Quest Data Quality V3.1.3

The User Guide

Prepared by

Introduction

Quest Data Quality | Modern Data Quality Platform

Quest Data Quality is a unified platform that streamlines data management by combining Data Quality, Data Observability, and Data Discovery. It offers automated monitoring with over 50 checks to ensure data health and performance, robust tools for maintaining data quality and metadata management, and efficient issue resolution workflows. The user-friendly interface and role-based functions cater to diverse needs, while advanced AI capabilities enhance data quality checks and metadata enrichment. Leveraging large language models (LLMs) and Generative AI (GenAI), Quest Data Quality automates complex tasks and supports a conversational approach, making it an essential tool for modern data teams.

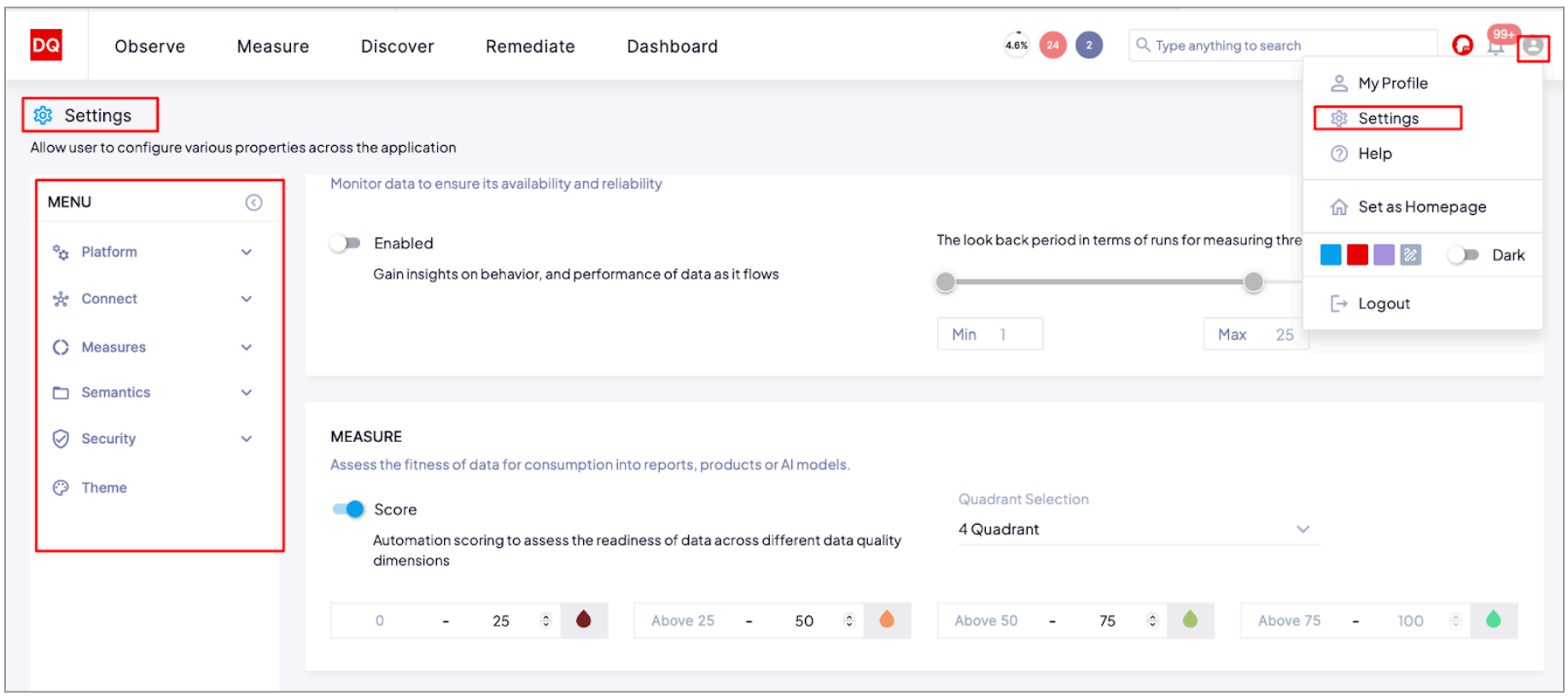
Why Quest DQ?

1. **Comprehensive Data Quality Lifecycle Management:** Quest DQ excels in managing the entire data quality lifecycle, from identifying unknown issues and measuring known ones using a semantic layer to providing detailed root cause analysis and remediation workflows. This end-to-end approach ensures not just detection but also the resolution of data quality issues. Quest DQ also goes beyond alerting by offering detailed analysis and push-down capabilities for bad data remediation, including circuit breaker support to stop problematic pipelines or workflows.
2. **Advanced Automated Monitoring and Anomaly Detection:** With over 50 automated data observability checks at various levels, Quest DQ leverages deep data profiling, AI/ML time-series analysis, and the latest anomaly detection algorithms. This ensures robust monitoring and significantly reduces manual effort in issue identification.
3. **Customizable and Contextual Alerts:** Quest DQ offers a rich, native semantic layer that helps in reducing alert fatigue by prioritizing and contextualizing alerts. Users can define specific thresholds and conditions, ensuring relevant and timely notifications, making it easier to address the most critical issues first.
4. **Robust Data Lineage and Schema Management:** Quest DQ provides detailed lineage tracking, automated schema drift detection, and comprehensive impact analysis. This capability helps in troubleshooting, maintaining data consistency, and understanding how data changes affect downstream processes, facilitating informed decision-making.
5. **Scalability, Integration, and User Experience:** Quest DQ' cloud-native architecture supports high-volume ingestion and processing with elastic scaling, ensuring efficient operation even with growing data volumes. It seamlessly integrates with existing data ecosystems and offers an intuitive user interface, extensive documentation, and 24x7 global customer support, making it accessible and user-friendly for both business and technical users.

Admin Setup

Understanding Platform Settings

The Settings menu allows the user to configure various properties related to the Quest DQ platform. To go to **Settings**, go to your user avatar, as shown on the top right in the figure below. In the drop-down, click on the **Settings** button.

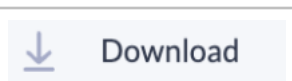


The Menu on the left shows the various settings that the user can configure concerning the following aspects:

- **Platform**: It has settings related to the platform configuration, repository, and utility
- **Connect:** This has the settings related to different connections and integrations active on the user's platform.
- **Measures:** Measures are specific metrics or indicators that are offered by Quest DQ for assessing the quantity and quality of data. These measures eventually help in tracking the improvements and make the data meet the required standards.

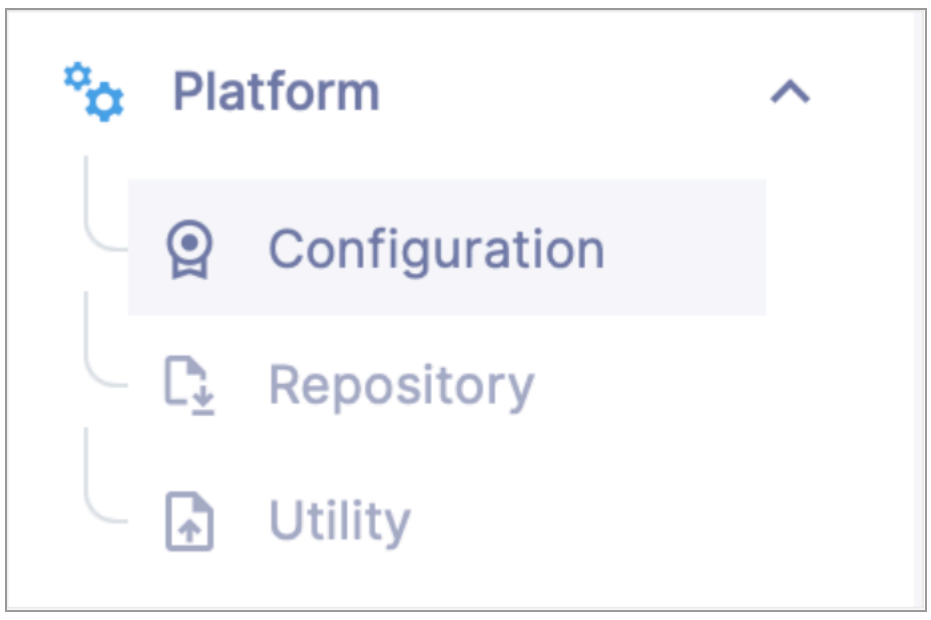
- **Semantics:** Semantics in Quest DQ implies the interpretation of various elements in a data set or a system. It helps ensure the processed data is meaningful and consistent.
- **Security:** Security in data quality refers to the practices implemented for protecting the data, avoiding breaches, preventing data loss maintaining its confidentiality, availability, and consistency.
- **Theme:** The theme-related settings in Quest DQ help for

The above aspects are discussed in detail in the sections below. Before going on to the menu, the following is a set of common operations and the icons present for each of the Settings Menu screens, listed out at the start to help the users understand the operational importance of each icon.

Icon	Description
	The Search button helps the user search specifically under the available set of columns.
	By clicking on this button, the user can switch to the tile view from the list view in a given dashboard.
	This is an Add New button to add a new row for a given dashboard entity.
	This allows the user to download the list in a given dashboard in .csv format.
	The column settings help the users to select the columns that are to be visible or hidden for a given dashboard.
	This button is used to edit the existing setting, generally present under the **Action** column.
	This button is used to delete the existing row in a given dashboard.

Platform - Setting up the platform

The platform-level settings help the user to configure the attributes for the platform and manage the repository and the utility-related settings. To access the platform-related settings, go to **Settings** > **Platform.**

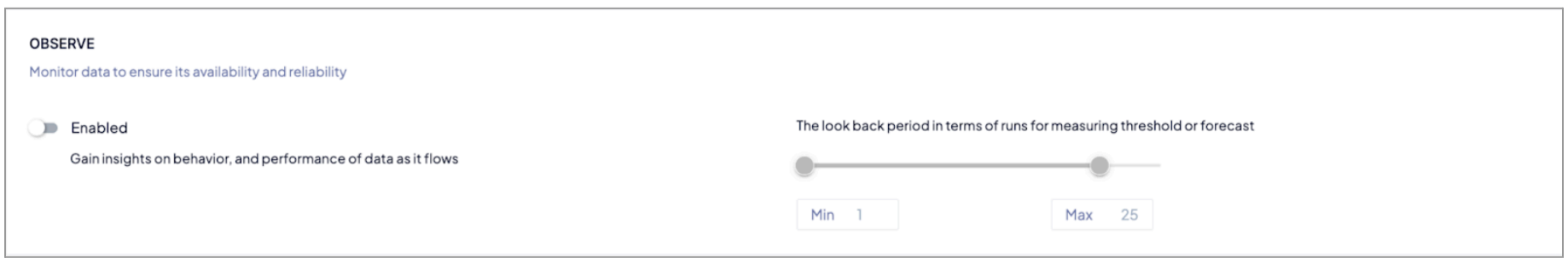


Configuration

The Configuration page has the settings to configure the Observe, Measure, Discover, and Remediate dashboard-related attributes. The configuration settings for each of the dashboards are explained in the sections below with images.

Configuring the 'Observe' Tool

The Observe dashboard toggle button, when enabled, will give the user will have the ability to get alerts for measures based on previous data. The look-back period for the previous data can be set from the slider on the right, as shown below.



Anomaly Threshold Calculation Example

In the anomaly detection process, the expected threshold for a current run is determined based on historical data within the anomaly window defined by the Max setting.

Example Scenario:

Current Run: run_id_10 with a value of 80.
Anomaly Max Setting: Configured to use the last 5 runs for calculating the expected threshold.

Threshold Calculation: The system identifies the last 5 runs, starting from run_id_10 and sliding back to include run_id_9 to run_id_5.

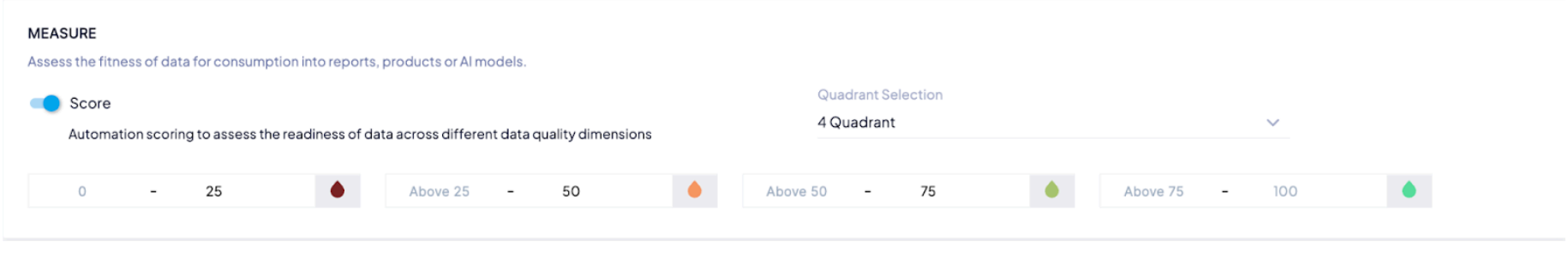
The values for these runs are:
run_id_9: 11
run_id_8: 38
run_id_7: 67
run_id_6: 3
run_id_5: 30

These values are used to compute the expected threshold for run_id_10.

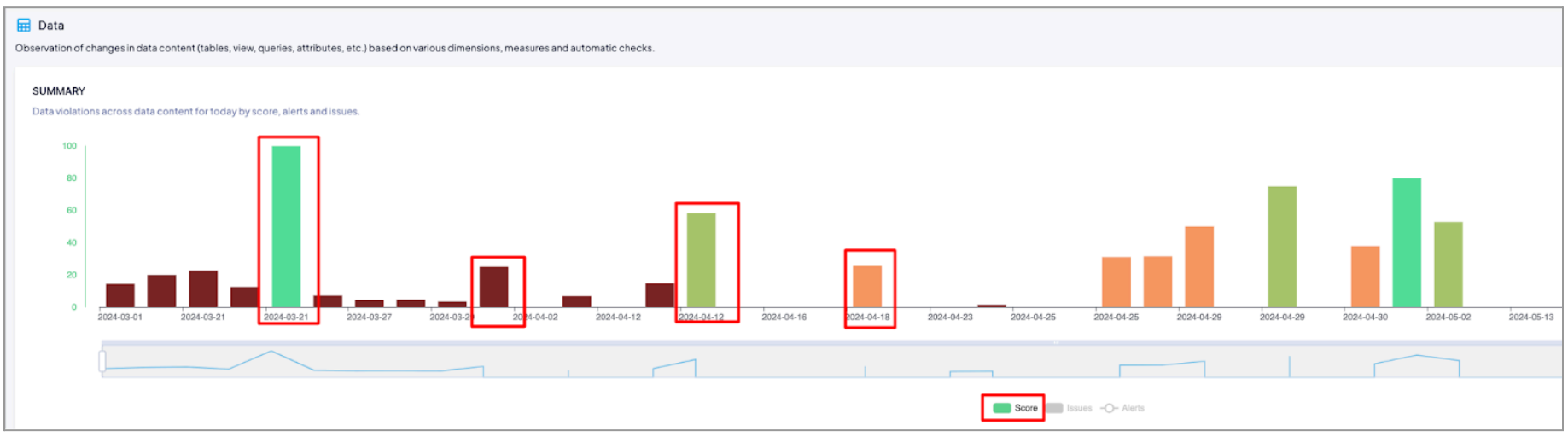
Key Insights: The Max Setting ensures that only the most recent runs within the defined window size are considered, providing a robust and consistent basis for calculating anomalies.
If fewer runs are available (less than the Min Setting), the system will not compute the expected threshold or generate alerts.
This process ensures that anomalies in the current run are detected by comparing them against an accurate and dynamically calculated expected threshold based on recent performance trends.

Configuring the 'Measure' Tool

- The **Score toggle button** under the **Measure-**related settings is used to set up threshold scores and color coding for threshold values for bar chart visualization in the. When the toggle button adjacent to Score is switched on, the **Quadrant Selection** is the next step.



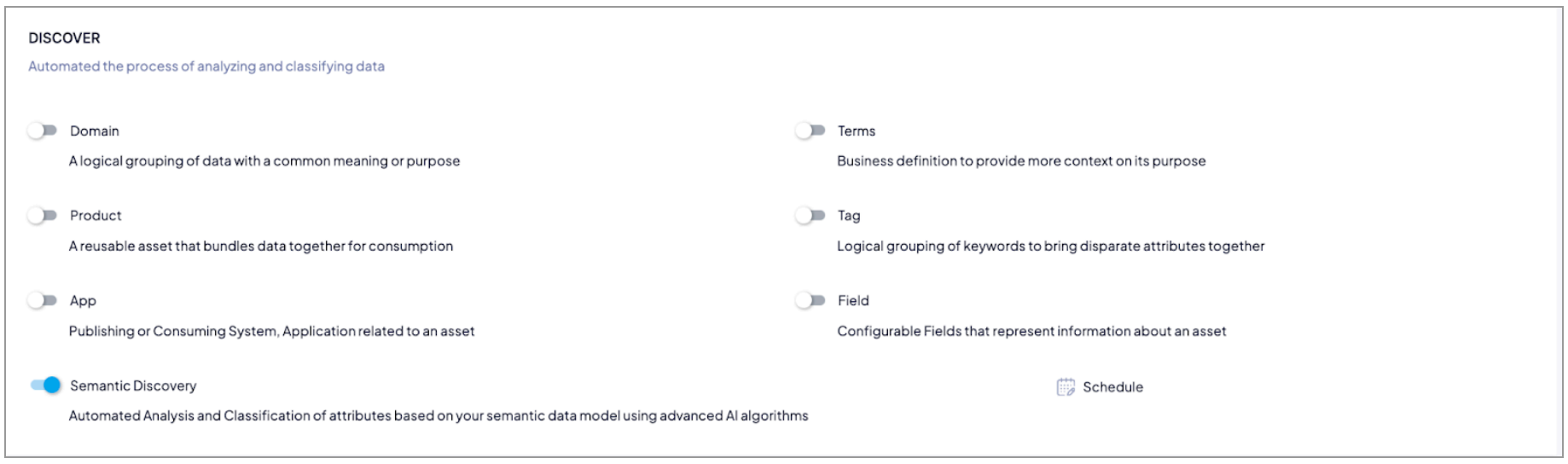
- The **Quadrant Selection** is a selection of threshold values and color codes assigned to those values, which would then reflect as shown at the bottom in the above figure.
- An example of how these quadrants are selected is shown in the example below. It is a summary of the Data when the user goes to **Observe** > **Data Summary.**



- Compare those red-highlighted bars in the bar graphs above with the threshold measures to know the color codes and their functions.

Configuring the 'Discover' Tool

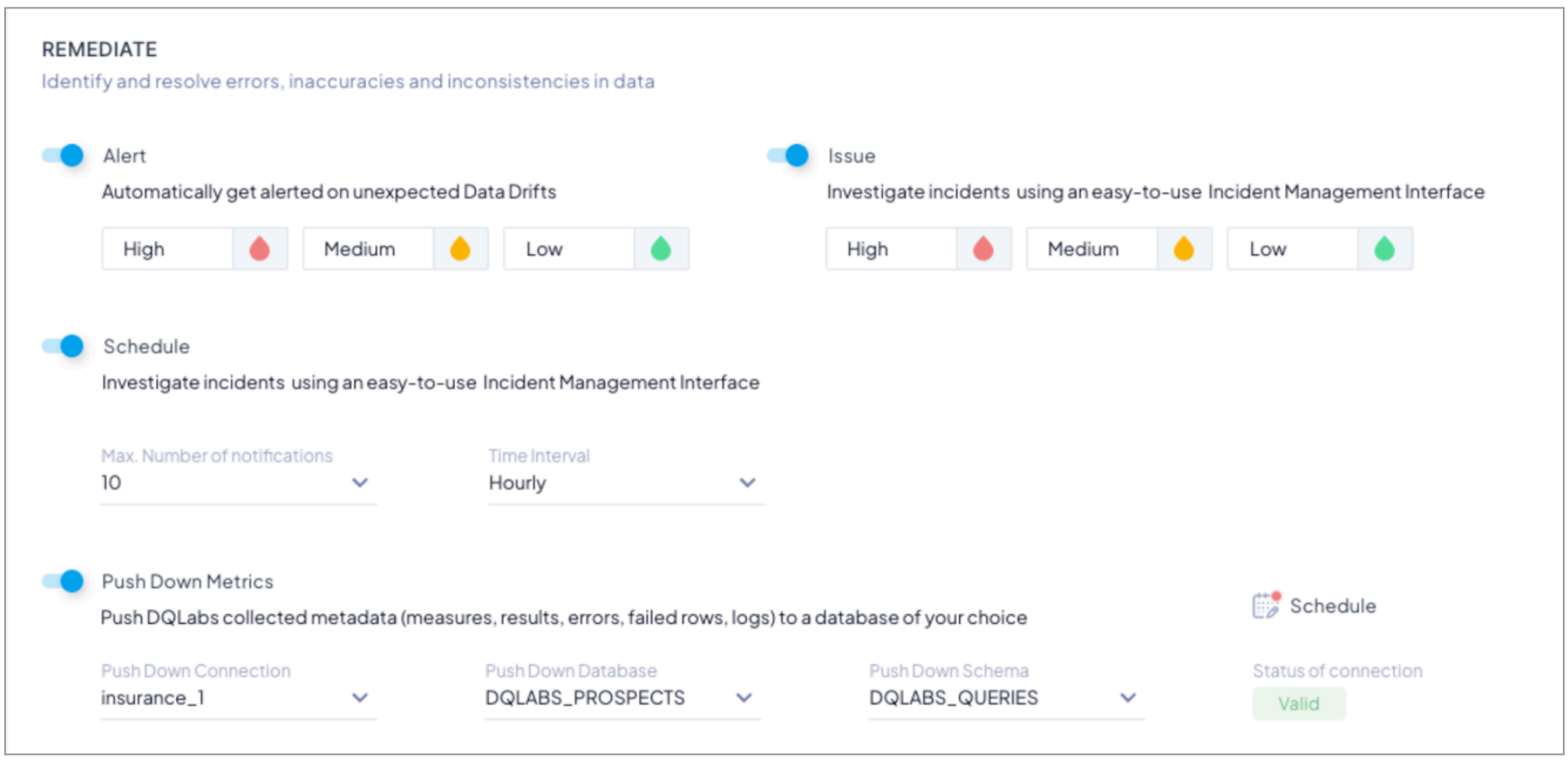
The Discover-related settings form the basis on which the data can be classified first and then can be analyzed. Each of these attributes for classification can be controlled on the basis of a toggle button.



Term	Explanation
Domain	The domain filter helps with logical grouping of data with a common meaning or purpose.
Terms	The business terms/definitions can be filtered using this toggle button.
Product	The product-related details present in the asset data can be filtered using this filter.
Tag	Tag-based filtration of information can be classified using this filter.
App	The application-related details in an asset data table can be filtered using the toggling feature.
Field	The field-related data in an asset can be filtered by turning this filter on.
Semantic Discovery	The advanced AI algorithms are used to classify the attributes and perform automated analysis on the semantic data model.

Configuring the 'Remediate' Tool

The Remediate dashboard settings principally allow the user to set alerts, raise issues, and configure metrics for remediation by the users.



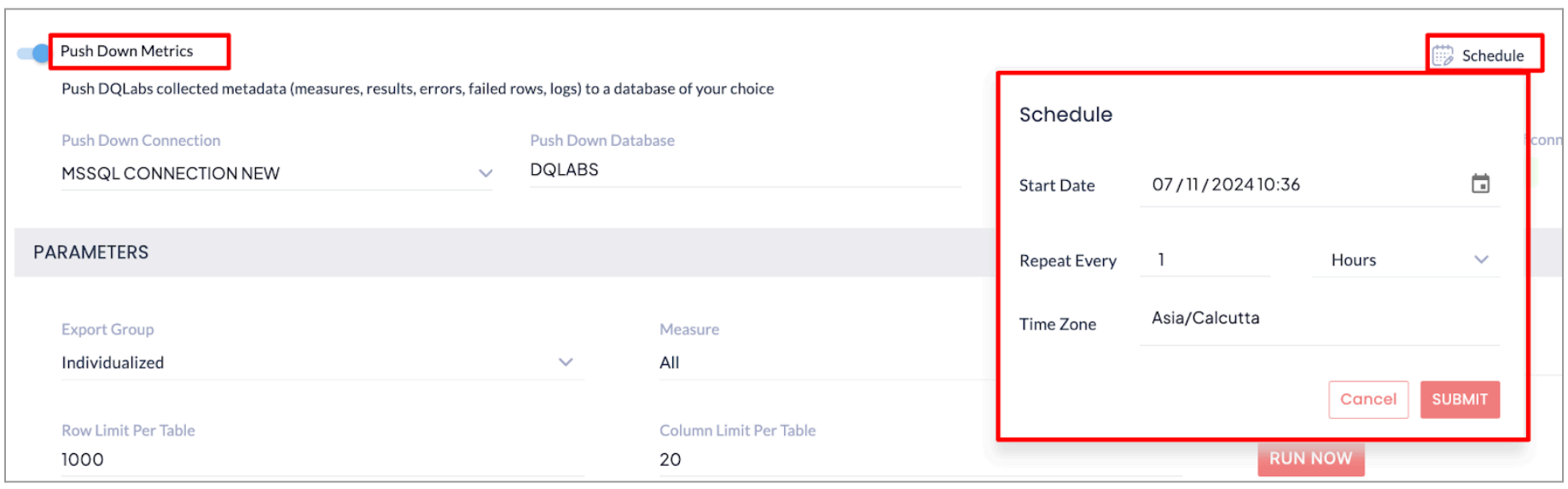
The Remediate configuration settings consist of the following attributes:

Term	Explanation
Alerts	The alerts toggle button, when switched on, enables the alerts that the user would get in circumstances of unexpected data drifts in the system. The alerts are labelled as High, Medium, and Low, based on their criticality. Each of these can be coded as per user preference.
Issues	The issues can be marked by the user as high, medium, and low by turning this toggle on.

Schedule	By switching this toggle on, the user can schedule the maximum number of notifications to be received in a given time interval. For example, 10 maximum notifications every hour can be set up as follows: ● Max. Number of notifications: 10 ● Time interval: Hourly
Push Down Metrics	**Push Down Metrics:** The push-down metrics-related settings determine the attributes that govern the metadata push-down from Quest DQ to the database of your choice.
	Push Down Connection: The pushdown connection is the connector with which the corresponding Quest DQ metadata is to be shared.
	Push Down Database: The push-down database is the Quest DQ database from which the metadata is to be pushed to the customer's database.
	Push Down Schema: Push-down schema is the schema that originates from Quest DQ database, gives relational information about the datasets present in the database, generated from the metadata received from the customer's database.
	Status of connection: The status of the connection shows whether the connection between Quest DQ and the intended database is valid.

- **Schedule - Push Down Metrics:**

The data push-down can be scheduled from the Schedule drop-down, as highlighted below.

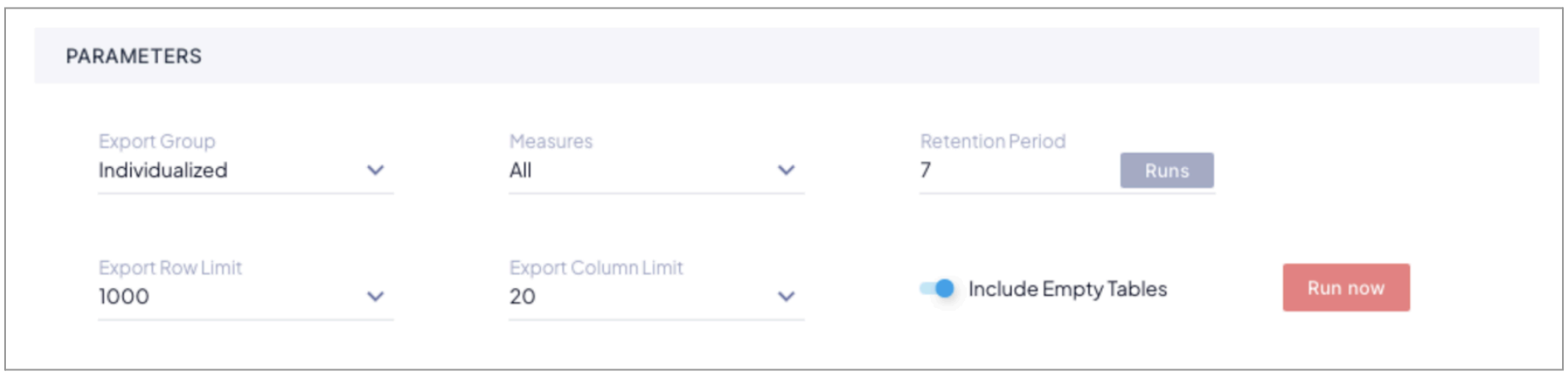


The Schedule drop-down can be used for setting up the push-down data schedule. The field-level information is described below.

Term	Explanation
Start Date	The start date for the push-down can be set up here.
Repeat Every (Repetition frequency)	The repetition frequency can be set up as minutes, hours, days, weeks, and months.
Time Zone	The time zone can be selected from the set of drop-down, for example, Asia/Calcutta

Parameters:

The parameters menu is used to define the settings that are related to the push-down metrics.



- **Export Group**:

The export group for the metadata that is to be pushed down can be selected from the drop down.

Term	Explanation
Individualized	Select this option to export all three data types viz., Measures, Assets, and Domains
Measures	Select this option to export Measures
Assets and Measures	Select this option to export assets
Domains	Select this option to export Domains

- **Measure:**

The Measure is applicable for the push-down data that can be selected from this drop-down option.

Term	Explanation
All	Select this option to export all the measures
Auto Measures	Select this option to export auto measures (auto-measures are out-of-the-box measures that are configured by Quest DQ)
Custom Measures	Select this option to export the custom measures

- **Retention Period:**

The retention period is the period for which the data is to be retained. The retention period can be changed in terms of runs, days, and months. Click on the button highlighted below to switch between days, months, and runs.

- **Export Row Limit:**

The row limit per table is the number of rows of data whose metadata is pushed down.

- **Export Column Limit:**

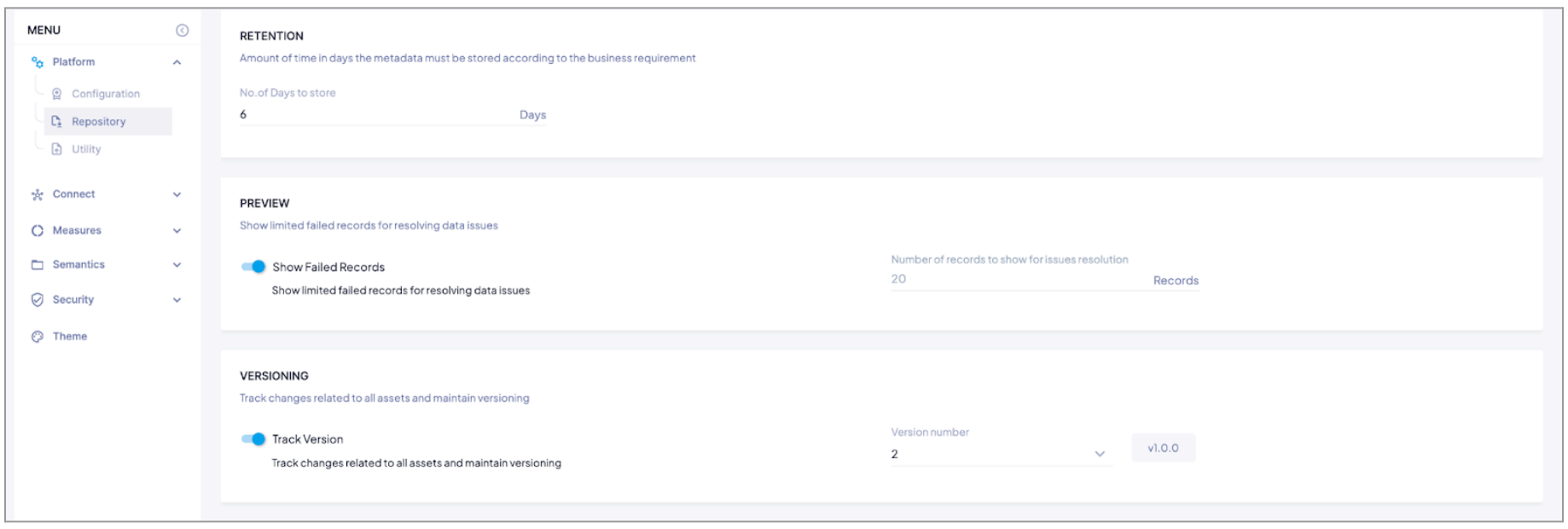
The custom limit per table is the number of tables whose metadata can be pushed down.

- **RUN NOW**:

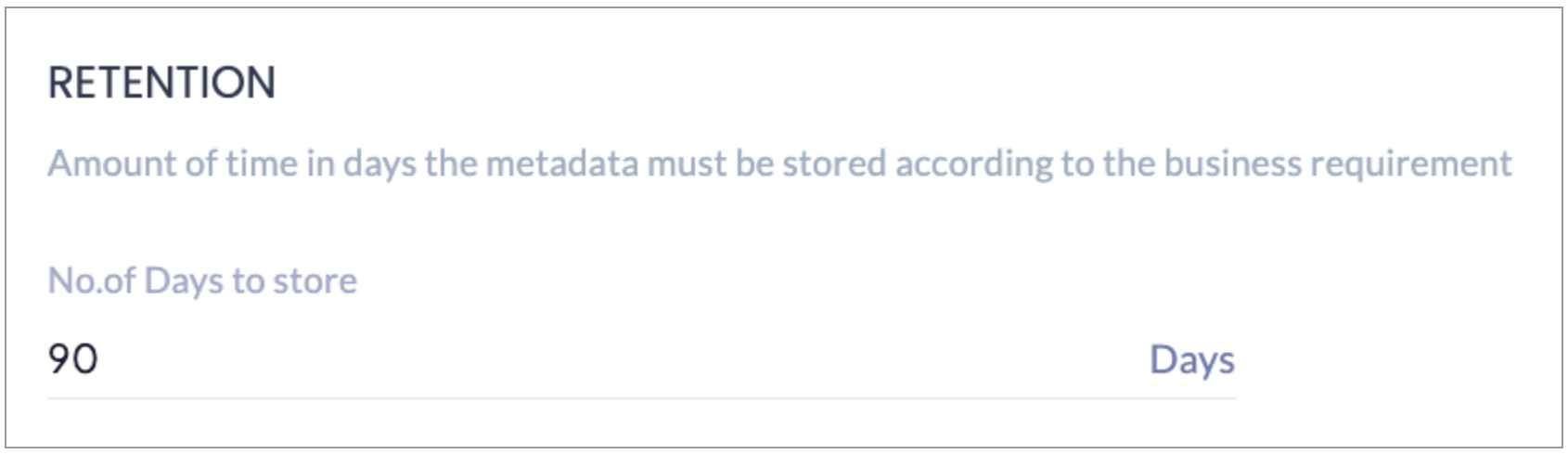
The Run Now button is used to run the push-down metrics, as per the parameters.

Repository

The data repository-related settings can be handled using the Repository settings dashboard as shown below.



- **Retention**: It is used to specify the time for which the metadata is to be stored. Users can enter the **number of days to store** the metadata.



- **Preview**: It is used to specify whether the failed records for resolving the data issues are to be shown or not to be shown by using the toggle on or off respectively. The **number of records to show the issue resolution** is currently limited to 20.

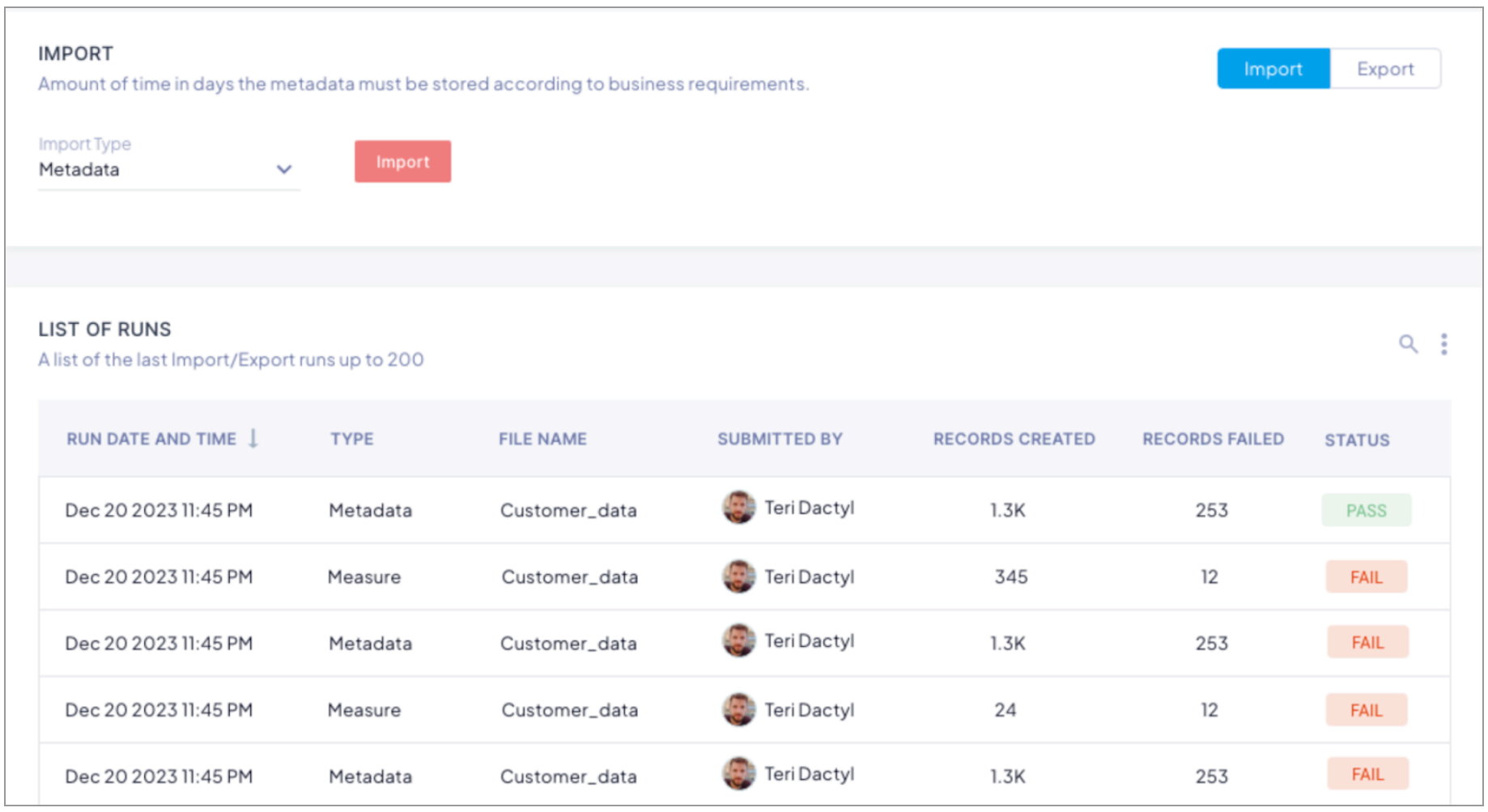
- **Versioning**: The versioning-related toggle is used to track the version that is related to the asset and maintain versioning for each incumbent asset.

Utility

The Utility-related settings (**Platform** > **Utility**) help the user with importing or exporting the data in the form of measures, metadata information, and users.

Import Utility

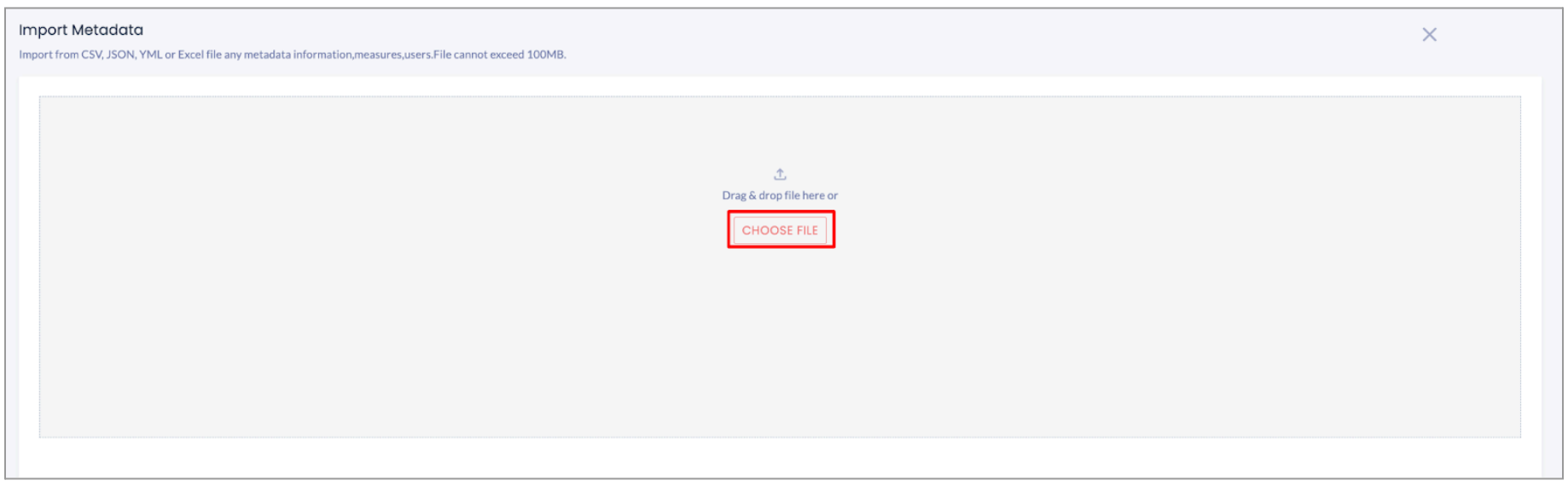
The Import Utility is used for importing the CSV, JSON, YML or Excel file with metadata information, measures, and users. The file to be imported should not exceed 100MB.



