for reference



Step 3

To Import the mapping.

Url=<http://10.200.104.100:8080/app/rest/pelican/persistmapping>

Method=POST

Header

Content-Type application/json

Authorization bearer 11c963b7-bb8f-4bdd-9588-20c7339324d8

Body

```
{
  "tableComparisionResultList": [
    {
      "resultName": "Mapped Tables",
      "tableComparisionResultList": [
```

Response from step 3 of Export API

```
{
  "tableMappingId": null,-----change set as null
  "mappingConfigId": null,-----change set as null
  "targetTablename": "BLOOMIES_OF_R_SALES_BY_STR_SUMM_T",
```

"sourceDataStoreId": 4,-----change as per 2nd API call from import process

```
"sourceDbname": "pelican",
"sourceTablename": "BLOOMIES_OFR_SALES_BY_STR_SUMM_T",
"comparisionDescription": null,
"tableMatching": false,
"approximateMatch": false,
"sourceTableFilter": "ITEM_QTY<1",
"targetTableFilter": "ITEM_QTY<1",
"sourceDatastoreName": "Teradata_Copy",
"targetDatastoreName": "bq_ftd",
"tableMappingName": "BLOOMIES_OFR_SALES_BY_STR_SUMM_T_1622096947844",
"tableMappingDescription": "Mapping for
BLOOMIES_OFR_SALES_BY_STR_SUMM_T_1622096947844",
"deleteColumnMappingList": null,
"resetColumnMappings": false,
"tags": [],
"columnComparisionResultList": [
  {
    "columnMappingId": 1445,
    "targetTableColumnName": "ITEM_QTY",
    "targetTableColumnDatatype": "INT64",
    "sourceTableColumnName": "ITEM_QTY",
    "sourceTableColumnDatatype": "integer",
    "comparisionDescription": null,
    "columnMatching": true,
    "ignore": false,
    "uniqueKeyColumn": false,
    "createTimestampField": false,
    "modifyTimestampField": false,
    "columnFilter": null,
    "sourceColumnLogicalDatatype": "Non Decimal Number",
    "targetColumnLogicalDatatype": null,
    "expressionName": "ITEM_QTY",
    "sourceColumnLength": 11,
    "sourcePrecision": null,
    "targetColumnLength": 11,
    "targetPrecision": null,
    "initialSourceColumnname": "ITEM_QTY",
    "initialTargetColumnname": "ITEM_QTY",
    "initialSourceColumnDatatype": "integer",
    "initialTargetColumnDatatype": "INT64",
    "initialSourceColumnLength": 11,
    "initialTargetColumnLength": 11,
```

```

    "initialSourceColumnPrecision": null,
    "initialTargetColumnPrecision": null,
    "defaultSourceExpression": false,
    "defaultTargetExpression": false,
    "columnMappingStatus": 1,
    "incompatibleNumericDatatype": false,
    "approximateMatch": false
  },
  {
    "columnMappingId": 1447,
    "targetTableColumnName": "FORMAT(\\'%.*f\\',2,TXN_AMT)",
    "targetTableColumnDatatype": "STRING",
    "sourceTableColumnName": "TRIM(LEADING '0' FROM CAST((TXN_AMT(format
'999999999.99')) AS VARCHAR(13)))",
    "sourceTableColumnDatatype": "VARCHAR",
    "comparisionDescription": null,
    "columnMatching": true,
    "ignore": false,
    "uniqueKeyColumn": true,
    "createTimestampField": false,
    "modifyTimestampField": false,
    "columnFilter": null,
    "sourceColumnLogicalDatatype": "Decimal Number",
    "targetColumnLogicalDatatype": "Decimal Number",
    "expressionName": "TXN_AMT",
    "sourceColumnLength": 9,
    "sourcePrecision": null,
    "targetColumnLength": 9,
    "targetPrecision": null,
    "initialSourceColumnname": "TXN_AMT",
    "initialTargetColumnname": "TXN_AMT",
    "initialSourceColumnDatatype": "decimal",
    "initialTargetColumnDatatype": "NUMERIC",
    "initialSourceColumnLength": 9,
    "initialTargetColumnLength": 9,
    "initialSourceColumnPrecision": 2,
    "initialTargetColumnPrecision": null,
    "defaultSourceExpression": false,
    "defaultTargetExpression": false,
    "columnMappingStatus": 1,
    "incompatibleNumericDatatype": false,
    "approximateMatch": false
  },
  {