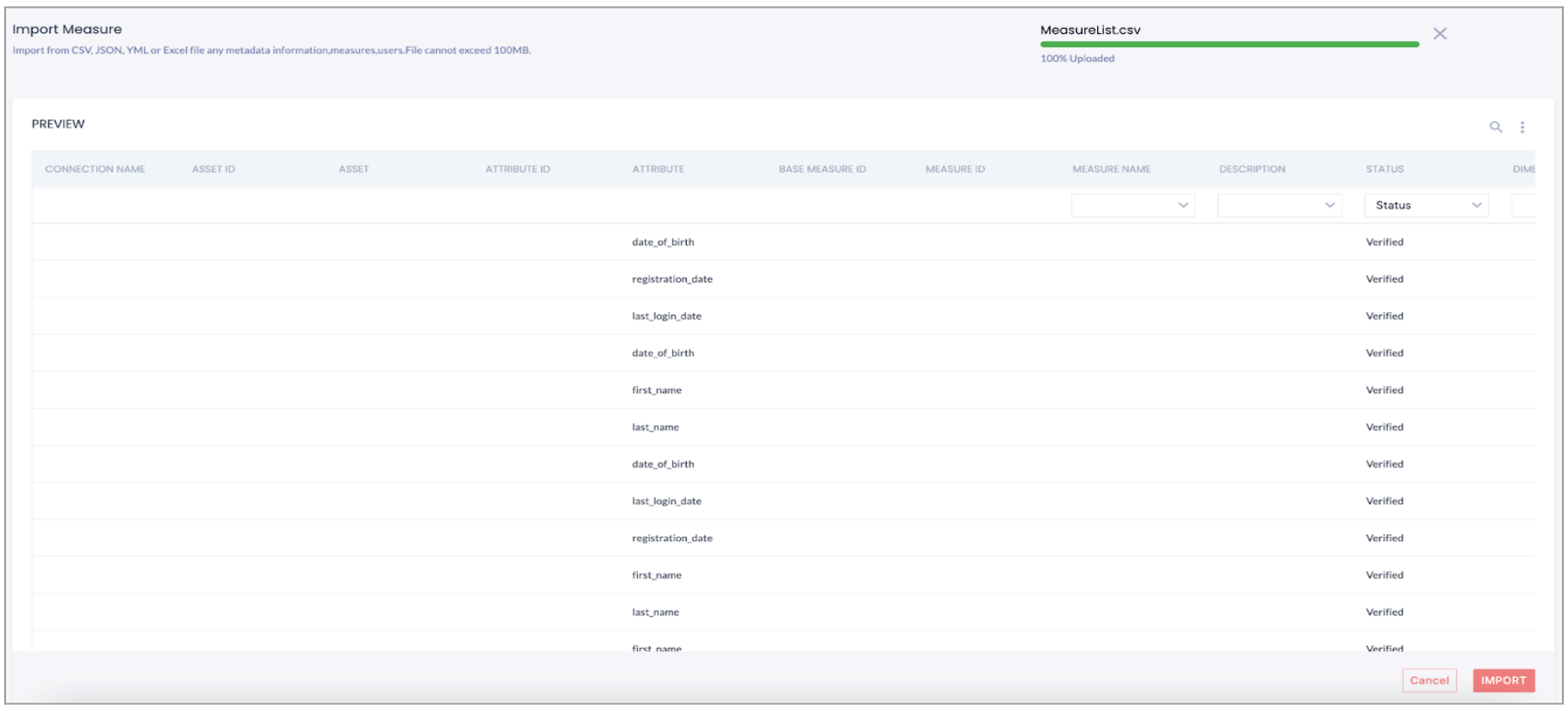
To import the information, conduct the following steps:

1. Go to the Import tab as shown below. Select the **Import data type** from the set of drop-down as shown in the below figure. Further, click on IMPORT.

2. The user is then taken to the file upload screen as shown below. Click on the **Choose File** button. Select the appropriate file from the local drive to be uploaded.

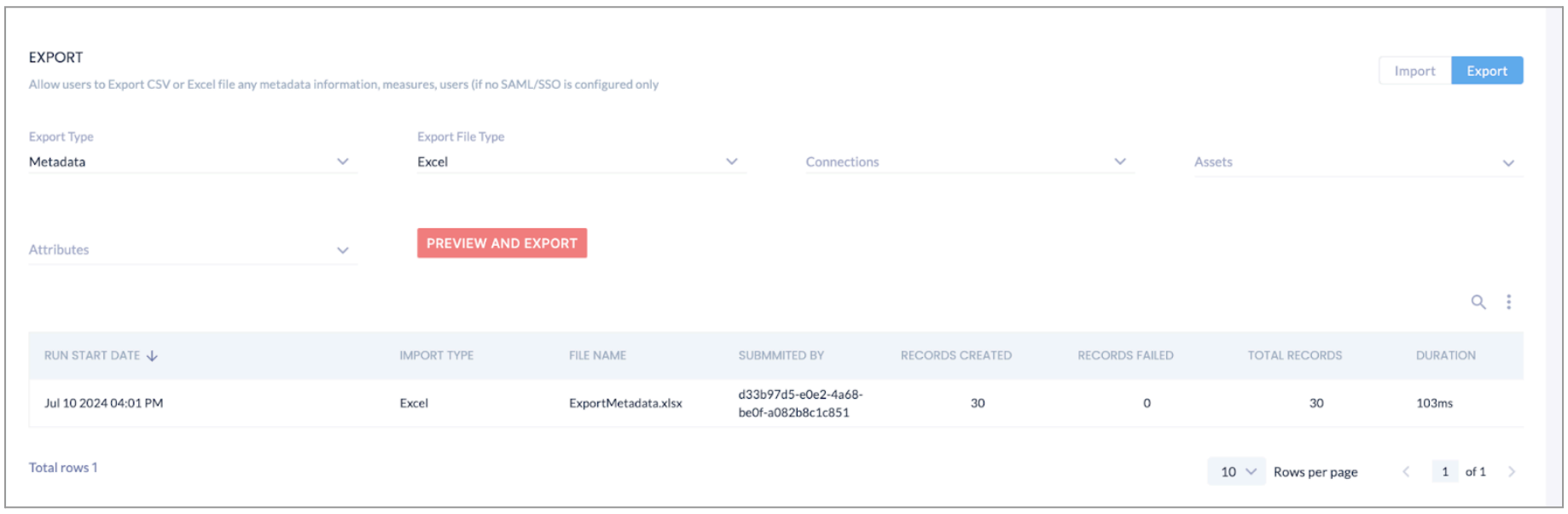


3. On uploading a file, the imported data can be viewed as shown in the below example. Click on the Import button to import the data.

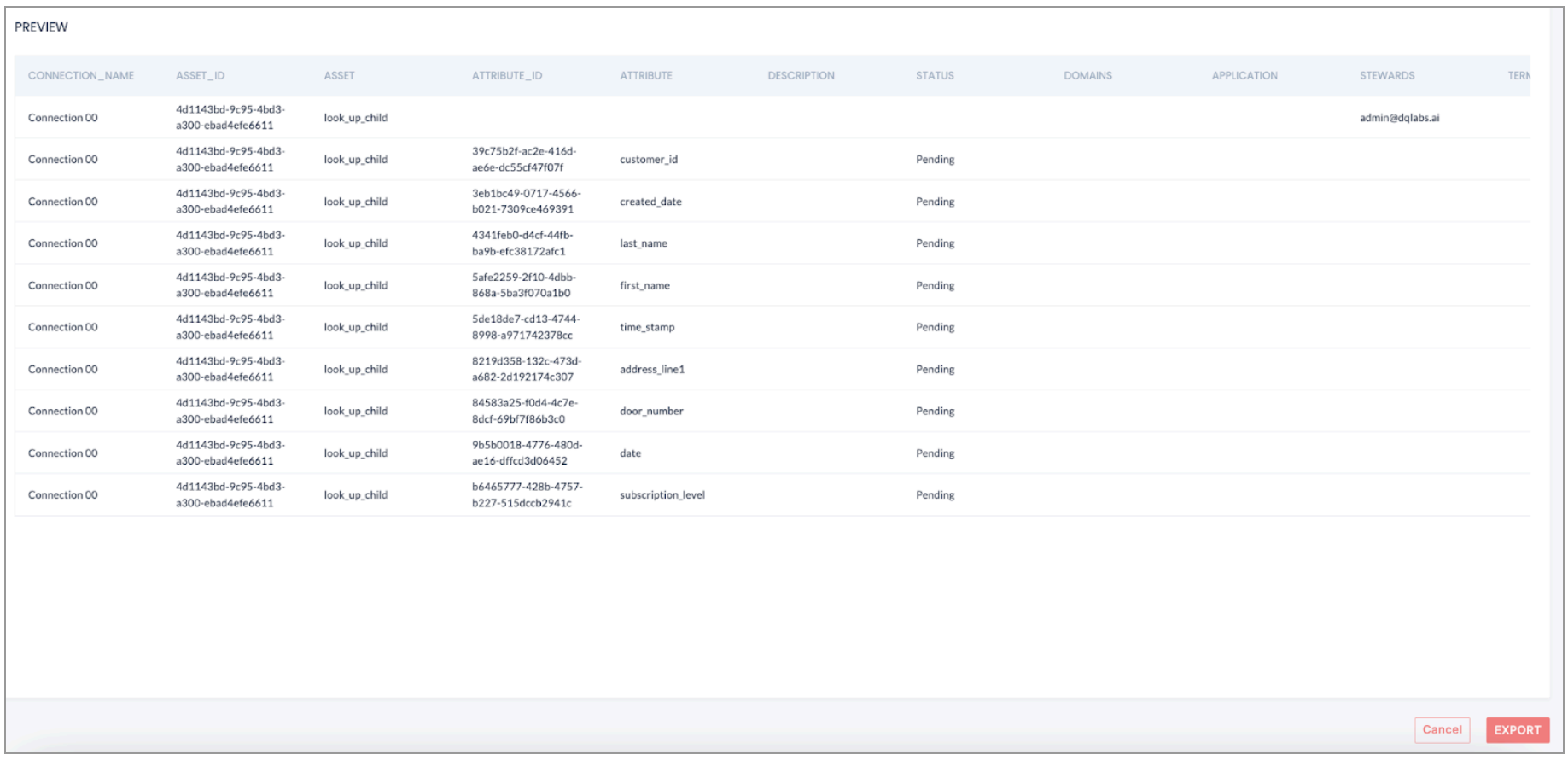


Export Utility

The users can export the CSV or Excel file for any metadata information, measures, users related data using the Export Utility.



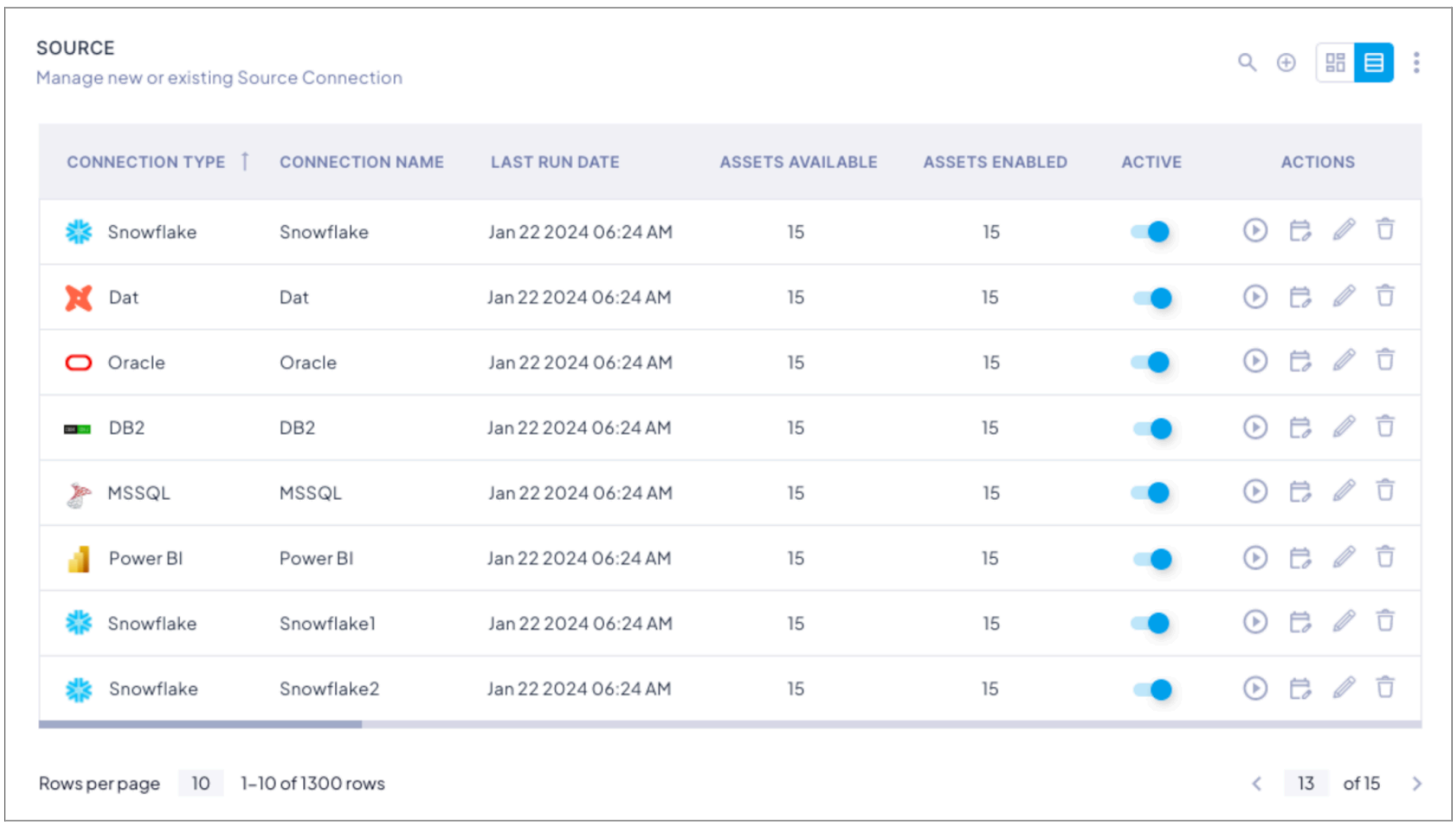
- Select the **Export Type** from the drop-down first. Next, select the **Export File Type** either Excel or CSV.
- Select **Connection** from the available set of dropdowns. After the selection of Connection, select the **Assets**, and further select the Attributes. Multiple selections can be made for Connections, Assets, and Attributes.
- After selecting the appropriate details, click on Preview and Export, the Preview of the data is shown as shown in the below example.



- The file is then downloaded on the local disk via the browser.

Connect

The Sources screen is used to add or manage the existing connections with Quest DQ



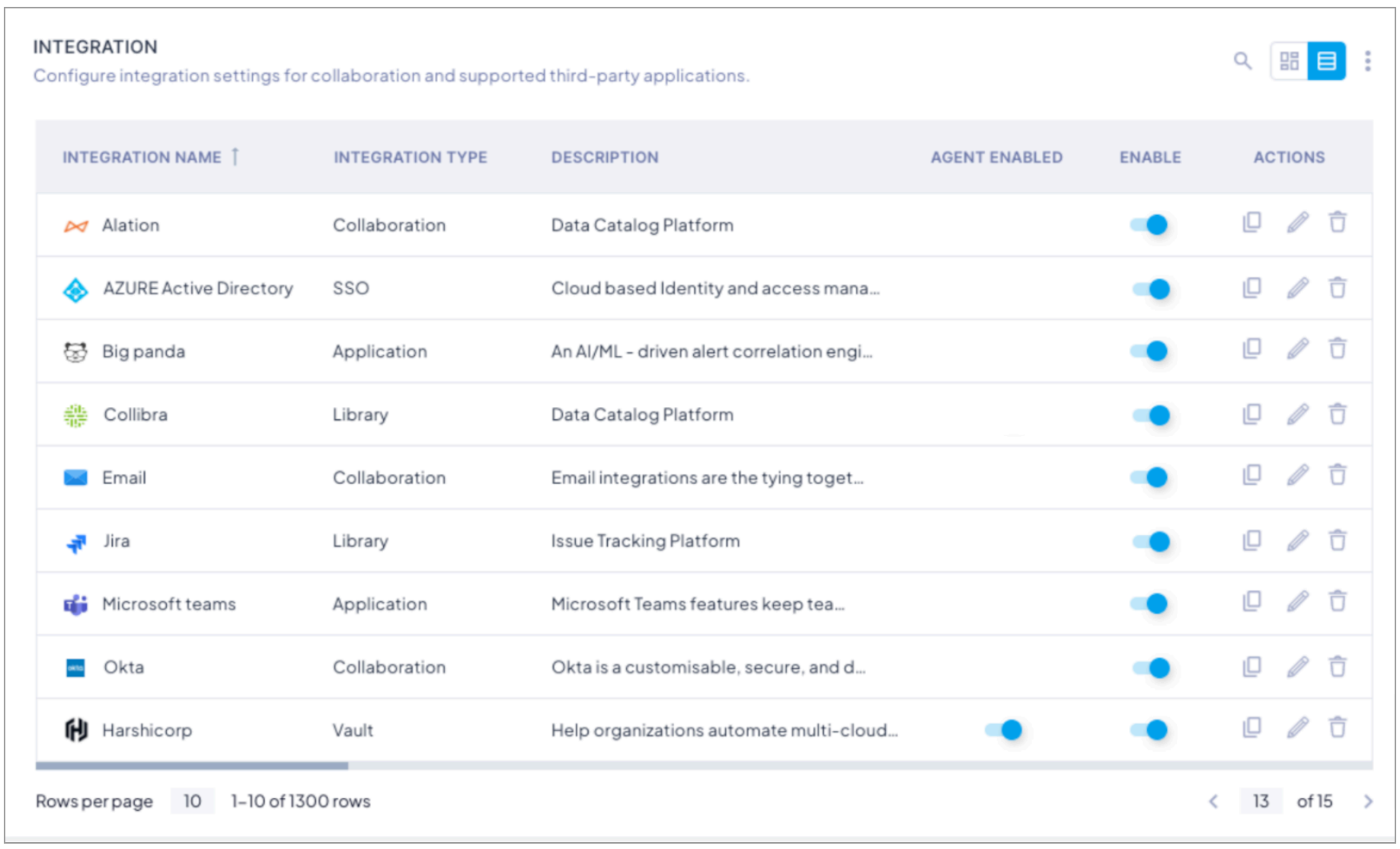
- **Enabling and disabling the connection**: For enabling and disabling the connection, under the ENABLED column, check the toggle button. If the toggle button is ON, the connection is enabled.

Term	Description

Connection Name	This shows the name of the connection as per the metadata received from the connection.
Connection Type	The Connection Type shows the data warehouse from which the connector information is fetched.
Run Date	The last job run date is displayed under this column.
Asset Available	The Asset Available shows the name of the assets available for a given connection.
Asset Enabled	Of the available assets, the asset enabled shows those that are enabled for the given connection, for use.

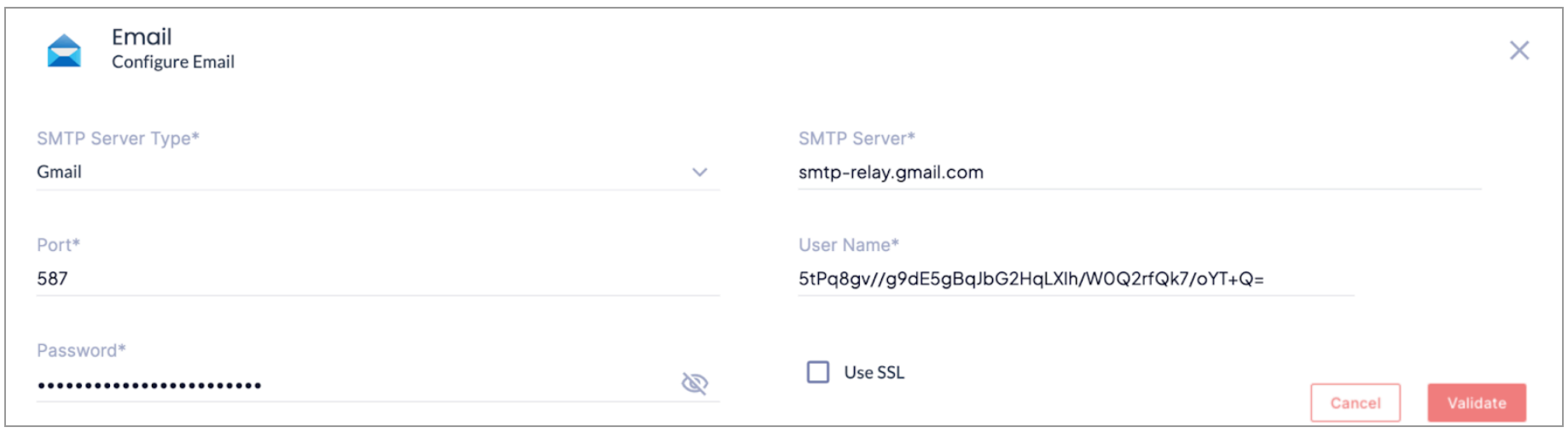
Integrations

Type	Connector	Description
Collaboration	Big Panda	IT operations platform that uses machine learning to automate incident management and alert correlation
Collaboration	Jira	Project management and issue tracking tool from Atlassian, commonly used for agile software development.
Notification	Slack	Team collaboration tool offering real-time messaging, file sharing, and integrations with various productivity apps
Notification	Teams	Microsoft's collaboration platform provides chat, video conferencing, file sharing, and integration with Office 365 apps.



The following is the set of Integrations that are available on the Quest DQ platform. To initiate an integration from the available set of options,

1. Click on the integration name row, for example, an Email integration is to be added. Click on the Email under Integration Name, and a form opens as shown below.



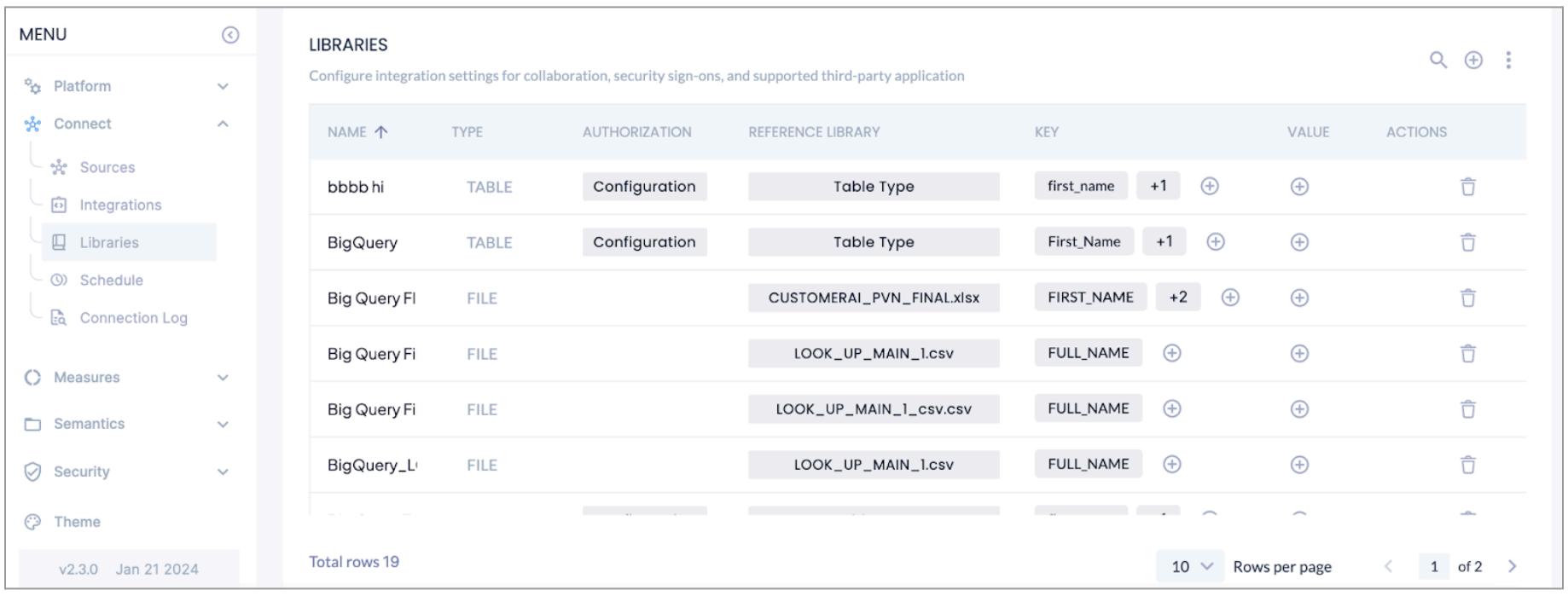
2. Enter the field-level information for configuring the email-related integration, and further click on the Validate button. When the connection gets validated successfully. The Agent Enabled column shows the toggled-on, as shown under the Integration landing page.

The integrations and the field-level information that are required for the integration to be activated are tabulated below.

Integration Name	Field Level Information Required
Big Panda	• URL • App Key • Organization Token • Push Alerts (High, Medium, Low) • Push Issues
Email	• SMTP Server Type • SMTP Server • Port • User Name • Password
Jira	• API Endpoint • User Name • API Key • Project ID • Enable WebHook (Checkbox)
Microsoft Teams	• WebHook URL

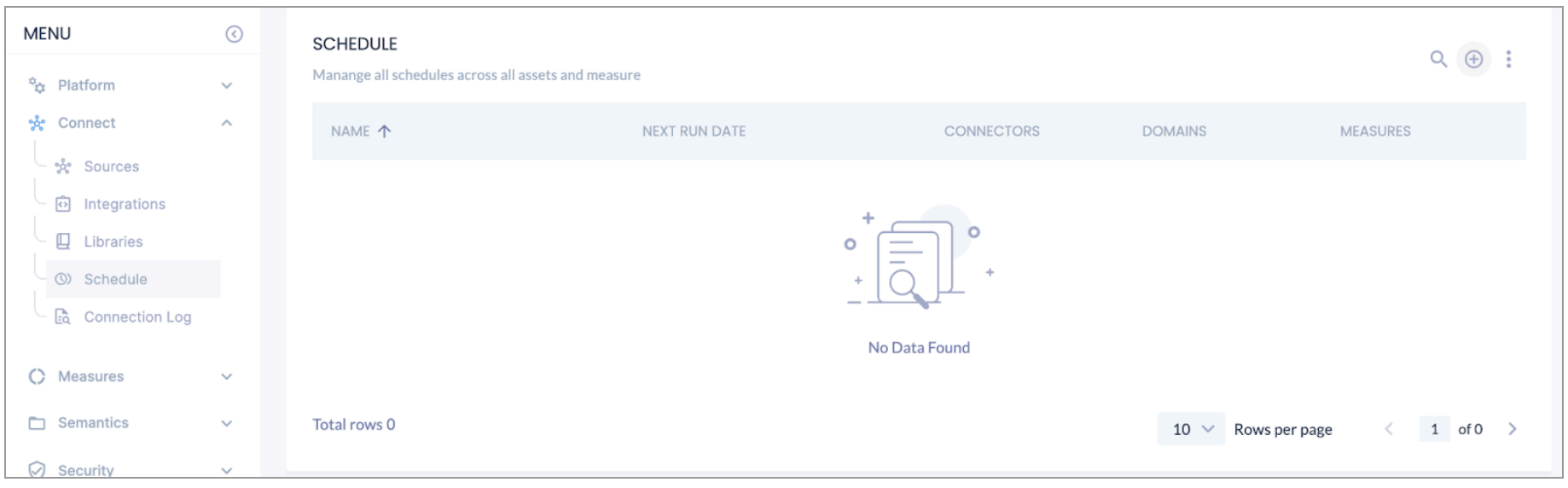
Libraries

Libraries can be used to add reference data for lookup measures creation.

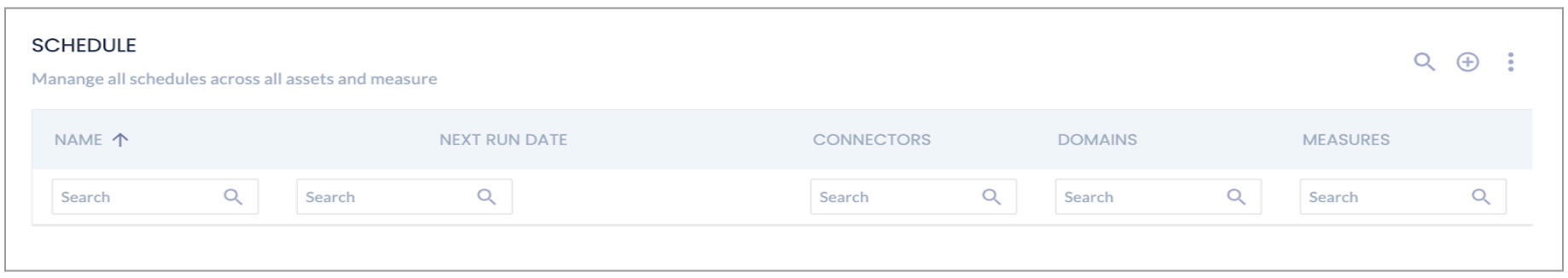


Schedule

The schedules across all the assets and measures can be set up. Go to **Connect** > **Schedule** for setting up the same.



To search for an entry about the schedule in the existing column, click on the button to search in the intended column in the search fields, as shown below.



Adding a Schedule

To add a schedule, click on the button as shown below. A form field, as shown below, appears.

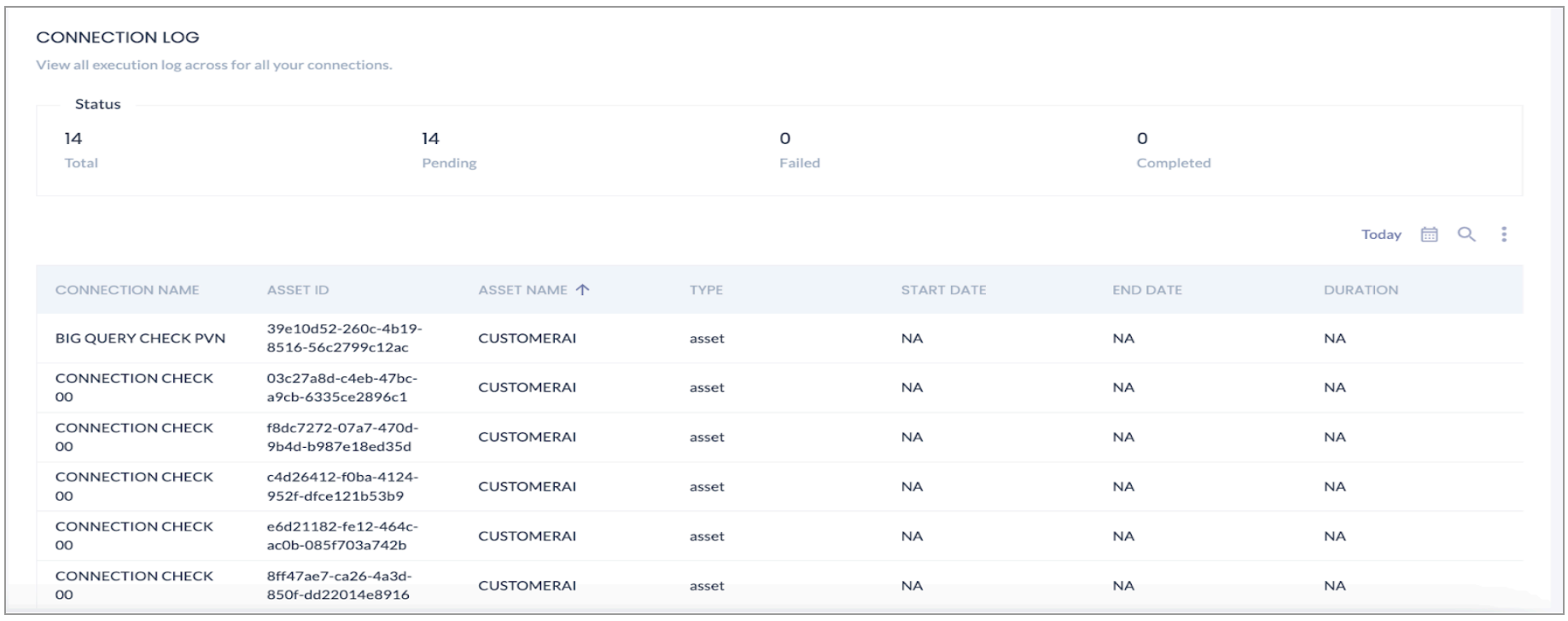
The form fields are explained below.

Term	Description
Name	Add the name of the schedule here; it is a mandatory field.
Start Date	The start date and time for the schedule can be added here.
Repeat Frequency	The repeat frequency for the schedule can be added here

- **Repeat Every**: The Repeat Every drop-down shows the time interval to be selected for the schedule frequency to be executed. It can be in days, hours, minutes, weeks, or months. It is a mandatory selection, which has Days as the default selection

Connection Log

The Connection Log shows the progress of existing connections of various connectors with Quest DQ, along with their start date, end date, status, etc.



- **Status**: The status shows the count of connections in terms of **Total** (total connections), **Pending** (Pending to be completed), **Failed** (failed connections), and **completed** (connection completed).
- **Connection Name:** The connection name is populated under this column.
- **Asset ID**: The asset ID assigned to the particular asset as per the metadata associated with the asset, is populated under this column.
- **Asset Name**: The asset name as per the metadata is populated under this column.
- **Type**: The data type for the given connection is shown under this column.
- **Start Date:** The Connection start date is populated under this column.
- **End Date:** The Connection end date is populated under this column.
- **Duration:** The duration for which the connection is active is shown under this column.
- **Total tasks:** The total number of tasks that are present for a given connection is shown in this column.
- **Completed Tasks:** The tasks whose 'observe' and 'measures' related computations are completed are termed completed tasks.
- **Failed Tasks:** The failed tasks corresponding to the given connection are populated under this table.
- **Status:** The status of the jobs corresponding to a given connection is shown here.
- **Actions:** For a given connection, there are two actions that can be performed**, Run Now** and **Kill Job.**
- **Run Now:** The job/s corresponding to a given connection can be run using this button.
- **Kill Job**: To kill the job corresponding to the given connection, press this button.
- **Category, Attribute, and Measure:** Click on the drop-down arrow under the Action column corresponding to a given connection.

Semantics

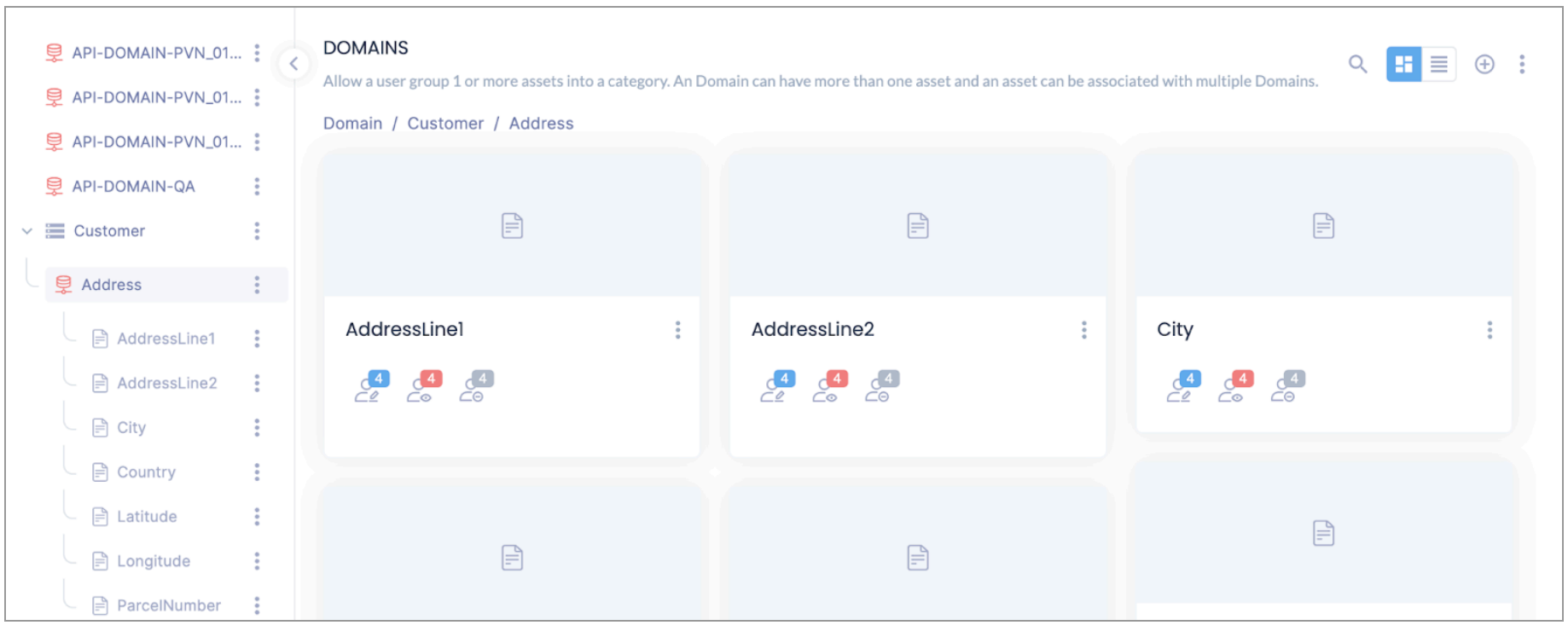
The Discover semantics page allows users to view the assets in the platform by their semantic layer definition, such as Domains, Applications, and Tags. The semantics view allows the users to use the sidebar to expand and navigate through subcategories of the selected semantics layer definition and view its assets.

A semantic overview page contains the following information: Summary, List of assets, and Assets grouped by asset type and attributes. The user can sort, search, and filter the assets in the list using the list functions. The user will be able to switch between different semantic definitions by clicking on the tab option in the left-side menu.

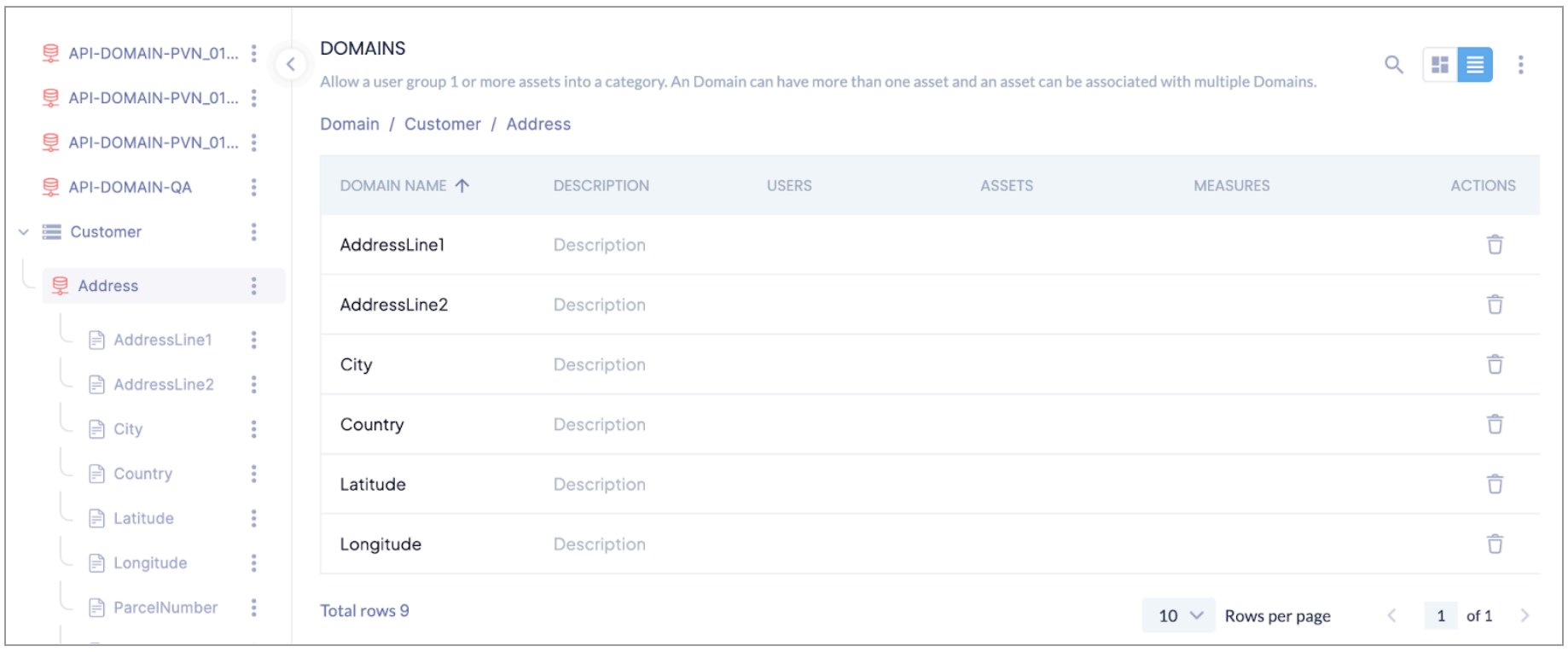
Domains

The domains allow the user to group one or more assets into a category. A domain can have more than one asset, and an asset can be associated with multiple domains.

In **grid view,** the domains are displayed as follows:



In **list view**, the domain list looks like the following:

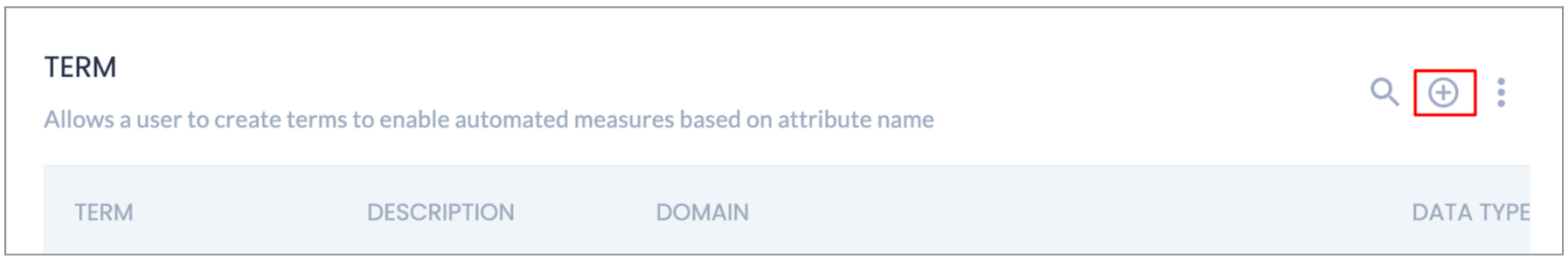


Terms

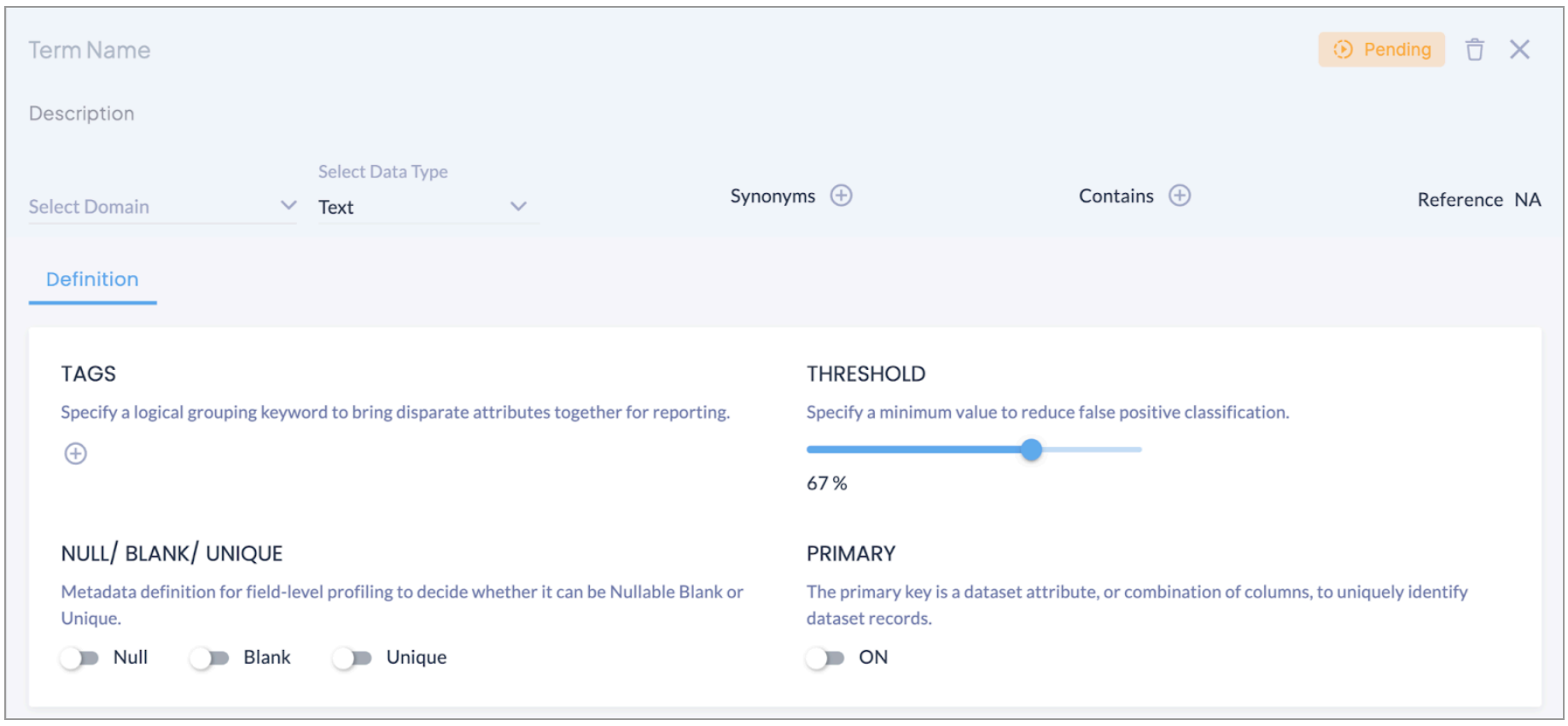
The terms allow the users to create terms that allow for setting up automated measures that can be linked to attributes in a given asset. To access the terms-related settings, click on the user icon, and a drop-down menu opens. Further, click on **Settings** > **Semantics** > **Term**.

Creating a new term

To create a new term, click on the + button as shown below.



When the user clicks on the + button, a form opens as shown below. To start with, enter the **term name** and the **description** for the term.



Further, select the relevant domain from the **Select Domain** drop-down. Select the data type for the domain. The available set of data types is the following:

Data Type	Description
Bit	The data type is used typically to save Boolean values
Text	The text type supports storing string values
Integer	The whole number values without the fractional ones can be saved under this category
Numeric	It stores the exact numerical values, especially where computed values are involved
Money	The currency-related values can be stored under this data type
Date	The date-related information can be stored under this data type
Time	The time-related data can be stored under this data
DateTime	Represents the datetime datatype
DateTimeOffset	Represents the datetime datatype with timezone
binary	Represents the variant datatype

- **Tags** - Associated tag of the attribute
- NULL/BLANK/UNIQUE - Defines the NULL/BLANK/UNIQUE/properties of an attribute
- PRIMARY - Defines the Primary key properties of an attribute
- Threshold - The percentage of records of an attribute that should contain the defined properties in order to auto-map the semantic field type to the given attribute
- Length - Defines the MIN and MAX length of an attribute
- Range - Defines the MIN RANGE and MAX RANGE of an attribute (Only applicable for Integer and Numeric data types)
- Patterns - The record patterns that are in the given field type
- Enumeration - The valid values in the given field types
- Measures - Custom measures that should be applied to the attribute
- Linked Assets - The semantic is linked to the attribute, and it identifies

After defining the business term with required properties, descriptions, rules, measures, and linked assets, the user can review and change the status to “Verified”.

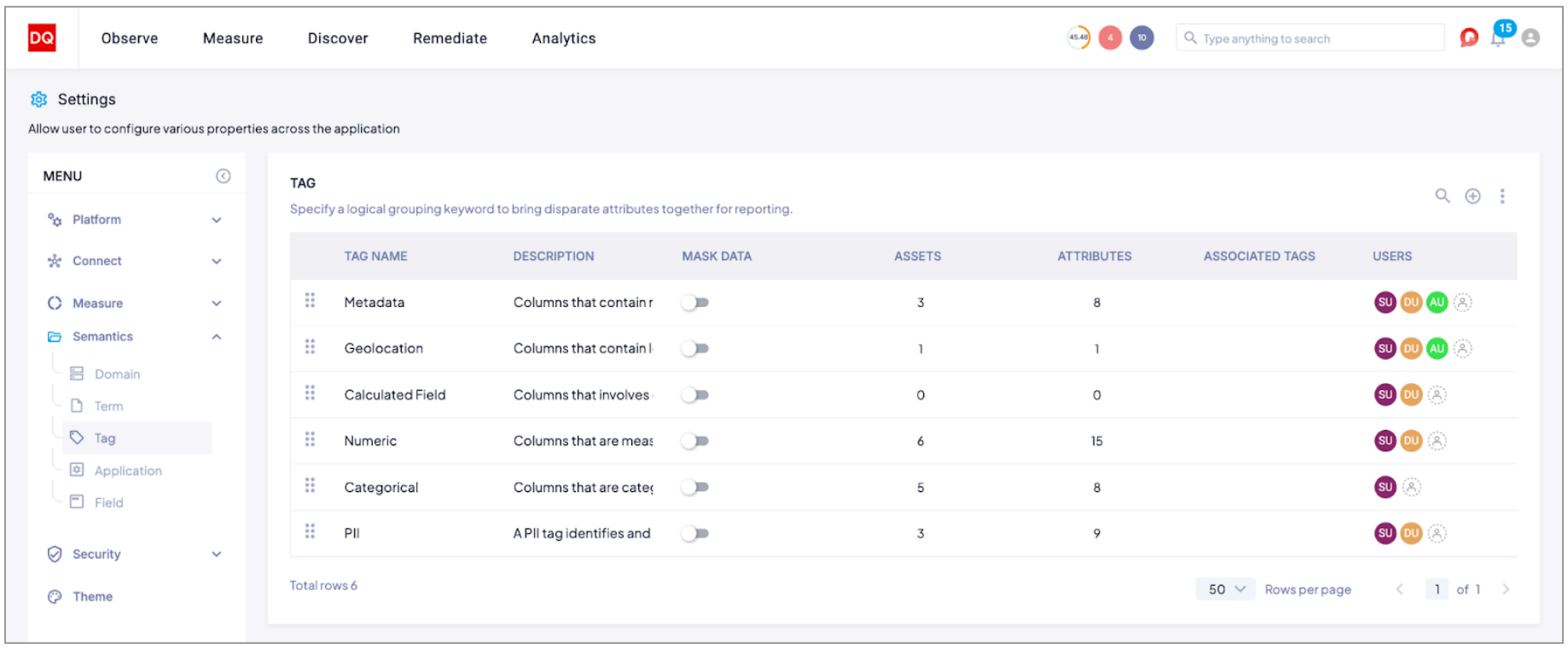
Tags

Tags are used to specify a logical grouping keyword to bring disparate attributes together for reporting.

A tag can be created by clicking on the + icon in the top right corner below the fields.

- Add name

- Add description
- Usage
- Linked attributes
- Associated tags
- Color
- Select action to save or delete

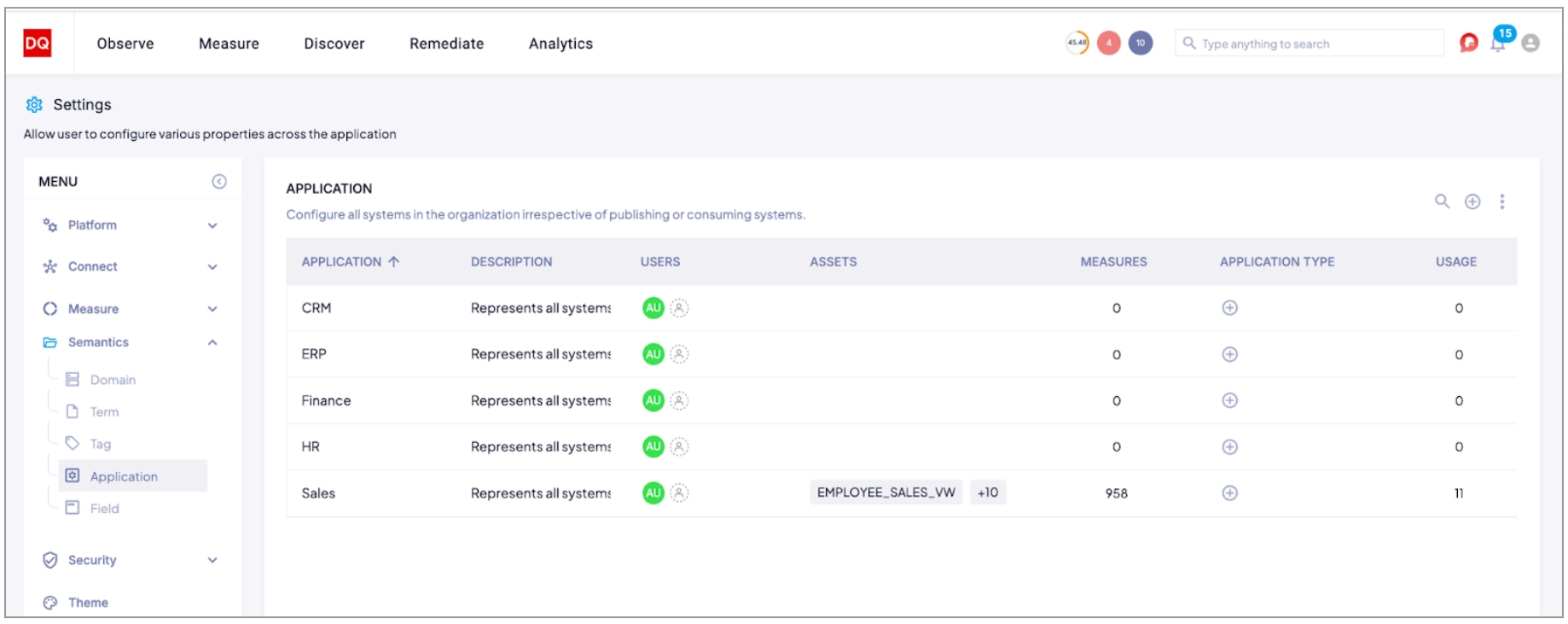


Application

Quest DQ provides the ability to group the datasets by application name, which can represent the source system from which the data is brought in.

An application can be created by clicking on the + icon in the top right corner below the fields.

- Add name
- Add description
- Assign type
- Usage
- Linked asset
- Add color
- Select action to save or delete



Security

The Security-related settings in Quest DQ are intended for features, capabilities, best practices, and authorizing access for protecting sensitive data and maintaining data integrity throughout the system. The following sections explain the settings that the user can use to enhance the security of the Quest DQ application. Data security is crucial for protecting sensitive and confidential information and for maintaining the integrity and availability of data.

Quest DQ security is the practice of protecting sensitive and confidential information from unauthorized access, use, disclosure, disruption, modification, or destruction. Data security can be achieved through a combination of physical, technical, and administrative controls.

By selecting Security under the settings tab, the user can see all the users across the organization and their permissions for the portal.

Roles

Roles allow the admin users to define a set of permissions or access rights assigned to individuals based on their job responsibilities or functions within an organization. These roles help regulate who can view, modify, or manage different parts of the platform.

By default, there are the following roles with respective permissions:

MENU	PAGE	SECTIONS	ADMIN	STEWARD	ENGINEER	USER
HOME			ON	ON	ON	OFF
	Shortcuts		ON	ON	ON	OFF
	Stats		ON	ON	ON	OFF
OBSERVE			ON	ON	ON	OFF
	Data		ON	ON	ON	OFF
		Details	ON	ON	ON	OFF
		Measures	ON	ON	ON	OFF
		Attributes	ON	ON	ON	OFF
		Usage	ON	ON	ON	OFF
		Lineage	ON	ON	ON	OFF
		Custom	ON	ON	ON	OFF
		Conversations	ON	ON	ON	OFF
		Query Assets	ON	ON	ON	OFF
		Approve (Verified or Deprecated)	ON	ON	ON	OFF
	Pipeline		ON	ON	ON	OFF
		Runs	ON	ON	ON	OFF
		Jobs	ON	ON	ON	OFF
		Tasks	ON	ON	ON	OFF
		Tests	ON	ON	ON	OFF
		Attribute	ON	ON	ON	OFF
		Complied SQL	ON	ON	ON	OFF
		Lineage	ON	ON	ON	OFF
	Report		ON	ON	ON	OFF
		Overview	ON	ON	ON	OFF
		Lineage	ON	ON	ON	OFF
			ON	ON	ON	OFF
	Usage		ON	ON	ON	OFF
MEASURE			ON	ON	ON	OFF

		Validate SQL	ON	ON	ON	OFF
		Approve (Verified)	ON	ON	ON	OFF
		Publish (Active or Inactive)	ON	ON	ON	OFF
		View Exceptions	ON	ON	ON	OFF
DISCOVER			ON	ON	ON	ON
	Assets		ON	ON	ON	ON
	Semantics		ON	ON	ON	ON
	Converse		ON	ON	ON	ON
REMEDIATE			ON	ON	ON	OFF
	Alerts		ON	ON	ON	OFF
	Issues		ON	ON	ON	OFF
SETTINGS			ON	ON	ON	OFF
	Platform		ON	ON	ON	OFF
		Configuration	ON	ON	ON	OFF
		Repository	ON	ON	ON	OFF
		Utility	ON	ON	ON	OFF
	Connect		ON	ON	ON	OFF
		Sources	ON	ON	ON	OFF
		Integrations	ON	ON	ON	OFF
		Libraries	ON	ON	ON	OFF
		Schedule	ON	ON	ON	OFF
		Connection Log	ON	ON	ON	OFF
	Measures		ON	ON	ON	OFF
		Auto	ON	ON	ON	OFF
		Advanced	ON	ON	ON	OFF
		Profiling	ON	ON	ON	OFF
		Dimensions	ON	ON	ON	OFF
	Semantics		ON	ON	ON	OFF
		Domain	ON	ON	ON	OFF
		Term	ON	ON	ON	OFF
		Tag	ON	ON	ON	OFF

		Application	ON	ON	ON	OFF
		Product	ON	ON	ON	OFF
		Custom	ON	ON	ON	OFF
	Security		ON	OFF	ON	OFF
		Roles	ON	OFF	ON	OFF
		Users	ON	OFF	ON	OFF
		SSO	ON	OFF	ON	OFF
		API	ON	OFF	ON	OFF
		License	ON	OFF	ON	OFF
		Audit Log	ON	OFF	ON	OFF
		Activity Log	ON	OFF	ON	OFF
	Theme		ON	ON	ON	OFF
DASHBOARD			ON	ON	ON	OFF
ACTIONS						
	Create/Edit/Delete Measure		E	V	V	N
	Create/Edit/Delete Semantics		E	E	V	N
	Create/Edit/Delete Sources		E	E	V	N
	Create/Edit/Delete Integrations		E	V	V	N
	Create/Edit/Delete Libraries		E	V	V	N
	Create/Edit/Delete Schedules		E	V	V	N
	Create/Edit/Delete Dashboard		E	V	V	N
	Create/Edit/Delete Issues		E	E	V	N
	Run Jobs		E	E	V	N

The admin/privileged user will be able to create new roles by clicking on the "+" icon. The roles in Quest DQ have the following two sections:

- Features/Module - Provides access to features/module by enabling the toggle
- Actions - Provides access to view, edit, or modify a detail in the page

Menu	**Tooltip**
HOME	Access to the HomePage
Shortcut	Ability to add a shortcut on the homepage
Stats	Ability to view metrics on the homepage
OBSERVE	Access to the observability module
Data	Ability to access data observability
Measure	Ability to access measures in the asset
Attributes	Ability to access attributes in the asset
Usage	Ability to access usage for the asset
Lineage	Ability to access lineage for an asset
Custom	Ability to access custom fields in the asset
Conversation	Ability to access conversations in the asset
Query Asset	Ability to query an asset from the query preview
Approve	Ability to change the status of the asset
Pipeline	Ability to access pipeline observability
Run	Ability to access runs in the pipeline observability module
Job	Ability to access jobs in the pipeline observability module
Task	Ability to access tasks in the pipeline observability module
Test	Ability to access tests in the pipeline observability module
Attribute	Ability to access the attributes for a task
Source Code	Ability to access the source code for a task
Lineage	Ability to access the lineage for a task
Report	Ability to access the report observability module
Overview	Ability to access the overview page of a report
Lineage	Ability to access the lineage for a report
Usage	Ability to access the usage for a report
MEASURE	Access to the measure module
Validate SQL	Ability to validate and preview records for a measure query
Approve (Verified)	Ability to mark a measure as verified
Publish (Active or Inactive)	Ability to change the status of the measure
Measure Preview	Ability to view record preview for the measure
Query	Ability to view the source SQL for the measure

DISCOVER	Access to the discoverability module
Assets	Ability to access the asset discover page
Semantics	Ability to access the semantics page
Converse	Ability to access the GEN AI co-pilot
REMEDIATE	Access to the remediation module
Alerts	Ability to access the alerts page under remediation
Issues	Ability to access the issues page under remediation
Dashboard	Ability to access the dashboard module
SETTINGS	Access to the settings module
Platform	Ability to access platform settings
Configuration	Ability to modify platform configurations
Repository	Ability to modify repository configurations
Utility	Ability to access import and export functionality
Connect	Ability to access connections in the platform
Sources	Ability to access data source connections
Integrations	Ability to access integrations
Libraries	Ability to access reference libraries
Schedule	Ability to access master schedules
Connection Log	Ability to access consolidated job logs
Measure	Ability to modify measures in the platform
Auto	Ability to manage OOTB measures in the platform
Advanced	Ability to manage advanced measures in the platform
Profiling	Ability to access profiling settings
Dimension	Ability to manage and measure dimensions
Semantics	Ability to access semantic layer definitions
Domain	Ability to access domains
Term	Ability to access terms
Tag	Ability to access tags
Application	Ability access applications
Product	Ability to access products
Field	Ability to access custom fields
Security	Ability to access security settings
Role	Ability to access roles and permissions
User	Ability to access users
SSO	Ability to set up SSO/SAML integrations
API	Ability to access Quest DQ API

License	Ability to manage license
Audit Log	Ability to access audit logs
Activity Log	Ability to access activity logs
Theme	Ability to manage theme settings
ACTION	Definitions of what actions a role can perform in a module
Asset	Ability to manage assets
Measure	Ability to manage measures
Semantics	Ability to manage semantic definitions
Sources	Ability to manage datasource connections
Integrations	Ability to manage third-party tools integrations
Libraries	Ability to manage reference libraries
Schedule	Ability to manage the master schedule
Dashboard	Ability to manage dashboards
Alerts	Ability to manage alerts
Issues	Ability to manage issues
Run Job	Ability to run jobs at all levels
API	Ability to manage Quest DQ APIs

User

Manage user across the organization and their related permissions for the portal. It shows

- Total users
- Active users
- Inactive users
- Outstanding Invites

It allows you to invite the user by entering:

- Email
- Role - Role can be assigned based on the access needed

The association allows the users to specify which assets users should have access to based on the semantic definition. A user can have either full access or no access to a semantic definition, such as domains, applications, or tags. If a domain is selected under full access for a user, then the role will have access to only the asset mapped under the domain. If the domain is selected under no access, then the user will not have access to any of the assets under the domain.

The admin/privileged user can modify the invite email before sending using the view invite option, also there is the ability to search for a user.

At the end, the user can view/edit all the users in a table with the ability to filter out the columns, which are:

- First Name - First name of the user
- Last Name - Last name of the user
- Title - Title of the user
- Email - Email address of the user
- Role - Assign role for user
- Associations – Restrict the permission of the user to a specific Semantics
- Active - Active/inactive user
- Action - To delete the user.

SAML/SSO

This section provides the details on the SSO SAML integration and Role-Based Security Access

Azure Active Directory

Quest DQ allows you to integrate your existing Azure Active Directory identity provider and access the platform using Single Sign On. Using SAML, all users in the domain will be able to log in to the sign-in page of Quest DQ.

Quest DQ uses email as the claim information, and you need to create a federation.xml file in your SAML provider and then update it in the Quest DQ platform. Refer to the SSO integration section for more details.

API Settings

The API settings allow the admin/privileged user to manage the Quest DQ API. The user creates an API by clicking on the "+" icon and by providing the name of the API and expiry date, the client ID and client secret are generated. Using these keys, the user can make calls to the Quest DQ API.

License

The license key provides the ability to manage the license. The admin/privileged user can provide the license key and activate the license for the portal. The user can view the package name, package mode, start date and time, and end datetime of the license with the license key

Audit Logs

Quest DQ also provides Audit trail abilities, which are important for security because they allow administrators to track and review actions performed by users. The audit trail can also be used for compliance purposes, as it provides a record of all actions performed within the Quest DQ tool.

Quest DQ provides the ability to track user activity in the platform and provides an audit history on each asset with respect to what changes are made to the asset. This helps in monitoring the activity in the platform

The audit trail can be useful in the following scenarios:

1. Security: Audit trails provide a record of all user activities, which can be used to detect and investigate incidents.

2. Accountability: Audit trails help establish accountability by providing a clear record of who performed what actions, when, and from what asset.

3. Auditing and reporting: Audit trails can be used for auditing and reporting purposes, providing valuable insights into user activities and system usage patterns.

4. Troubleshooting: Audit trails can be useful for troubleshooting issues, as they provide a record of all relevant activities that took place leading up to the issue.

5. Data integrity: Audit trails help ensure the integrity of data by providing a record of all changes made to it.

The following activities are tracked in the user activity :

MODULE	ACTION
Asset	User adds a domain
	User removes a domain
	User adds an application
	User removes an application
	User adds an identifier key
	User removes an identifier key
	User-enabled semantic term
	User adds Description
	User updates description
	User creates a conversation(Under assets and issues)
	User Replies to a conversation(Under assets and issues)
	User edits a conversation (Under assets and issues)

	User deletes a conversation(Under assets and issues)
	User mentions another user in the conversation(Under assets and issues)
	User creates a measure
	User modifies a measure
	User deletes a measure
	User runs a job
	User triggers semantics discovery
	User schedules the asset
	User removes a schedule
	User updated status
	User adds a column
	User deletes a column
	User changes the status of the attribute
Attribute	User adds Description
	User updates description
	User enables- primary key toggle
	User disables primary key toggle
	User adds a term
	User removes a term
	User adds a tag
	User removes a tag
	User enables advanced profiling
	User runs a job
	User creates a measure
	User modifies a measure
	User deletes a measure
	User changes the status of the attribute
Stand-alone measure Measure	User creates a measure
	User modifies a measure
	User deletes a measure
	User changes the status of the measure
Search	User searches for an asset
	User filters an asset
Alert	User Marks as Alert as Normal
	User Marks as Alert as Outlier
Issue	User creates an issue

	User updates properties for an issue
	User delets an issue
	User changes the status of an issue
	User changes the priority of an issue
Settings	User creates a connection
	User updates a connection detail
	User deletes a connection
	User deactivates a connection
	User creates an asset
	User updates asset details in the connection page
	User adds/deletes database
	User adds/deletes schema
	User selects/deselects a column from the asset page
	User deletes an asset
	User unchecks an asset from the connections page
	User creates an integration (Collaboration/SSO/Apps)
	User updates an integration (Collaboration/SSO/Apps)
	User deletes an integration (Collaboration/SSO/Apps)
	User deactivates an integration (Collaboration/SSO/Apps)
	User creates a Library (Table/File)
	User updates a Library (Table/File)
	User deletes a Library (Table/File)
	User deactivates a Library (Table/File)
	User creates a master schedule
	User updates a master schedule
	User deletes a master schedule
	User deactivates a master schedule
	User adds a semantics(Domain/Term/Tag/Applications/Fields)
	User updates a semantics(Domain/Term/Tag/Applications/Fields)
	User deletes a semantics(Domain/Term/Tag/Applications/Fields)
	User adds a subdomain
	User updates a subdomain
	User deletes a subdomain
	User adds a subtag
	User updates a subtag
	User deletes a subtag

	User invites a user from settings
	User updates user details form settings
	User changes password from settings
	User invites from the import user
	User deletes/inactivates as user
	User creates a role
	User updates a role
	User deletes a role

Activity Logs

The user activity section provides the following levels of audit trail of a user based on his/her activity in the Quest DQ platform. Navigate to security under the settings page and click on the “Activity“ tab.

The user activity information table shows the following details:

Column Name	**Description**
User	Shows the name of the user
Last Logged In	The date and time that the user recently logged in
No. Of Audit Logs	Number of actions performed inside Quest DQ
Login Count	Number of times the user successfully logged into the platform
Avg Duration (mins)	Average time spent inside Quest DQ in minutes
Min Duration (mins)	The minimum time spent inside Quest DQ in minutes
Max Duration (mins)	The maximum time spent inside Quest DQ in minutes

The user activity table can be sorted based on the column name. Expanding each user provides the information for each session, i.e, for each login

Column Name	**Description**
Session Start time	The date and time of the login
Session End time	The date and time of logout for that session
IP Address	The IP address from which the Quest DQ was accessed
Browser	The browser that was used to log in to Quest DQ
Audit Information	Displays all the activity that the user performed during the session with the following details ● Created Date ● Connection Name ● Asset Name ● Attribute Name ● Audit Log - Action performed

The user can download the audit information by clicking on the download icon on the activity page

Themes

The theme-related settings help the users to set up the appearance, style, and reporting-related settings that help the users to tweak the visualization of the platform. The settings have the following three tabs, explained with screens in the sections below:

Appearance

The logo to be added that is to be shown on the platform screens can be set up here.

- **Custom Logo**: The custom logo can be uploaded by clicking on the **Upload Logo** button.
- **Tagline**: The tagline for the branding can be added here. This tagline will appear below the logo on the logo page.
- **Customize Login Page Background**: The Customize Login Page Background section can be used to provide a background image/banner and content on the login page

To add the attributes, click on the button, to open the form for adding the login page background. The page looks as follows:

- Under the **Add Content** header, in the text area, add the content that would be displayed to the user who would be logging on to the platform.
- If you want to add the **background image**, it can be uploaded by clicking on the **+Upload** button. Click on the **Save** button after adding the intended details.

Style

- The Style tab helps users to edit the element-level features visible to the user on the screen. These elements include the font texts, buttons on the screen, tables, etc.
- The Element column has a set of drop-down, as shown, for example, **Typography**.
- The typography has the heading font, size, and color-related settings as shown below.

The preview of the changed settings can be viewed under the Preview Theme on the extreme right pane of the screen.

Reporting

The settings for PDF reporting can be handled from this dashboard

Glossary

- **Asset** - Properties of the business that contain data. Can include Data Sources, Tables, Views, or Attributes.
- **Attribute** - A field in the data set
- **Alert** - An automated warning when an asset does not align with targeted parameters set by the platform or user
- **Issue** - A manual log of asset inconsistency with targeted parameters set by the platform or user
- **Conversations** - Feature allowing users to rate and comment on assets, and host conversation threads as well as one-off comments
- **View** - the result of a stored query pertaining to one or more assets
- **Domain** - refers to a logical grouping of related data that has meaning to the business
- **Application** - system related to an asset. It could be a publishing or consuming system
- **Identifier** - Primary key or Composite key to identify duplicates
- **Term** - Business definition to define taxonomy, provide more context on its purpose
- **Tag** - specify a logical grouping keyword to bring disparate attributes together for reporting
- **Semantics** - Data discovery that uses advanced algorithms and machine learning to automate the process of analyzing and classifying attributes based on the semantic data model
- **Reporting** - allows for data collection on failed rows (either full or preview data only) around the primary key, composite key, and/or all attributes in case of failure
- **Versioning** - allows the user to version control and gather audit logs; highlights changes. Anomaly - specifies the number of runs to look back for measuring the threshold or forecasting. Connection - the means of connecting to a database
- **Flat files** - stagnant files that can be uploaded and processed, such as XML and CSV files
- **Data Health** - measures of data quality that include Completeness, Validity, Uniqueness, and Timeliness
- **Data Distribution** - definable measures that relate to the contents of the data, such as Validity
- **Data Statistics** - definable measures to figure out the suitability of data for its intended applications in data analytics, data science, or machine learning
- **Data Frequency** - definable measures to understand the format and enumeration of data assets
- **Duplicates** - The total number of duplicates in an asset or attribute for the given key
- **Freshness** - the freshness of the data in a given asset
- **Schema** - the total number of columns
- **Volume** - the total number of rows
- **Widgets** – Widgets are visual representations of certain Health, metadata, and other metric data points that have been extracted from Assets that a user has set up on Quest DQ.

OBSERVE

Data

The data module focuses on ensuring that the specific tables/views and query-based assets are functioning correctly, with reliable, accurate, and timely data.
Key aspects of data observability for data assets in Quest DQ include:

1. **Data Quality**: Ensuring the data in the asset meets defined quality standards based on the applied measures in Quest DQ.
2. **Alerts:** Anomaly detection based on continuous and automated data observability monitoring
3. **Issues**: Tickets created to manage and resolve the raised alert

The Observe Data module has the following sections, which allow the users to infer the required information about the data assets in the organization

Component	Description
Trend Chart	Allows the users to view the trend in DQ Score, Alerts, and Issues in the platform over a period of time
Analysis Chart	Allows the users to view the top assets with issues that the user should focus to improve the overall health of the assets
List of Assets	Allows the users to quickly view the assets in the platform

Summary

The summary of data violations across the asset data for today in terms of score, alerts, and issues. The data summary, that is, the chart shown as a summary to you, shows the data violations in terms of data quality score, the issues raised for the data violations, and the alerts raised for the data violations. There are two important visualizations here– Trend and Analysis.

Trend

The summary trend chart is a combination chart that represents the DQ score of each asset in a bar, and the count of alerts and issues is represented by separate line charts. The trend chart also contains a slider that allows users to view the required bar for a particular asset by moving the slider. The user can also filter the assets in the trend chart and the list by using the date filter.

Component	Description
X-axis	The date is represented on the X-axis
Y-Axis	DQ Score, Alerts, and Issues are represented on the Y-axis
Bar	Represents the DQ score of the asset and is colour-coded based on the quadrant configuration in settings
Red Line	Represents the count of issues
Grey Line	Represents the count of alerts

Operation:

- Hover over each of the vertical bars on the bar graph to check for the date-wise scores, issues, and alerts for analyzing the date-wise trends for the data quality insights.
- To change the date span for the graph, drag the slider between the appropriate dates to isolate the date span to be analyzed. The slider is shown below.
- The graph can be filtered for scores, issues, and alerts using the filter buttons at the bottom center, as shown below. All three can be visualized simultaneously on the Trends chart.
- The Search button can be used to filter search results and thereby visualize the filtered list table in the chart.
- The users can click on the bar in the chart to filter the results in the asset list below

List of Data Assets

The asset list view provides more details of the assets in a list format and allows users to search, sort, and filter the assets. The list is more functional than the grid view in terms of functionality. The list of data assets shows the connection-wise assets, DQ score, application, domain, and schema-related details.

Operations:

- The user can view the following details in the list and search, sort, and filter them same way in the list.

Term	Description
Asset	The asset name is shown under this column.
Connection Name	The Connection Name is shown under this column.
Score	The score for an asset is shown under this column.
Asset Type	The asset type is shown under this column.
Account	The data warehouse account details are populated under this column.
Application	The application mapping for the given connection can be added here by clicking on the button.
Domain	The domain related to assets that are added for a corresponding domain is added.
Schema	The schema for a given asset is shown across the asset under this column.
Alerts	The alerts generated for a given asset are populated under this column
Issues	The issues created for a given asset are populated under this column
Asset Description	The asset description for a given asset can be added for
Connection Type	The connection type column shows the connection source from which the given asset is fetched
Connection	The connection name, as per the source connection, is populated under this column
Custom Field	If a custom field is added for a given asset in the row, the corresponding name of the custom field is populated under this column
Data Size	The data size for a given asset is populated under this column
Database	The database name under which the corresponding task is maintained at the Quest DQ end is shown under this column
Last Run	The time and date-related details about the latest previous run are populated under this column
Measure	The count of measures computed for the given asset is populated under this column
Next Run	The details of the scheduled next run, if any, are populated under this column
Product	The product associated with the asset, if any, is shown under this column
Rows	The number of rows present in an asset is shown under this column
Schema	The name of the schema associated with the asset is shown in this column
Schedule	The Schedule, if any, added for a given asset to be updated from the source is shown under this column
Score	The DQ score for the given asset is shown in this column
Tags	The tags associated with the asset, if any, are shown under this column
Terms	The terms associated with the asset, if any, are shown in this column
Unique Identifier	The unique identifier for the asset, as imported from the source, is shown under this column

- The **Search** button can be used to filter search results and thereby visualize the filtered list table in the chart.
- You can download the data assets in .csv format from the list of data assets' meatball menu as shown below.
- The Save View button, as shown in the above meatball menu, helps you to save the filtered data assets view for accessing the filtered assets readily. Click on Save view– enter the name for the view, let's say ***Test View*** as an example, and enter the description for it. The view gets saved for use as shown below. Click on it to access the filtered view readily.

Pipeline

Pipeline observability refers to the process of monitoring and gaining visibility into the health, performance, and reliability of data pipelines. A data pipeline is a set of processes or workflows that transport, transform, and load data from one system to another, typically from a source system to a destination system. Pipeline observability helps ensure that these processes run smoothly, detect issues early, and ensure the timely and accurate delivery of data.

Quest DQ monitors the following key components of a pipeline to identify issues and provide faster resolution

Component	Description
Runs	Monitors the run of each task, model, and test in the pipeline
Jobs	Monitors a specific unit of work within the pipeline that operates on the data
Tasks	Monitors the definition of the work, such as transformation, extraction, and loading
Test	Monitors the validations embedded within the pipeline to ensure that the data being processed meets specific criteria for quality, accuracy, and consistency
Lineage	Allows the users to view the flow of data from source to target

In Quest DQ, the observe Pipeline module provides a summary of the following components: Tasks, Jobs, Runs, and Tests of a pipeline. Each component is grouped under separate tabs, allowing users to view each component in detail. Each tab contains the summary chart and the list of objects based on the selected component.

Runs

The Runs tab contains the details of all the pipeline runs with their status. A run can contain the following Objects: Tasks and Tests. The run summary chart shows the details of the tasks and tests that have been run on the applied date filter, including the duration and status of the run.

Summary

The run summary shows the graph of the executed runs for a given period of time. The duration of the data can be set for today, for the last 3 days, 7 days, 30 days, or all. The time period can be changed by the calendar drop-down as shown below. This filter is available at the top right corner of the graph.

Component	Description
X-axis	The date is represented on the X-axis
Y-Axis	Duration, Tasks, and Tests are represented in the Y-axis
Bar	Represents the Duration of the run and is color-coded based on the status of the run
Red Line	Represents the count of Tasks
Grey Line	Represents the count of Tests

An example of the Run-Summary page is shown below. The runs can be analyzed on the graph on three different bases.

The Y axis shows the following parameters, which can be switched on and off as a filter from the buttons that are present below the graph at the center, shown in the figure below.

- **Duration**: The duration for which the Run was executed. Click on the button next to Duration in the legend to know the scale on the Y-axis for Duration.
- **Task**: The count of tasks that were performed for the execution of a run. To know the scale on the Y-axis for Tasks, click on the button titled Tasks in the legend.
- **Test**: The count of tests performed on a given run. To know the scale on the Y-axis for Tests, click on the button titled Tests in the legend.

The X-axis shows the timeline for which these parameters are analyzed. A user can observe the run execution over time for a given duration from this summary.

List of Runs

The bottom section of the page contains the list of runs across all configured data pipelines, which shows the details of the run duration, start and end time, and status of each run. Clicking on each run will take the user to the run detail page. The list will be populated based on the applied filter, and the users can search, sort, and filter the table to view the desired runs.

The tabulated list consists of the following column headers:

Term	Description
Alerts	The alerts generated against the tasks are populated under this column
Commit Merge Date	The commit merge date is the date on which a specific commit was merged on to main branch or any other target branch
Commit SHA	Commit SHA is the unique identifier for a specific commit in the Git repository
DBT Tags	DBT tags help label and organize specific sets of data for modeling, testing, and for other such data management operations
Issues	The issues linked to the tasks are shown in this column
Job	The job is a scheduled set of triggered operations
Job Duration	If there is a duration set for a given job to be executed. The duration in seconds for a given job is shown under this column
Job End	The date and time-related details for the job when it ended are shown under this column
Job Start	The date and time-related details for the job when it started are shown under this column
Run ID	The run ID for the given run in the pipeline is displayed under this column
Run Status	The run status (success or failure) for a given run ID is shown under this column
Status	The status of the run (pending or completed) is shown under this column
Run Duration	The task duration for the job is shown under this
Run End	The date and time-related details about the run start point are populated under this column
Run Start	The date and time-related details about the run endpoint are populated under this column
Tests	The tests associated with the given run ID, if any, are shown under this column

Jobs

A **job** refers to a specific unit of work or task within the pipeline that performs an operation on the data. Jobs are the individual steps or processes in the pipeline that handle tasks such as extracting, transforming, or loading data. They are typically scheduled or triggered to run automatically at specified intervals or in response to certain events. A job is a component in pipelines that contains the schedule information and the list of tasks/models that have to be executed with the job is run manually or by a schedule.

Summary

The Job tool can help the user with the following:

- It helps you to have an overview of the statistics about the duration required for the job to be completed in terms of average, maximum, and minimum duration. It also helps in analyzing the rate of execution of the jobs in the data pipeline.
- You can filter the list of jobs at the column level in the list of jobs and generate insights for a specific set of jobs to analyze the trends.