```

```

    "columnMappingId": 1446,
    "targetTableColumnName": "FORMAT(\"%.*f\",2,DISC_AMT)",
    "targetTableColumnDatatype": "STRING",
    "sourceTableColumnName": "TRIM(LEADING '0' FROM CAST((DISC_AMT(format
'999999999.99')) AS VARCHAR(13)))",
    "sourceTableColumnDatatype": "VARCHAR",
    "comparisionDescription": null,
    "columnMatching": true,
    "ignore": false,
    "uniqueKeyColumn": false,
    "createTimestampField": false,
    "modifyTimestampField": false,
    "columnFilter": null,
    "sourceColumnLogicalDatatype": "Decimal Number",
    "targetColumnLogicalDatatype": "Decimal Number",
    "expressionName": "DISC_AMT",
    "sourceColumnLength": 9,
    "sourcePrecision": null,
    "targetColumnLength": 9,
    "targetPrecision": null,
    "initialSourceColumnname": "DISC_AMT",
    "initialTargetColumnname": "DISC_AMT",
    "initialSourceColumnDatatype": "decimal",
    "initialTargetColumnDatatype": "NUMERIC",
    "initialSourceColumnLength": 9,
    "initialTargetColumnLength": 9,
    "initialSourceColumnPrecision": 2,
    "initialTargetColumnPrecision": null,
    "defaultSourceExpression": false,
    "defaultTargetExpression": false,
    "columnMappingStatus": 1,
    "incompatibleNumericDatatype": false,
    "approximateMatch": false
  }
],
  "hiveDataStoreId": 3,-----change as per 2nd API call from import process(use bq data
store id here)
  "hiveDbname": "Test",
  "comparisionResult": null
}
],
{
  "resultName": "Partially mapped Tables",