The Pipeline Job page contains the following two summary charts

- Execution Summary Chart
- Duration Summary Chart

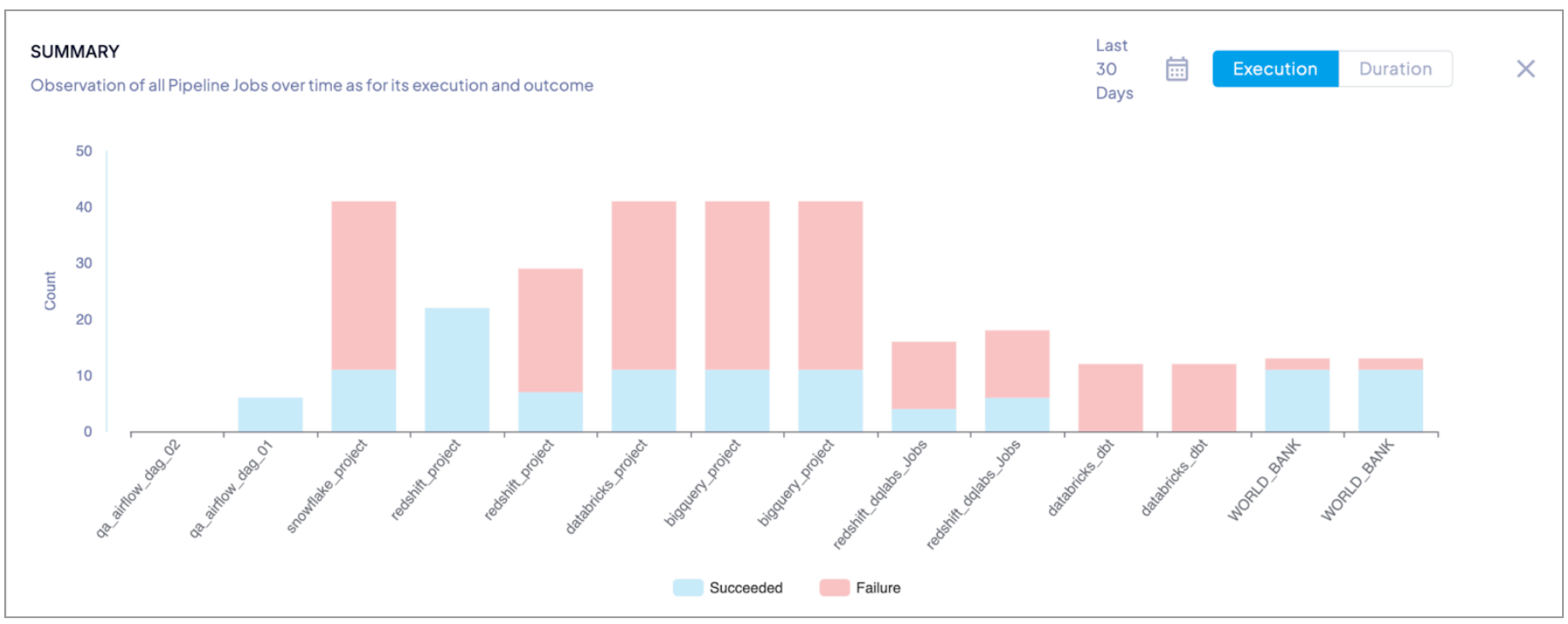
This allows the users to view the details a pipeline jobs at a glance and filter the jobs and view their respective details:

Execution Summary Chart

The execution summary chart contains the details of all the pipeline jobs with a count of total successes and failures based on the applied date time filter, and the execution is represented in a bar graph. Failed jobs are represented in red, and success is represented in blue.

Component	Description
X-axis	Jobs are represented on the X-axis
Y-Axis	The count of jobs is represented on the Y-axis
Bar	Represents the count of Jobs and is color-coded based on the status of the run by the applied date filter
Red Bar	Represents the count of failed Tasks/Tests in the job
Blue Bar	Represents the count of Tasks/tests that succeeded in the job

An example of a graph showing the analysis of the execution of the jobs is shown below. The X-axis shows the name of the jobs, and the Y-axis shows the count of the tasks completed for the given job.



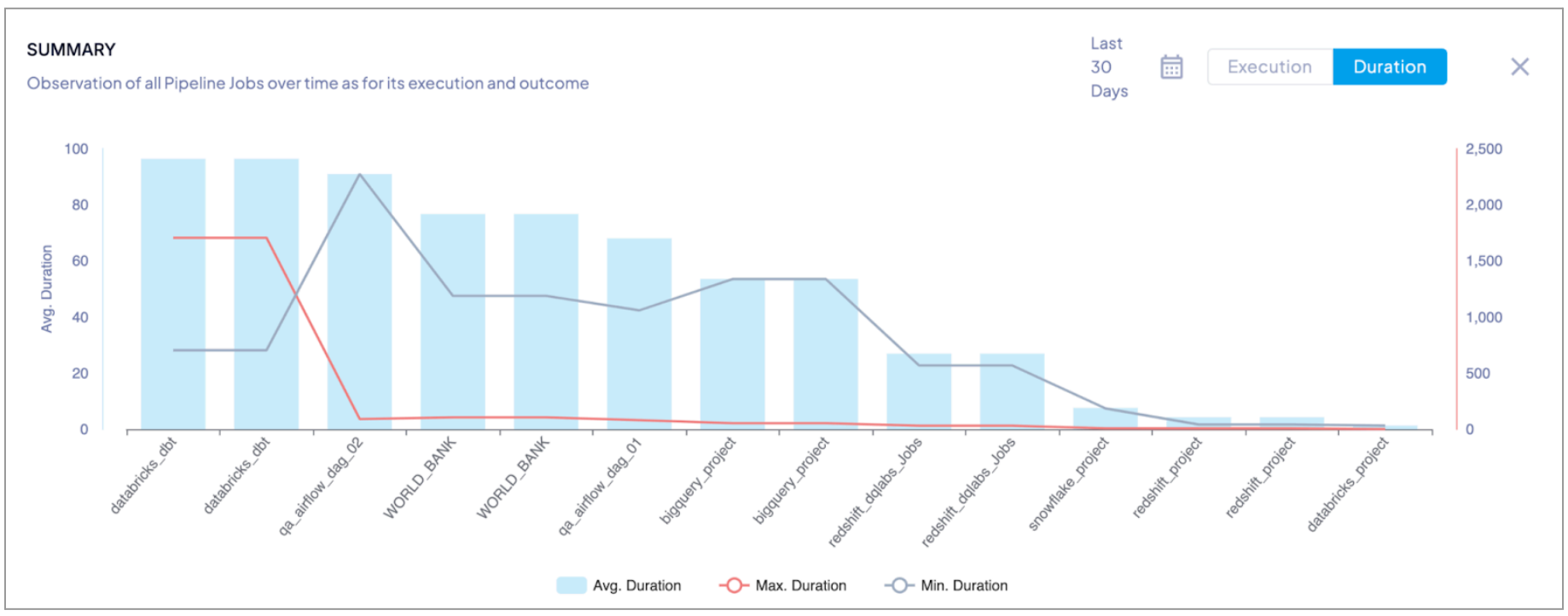
Duration Summary Chart

The duration chart is represented in a combination chart with bars representing the average duration of a job based on the applied date filter and lines representing the min and maximum duration of the job.

Component	Description
X-axis	Jobs are represented on the X-axis
Y-Axis	Average Duration, Min Duration, and Max Duration are represented in the Y-axis
Bar	Represents the average duration of the job based on the applied date filter
Red Line	Represents the Maximum duration of the job based on the applied date filter
Grey Line	Represents the Minimum duration of the job based on the applied date filter

The Duration-related graph shows the jobs on the X-axis and the average time taken (average duration) for the completion of the job on the Y-axis. For a better drill-down on optimizing the duration for the job, the duration is shown as:

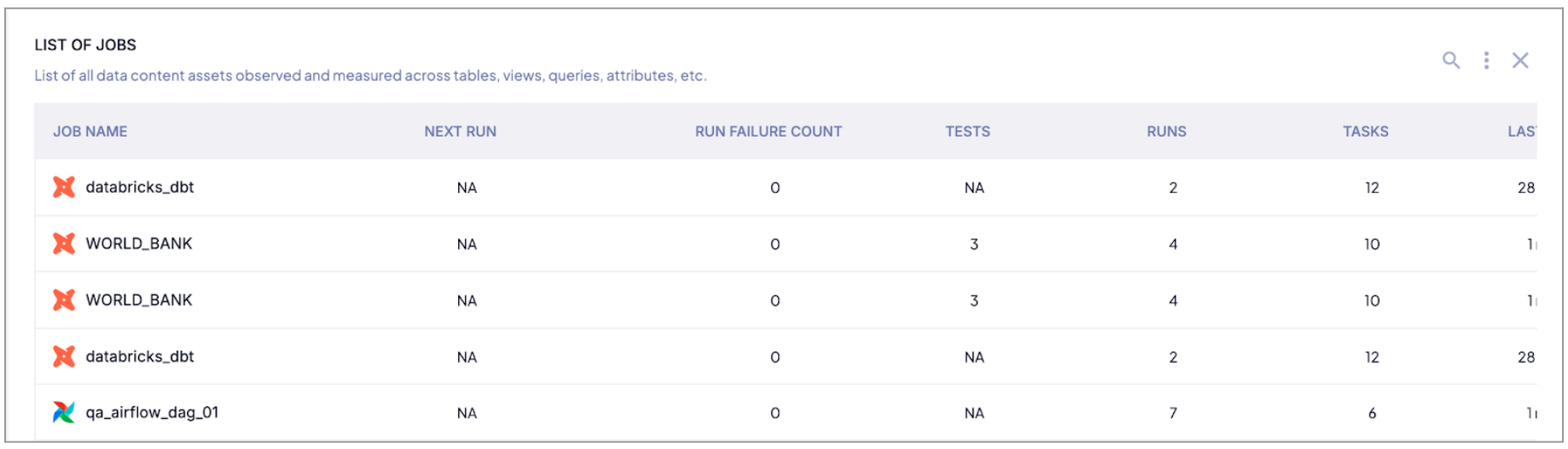
- Average Duration (Avg. Duration),
- Maximum Duration (Max. Duration)
- and Minimum Duration (Min. Duration).



These variations in the duration can be altered by clicking on the calendar button. By clicking on the calendar button as shown below, the span for the input data can be changed.

List of Jobs

The list of jobs displays all the jobs configured on the platform across all pipeline data sources. Clicking on the job name will take the user to a job detail page. The users can search, sort, and filter the jobs in the list.



Term	Description
Avg Duration	The average duration taken for executing a job is populated under this column
Connection Name	The connection name corresponding to the job is populated under this column
Environment ID	The environment ID for the given connection and the job is populated under this column
Last Duration	The latest last duration for completion of a job is shown here
Last Run Status	The status of the last run is shown under this column
Max Duration	The maximum duration required for the given job to be completed is shown in this column
Min Duration	The minimum duration required for the given job is shown in this column
Next Run	The next run scheduled if any, the details about the same are shown under this column

Project ID	The project ID for the given job is shown under this column
Run Failure Count	The run failure count is shown under this column
Runs	The job that runs for a given job are populated under this column
Tasks	The tasks related to the job are populated under this column
Tests	The tests related to the jobs are populated under this column

Tasks

A Task is a component in pipelines that contains the definition of what operations a pipeline should perform when being executed when the job is run manually or by a schedule.

Summary

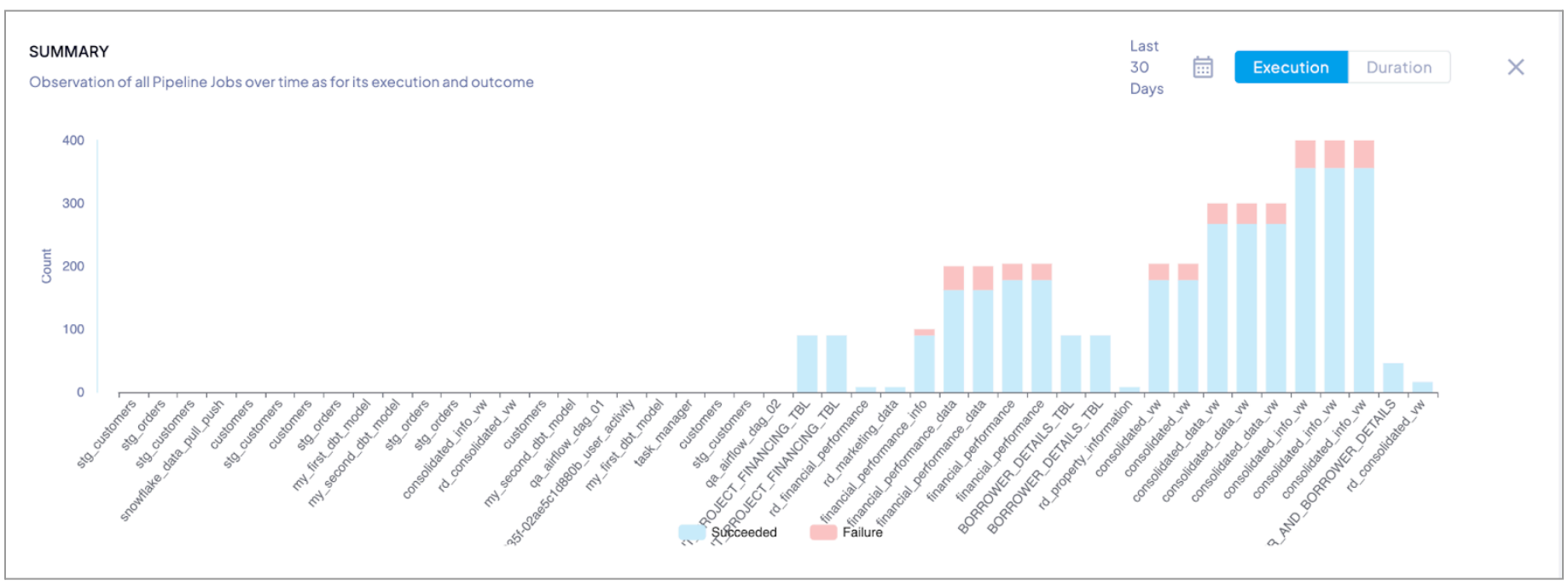
The Pipeline Task page contains the following two summary charts

- Execution Summary Chart
- Duration Summary Chart

This allows the users to view the details of pipeline tasks at a glance and filter the tasks, and view their respective details

Execution Summary Chart

The task summary section shows you the observations computed by Quest DQ for the various tasks under consideration in the Data Pipeline



The X-axis shows the tasks under consideration. The Y-axis shows the count of the tasks. The filter for the count is the Success and Failure of the tasks completed.

The buttons at the bottom center show the filter for the Success and Failure count of tasks. Click on the button to activate the specific visualization filter with color code.

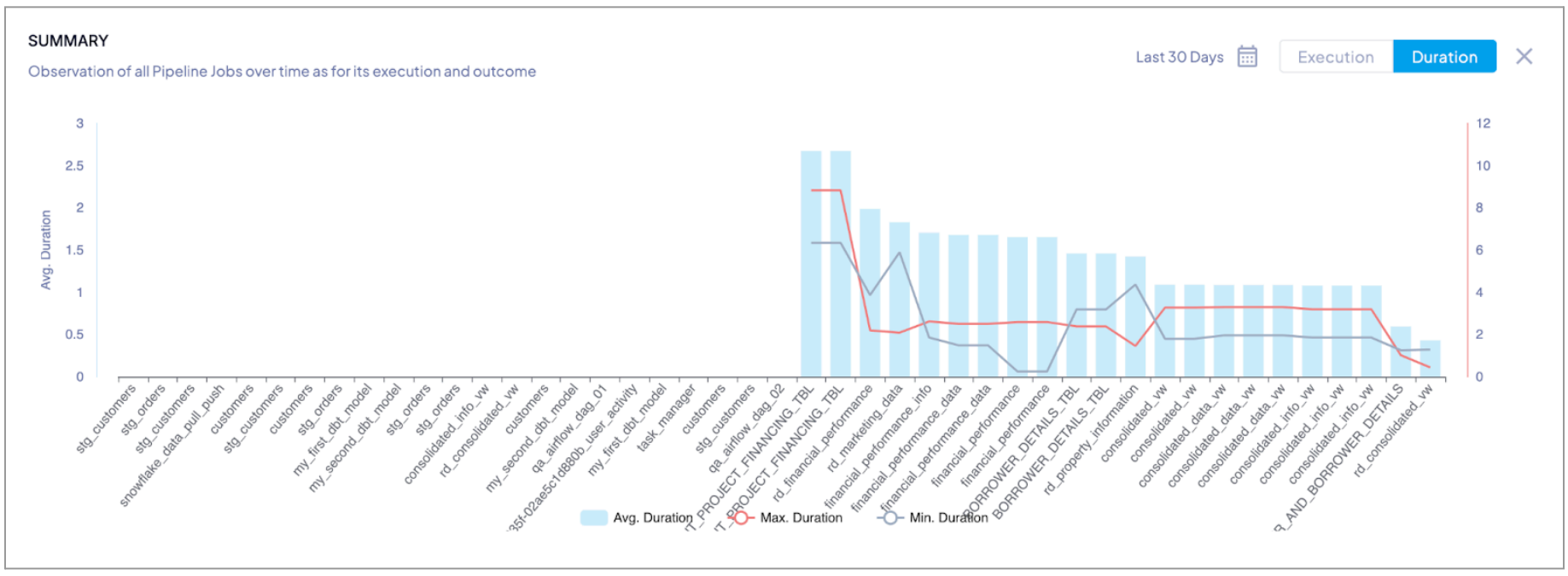
Component	**Description**
X-axis	Tasks are represented on the X-axis
Y-Axis	The count of task runs is represented in the Y-axis
Bar	Represents the count of tasks and is color-coded based on the status of the run by the applied date filter
Red Bar	Represents the count of failed Tasks
Blue Line	Represents the count of Tasks that succeeded

The X-axis shows the tasks under consideration. The Y-axis shows the count of the tasks. The filter for the count is the **Success** and **Failure** of the tasks completed.

The buttons at the bottom center show the filter for the Success and Failure count of tasks. Click on the button to activate the specific visualization filter with color code.

Duration Summary Chart

The Duration-related graph shows the tasks on the X-axis and the time taken (duration) for the completion of the job on the Y-axis.

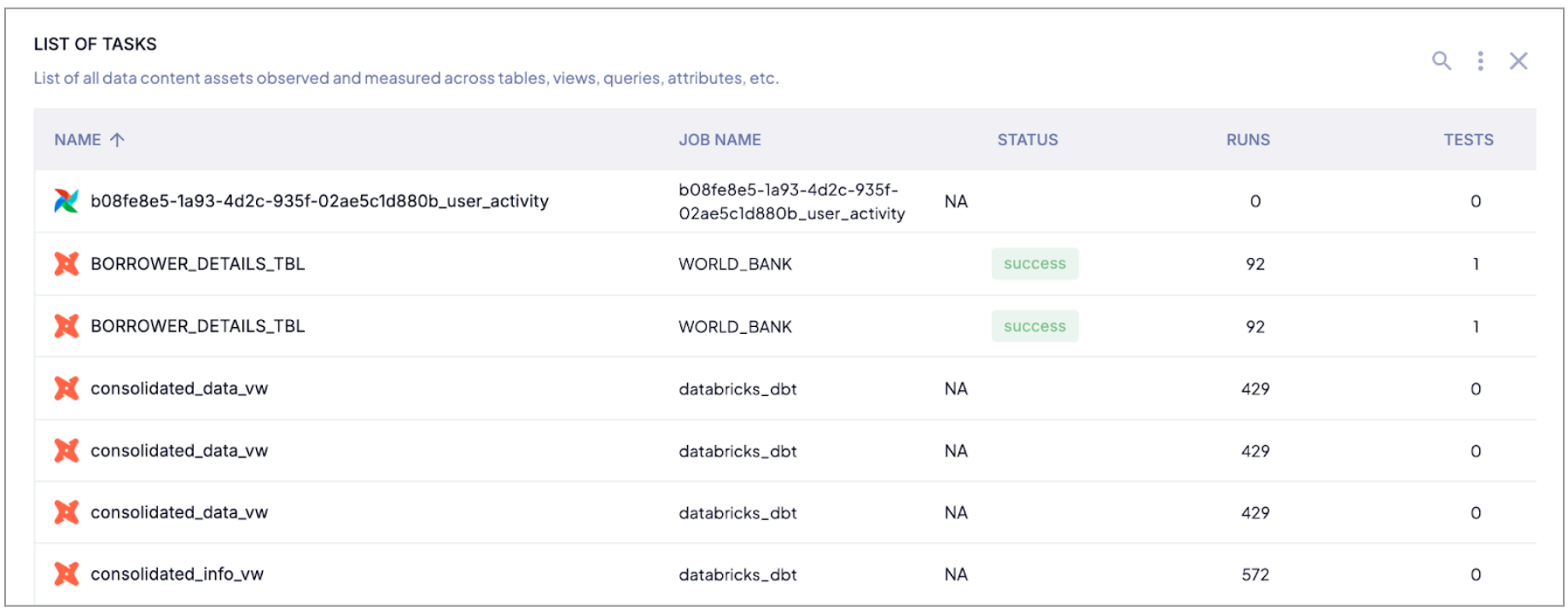


For a better drill-down on optimizing the duration for the task, the duration is shown as

- Average Duration (Avg. Duration),
- Maximum Duration (Max. Duration)
- and Minimum Duration (Min. Duration)

List of Tasks

The information about the tasks and related information is shown in this section. The bottom of the page lists all the tasks configured in the platform across all pipeline data sources. Clicking on the task name will take the user to a task detail page. The users can search, sort, and filter the tasks in the list.



The column headers present under the list of tasks are shown here:

Term	**Description**
Name	The name of the task is populated under the column
Alerts	The alerts corresponding to the tasks are shown under this column
Application	The Application under which the corresponding task is listed is shown in this column

Connection Name	The Connection Name for the given task is shown under this column
Database	The database name under which the corresponding task is maintained at the Quest DQ end is shown under this column
Domain	The Domain associated with the task is shown under this column
Environment ID	The environment ID for the task is shown under this column
Error Records	The error records for the given task are populated under this column
Job Name	The job name corresponding to the task is shown here
Project Name	The project name for the task is shown under this column
Run Date	The date and time at which the task runs
Runs	The recent 7-day runs are based on success and failure
Status	Indicates if the run has succeeded or failed
Tests	The number of tests related to the task

Test

A Test is a component in pipelines that contains the conditions to check when a task has finished pushing the data. The output of a test can either be pass or fail.

Summary

The pipeline test page contains the following two summary charts

- Execution Summary Chart
- Duration Summary Chart

This allows the users to view the details of pipeline tests at a glance and filter the tests based on status, and view their respective details

Execution Summary Chart

The execution summary chart contains the details of all the pipeline tests with the count of total success and failure based on the applied date time filter, and the execution is represented in a bar graph. Failed tests are represented in red, and success is represented in blue.

Component	Description
X-axis	Tests are represented in the X-axis
Y-Axis	The count of test runs is represented in the Y-axis
Bar	Represents the count of Tests and is color-coded based on the status of the run by the applied date filter
Red Bar	Represents the count of failed Tests
Blue Bar	Represents the count of Tests that succeeded

Duration Summary Chart

The duration chart is represented in a combination chart with bars representing the average duration of a test based on the applied date filter and lines representing the min and max duration of the test.

Component	Description

X-axis	Tests are represented in the X-axis
Y-Axis	Average Duration, Min Duration, and Max Duration are represented in the Y-axis
Bar	Represents the average duration of the test based on the applied date filter
Red Line	Represents the Maximum duration of the test based on the applied date filter
Grey Line	Represents the Minimum duration of the test based on the applied date filter

List of Tests

The list of tests consists of tests that were executed as per the run date. An example of the list of tasks is shown below:

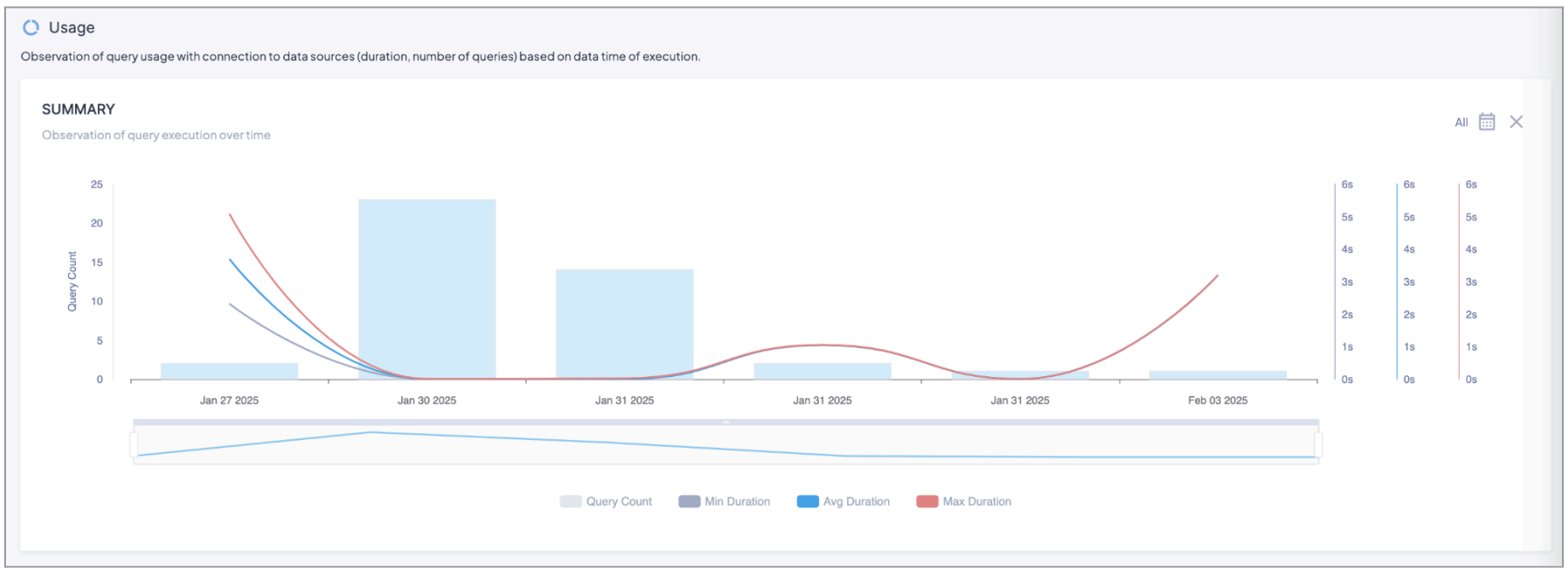
The column headers that are present under the list of tests are the following:

Term	**Description**
Test Name	The test names are populated in this column
Error	The error-related details are shown in this column
Runs	The Runs/set of runs related to the test are shown under this column
Status	The Status for the test (success or failed) is shown under this column
Tags	The tags associated with the test name are shown under this column
Test Duration	The duration required for completion of the test is shown here
Test End	The test end date and time-related details are shown in this column
Test Start	The test start-related date and time details are shown in this column

Usage

The Observe-Usage tool is used to analyze the overall usage of the data by the user. The details about the tools are explained below. The Usage Summary shows the details about the query execution counts and duration for which the query execution takes place. This is a standalone usage metric analysis tool. For asset-level usage, refer to the Asset - Usage section above. The Usage page contains the details of the time taken to execute the query in the source data warehouse for all configured connections. The usage summary chart shows the details of the time taken and the number of queries executed over a period of time based on the applied date filter.

The summary chart is a combination chart with displays the query count for an asset in bars and represents the duration in line.

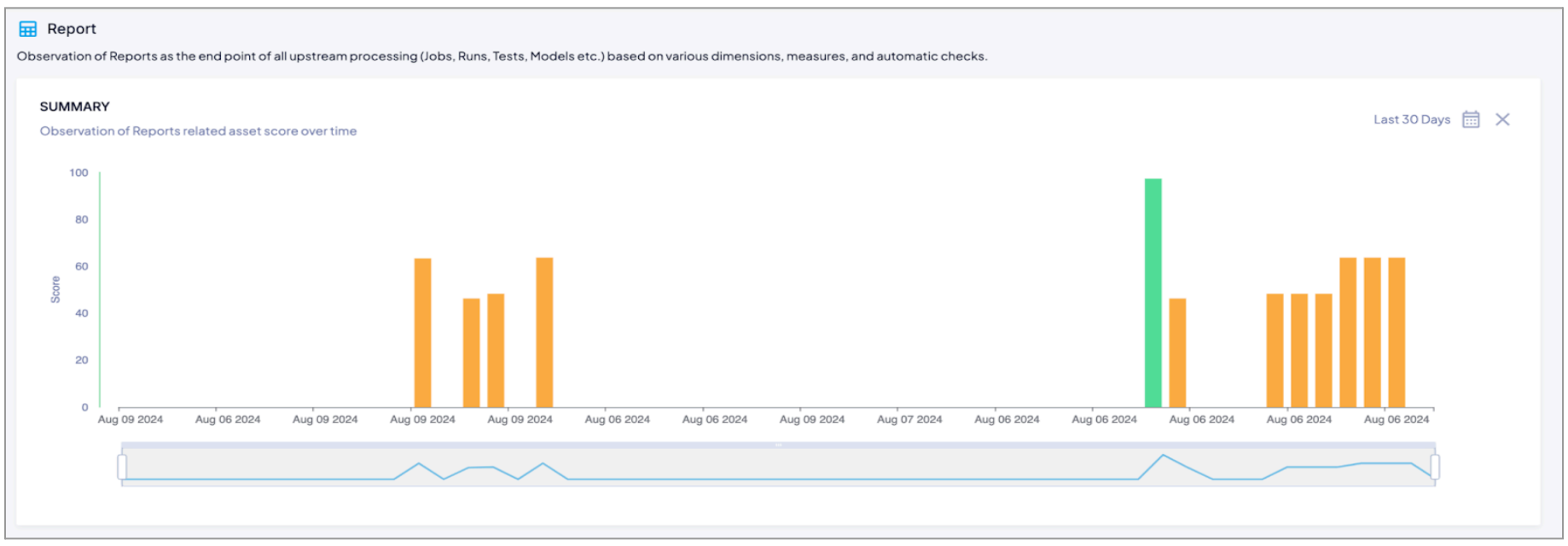


The Y-axis shows the query count and duration as highlighted above. The X-axis shows the

Component	Description
X-axis	The date is represented on the X-axis
Y-Axis	Query Count, Average Duration, Min Duration, and Max Duration are represented on the Y-axis
Bar	Represents query count for an asset run over a period of time
Red Line	Represents the Maximum duration of the queries for the asset
Grey Line	Represents the Minimum duration of the queries for the asset
Blue Line	Represents the Average duration of the queries for the asset

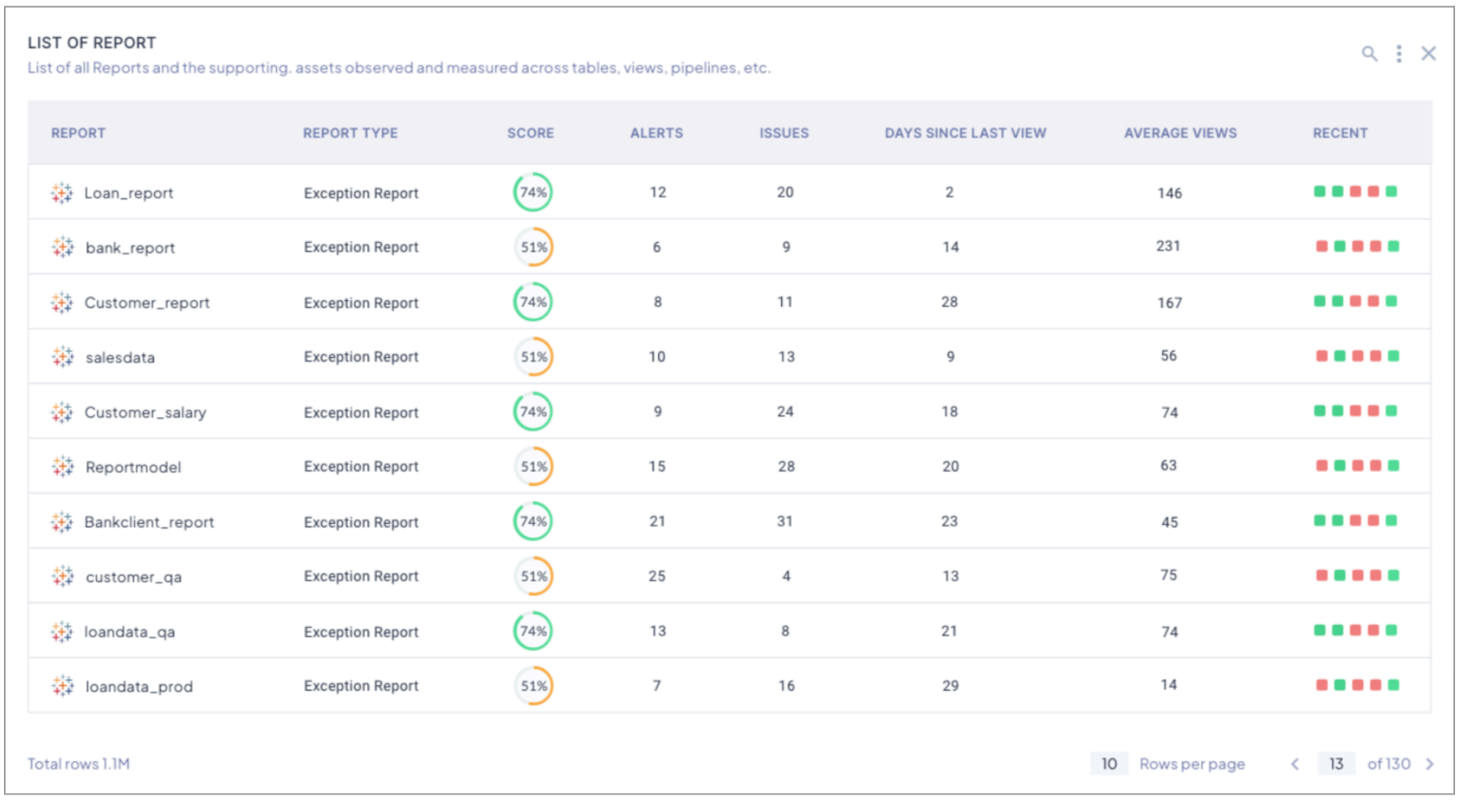
Report

The reports page displays all the BI reports configured in the platform, and the report summary chart displays the DQ score for the report run over a period of time based on the applied filter. The DQ score for a report is the aggregated average score of all the data assets that the report is linked to from the details page.



Component	Description
X-axis	The date is represented on the X-axis
Y-Axis	DQ Score for the reports represented on Y-Axis
Bar	Represents the DQ score of the report and color-coded based on the quadrant configuration in settings

The bottom of the page displays the list of all the reports in the platform across all configured BI data sources. The user can search, sort, and filter the list of reports in the list. The user will be taken to the report detail page by clicking on the report name in the list.



MEASURE

Measures in Quest DQ are used to monitor the various metrics across the assets and attributes in Quest DQ. The user can configure out-of-the-box measures or create a custom measure in Quest DQ. These measures can be used in data quality scoring.

Data quality rules are a set of guidelines or standards that are used to ensure that data is accurate, complete, and consistent. These rules can be applied to data at various stages of its lifecycle, such as during data entry, data validation, or data analysis. Examples of data quality rules include:

- Data must be entered in a specific format, such as a date in "MM/DD/YYYY" format
- Data must be within a certain range, such as a product price between $0 and $1000
- Data must be unique, such as a customer ID or email address
- Data must be verified, such as a phone number or email address
- Data must be consistent across different sources or systems.

Data quality rules are important for ensuring that data is reliable and can be used for reporting and analytics. They help to ensure that data is consistent and accurate, which can improve the efficiency and effectiveness of business processes.

Different types of measures can be applied to the data assets. The rules are listed below:

- Auto Measures
- Custom Measures

Auto Measures

Quest DQ comes with out-of-the-box data quality measures that can be applied across the data assets in the organization. The following types of rules can be applied to measure the required metric and monitor data quality. Auto measures are automated, system-driven checks and metrics that continuously evaluate various aspects of data quality without requiring human involvement for each assessment.

The auto measures present under the Auto Measures screen are listed and explained below.

Measure Type	Rule Type	Description
Reliability	Out of the box	The reliability measures help in evaluating the data stability and predictability.
Distribution	Out of the box	The distribution measures analyse the spread and pattern of the data within a given dataset.
Frequency	Out of the box	Frequency measures help identify various patterns– expected and unexpected or certain data values in a given dataset
Statistics	Out of the box	The statistical measures are mathematical techniques used for analysing and summarising the data characteristics

The following out-of-the-box rules are available in Quest DQ to measure and monitor data quality. The admin can enable these rules from the settings page or under each attribute. The rules are further categorized into advanced and basic measures, which allow the users to turn ON profiling based on their needs.

Reliability Measures

Name	Description	Level	Dimension
Duplicates	Counts the number of rows that are identical based on either the primary key or the composite keys defined by the user.	asset	Uniqueness
Freshness	Computes how up-to-date your data is.	asset	Timeliness
Schema	Computes the total number of columns.	asset	Validity
Volume	Computes the total row count.	asset	Completeness

Distribution Measures

Name	Description	Level	Dimension
Alpha Numeric Count	Compute the count of all values with both alphabet and numeric. Included as part of the advanced profile.	attribute	Completeness
Blank Count	Compute the count of all blank values, meaning there is a value, but the field is blank. Included as part of the basic profile.	attribute	Completeness
Character Count	Compute the count of all values with one or more characters. Included as part of the advanced profile.	attribute	Validity
Digits Count	Compute the count of all values with one or more digits. Included as part of the advanced profile.	attribute	Validity
Distinct	Computes the number of unique values. Included as part of the basic profile.	attribute	Uniqueness
Duplicate	Computes the exact number of identical values. Included as part of the advanced profile.	attribute	Uniqueness
Inner Space	Compute the count of all values with space in between. Included as part of the advanced profile.	attribute	Validity
Leading Space	Compute the count of all values with whitespace before the value. Included as part of the advanced profile.	attribute	Validity
Max Length	Computes the maximum length across all values. Included as part of the basic profile.	attribute	Validity
Max Value	Computes the maximum value across all values. Included as part of the basic profile.	attribute	Validity
Min Length	Computes the minimum length across all values. Included as part of the basic profile.	attribute	Validity
Min Value	Computes the minimum value across all values. Included as part of the basic profile.	attribute	Validity
Negative Count	Computes the total occurrences of all negative counts. Included as part of the advanced profile.	attribute	Validity
Non Empty Count	Computes the total occurrences of non-null or non-empty values. Included as part of the advanced profile.	attribute	Completeness
Null Count	Compute the count of all values with no values. Included as part of the basic profile.	attribute	Completeness
Outer Space	Computes the number of values with whitespace before and/or after the value. Included as part of the advanced profile.	attribute	Validity
Positive Count	Computes the total occurrences of all positive counts. Included as part of the advanced profile.	attribute	Validity
Space Count	Compute the count of all values with spaces. Included as part of the basic profile.	attribute	Completeness

Special Character Count	Compute the count of all values with a Special character. Included as part of the advanced profile.	attribute	Validity
Trailing Space	Computes the number of values with whitespace at the end. Included as part of the advanced profile.	attribute	Validity
Whitespace Count	Computes the count of all values with whitespace. Included as part of the advanced profile.	attribute	Validity
Zero Values Count	Computes the count of all values with a value of zero. Included as part of the basic profile.	attribute	Completeness

Statistical Measures

Name	Description	Level	Dimension
Kurtosis	Computes whether the data are heavy-tailed (heavy outlier) or light-tailed (lack of outlier) relative to a normal distribution.	attribute	Accuracy
Mean	Computes the sum of all values divided by the total number of values.	attribute	Accuracy
Median	Computes the value in the middle number in a sorted (ascending or descending) list of all values.	attribute	Accuracy
Mode	Computes the average value, which describes where most of the data is located.	attribute	Accuracy
Q1	Computes the value separating the first quarter (25th percentile) from the second quarter (50th percentile) of the data.	attribute	Accuracy
Q3	Computes the value separating the third quarter (75th percentile) from the fourth quarter	attribute	Accuracy
Range	Computes the difference between the smallest and the largest value of the data	attribute	Accuracy
Skewness	Measures the deviation of a random variable's given distribution from the normal distribution. Left means Positive Skewness and Right means Negative Skewness.	attribute	Accuracy
Standard Deviation	Computes how dispersed the data is in relation to the mean. A low Standard deviation means data are clustered around the mean, and a high standard deviation indicates data are more spread out.	attribute	Accuracy
Sum	Calculates the total sum of all values.	attribute	Accuracy
The Margin of Error	Calculate how many percentage points your results will differ from the real population value	attribute	Accuracy
Variance	Computes the measure of how far a set of random numbers is dispersed from the mean.	attribute	Accuracy

Frequency Measures

Name	Description	Level	Dimension
Enum	Computes how many times a particular text or a number value occurs. Included as part of the advanced profile.	attribute	Validity

Length	Computes how data is distributed across various lengths by showing the number of occurrences for each length. Included as part of the advanced profile.	attribute	Validity
Length Range	Computes the minimum and maximum length across all the values. Included as part of the advanced profile.	attribute	Validity
Long Pattern	Auto-discovers all data patterns using acronyms such as A for alphabets, N for numeric, S for space, and special characters as it is. Repeating acronyms is avoided for easy reading. Included as part of the advanced profile.	attribute	Validity
Regular Expressions	Computes the occurrence of each conditional regular expression-based pattern defined or discovered automatically using business terms or manual configuration. Included as part of the advanced profile.	attribute	Validity
Short Pattern	Auto-discovers all data patterns using acronyms such as A for alphabets, N for numeric, S for space, and special characters as it is. Repeating acronyms are not avoided to identify all occurrences of patterns. Included as part of the advanced profile.	attribute	Validity
Value Range	Computes the minimum and maximum value range across all values. Included as part of the advanced profile.	attribute	Validity

Advanced

Quest DQ provides the ability to add custom measures to an attribute level or at the semantics level. Based on the complexity of the rule, complex rules are categorized into the following types. Advanced measures, types, and configurations are for additional quality assessment. Following is the list of advanced measures.

Conditional Measures

This allows the users to set up rules based on predefined conditional values. Quest DQ also provides the ability to add multiple conditional rules and rule groups as a single measure.

Query Measures

Quest DQ allows the user to use native Snowflake queries to create measures. For example, the user can use a query such as “Select * from Person.data where ID < 100“ or use joins to filter the data. The user can also define the polarity, condition, and value for which the output should be measured and monitored.

Behavioural Measure

Quest DQ offers behavioural analysis with simple configurations for time-series-based data. A time series is a sequence of observations recorded at regular time intervals. Depending on the frequency of observations, a time series may typically be hourly, daily, weekly, monthly, quarterly, or annual. If you have any such time series data, with simple configurations Quest DQ can auto-monitor and provide auto rules with thresholds. For this analysis, you need

- Time Dimension Attribute
- Categorical Attribute (one to many)
- Numeric Attribute

The numeric attribute benchmark can be measured in summation as one of the following options

- Average
- Sum
- Minimum
- Maximum
- Count
- Value

Lookup Measure

Quest DQ provides the ability to create measures that refer to a file or table to assess and validate data against reference data sets or predefined standards. The primary purpose of a lookup measure is to ensure that data values align with expected or allowed values, which enhances the accuracy and consistency of data.

Standalone Measures

Standalone measures allow the users to create measures without associating the measure with any assets, there are two types of measures that are currently supported by Quest DQ, and they are as follows:

- Query mode measure
- Comparison measure

Standalone query measure

Query mode measures are similar to custom measures that are created under an asset but in a stand-alone mode, the total record query is mandatory in order to define the total record scope of the query. Follow the steps below to add a standalone measure.

Standalone comparison measure

A comparison measure allows the users to create rules to compare between two assets of the same connection object. Quest DQ provides the ability to compare the following metrics against two assets.

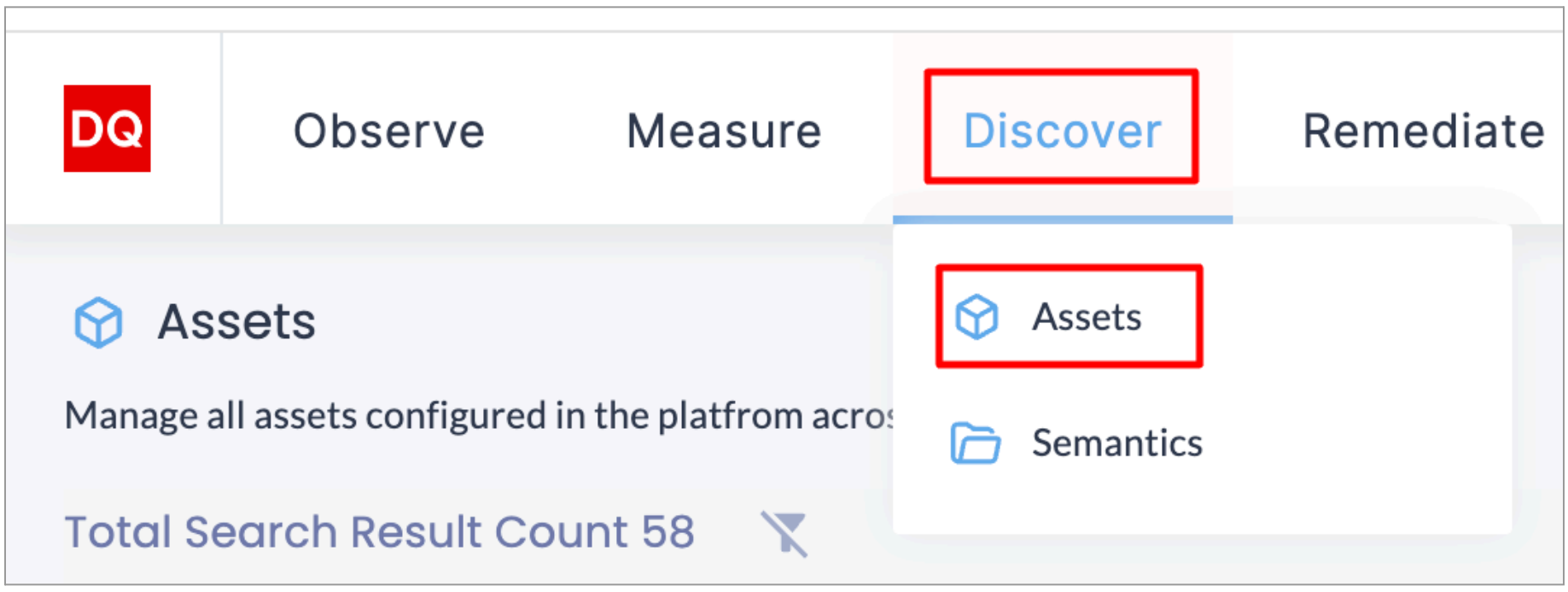
DISCOVER

Assets

The assets page allows users to view the list of all the assets in the platform. The user can filter the assets by their type and use the sidebar to sort the assets by their metadata definition, and also use other metadata properties such as created date, connection type, rating, etc..

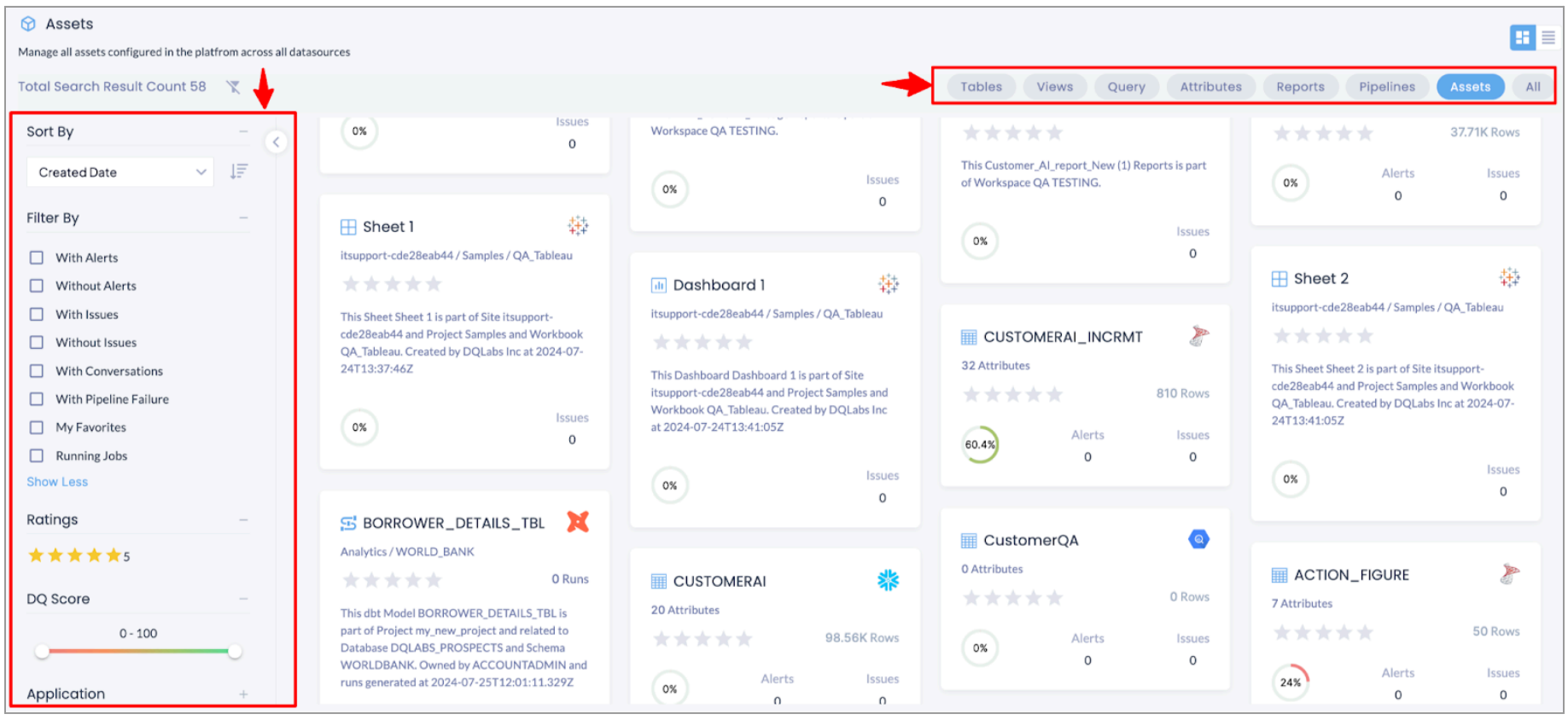
The asset page has two views: Grid view and List view. The grid view displays the assets in tiles and displays the following information: Asset Name, Number of Attributes, Number of Rows, DQ score, Alerts, and Issues. Depending on the type of asset, the information will change.

The user can land on the asset discovery page by clicking on the **Discover** button, as shown in the screen below.



The user lands on the following screen for asset-level discovery. There are two fronts on which the user can filter the asset-level data, as highlighted on the screen below.

- The data can be filtered by the data forms, that is, tables, views, queries, attributes, Reports, etc., from the top right as highlighted in the screen below.
- The data can also be filtered on the basis of the filter menu present on the left-hand side of the screen, as shown in the screen below.



- The data filters on the left column, as seen on the above screen, are tabulated below for reference.

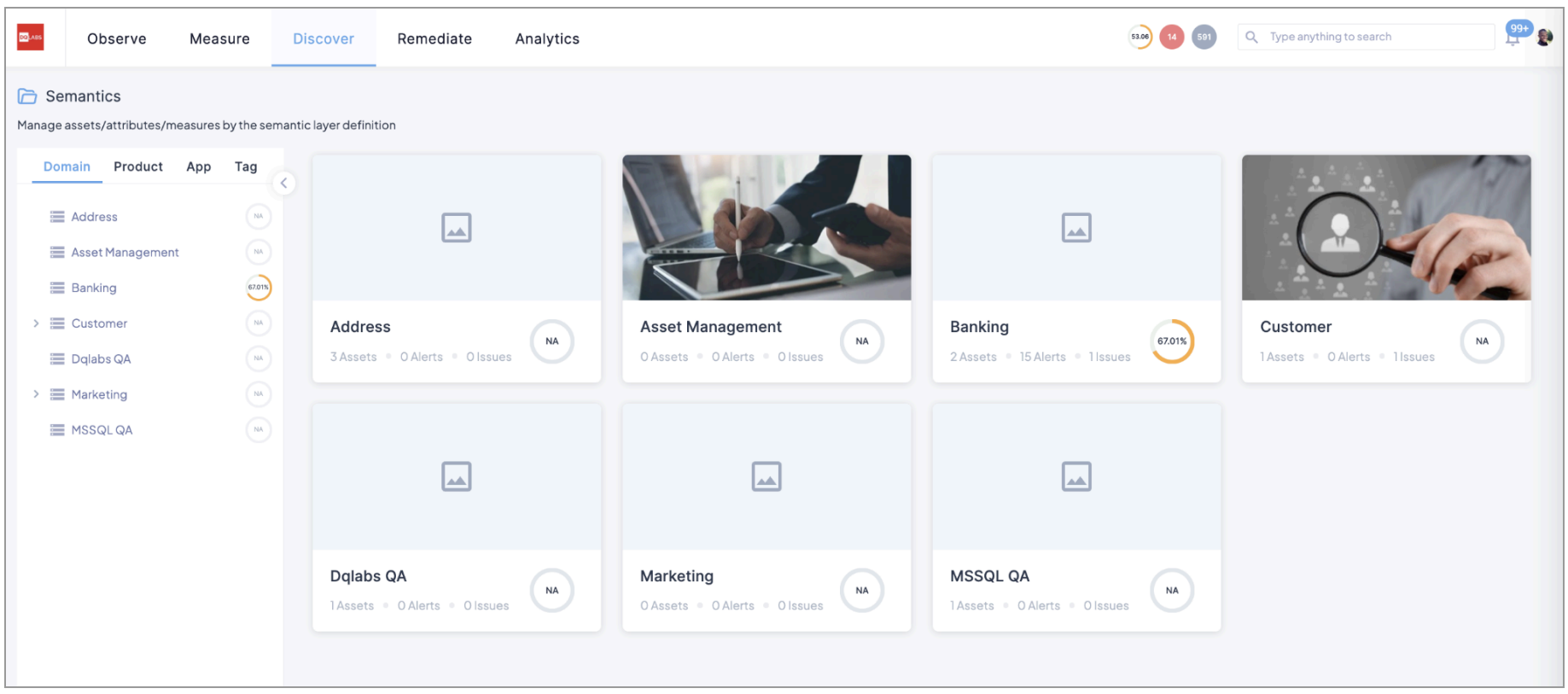
Filter	Description
Sort By	The Sort By drop-down has the filter criteria such as Name, DQ Score, Ratings, etc..

Filter By	The Filter By checkbox has criteria related to alerts, issues, pipelines, and runs
Ratings	The ratings related to the assets can be filtered from here.
DQ Score	The DQ Score slider filter helps with the asset filtering based on the DQ Score that is computed by Quest DQ
Application	The Application level data filter for assets can be selected from this drop-down.
Connection Type	The assets can be filtered based on the connection types from the drop-down.
Connection	This helps filter the assets based on the connection
Domain	This helps in filtering the assets based on the domain
Status	This helps in filtering the assets based on status
Tag	This helps in filtering the assets based on the tag
Term	This helps in filtering the assets based on the term

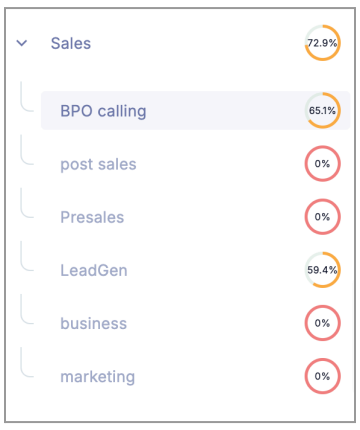
Semantics

The Discover semantics page allows users to view the assets in the platform by their semantic layer definition, such as Domains, Applications, and Tags. The semantics view allows the users to use the sidebar to expand and navigate through subcategories of the selected semantics layer definition and view its assets.

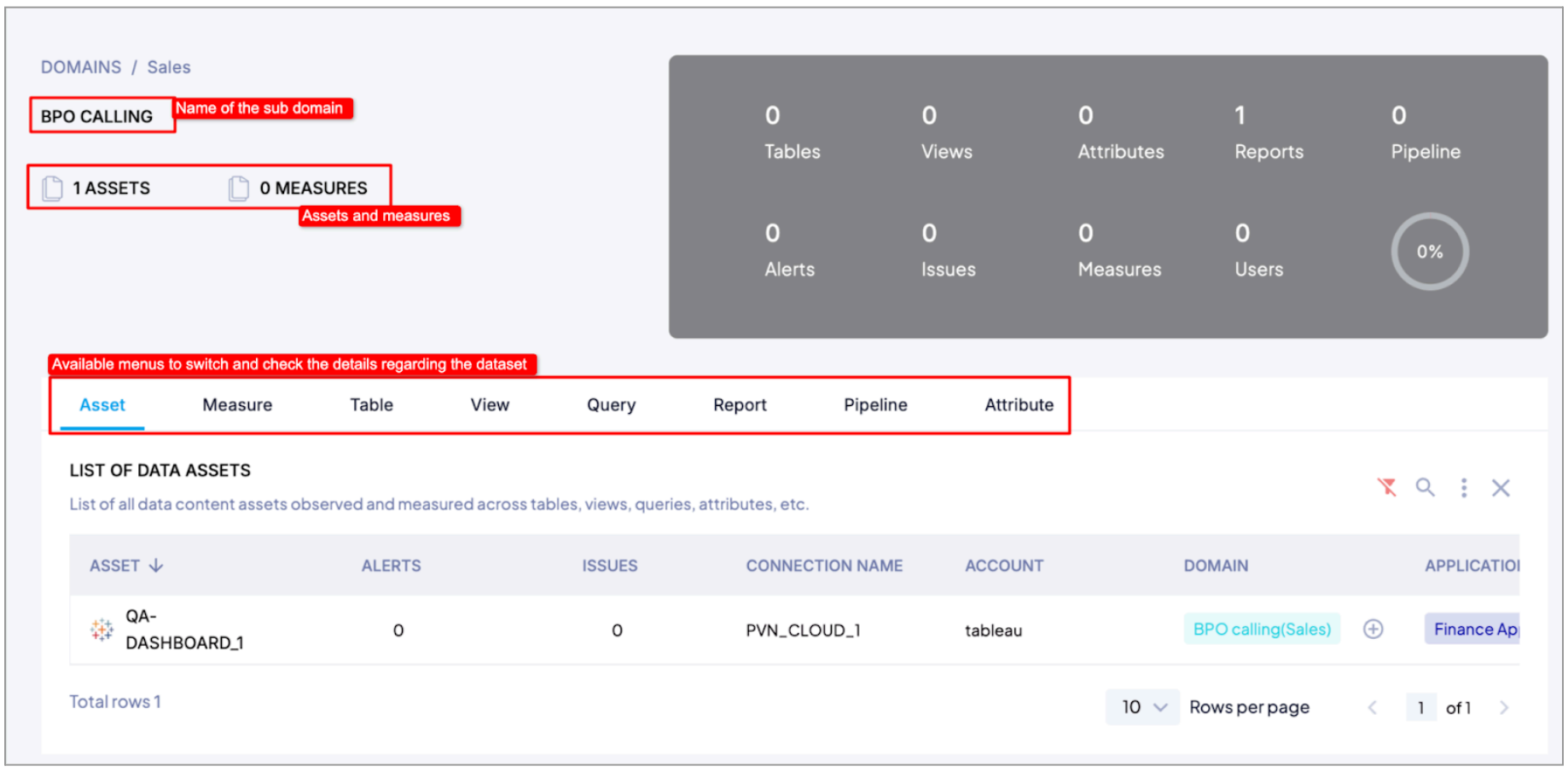
A semantic overview page contains the following information: Summary, List of assets, and Assets grouped by asset type and attributes. The user can sort, search, and filter the assets in the list using the list functions. The user will be able to switch between different semantic definitions by clicking on the tab option in the left-side menu. For discovering the domain-level classification of data in Quest DQ, go to **Discover** > **Semantics** > **Domains**.



- The **Domains** menu on the left allows for further drill-down for the data grouped under various subgroups.
- The Domains have various classes under them, and an example is shown below:



- The domain dataset has various subdomains associated with it, mapped as a set of dropdowns as shown in the above figure. For example, we are accessing the ***BPO calling***.
- Following is the screen that is visible to the user:



The menus and screens available for checking the details are as follows:

- **Asset:** When a user clicks on a given asset from the list of assets table, the user is taken to the asset tool as explained under List of Data Assets - Asset Level Details.
- **Measure:** The Measure tab shows the list of measures populated under the list of measures table that is computed across tables, views, queries, attributes, etc..
- **Table**: The Table shows the list of table-based assets.
- **View:** The view tab lists all the view-based assets
- **Query:** The query tab lists all the query-based assets
- **Report:** The reports related to the given domain are shown here.
- **Pipeline:** The list of tasks present under the pipeline for the given domain is presented here
- **Attribute:** The attribute tab lists all the attributes across all assets

REMEDIATE

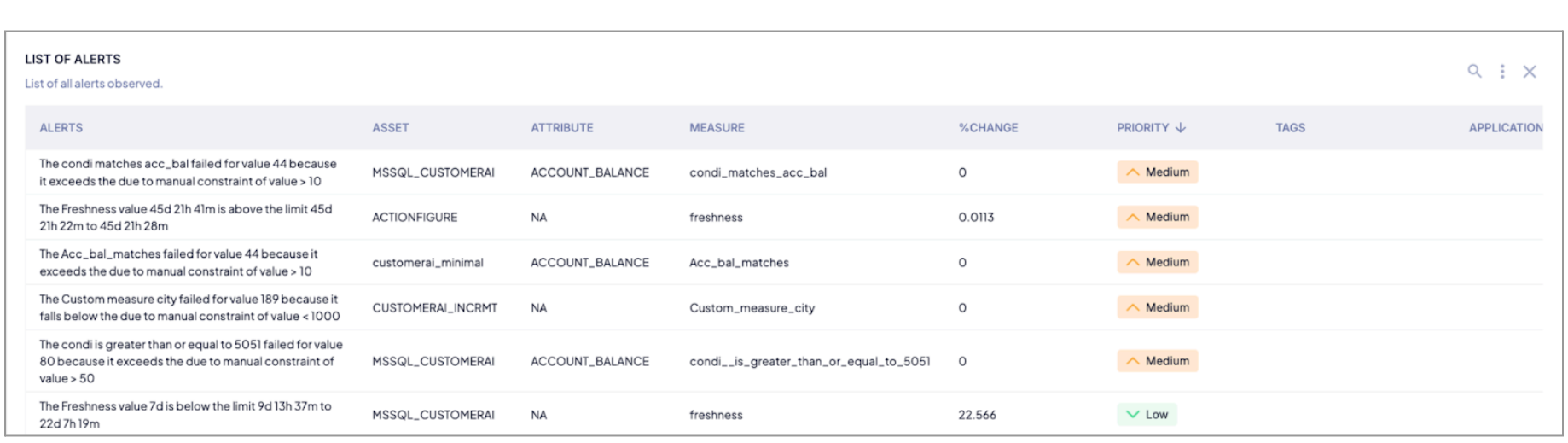
Alerts

The alerts page provides the details of the alerts across all assets in the platform. The alerts page has two sections: the summary chart and the list view with all the alerts in the platform. The summary chart represents the number of alerts raised over a period of time based on the applied date filter.

Component	Description
X-axis	The date is represented on the X-axis
Y-Axis	The number of alerts is represented on the Y-axis
Bar	Represents the Alert count for the particular date

The alert page also displays metrics on the number of alerts based on High, Medium, and Low priority, with a timeline chart that displays the count of alerts based on the created date. The user can apply filters in the list to view the count of alerts based on priority for required ranges.

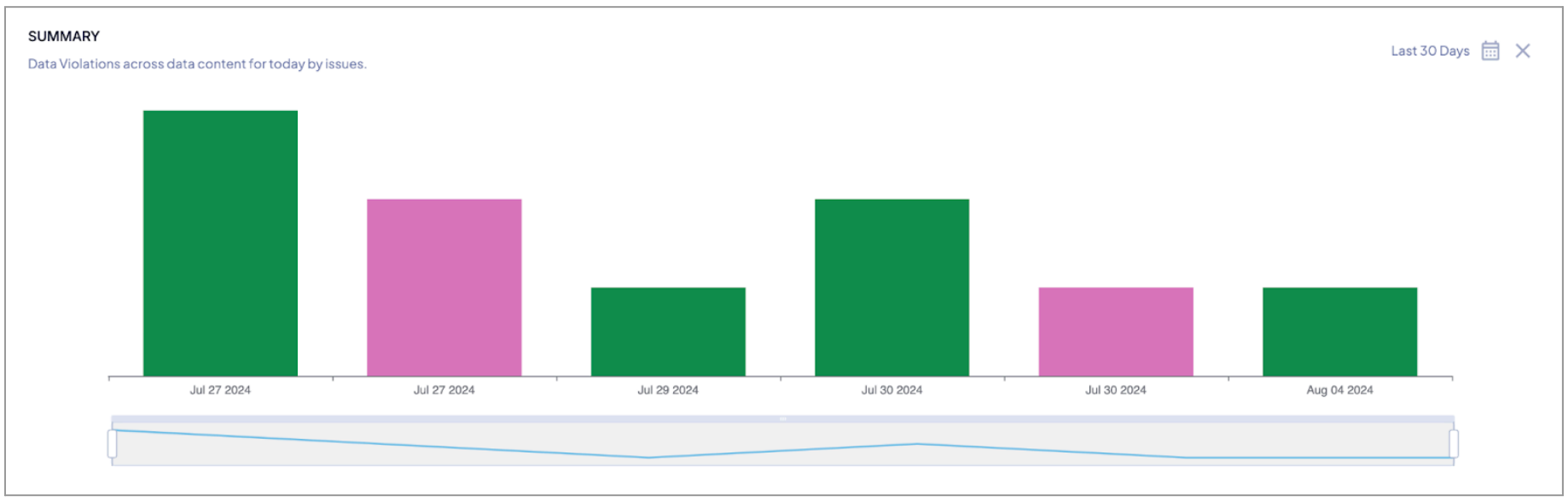
The list of alerts displays all the alerts raised in the platform. The users can view assets and attributes, measure the priority of the alert in the list, and can also choose other columns from the select column option. The admin/privileged users will also be able to create issues from the list page for desired alerts. The users can search, sort, and filter the alerts in the list using the list functionalities.



Issues

The issues are the problems that are identified in the data that are under consideration. These issues are populated under the **Remediate** > **Issues** tool. The issues are created based on the alerts that are generated when the measures are defined for the organization of data. Whenever there is a violation of a measure, an alert is generated. The issues are created based on the alerts.

The summary has a visual representation of the issues that are created for the data violations. The summary gives an idea of the severity of the issues for the incumbent data.



Component	Description
X-axis	The date is represented on the X-axis
Y-Axis	The number of issues is represented on the Y-axis
Bar	Represents the issue count for the particular date

The issue page also displays metrics on the number of issues based on each status of the issue with a timeline chart that displays the count of issues based on the created date. The user can apply filters in the list to view the count of issues based on the status of the issues

The list of issues displays all the issues created in the platform, the users can view assets, attributes, and measures, the priority of the issue in the list, and can also choose other columns from the select column option. The users can search, sort, and filter the issue in the list using the list functionalities.

Dedupe (Process)

The deduplication functionality enables users to detect and merge duplicate or similar records across spreadsheets and databases by leveraging machine learning techniques. It is especially valuable for organizations handling messy or inconsistent data, such as variations in names, addresses, or other key fields.

Functionality

- **ML-Based Matching Engine:** Automatically learns how to detect duplicates and similar records based on user-provided examples.
- **Guided Workflow:** A step-by-step interface for uploading data, training the model, reviewing results, and exporting clean datasets.
- **Customizable Field Matching:** Supports complex logic for comparing names, emails, addresses, business data, and more.
- **Scalable Integration:** Compatible with large-scale datasets from CRM, ERP, and marketing systems.

Use Cases

- Merging customer records from multiple sources
- Cleaning and enriching mailing lists
- Consolidating vendor and business data
- Preparing data for analytics, reporting, or CRM migration
- Resolving duplicates across merged databases

Business Problems Solved

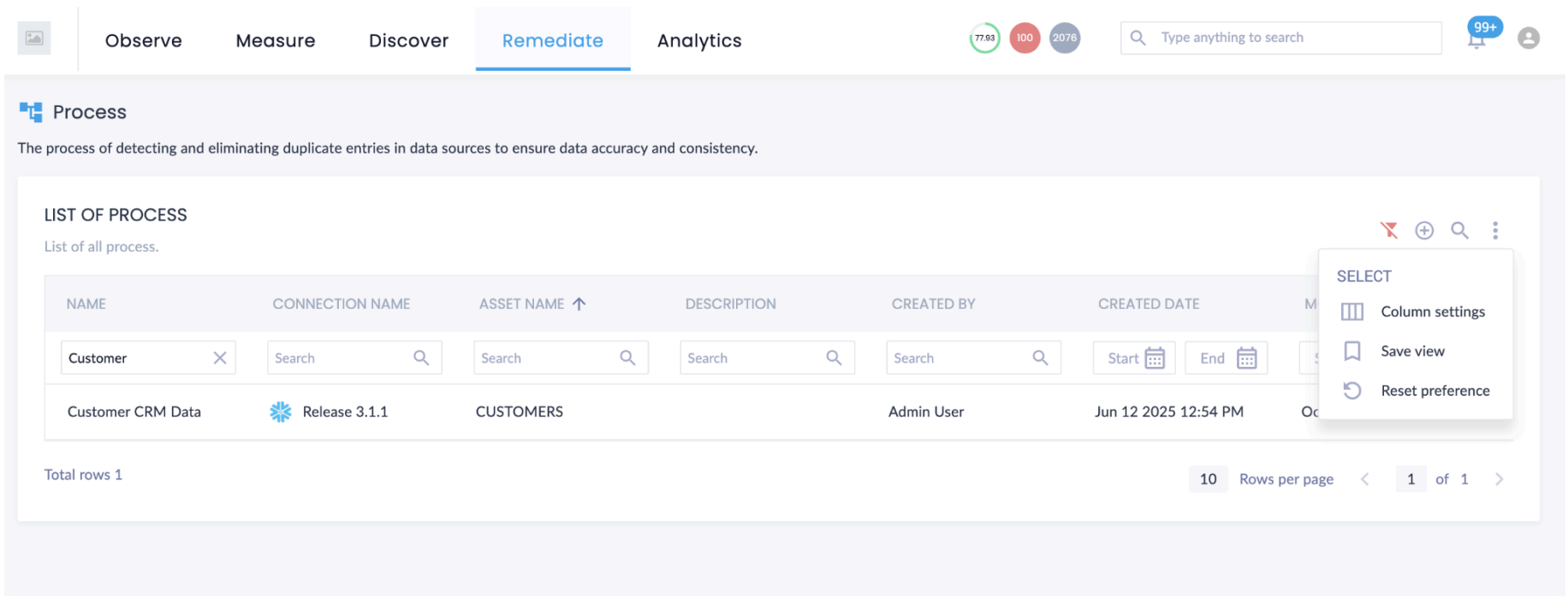
- Eliminates redundant or duplicate entries, improving data quality
- Saves time and labor through automation
- Enhances reporting accuracy and customer insights
- Reduces the cost of poor data quality in operations and marketing

These users can search, sort, and filter the list of Processes. Users can select which columns to display in the Process list using the select column option and can also drag and drop columns to reorder them.Users can create new Process by clicking the **Plus (+) icon.**

Column Settings-Lets users choose which columns to display and reorder them for a customized view of the measure list.

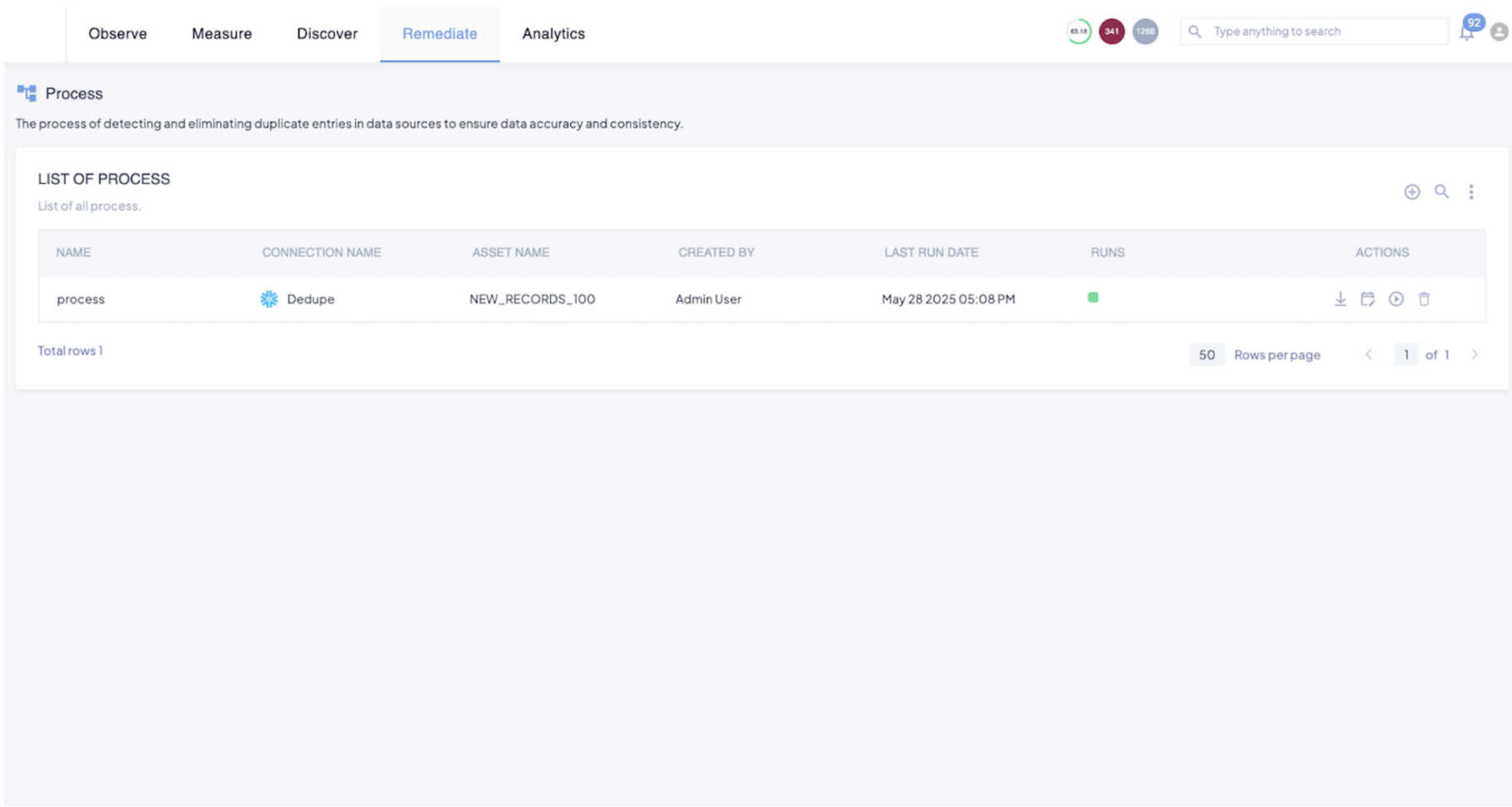
Save View-Allows users to save their filtered or sorted view of Processes for quick access.

Reset Preference-Restores the default layout by clearing all saved column settings.



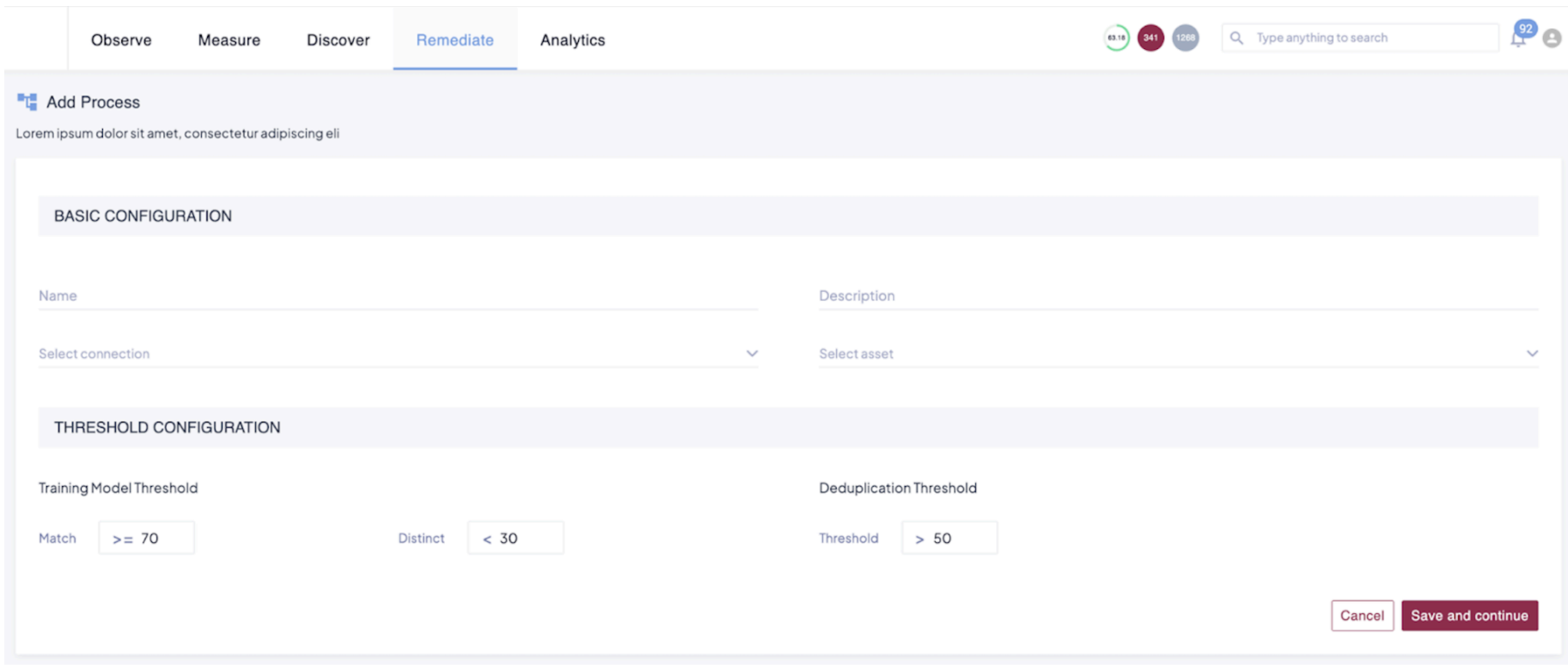
Follow the steps below to create a dedupe process in Quest DQ:

Step 1: Navigate to Remediate → Process



Step 2: Click on the "+" icon and provide the following details, save, and continue

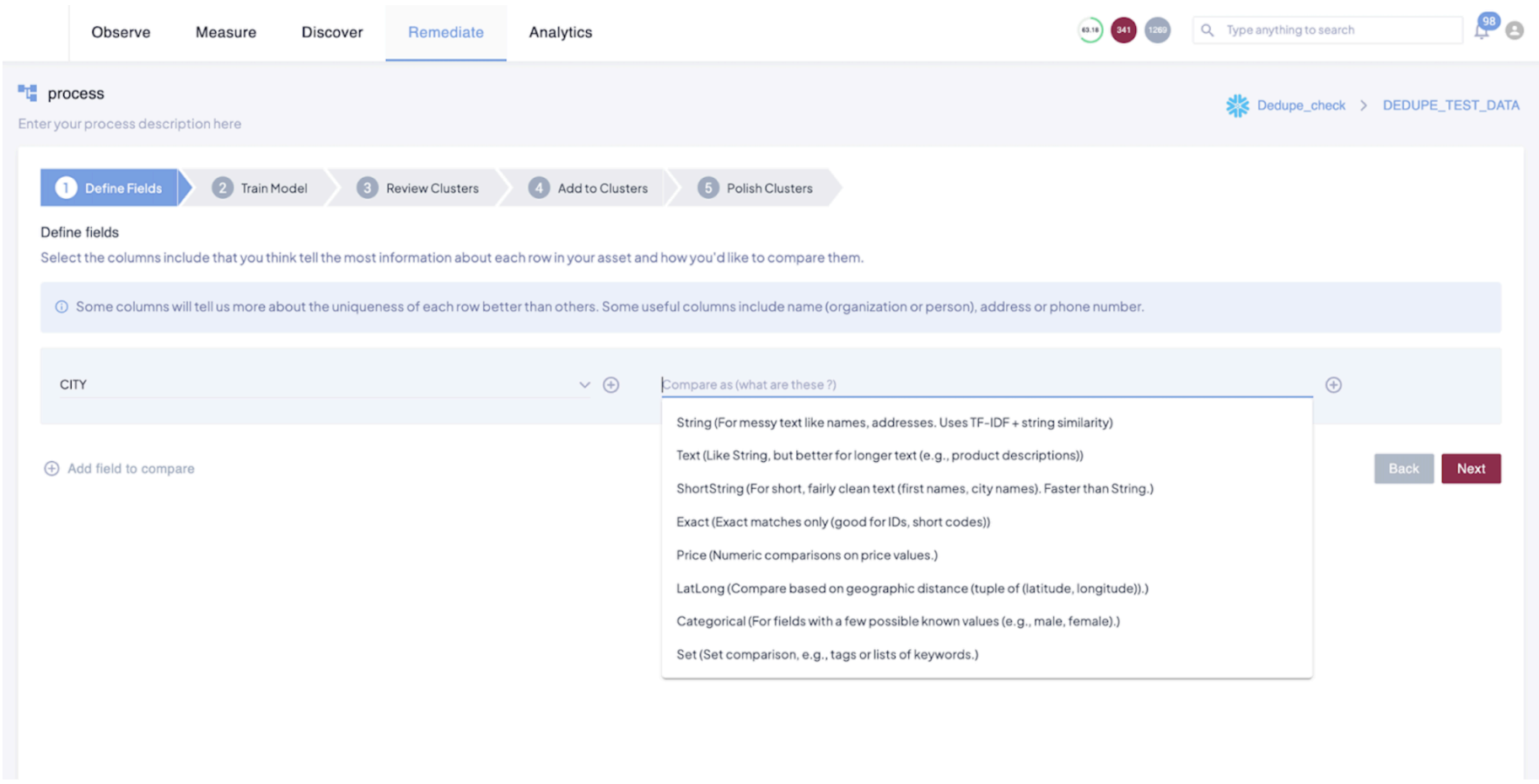
- Basic Configuration
  - Name
  - Description
  - Select Description
  - Select Asset
- Threshold Configuration
  - **Match Percentage** - Refers to the confidence score that two records refer to the same entity. Helps prioritize high-confidence matches for automated deduplication or user review.
  - **Distinct Percentage** - The complement of Match Percentage: this is the confidence that two records are different. Example: 15% match score = 85% distinct = low likelihood of duplication.
  - **Threshold Percentage** - A cutoff value used to decide what is considered a match.