```

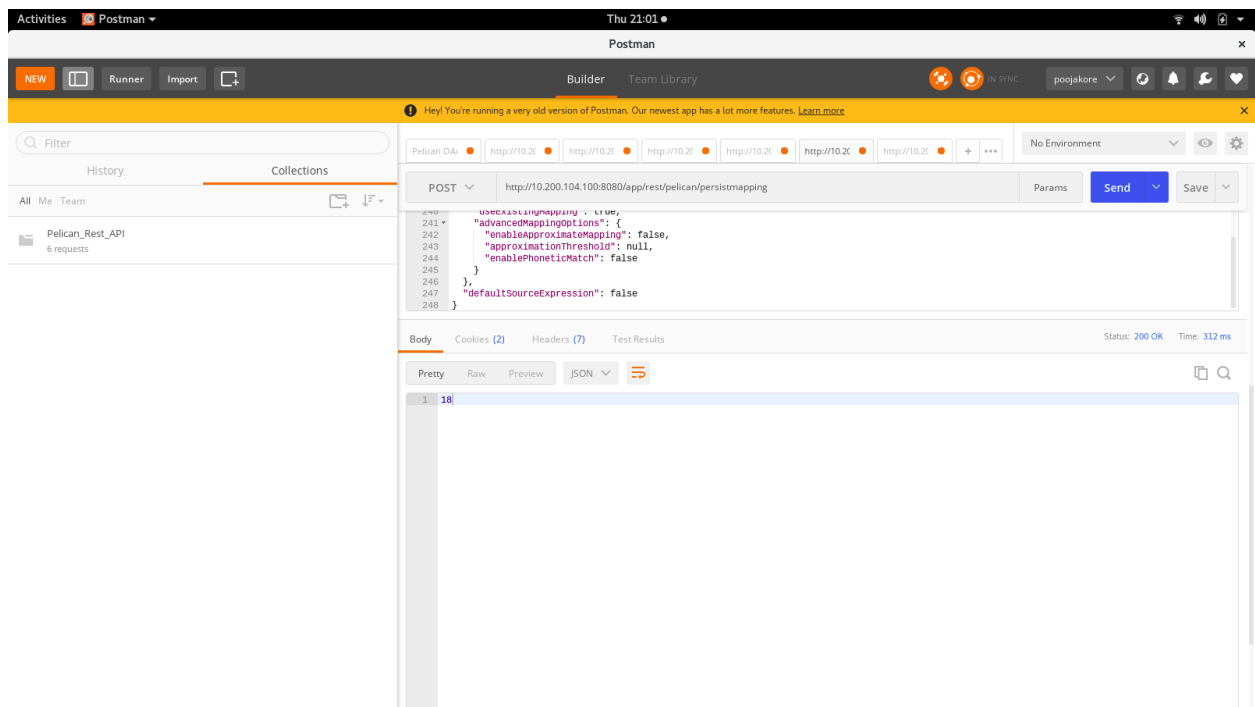
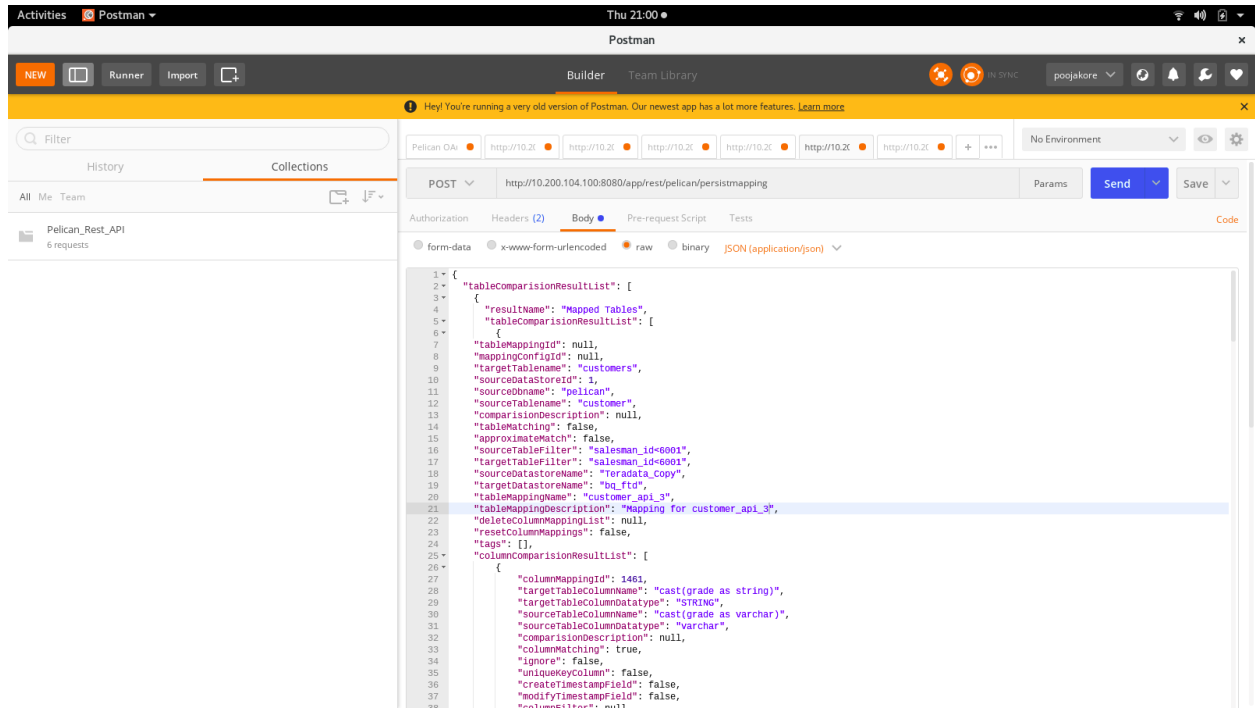
```

    "tableComparisionResultList": []
  },
  {
    "resultName": "Unmapped Tables",
    "tableComparisionResultList": []
  },
  {
    "resultName": "Previously mapped Tables",
    "tableComparisionResultList": []
  }
],
"pelicanMappingParameterDTO": {
  "pelicanMapConfigId": null,
  "sourceDataStoreId": 4,-----change as per 2nd API call from import process
  "sourceDbName": "pelican",-----set source database name
  "tableList": [
    {
      "tableName": "BLOOMIES_OFR_SALES_BY_STR_SUMM_T",-----set your table name
      example(BLOOMIES_OFR_SALES_BY_STR_SUMM_T)
      "dataStoreType": null,
      "databaseName": "pelican",-----set source database name
      "dataStoreId": 4,-----change as per 2nd API call from import process
      "selected": false,
      "dataStoreName": null,
      "columnMetadataMap": {},
      "approximateMatch": false,
      "logicalDatatypeToColumnNamesMap": {},
      "physicalDatatypeToColumnNamesMap": {}
    }
  ],
  "targetDataStoreId": 1,-----change as per 2nd API call from import process
  "targetDbName": "Test",-----set target database name
  "targetTableName": null,
  "sourceTableName": null,
  "editMapping": false,
  "resetMapping": false,
  "useExistingMapping": true,
  "advancedMappingOptions": {
    "enableApproximateMapping": false,
    "approximationThreshold": null,
    "enablePhoneticMatch": false
  }
},
"defaultSourceExpression": false