Step 3: Define fields in this section by selecting the attributes and selecting the compare as an option, as described below, and proceed to the next

- Choose the attributes (columns) to consider when identifying duplicates.
- Typical fields: Name, Email, Phone, Address, Company, etc.
- For each field, specify the type:
  - **String**: For names, business names, etc.
  - **Address**: For structured postal addresses.
  - **Set**: For unordered lists like tags or categories.
  - **Exact**: For IDs or categories that must match exactly.

- **Text:** For longer texts, such as descriptions
- **Price:** For numeric comparison
- **Latlong:** For geographic distance
- **Categorical:** For possible known values

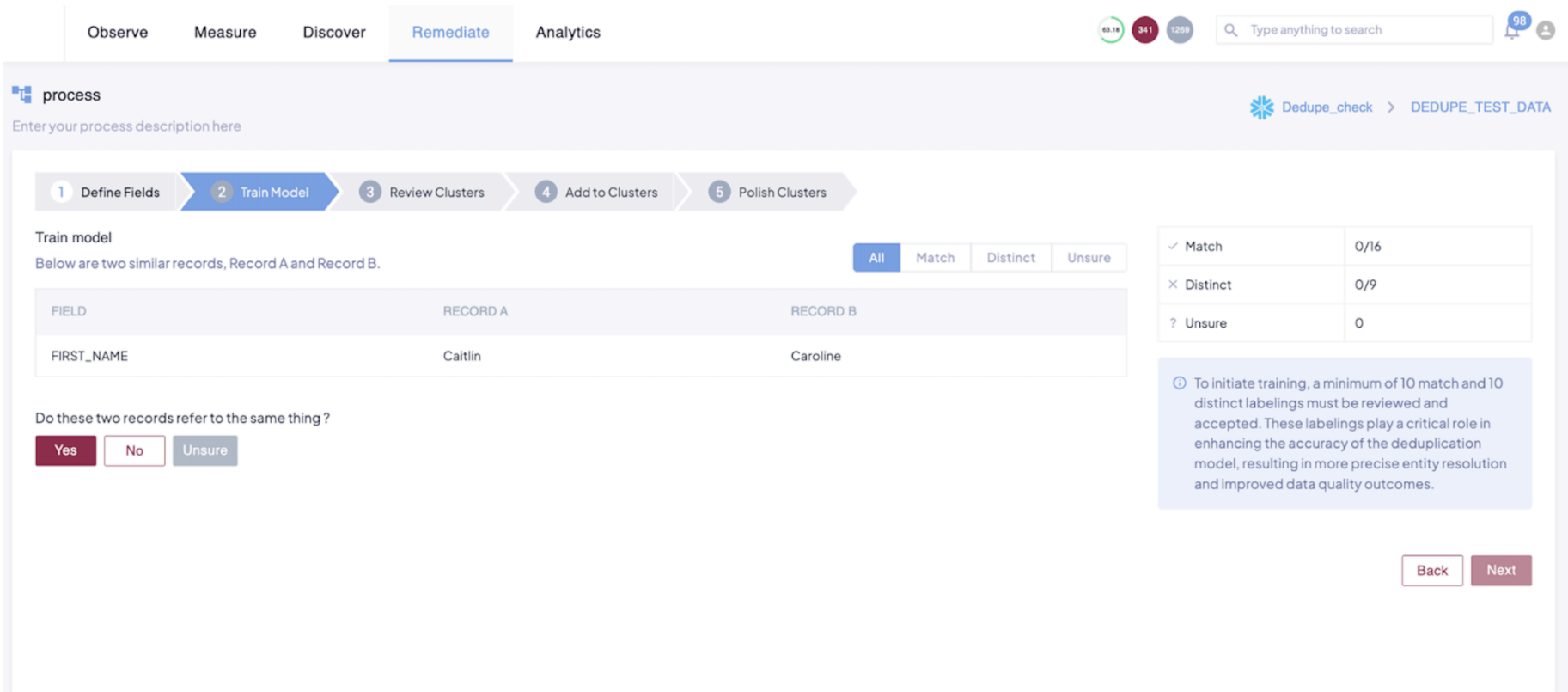


The user can add multiple columns to compare.

Step 4: This section will ask the users to match the records for Distinct and duplicates

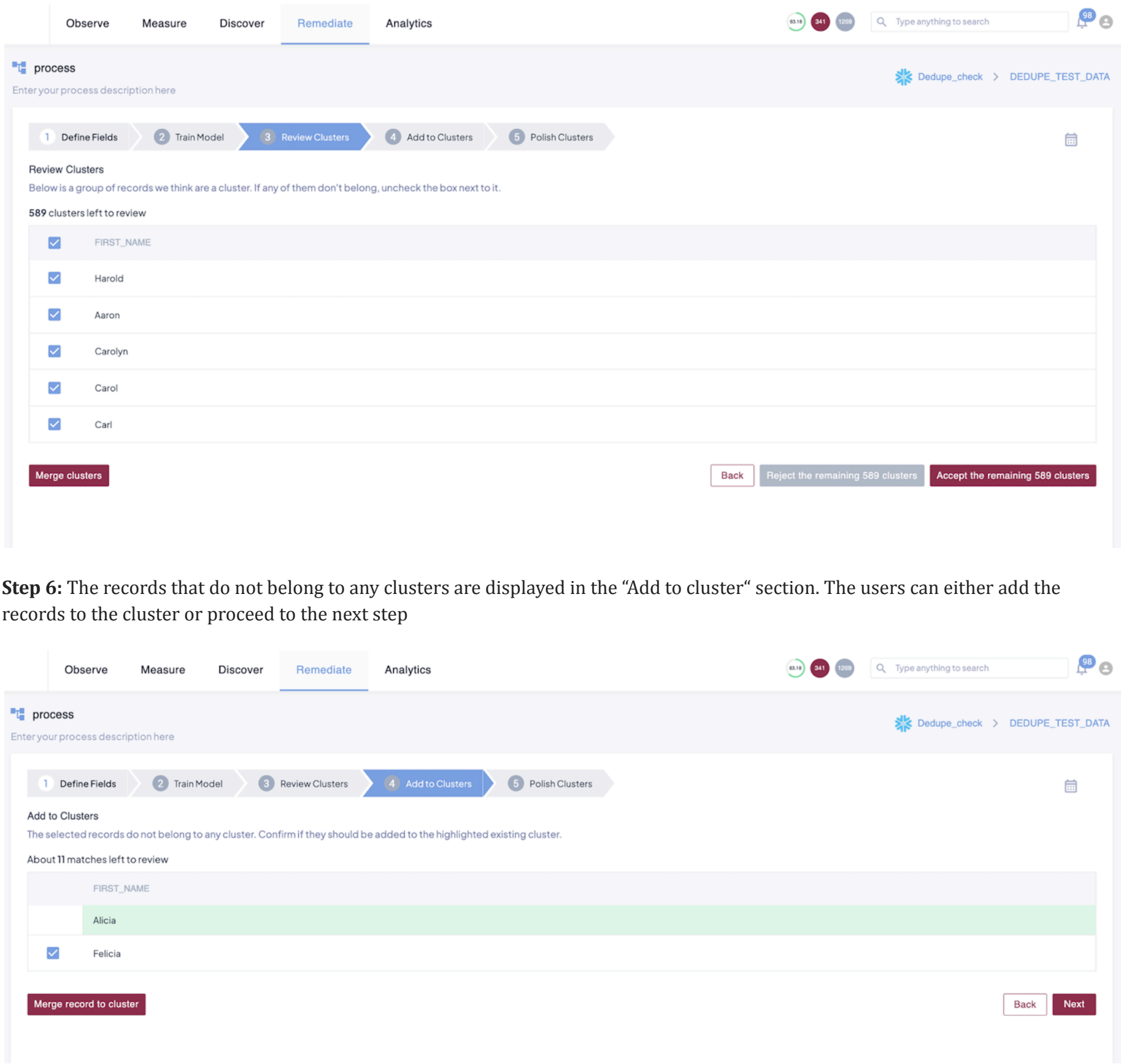
- The system will show you pairs of records and ask: Are these the same?
- You manually label pairs as:
  - **Match** (same entity)
  - **Distinct** (different entities)
  - **Unsure** (skip or decide later)
- The more you label, the smarter the model gets.
- Quest DQ uses active learning to focus on the most informative examples.

Click “Yes” to Match and “No” to Distinct. A total of 10 records should be mapped for both Match and Distinct to proceed further.

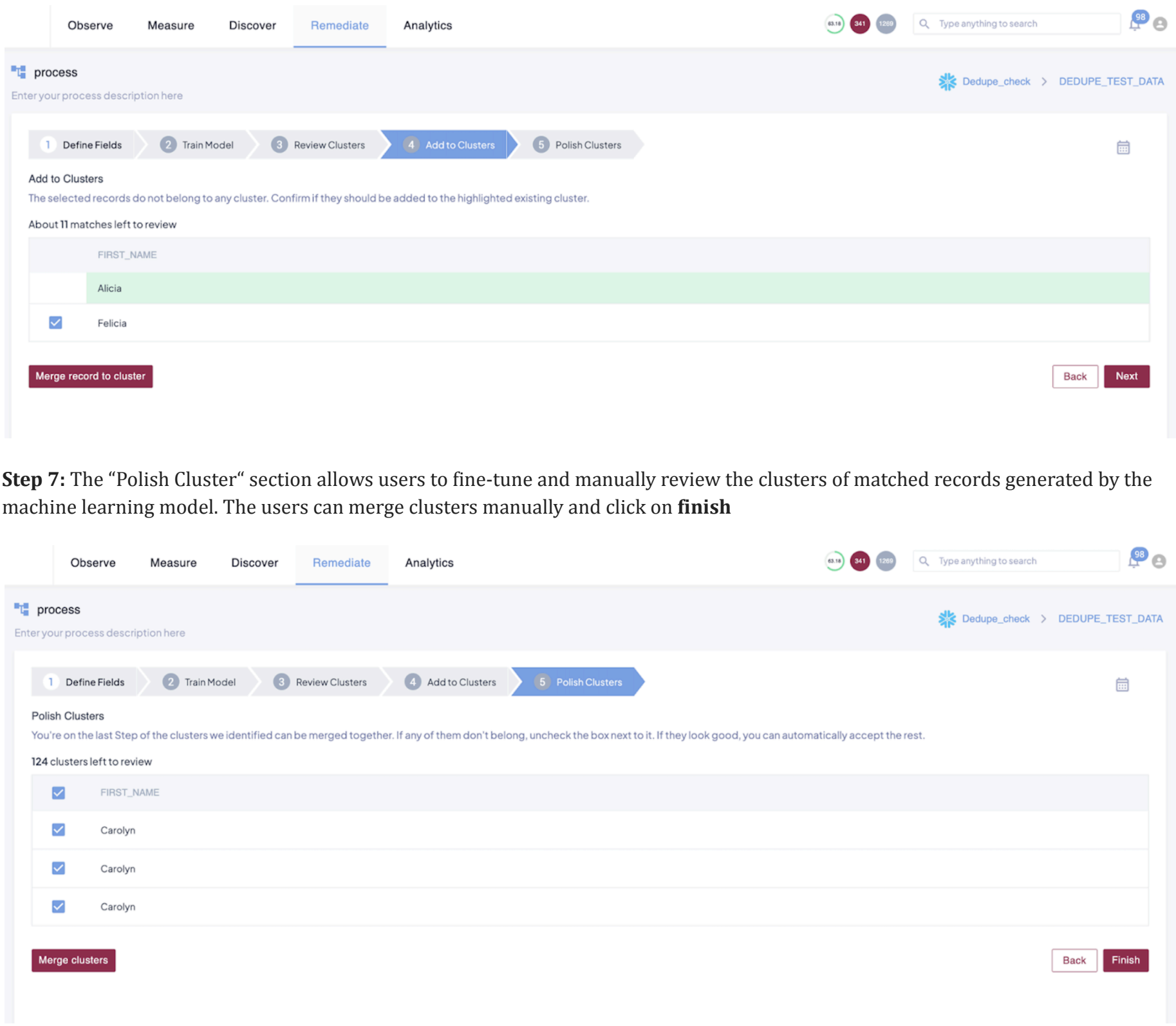


Step 5: The records that match are grouped as a cluster, and the user can review and confirm the records

- After training, Quest DQ shows **clusters of records** that are the same entity.
- Review and adjust clusters:
  - Merge or split as needed.
  - Confirm or reject system decisions.

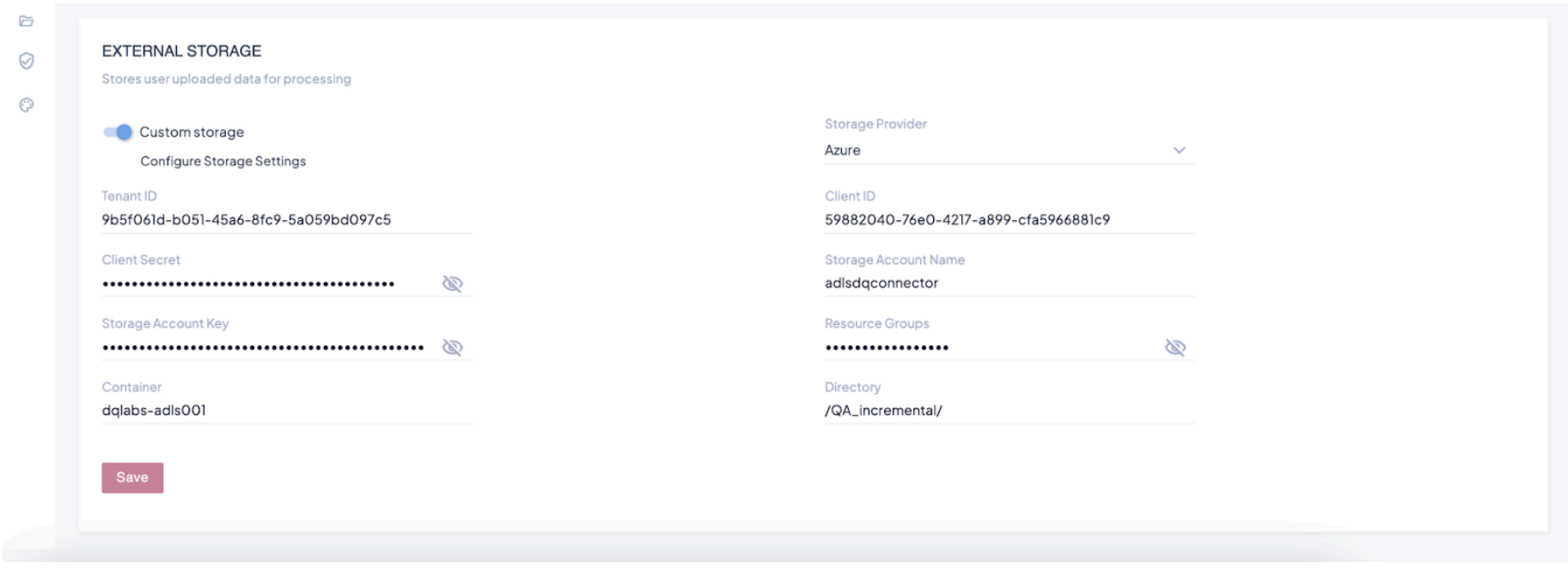


Step 6: The records that do not belong to any clusters are displayed in the "Add to cluster" section. The users can either add the records to the cluster or proceed to the next step



Step 7: The "Polish Cluster" section allows users to fine-tune and manually review the clusters of matched records generated by the machine learning model. The users can merge clusters manually and click on **finish**

Once the setup is complete, sample records will be downloaded to the local machine where the user can review the results. Also, the user can create schedules to run the process, and then processed files are located at the storage specified under the external storage section under settings



Limitation: The Dedupe functionality is currently supported only on Snowflake

DATA CONNECTORS

ADLS

Azure Data Lake Storage (ADLS) is a highly scalable and secure data lake solution provided by Microsoft Azure. It is designed to handle large volumes of data, making it ideal for big data analytics and storage needs. Quest DQ allows users to connect to ADLS and monitor the foundation health checks.

Current Implementation

Quest DQ leverages the Spark clusters to create iceberg tables for the connected files in ADLS, and the measure queries will be executed on the iceberg table created to get the metadata information. Once all the measures are executed and the metadata is extracted, the iceberg table will be dropped from the database.

Currently, the following measures are supported in ADLS connectors

- OOB Measures
- Conditional
- Query
- Comparison Measure

A user can provide the folder or file path that creates the ADLS connection, and each connection can create only one asset in Quest DQ with the user-specified column names and datatypes. Currently, the following file types are supported

- Parquet
- XML
- CSV
- JSON

Prerequisites

Whitelist IP

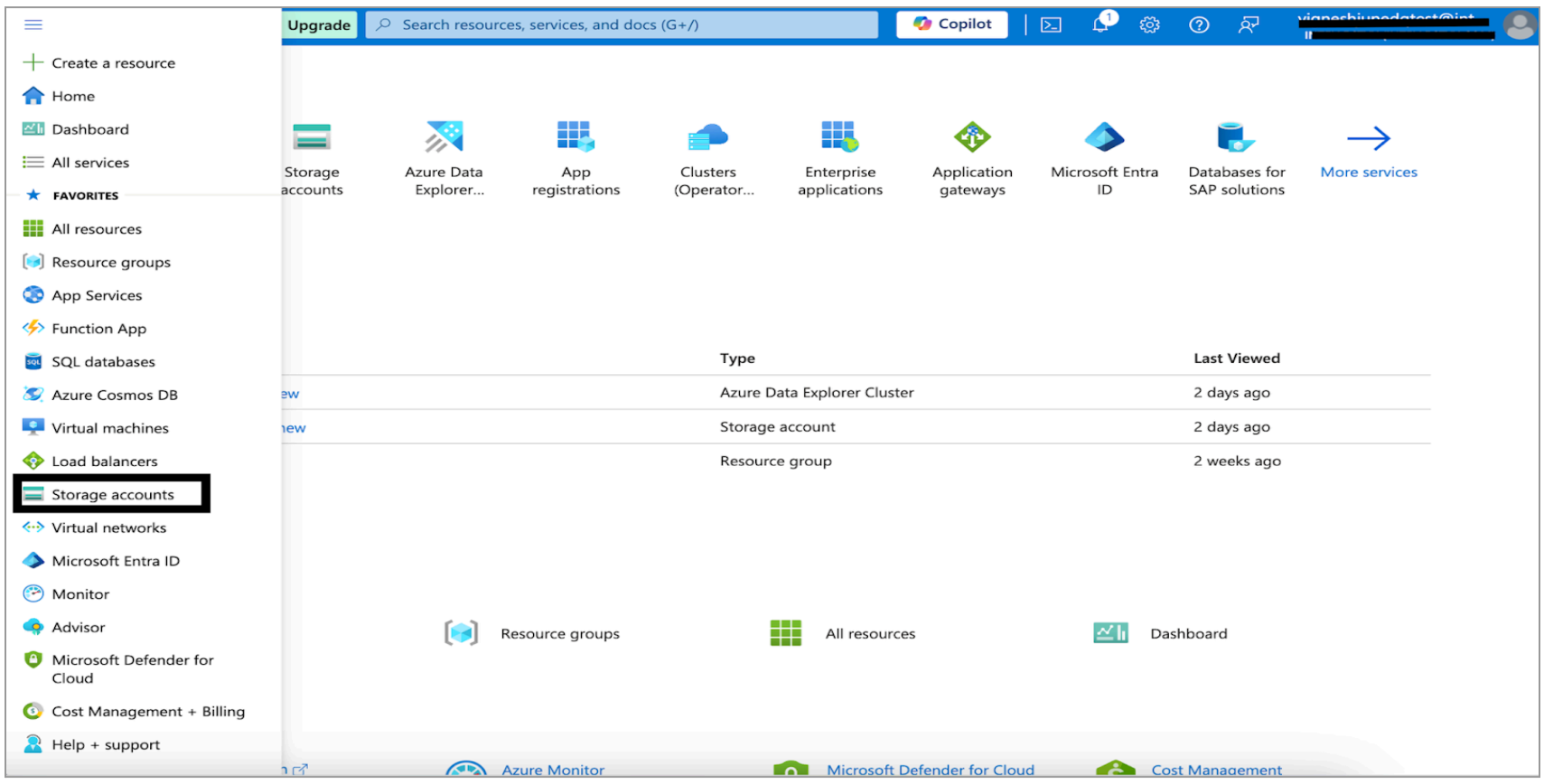
If your organization uses a whitelist to manage ADLS access, Quest DQ will only access your ADLS through IP. For assistance on whitelisting, kindly reach out to Support team.

Storage Accounts

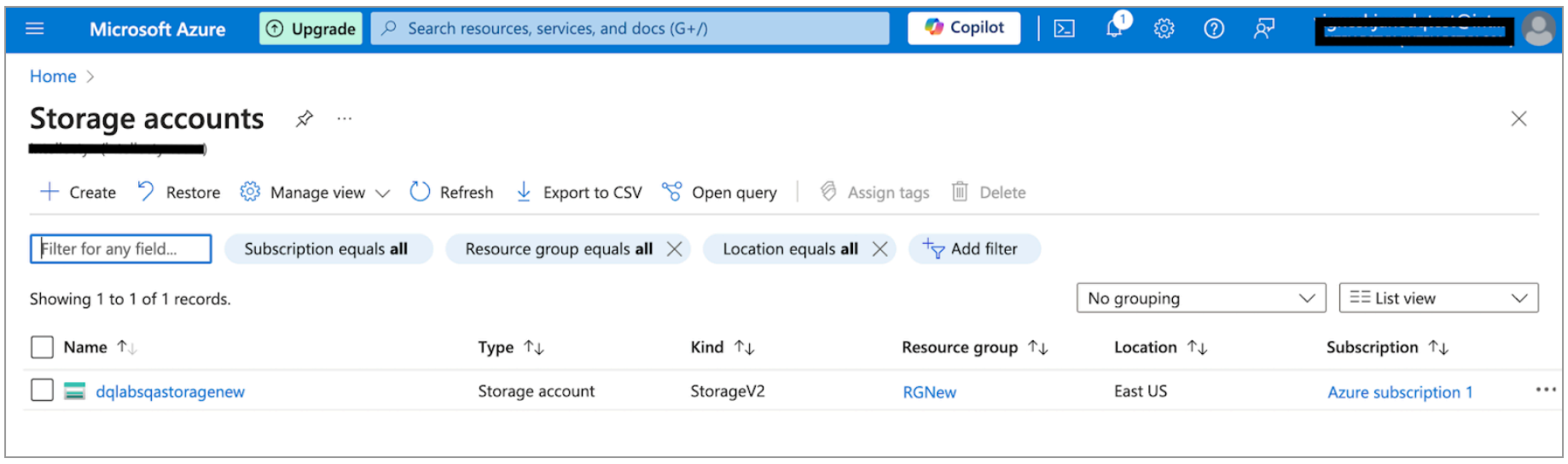
Please follow the steps below to create storage accounts in ADLS:

Step 1: Log in to the Azure Portal

Step 2: Click on the Left Menu to view the list of services available and select “Storage Accounts”

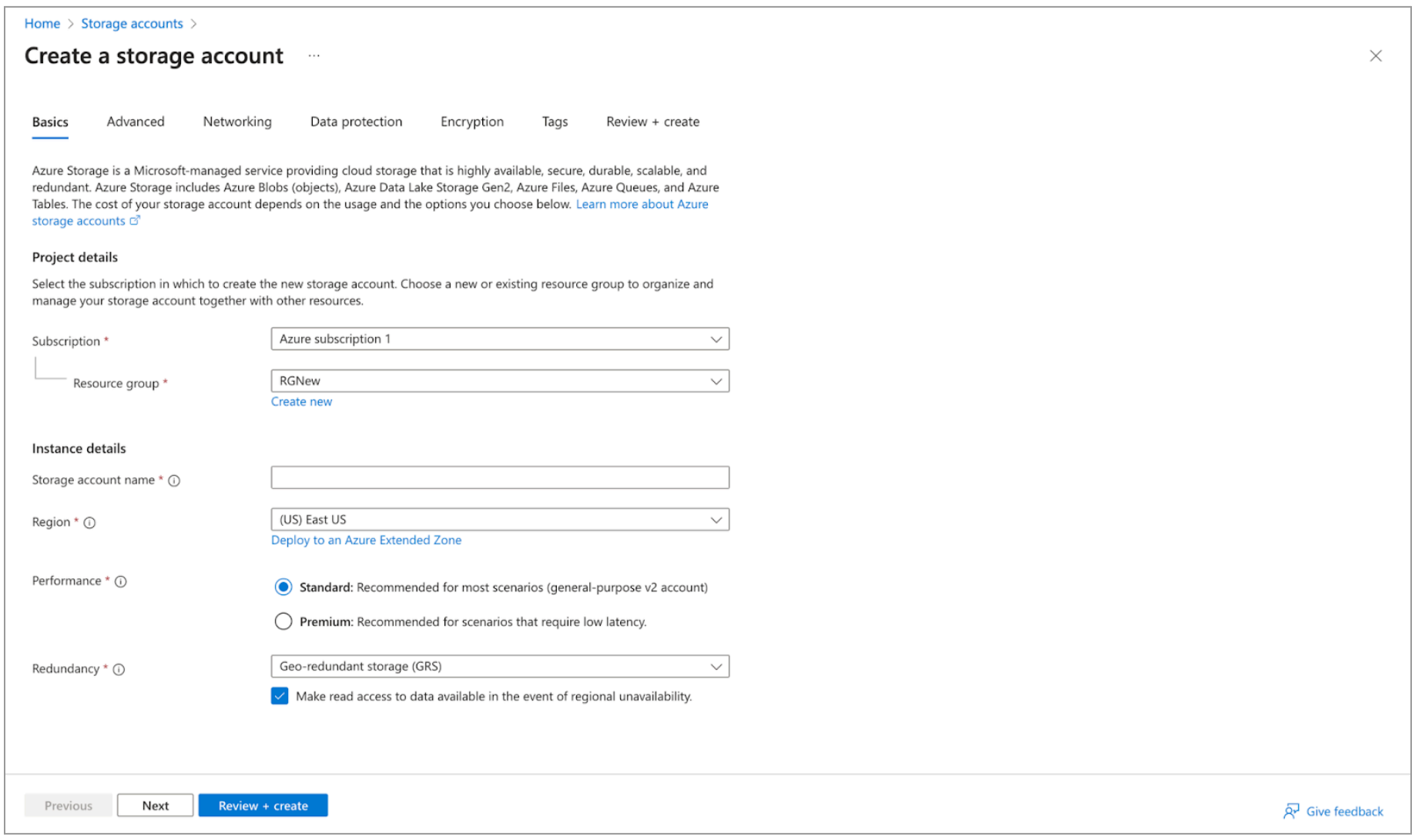


Step 3: On the Storage Accounts, Click on “Create“



Step 4: Provide the required details in the following tab based on the required specification:

- Basic
- Advanced
- Networking
- Data Protection
- Encryption
- Tag



Step 5: Finally, review the settings provided and click on “Create“

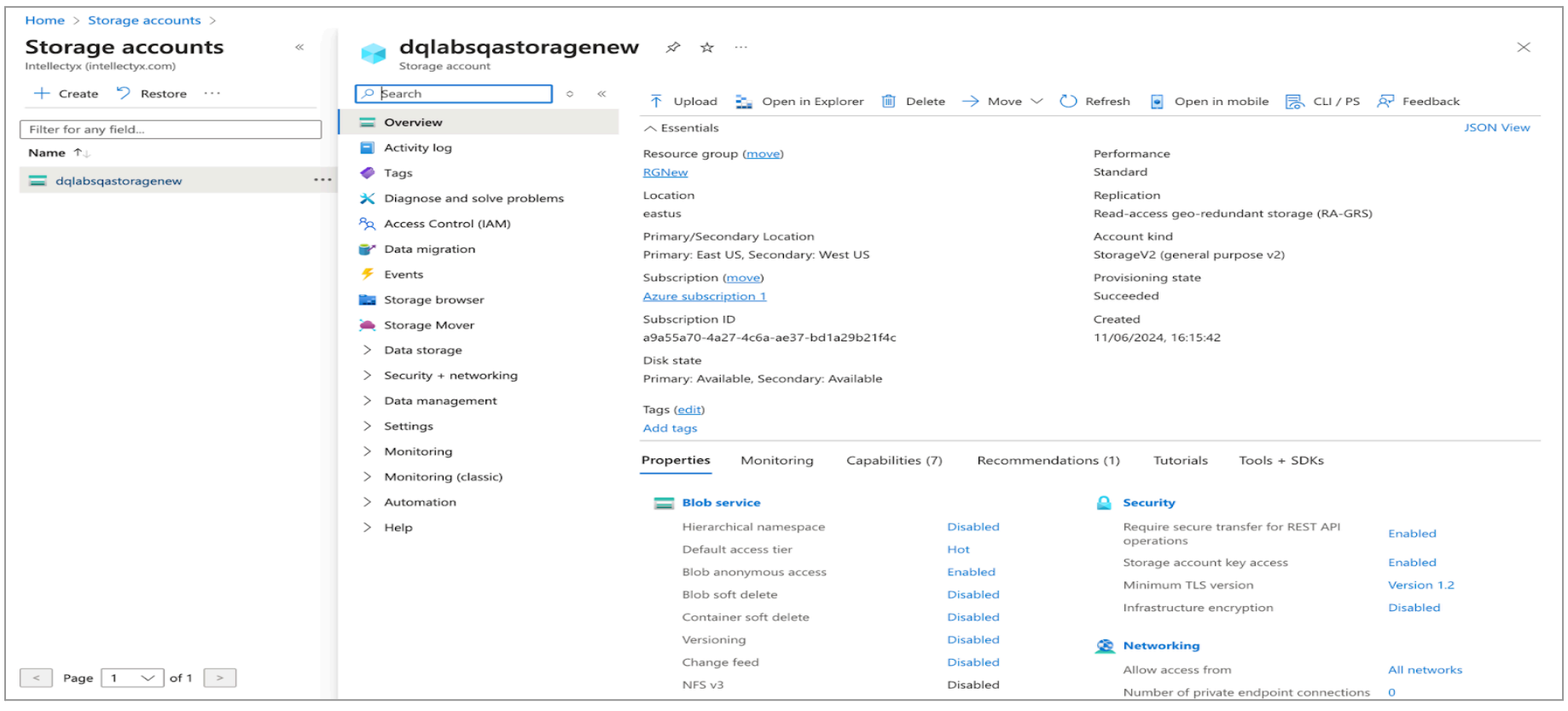
The created storage account will be listed on the storage account page.

Reference Link: https://learn.microsoft.com/en-us/azure/storage/common/storage-account-create?tabs=azure-portal

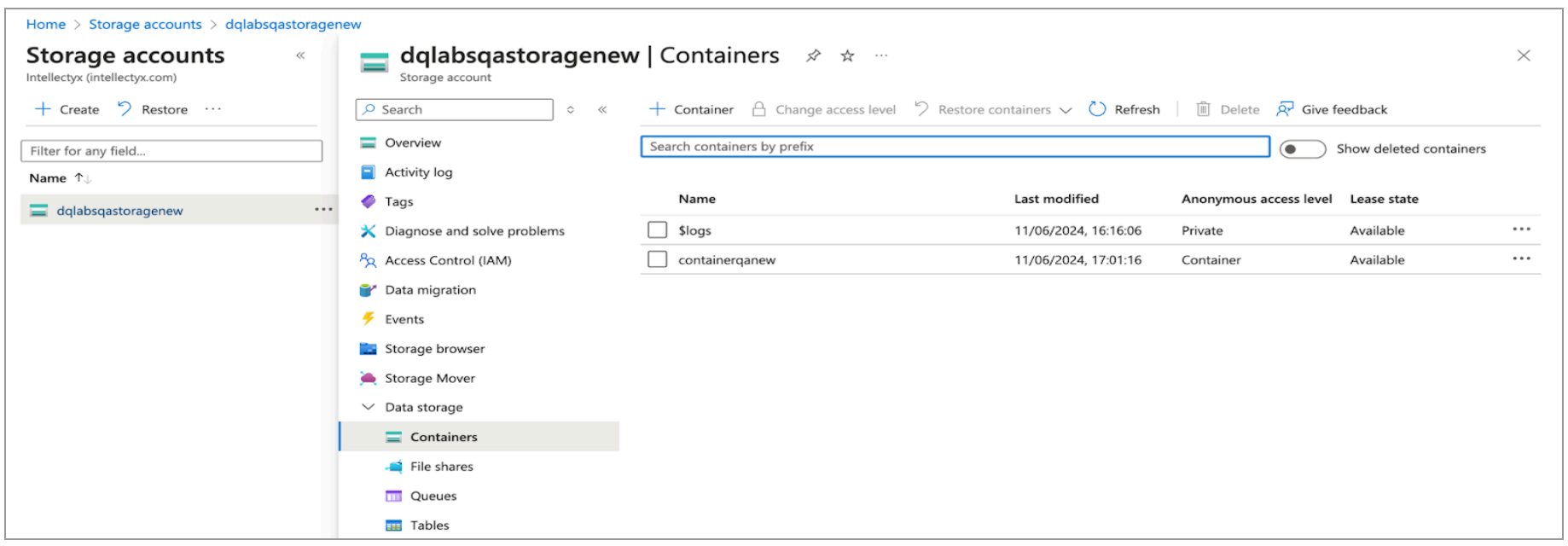
Containers:

Once the storage account is created, the user should create containers inside the storage account where the files can be stored. Please follow the steps below to create containers:

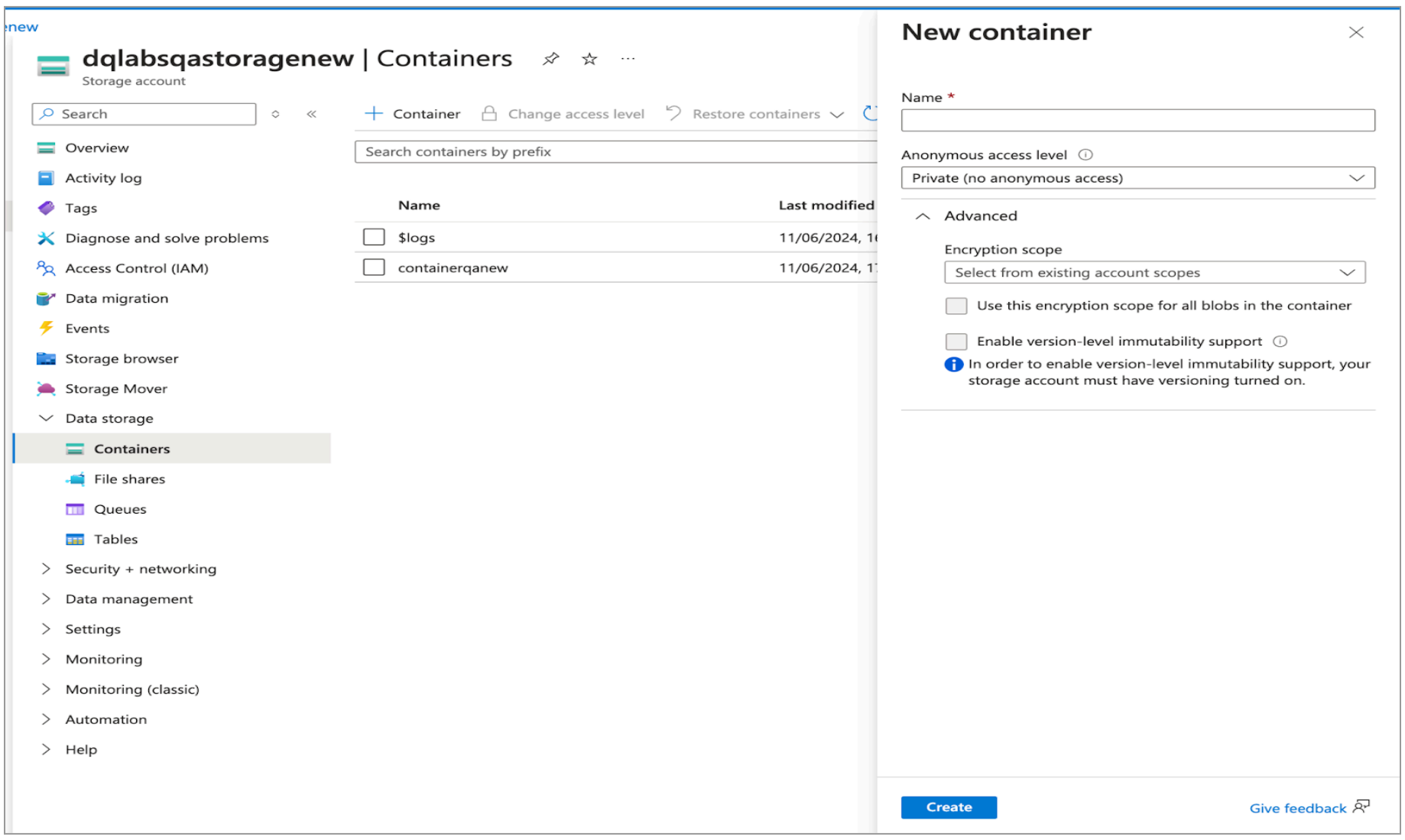
Step 1: Click on the storage account created



Step 2: Click on Data Storage → Container



Step 3: Click on the "+" Container button and provide the requested details, and click on Create



Step 4: Once created, the container will be listed in the containers page, and the users can add files and folders

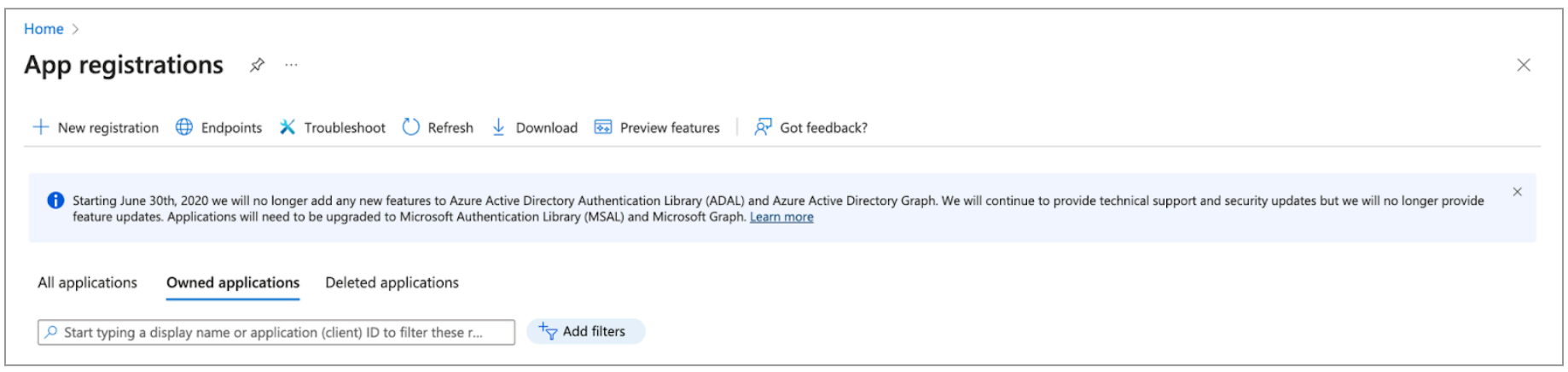
Reference Link: https://learn.microsoft.com/en-us/azure/storage/blobs/storage-quickstart-blobs-portal