```

```
}
Expected Result [Example]
Mapping id of imported mapping
```

Screenshot for Reference



14.2. Pelican V3 API

Step 1

OAuth API call

Url= <http://127.0.0.1:8100/oauth/token>

Method=POST

Header

content-type : application/x-www-form-urlencoded

accept : application/json

Body

username superuser@datametica.com

password admin

grant_type password

client_id ecac

client_secret ecac

Step 2

Pelican V3 api to get table mapping id

Url=

http://127.0.0.1:8100/api/rest/v3/pelican/mapping/db2/SAMPLE.DB2ADMIN/CUSTOMER_P2?targetDsName=BQ_Sam&targetDbName=qastage&targetTableName=customer_p2

Method=GET

Header

accept : application/json

content-type : application/json

Authorization : Bearer {oauth token from API call #1}

**** Replace source and target db name and table name above ****

Step 3

Pelican V3 API for mapping details

Url= <http://127.0.0.1:8100/api/rest/v3/pelican/mapping/{table mapping id}>

Method=GET

Header

accept : application/json

content-type : application/json

Authorization : Bearer {oauth token from API call #1}

**** Replace table mapping id in url from API call #2 ****

14.3. API to get Scheduler status

Step 1:

To get the authentication token.

OAuth API call

Url= <http://127.0.0.1:8100/oauth/token>

Method=POST

Header

content-type : application/x-www-form-urlencoded

accept : application/json

Body

username superuser@datametica.com

password admin

grant_type password

client_id ecacat

client_secret ecacat

Step 2:

API to get the datastore ID.

Url=<http://10.200.104.100:8080/app/rest/datastoreforui>

Method=GET

Header

Authorization bearer {access_token generated in step 1 }

Expected Result [Example]

```
{
  "id": 401,
  "name": "BigQuery",
  "dataStoreList": [
    {
      "dataStoreId": 1,
      "name": "bq_ftd",
      "description": "data store",
      "dataStoreTypeId": 401,
      "dataStoreTypeName": "BigQuery",
    }
  ]
}
```



```
"dataStorePropsList": [  
  {  
    "propertyId": 1,  
    "propertyName": "dataStoreHost",  
    "propertyValue": "https://www.googleapis.com//biquery/v2"  
  },  
  {  
    "propertyId": 2,  
    "propertyName": "dataStoreJdbcPort",  
    "propertyValue": "443"  
  },  
]
```

Step 3:

API to get the scheduler status.

Url= <http://10.200.104.138:8080/app/rest/schedulingConfigStatus/{dataStoreId}>

Ex. <http://10.200.104.138:8080/app/rest/schedulingConfigStatus/12>

Method=GET

Header

accept : application/json

content-type : application/json

Authorization : Bearer {oauth token from API call #1}

Expected Result [Example]

```
{  
  "status":"Completed",  
  "Id":12,  
  "schedulerName":"td_snowflake"  
}
```

**** Replace table mapping id in url from API call #2 ****

15. Automatic Backup

- The Pelican MYSQL Backups are taken automatically. They are scheduled weekly every Sunday at 8.30 am.
- Only 5 latest backup files are maintained at the designated location, as the oldest file gets replaced by the newest one.
- The backup location is as follows:
 - ❖ For docker and kubernetes:
`/usr/local/apache-tomcat-${TOMCAT_VERSION}/dbbackup`
 - ❖ For Jar based installation on VM:
`/(path of tomcat server)/dbbackup`

16. Frequently Asked Questions

1. How to view the validation result in Pelican?

The validation result can be viewed only for those tables whose table mapping is done. So, if you are a new user and the validation mapping has not been done yet then you need to follow the below given steps:

Step 1: First, you need to configure a data store.

Step 2: Followed by datastore configuration, you need to validate tables between source and destination datastore.

Step 3: Next, you need to configure the scheduler for the saved validation mapping

Step 4: Finally, go to Reports and click Validation Result. For more information, refer to the [Validation Result](#) topic.

2. Which are the various data stores the Pelican supports?

Pelican supports comparison between various data stores. For more information, refer to the [Data Store Support](#) topic.

3. How to validate tables between source and destination datastore?

You can validate tables between source and destination Datastore using Validation Configuration functionality. For more information, refer to the [Validation Mapping](#) topic.

4. What is the scheduler and how to configure it?

The Scheduler allows the user to execute processes at a regular time interval. In Pelican, the user can create a scheduler for a saved mapping; so that, after a time period the scheduler executes the process and it validates the source table with destination tables. For more information, refer to the Scheduler Configuration topic.

5. How to configure the email notification?

Email notification functionality allows the user to send an email automatically to various user on the execution of the scheduler. For more information, refer to the [Email Notification](#) topic.

6. How to update the pelican licence?

Once you received the pelican licence file, i.e. licence.pel follow the following steps.

Steps 1; log in to the Pelican

Step 2: Administration -> Add Licence

Step 3: Click on the upload and select the Pelican.pel from the file system.

Step 4: Again login to continue the usage.

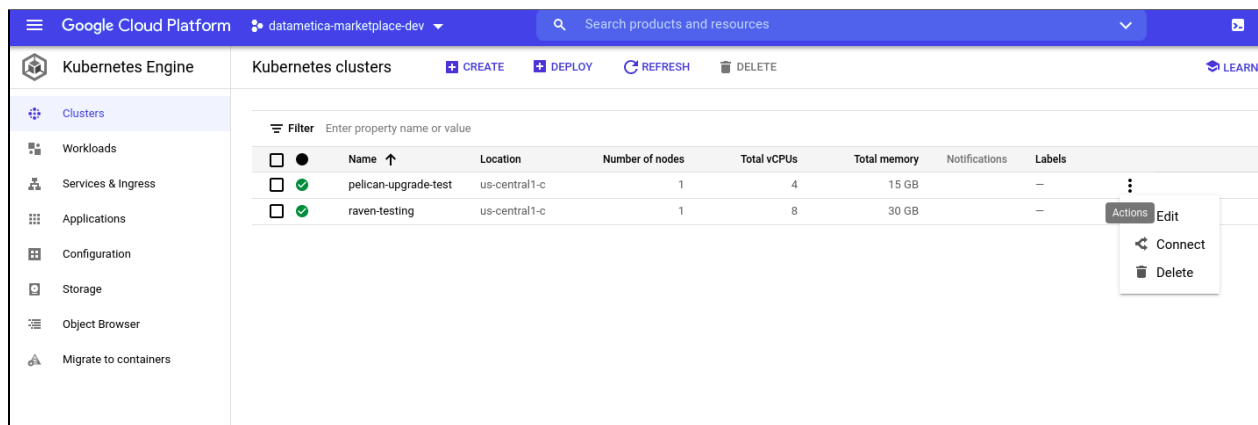
7. Who can create LDAP user?

Only the superuser has the right to create LDAP users. LDAP users can't create any new users. Only users with Admin access can create new LDAP users.