App Registrations

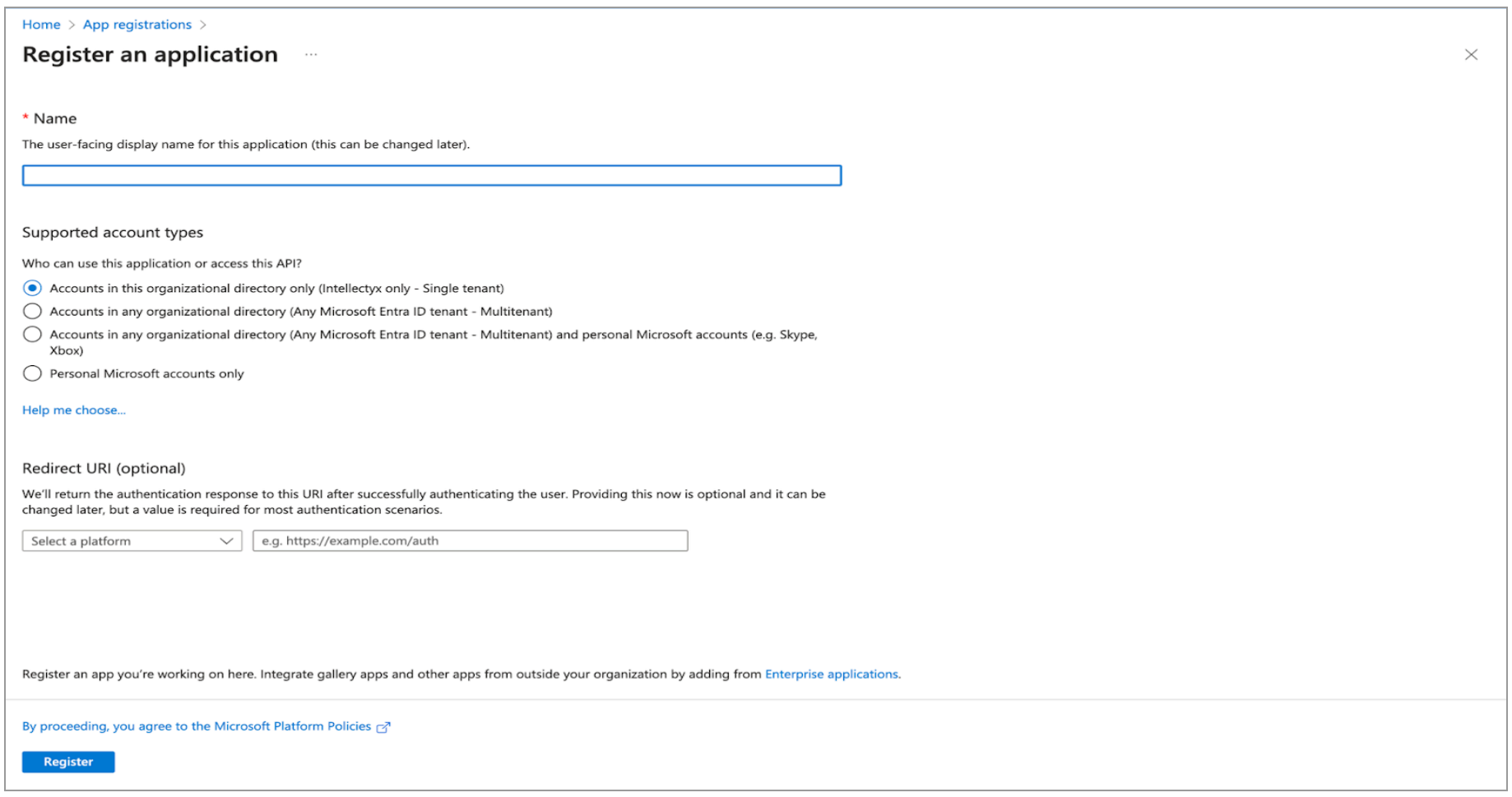
Once files are loaded in the storage account, the user has to create an app registration to generate a client secret. Please follow the steps below to generate a client secret:

Step 1: Navigate to App Registrations

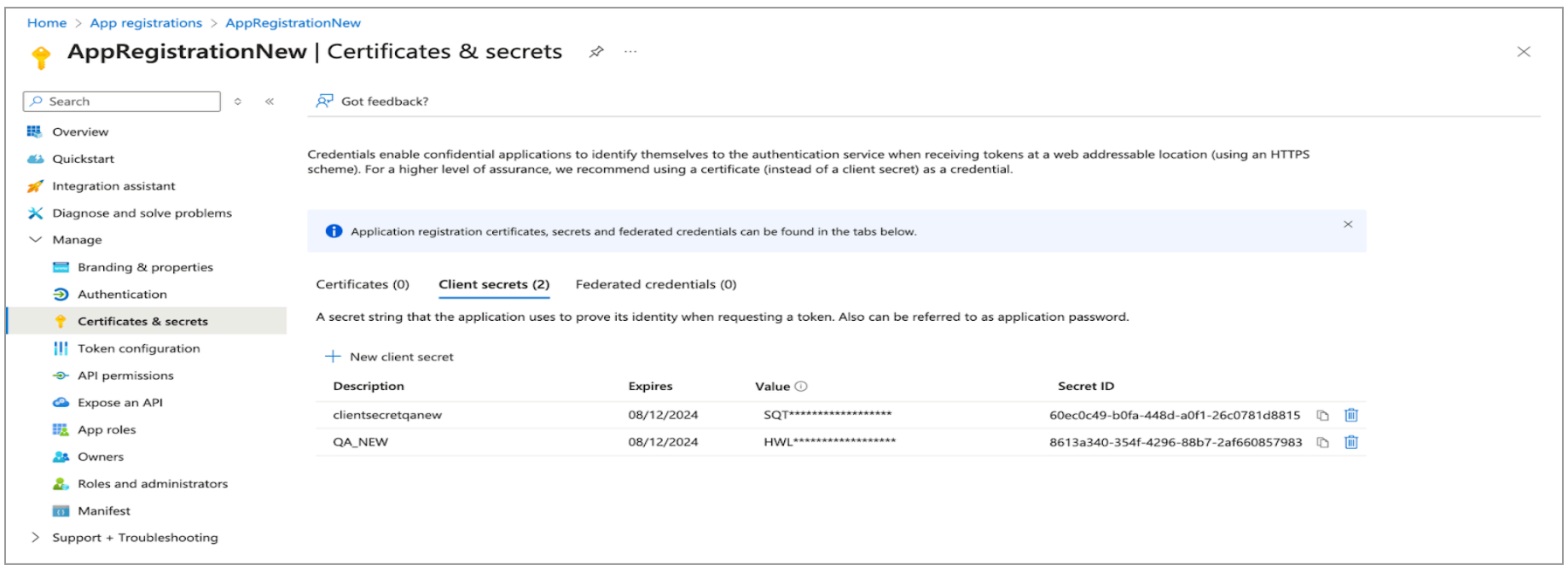
Step 2: Click on "+" New registrations



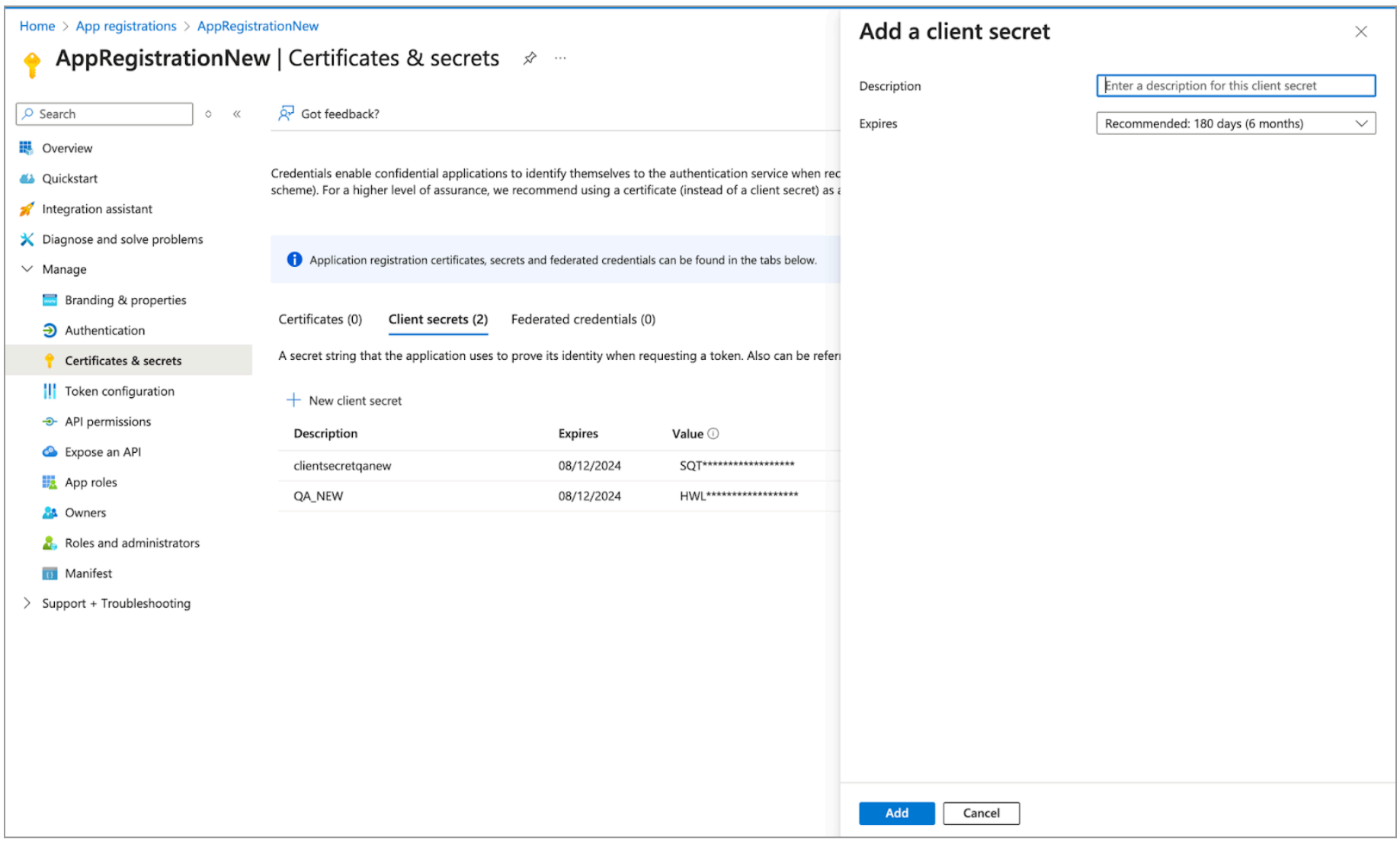
Step 3: Provide the requested details and click on "Register"



Step 4: Once the Registration is created, navigate to the created app registration → Manage → Certificates & secrets



Step 5: Click on “New client secret“ and provide the description and set the expiration date

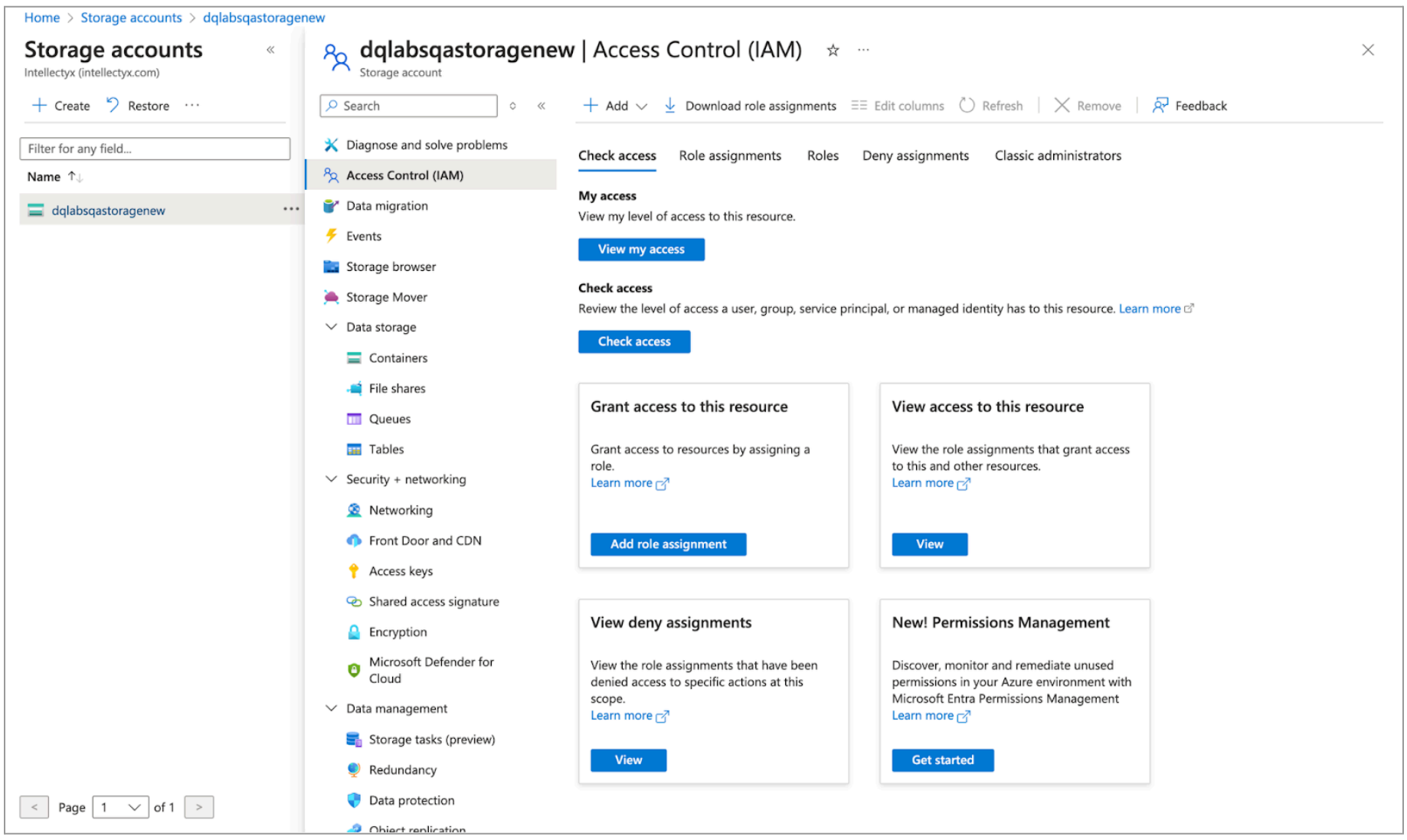


Step 6: Copy the generated secret value and save it for further use.

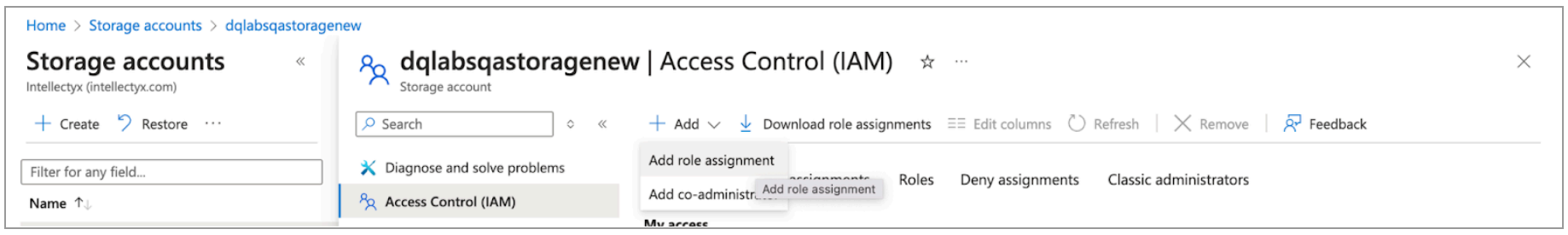
Assign Role in Storage Account

Once app registration is created, the user has to assign the Storage Blob Reader Role to the app registration under the storage account. Please follow the steps below to do the same:

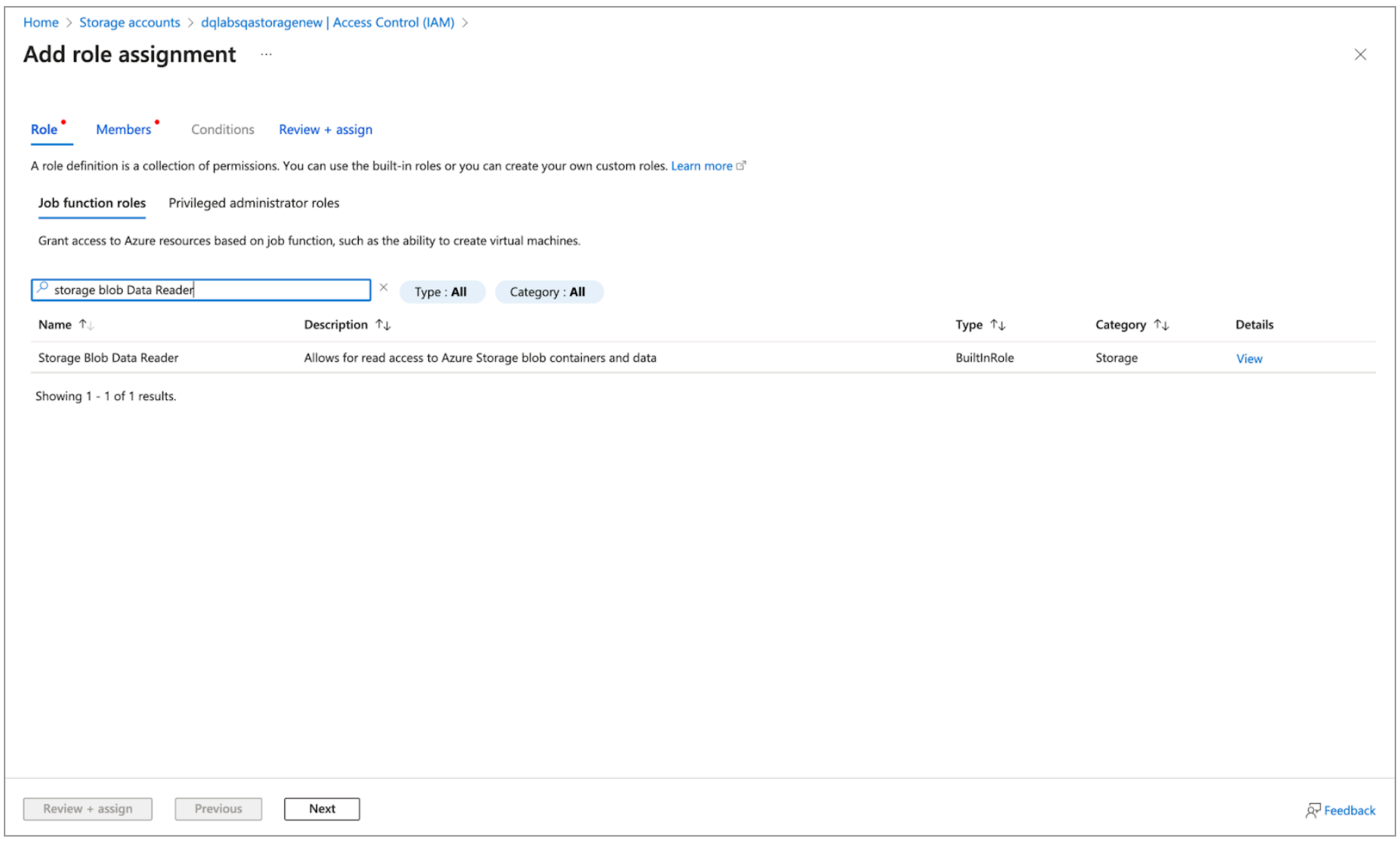
Step 1: Navigate to Storage Account → Select Created Account → Access Control (IAM)



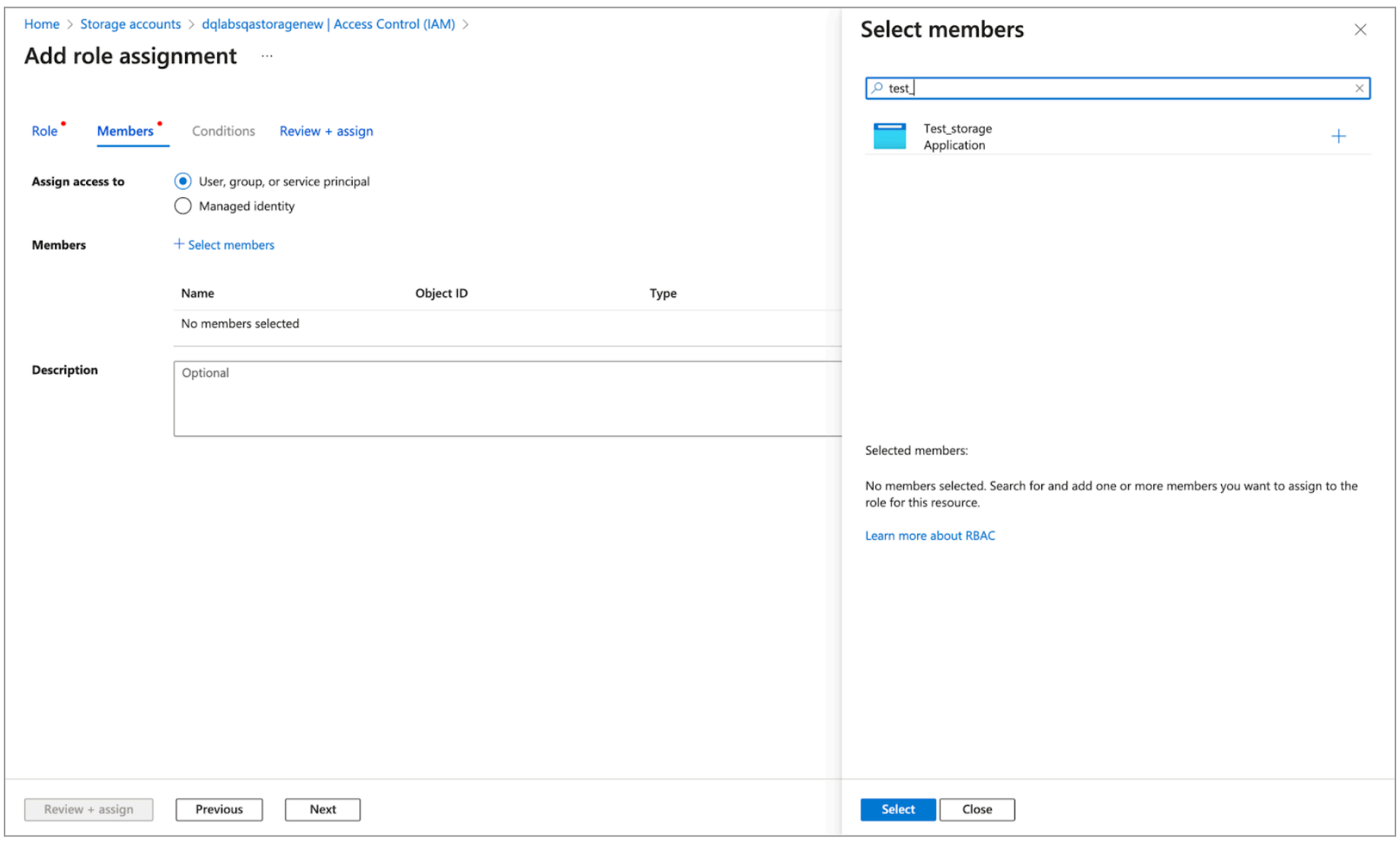
Step 2: Click on “+” Add role assignment



Step 3: Search for the "Storage Blob Data Reader" role and click on "Next"



Step 4: In the Members Tab, click "Select Members" and search for the created application, and click on Next



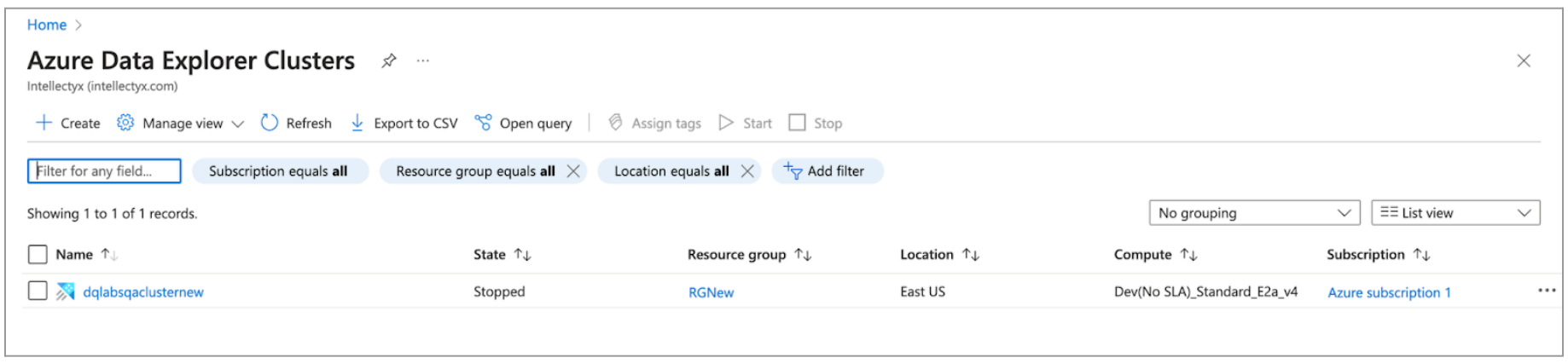
Step 5: Review and Assign as Contributor

Once the role is assigned, the storage account will be accessible by the generated client secret.

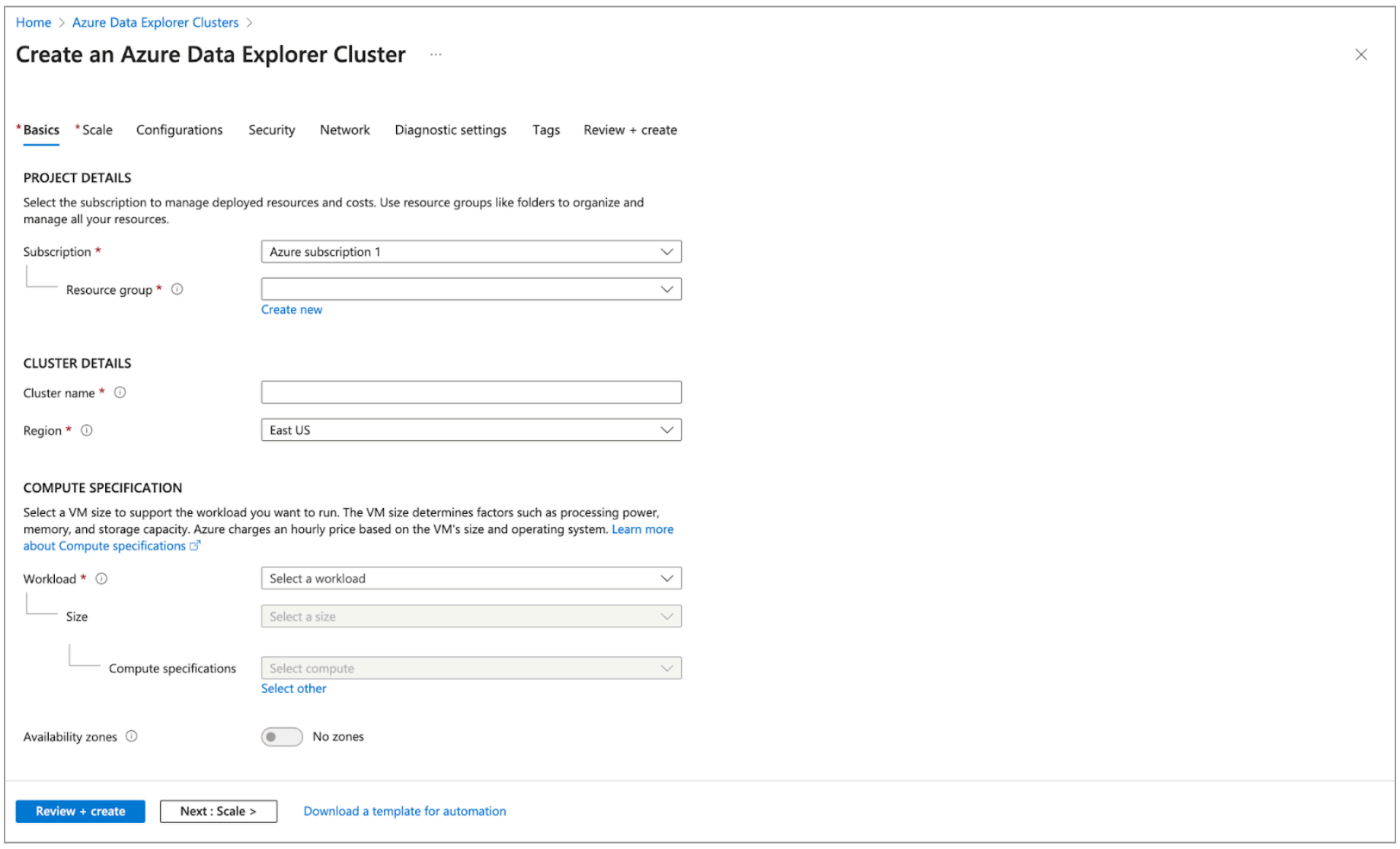
Azure Data Explorer Cluster

Quest DQ leverages the Azure data explorer cluster to connect to ADLS and fetch data, Please follow the steps below to create the cluster:

Step 1: Navigate to Azure Data Explorer



Step 2: Click on the "+" create button



Step 3: Provide the required details and click on create

Step 4: Once created, provide the following access in Access control (IAM) for the app registration

- Database investor
- Database user anonymous access in the container

A database must be created to store the external tables while processing

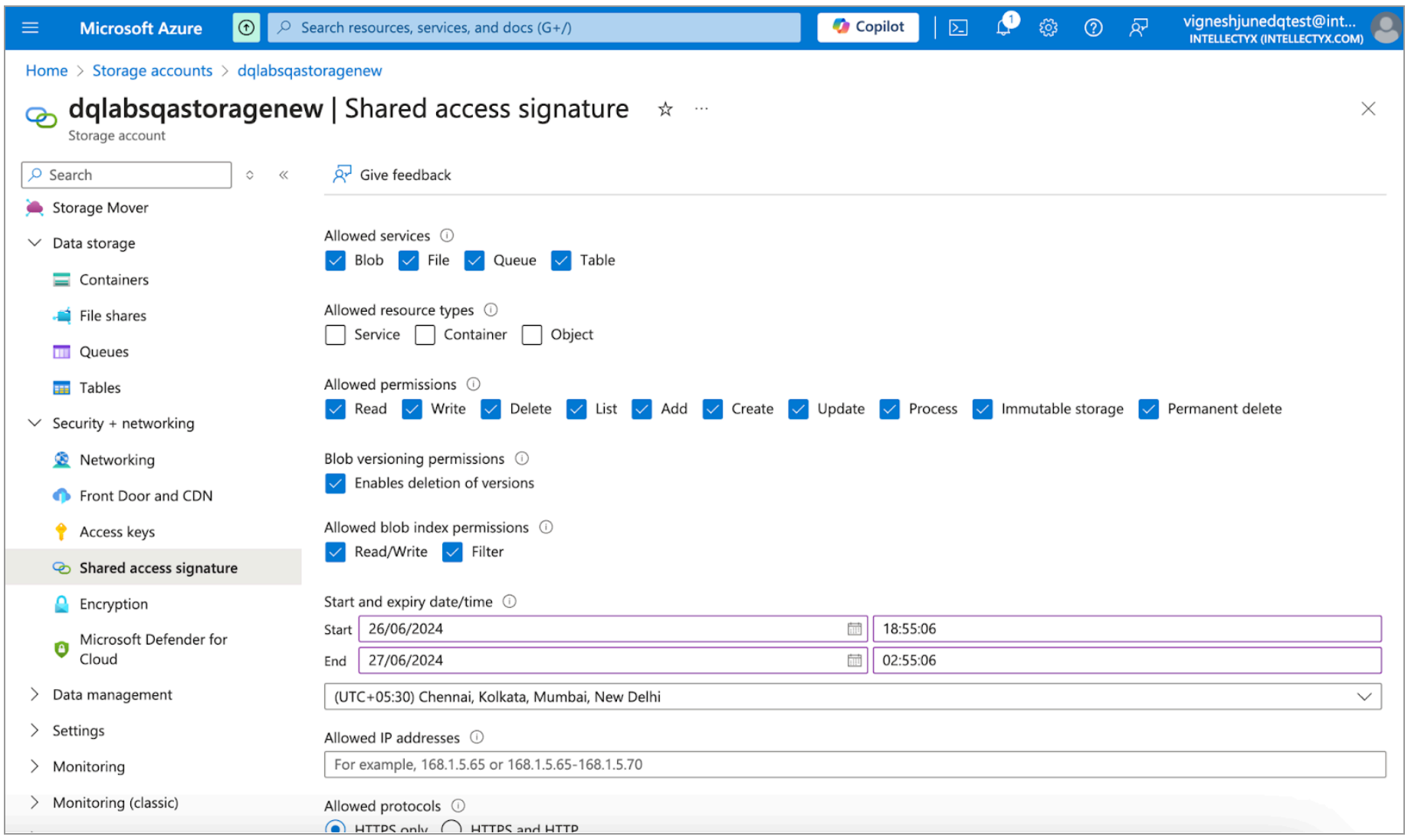
Reference Link:https://learn.microsoft.com/en-us/azure/data-explorer/create-cluster-and-database?tabs=free

Shared Access Key

Quest DQ uses SAS Token for authentication to storage accounts and kustos, please follow the steps below to create a SAS Token:

Step 1: Navigate to Storage Account → Security + Networking

Step 2: Click on Shared Access Signature



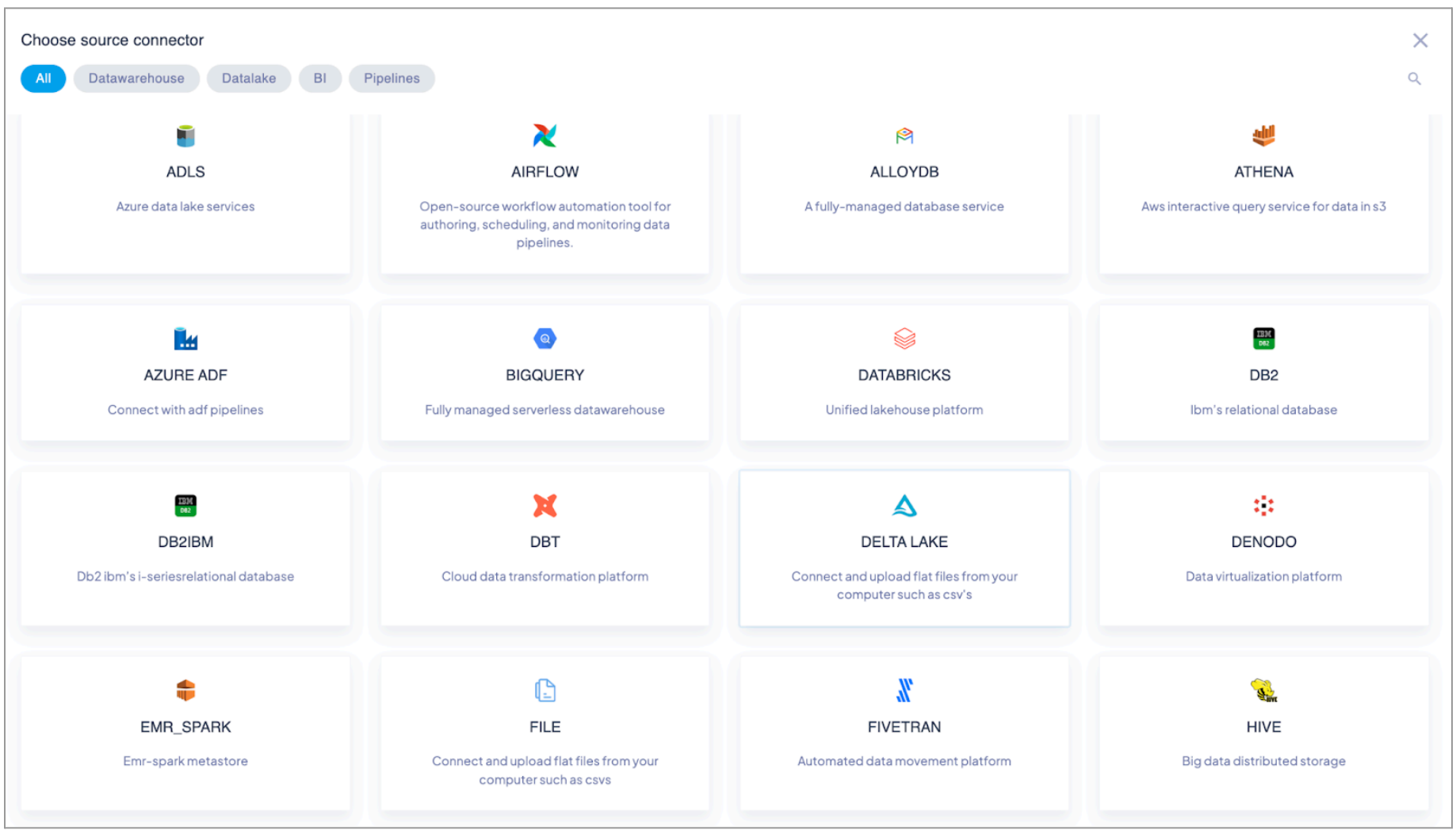
Step 3: Provide access to all services, all permissions, blob versioning, and blob index permission in the SAS, and generate the SAS

Step 4: Copy the key from Access keys

Connect to ADLS

Follow the steps below to connect to ADLS from Quest DQ and create assets:

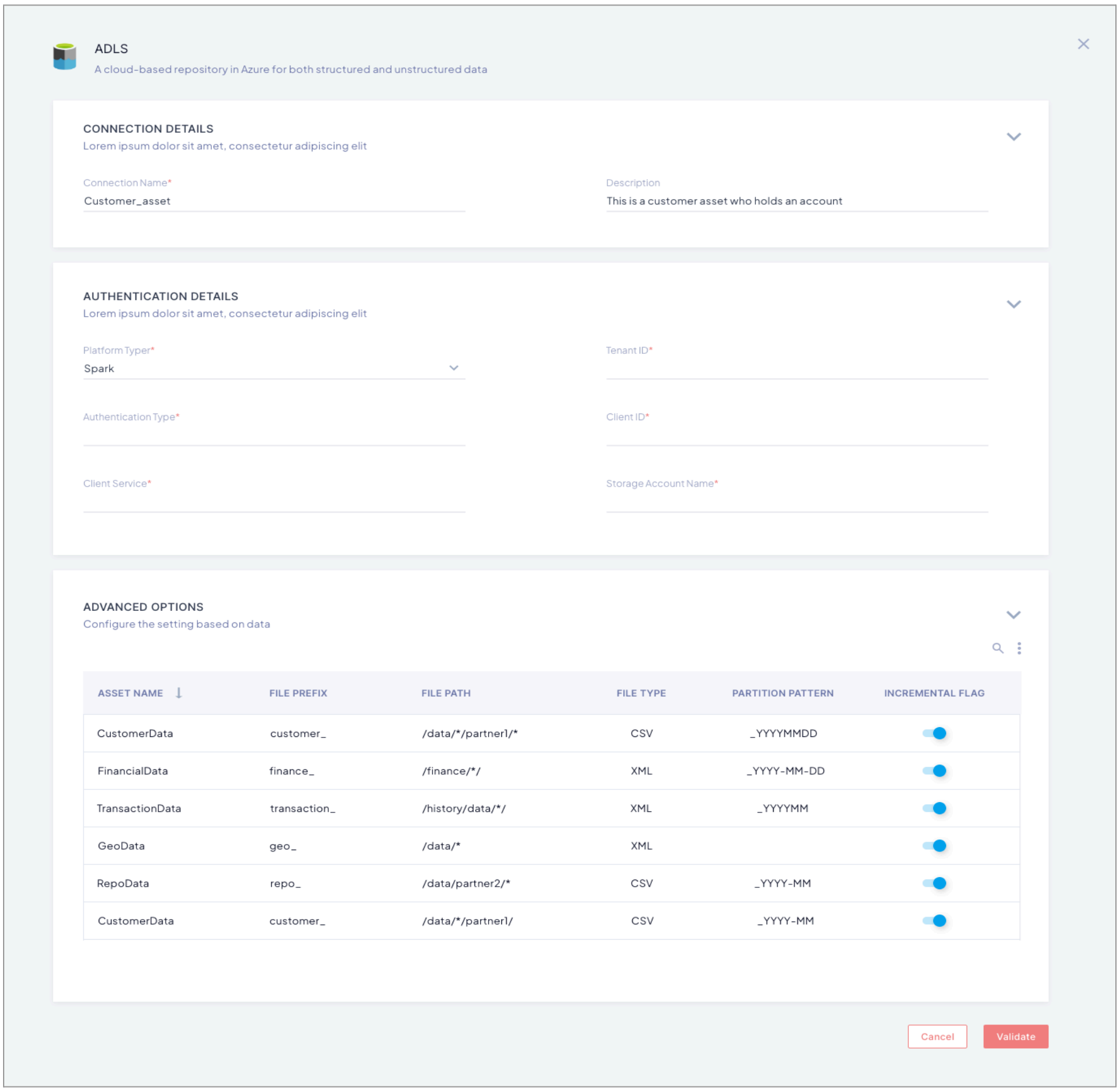
Step 1: Navigate to Settings → Connect → Source and click on the "+" icon



Step 2: Click on ADLS and provide the following details

- Connection Details
  - Connection Name
  - Description
- Authentication Details

    - Platform Type
        - Spark
    - Tenant ID
    - Authentication Type
    - Client ID
    - Client Secret
    - Storage Account Name
    - Storage Account Key
    - Containers
    - Directory
    - File Type
- Advanced Option - The admin/privileged user will be able to configure the asset details using the following details**(for incremental mode - Yet to be implemented)**
    - Asset Name - Name of the asset if different files get merged
    - File Prefix - Prefix of the file to look for
    - File Path - Location of where all files will be stored
    - File Type - CSV, Parquet, XML
    - Partition Pattern - Suffix of Date or Timestamp to identify files with the same prefix but with a partition pattern
    - Incremental Flag - If enabled, then the following two options will be available
        - Fingerprint Column - Which column to use for Fingerprint with the same options as Straightforward date, or option to use casting
        - Incremental Depth - How many days to use for incremental depth
- The admin/privileged user will be able to add multiple assets using the above configuration



Step 3: After providing the details, click on Validate, and once validation is complete, the user can connect to the assets.