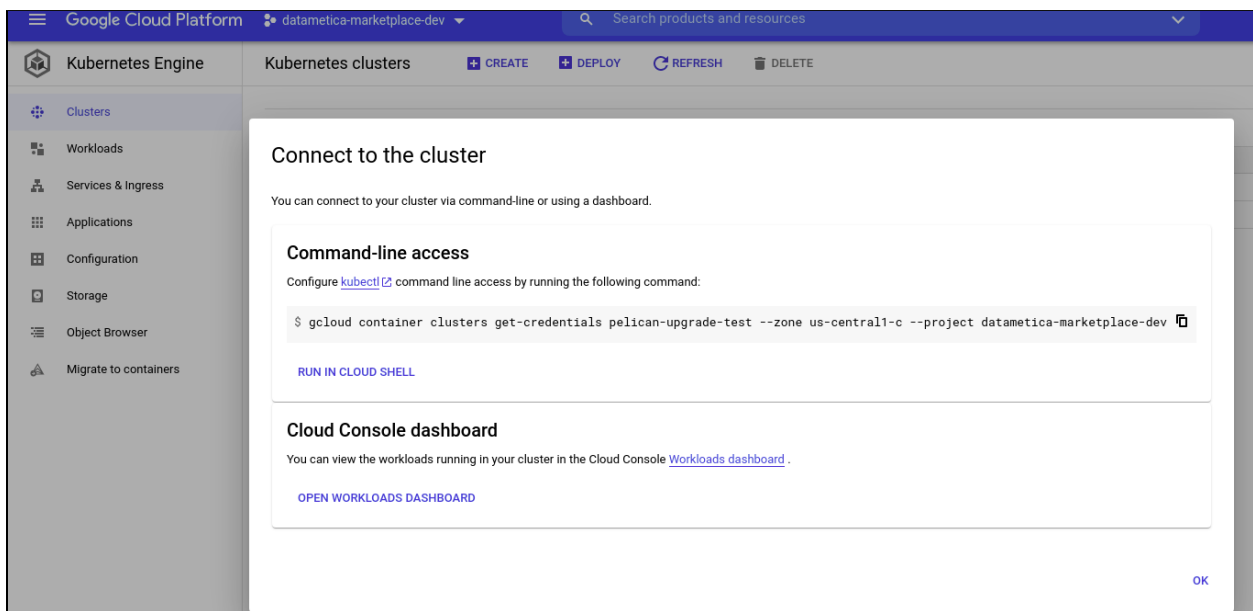
8. How to get GKE server(Pelican services) start-stop access permissions to QA Lead?

Step 1 - Go to GCP console > Search for GKE

Step 2 - Click on Connect your cluster



Step 3 - Click on Run in cloud shell



Step 4 - Run this command after login on cluster - **kubectl get pods**

Step 5 - Copy the web pod name and run this command to restart the service - **kubectl delete podname**

Note - Pelican service pod will be automatically started and old one will be deleted.

9. How to get Application log file url along with access permissions to QA user (catalinaout.log and application.log) ?

Step 1 - To see the application logs, we need to go inside the pelican pod. Run this command to go inside the pod - **kubectl exec -it podname bash**

Step 2 - Now you can view the logs on this file path -
/usr/local/apache-tomcat-8.5.65/logs/application_logs/log.log

10. How to add a lineage file?

To add a lineage file, login to the system, to go Administration and select Add lineage. The window will be displayed on the screen. For more information, visit [Add Lineage](#).

11. What is a CSV file?

The CSV file is a text file in which field values are separated by commas and each record is separated by a new line character. First row can optionally be a header for field values.

17. Glossary

- Three Level Hierarchy:

The three level DBMS architecture has following three levels:

- **External or User view**
- **Conceptual or Logical**
- **Internal or Physical**

External Level:

It is also called the view level as several users can view their required data by internally fetching it from the database with the help of internal and conceptual level mapping. This level is the top-most level of the three level database architecture.

Conceptual Level:

Also referred to as the logical level it describes the entire database design. The relationship between the data, schema of data etc. are defined at this level by the Database administrator. Security and database constraints are also defined at the conceptual level.

Internal Level:

The Internal level is also known as the Physical level. It describes how the database is actually stored in terms of the record layout of files and type of files (hash, b-tree, flat). The Internal level is the lowest level of database architecture.