The connected assets will be listed on the Quest DQ platform

AWS Athena

Amazon Athena is a service that enables data analysts to perform interactive queries in the web-based cloud storage service, Amazon Simple Storage Service (S3). Athena is used with large-scale data sets.

Amazon S3 is intended for online data and application preservation and backup on Amazon Web Services (AWS). With use cases including data storage, archiving, website hosting, data backup and recovery, and application hosting for deployment, Amazon S3 was developed to make web-scale computing easier for developers. With Amazon Athena, customers can utilize Structured Query Language (SQL) to examine data stored in Amazon S3. The tool is made for speedy, sophisticated, and ad hoc analysis.

Prerequisites

Whitelist IP

If your organization uses a whitelist to manage Athena access, Quest DQ will only access your Athena through IP. For assistance on whitelisting, kindly reach out to the support team.

Account Setup

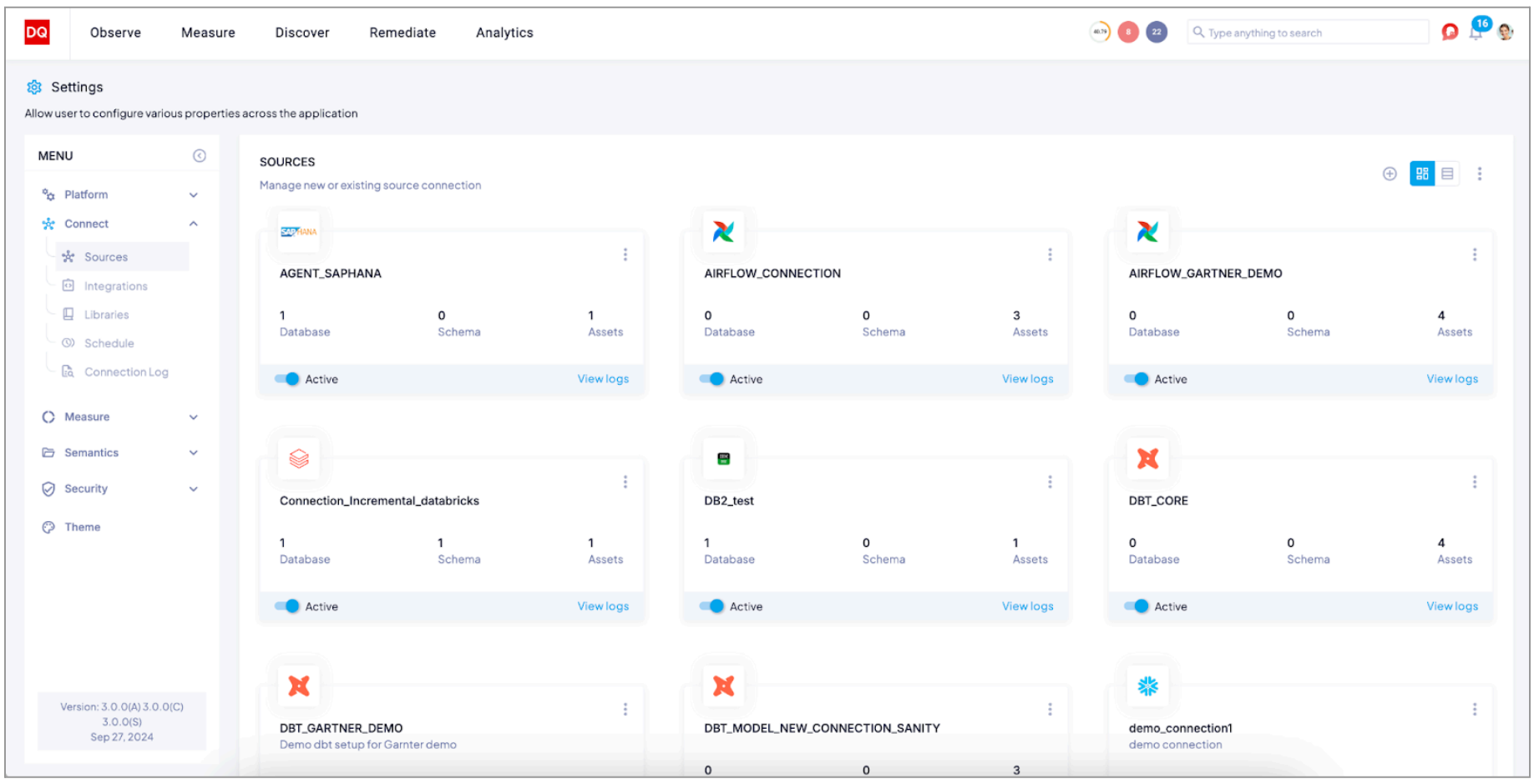
Following are the steps to create a service user and provide access to a database in AWS Athena:

- On the Console Home page, select the IAM service.
- In the navigation pane, select Users and then select Add Users
- On the Specify user details page, under User details, in User name, enter the name of the new user. This is their sign-in name for AWS, and click on Next
- On the Set Permissions page, specify how you want to assign permissions for this user. By selecting policies
- The user should have Athena, Glue, and S3 access to connect to Athena in Quest DQ

Connect to AWS Athena

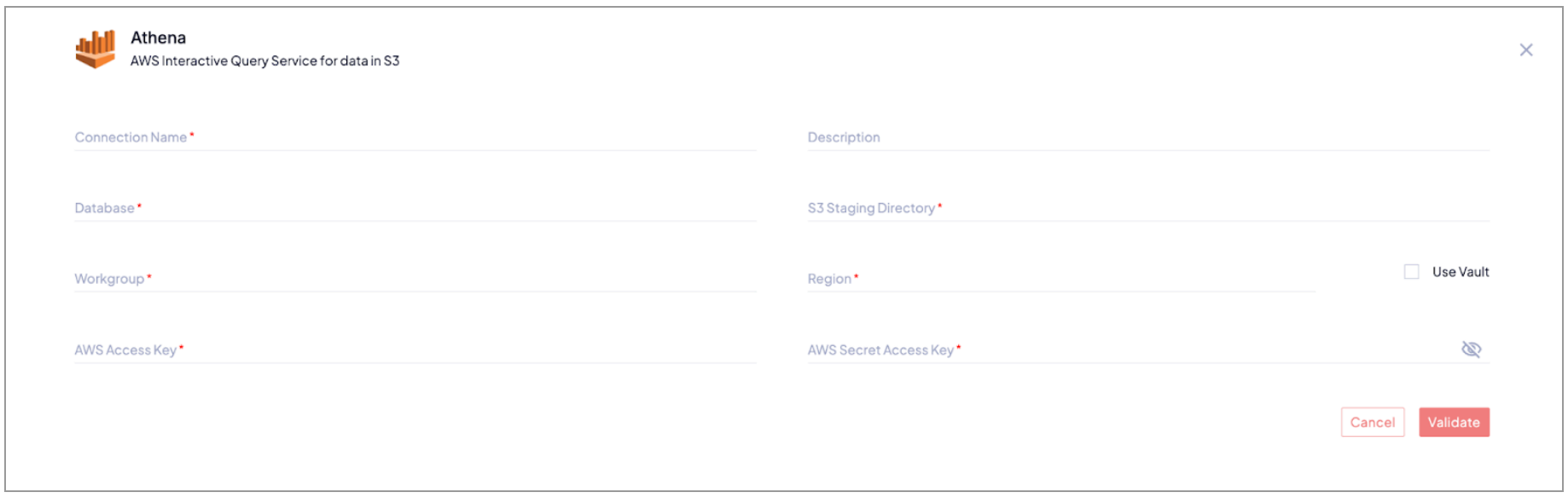
Step 1: Navigate to **Settings** > **Sources**

Step 2: Go to the + icon in the top right-hand corner of the screen



Step 3: Click on Athena and provide the following details

- Connection name (User Preference)
- Description (Can be used to describe the connection and its purpose)
- AWS Access Key
- AWS Secret Key
- Database
- S3 Staging Directory
- Region



Step 4: Validate it

Step 5: Once the connection is established, select the required schemas from the list of all available schemas and connect.

AWS EMR

Amazon EMR (previously called Amazon Elastic MapReduce) is a managed cluster platform that simplifies running big data frameworks, such as Apache Hadoop and Apache Spark, on AWS to process and analyze vast amounts of data.

Quest DQ provides the ability to connect to an EMR cluster and process data for data quality and other use cases in Quest DQ. Quest DQ supports the following EMR configurations:

- Hadoop 3.3.3
- Hive 3.1.3
- JupyterEnterpriseGateway 2.6.0
- Livy 0.7.1
- Presto 0.281
- Spark 3.4.1

Prerequisites

Whitelist IP

If your organization uses a whitelist to manage EMR access, Quest DQ will only access your AWS EMR through IP. For assistance on whitelisting, kindly reach out to the Support team.

Account Setup

User Permissions

Following are the steps to create a service user and provide access to a database in AWS EMR:

- On the Console Home page, select the IAM service.
- In the navigation pane, select Users and then select Add Users.
- On the Specify user details page, under User details, in User name, enter the name of the new user. This is their sign-in name for AWS, and click on Next
- On the Set Permissions page, specify how you want to assign permissions for this user. By selecting policies
- The user should have Athena, Glue, and S3 access to connect to EMR in Quest DQ

Create secret and AccessKey

- Go to the AWS management console, click on your Profile name, and then click on My Security Credentials
- Go to Access Keys and select Create New Access Key
- Click on Show Access Key and save/download the access key and secret access key

Dataset Access:

- The user must have select permission to the schema/external tables within the database to connect to Quest DQ.
- Grant USAGE permission to the new user by running the following command:

```
None
GRANT USAGE ON SCHEMA External_schema TO user;
```

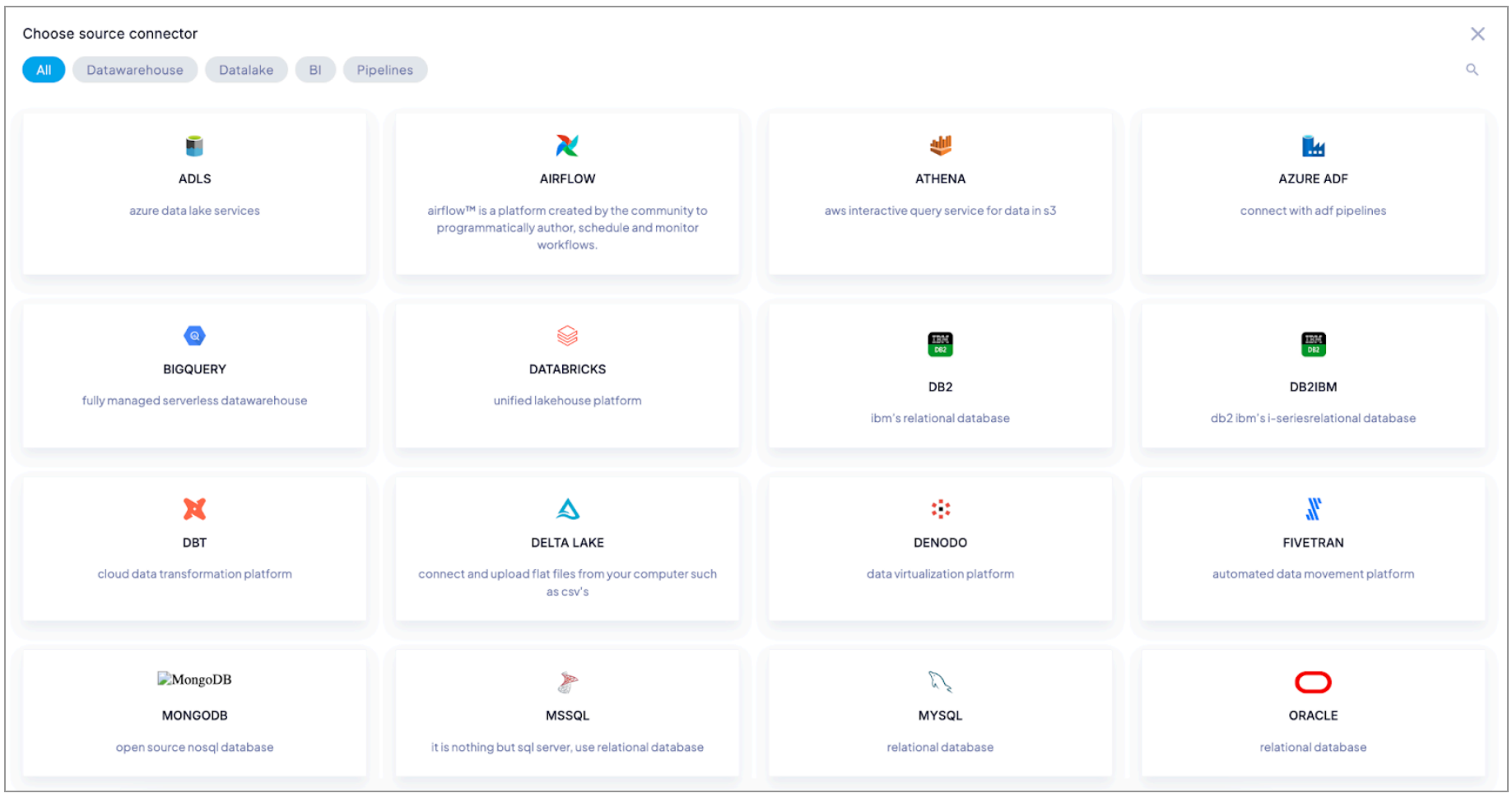
- Grant SELECT permission on the external tables by running the following command:

```
None
GRANT SELECT ON TABLE External_schema.Table TO user;
```

Connect to AWS EMR

Step 1: Navigate to Settings -> Sources

Step 2: Go to the + icon in the top right-hand corner of the screen



Step 3: Click on EMR and provide the following details

- Connection name (User Preference)
- Description (Can be used to describe the connection and its purpose)
- AWS Access Key
- AWS Secret Key
- Database
- Staging Directory
- Cluster-ID
- Region

Step 4: Validate it

Step 5: Once the connection is established select the required schemas from the list of all available schemas and connect.

Azure Synapse

Azure Synapse Analytics is a cloud-based analytics service that provides big data storage, processing, and analytics capabilities. It enables organizations to gain insights from their data by providing a serverless environment for data warehousing and big data analytics. Azure Synapse Analytics also includes built-in machine learning and artificial intelligence (AI) capabilities.

Prerequisites

The following prerequisites must be met to establish the connection between Synapse and Quest DQ

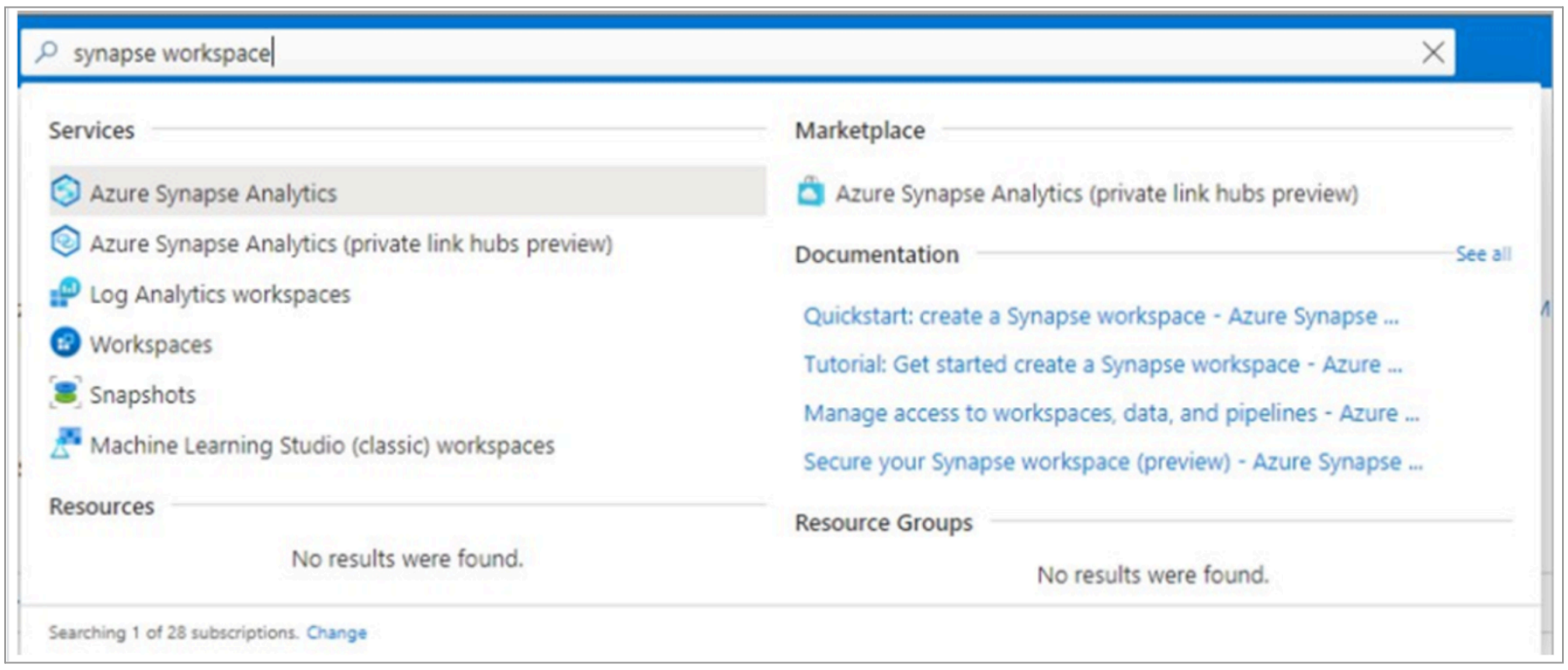
Whitelist IP

If your organization uses a whitelist to manage Synapse access, Quest DQ will only access your Synapse through a certain IP. Reach out to the Support team.

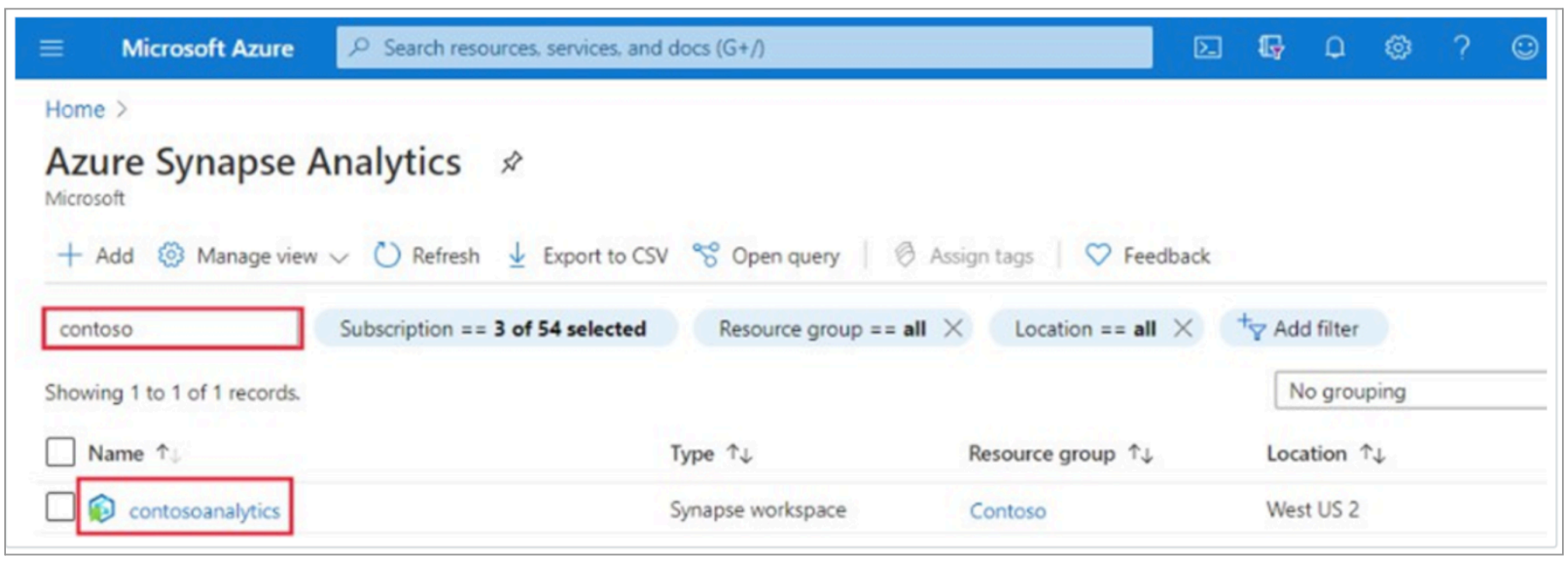
Account Setup

Create a dedicated SQL pool using Synapse Studio

Step 1: Log in to the Synapse Workspace and navigate to the Synapse Workspace

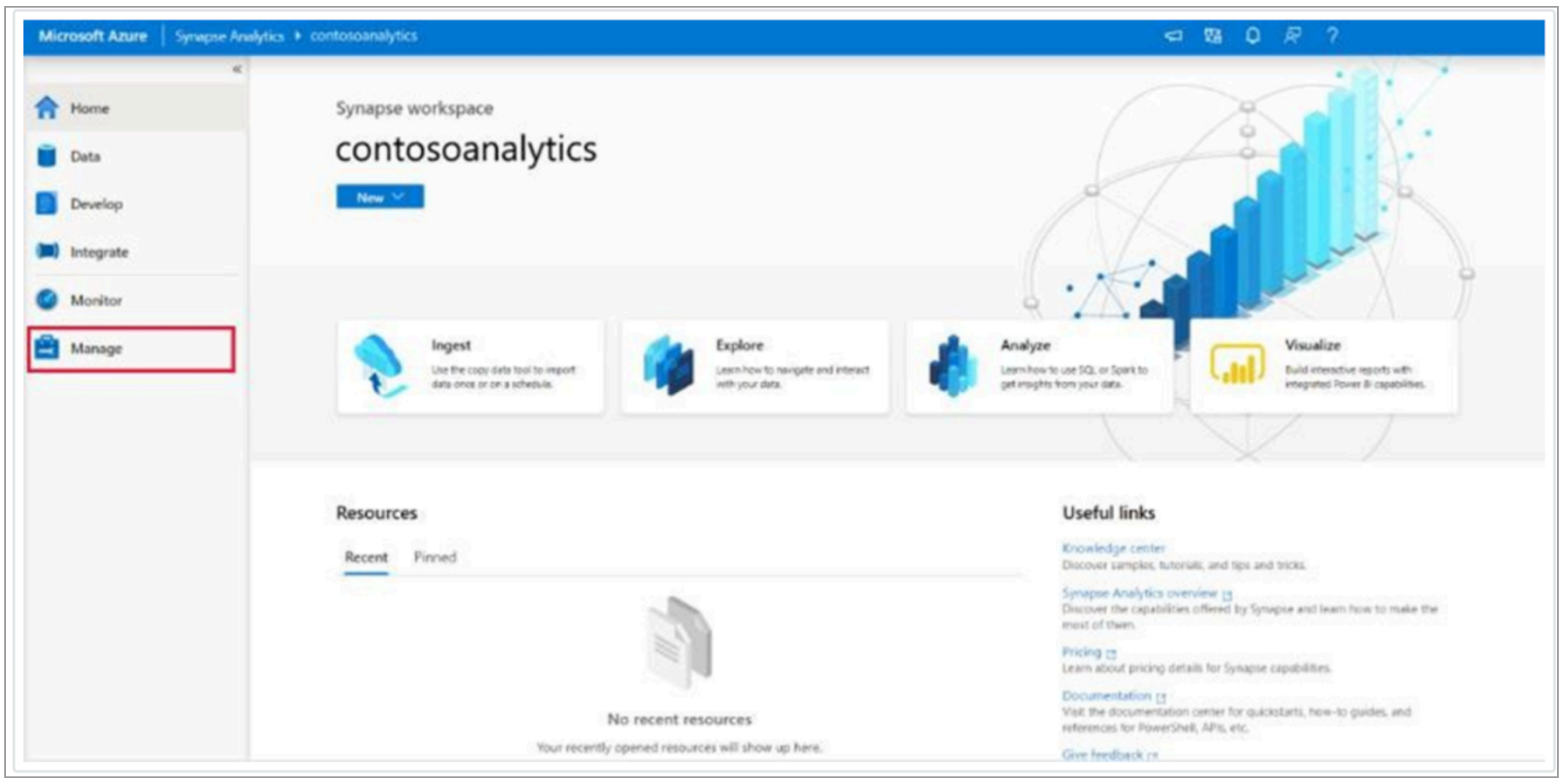


Step 2: From the list of workspaces, select ContosoAnalytics

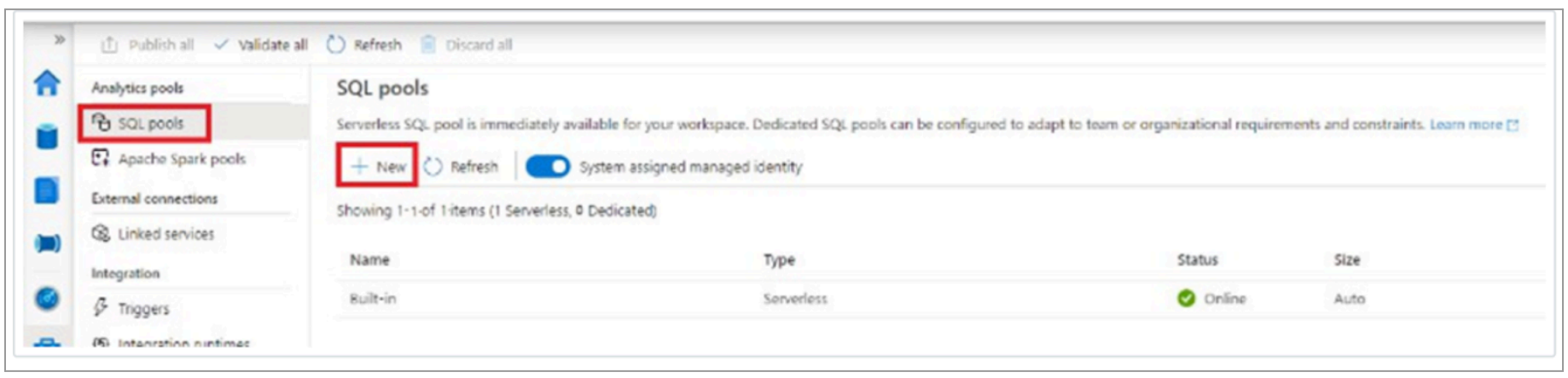


Step 3: Select the Workspace web URL to launch SynapseStudio from the workspace overview

Step 4: Select Manage in the Management hub left navigation panel

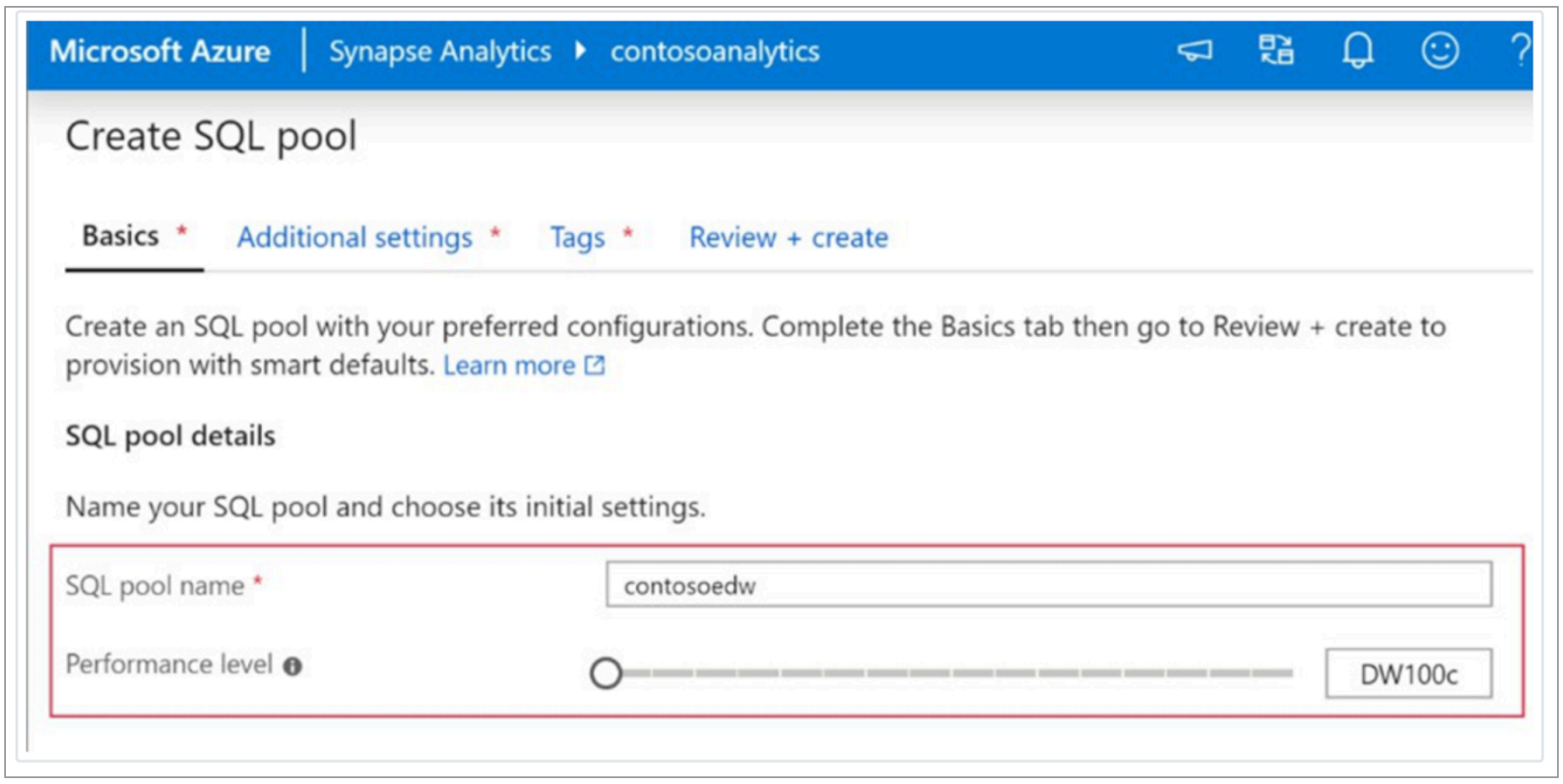


Step 5: Click on the "+ New" button to create a SQL pool

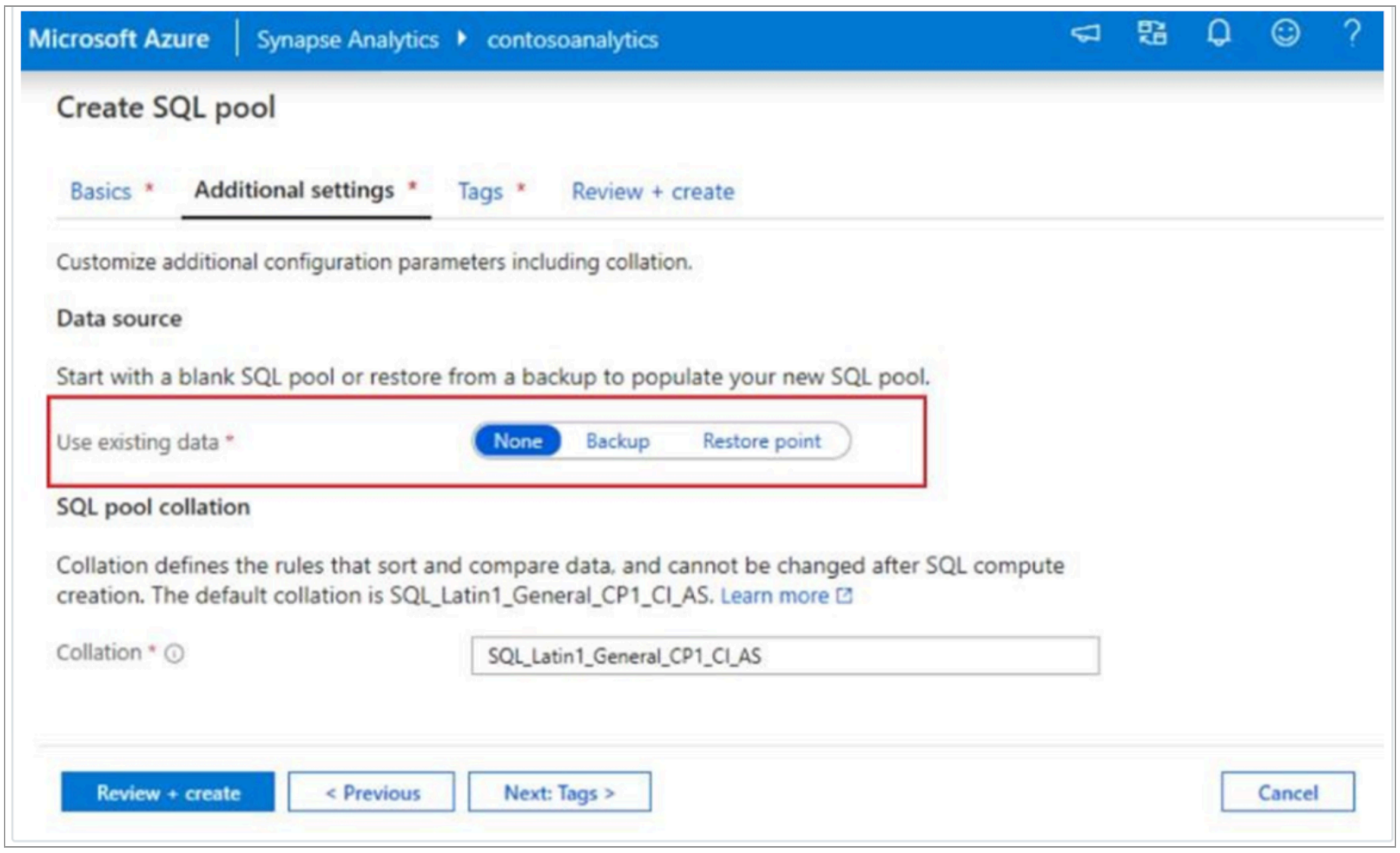


Step 6: Enter the following details on the basic tab

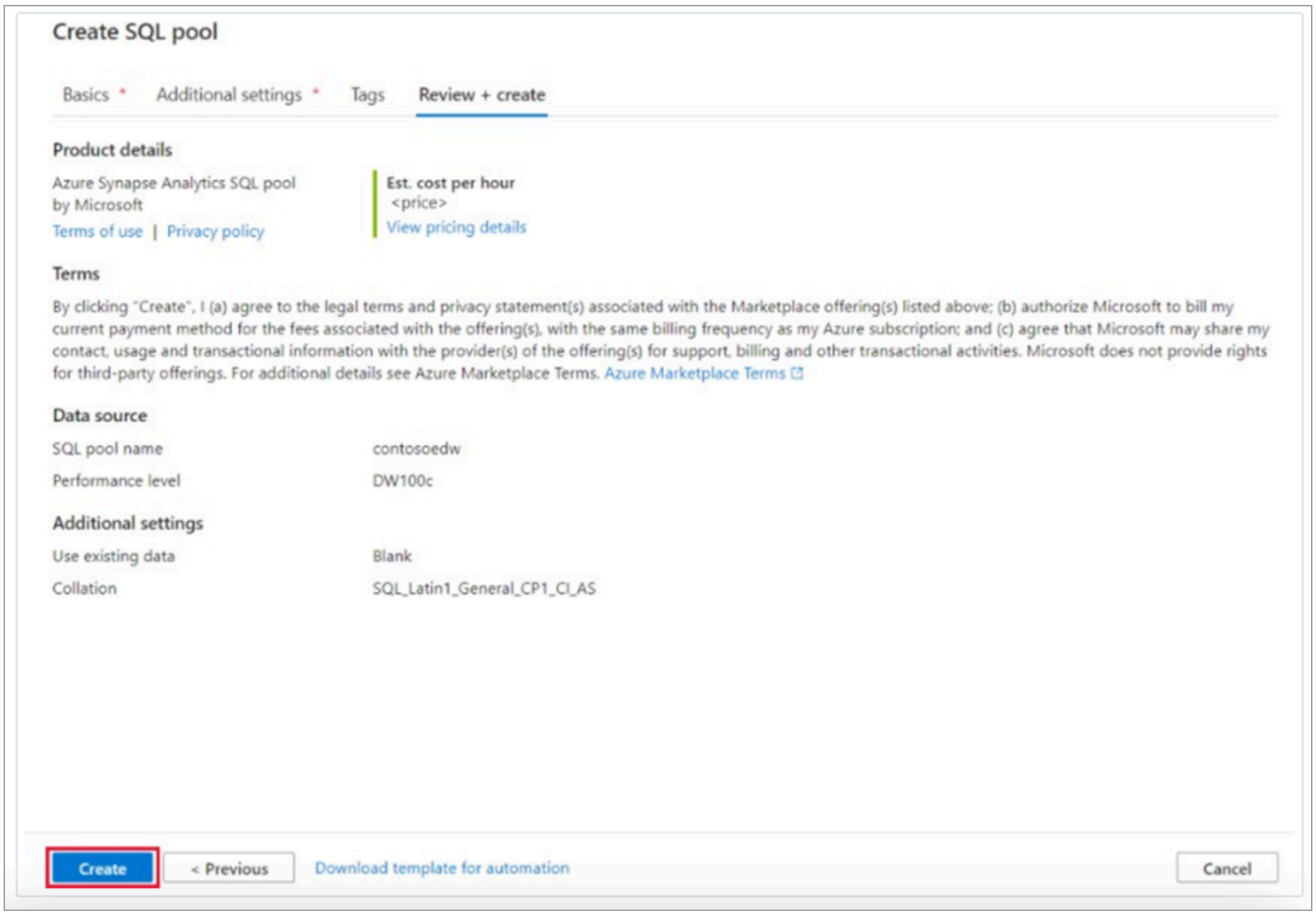
- SQL Pool Name
- Performance Level



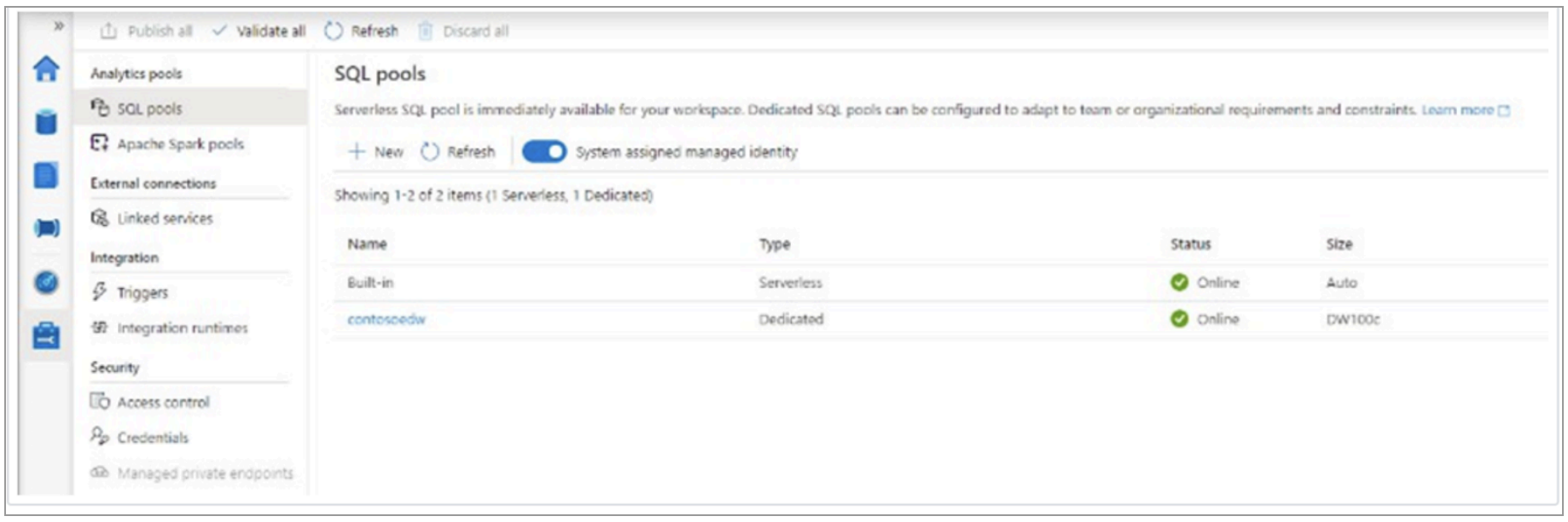
Step 7: Navigate to the "Additional Settings" tab and click on Review + Create



Step 8: On the Review + Create tab, verify the information, and click on Create



Step 9: Once the SQL pool is created, it will be available in the workspace



Create a Service User

To create a service user and provide access to a database in Synapse, you can follow these steps:

- Log in using the administrator account and connect to the master database
- Run the following SQL command to create a user

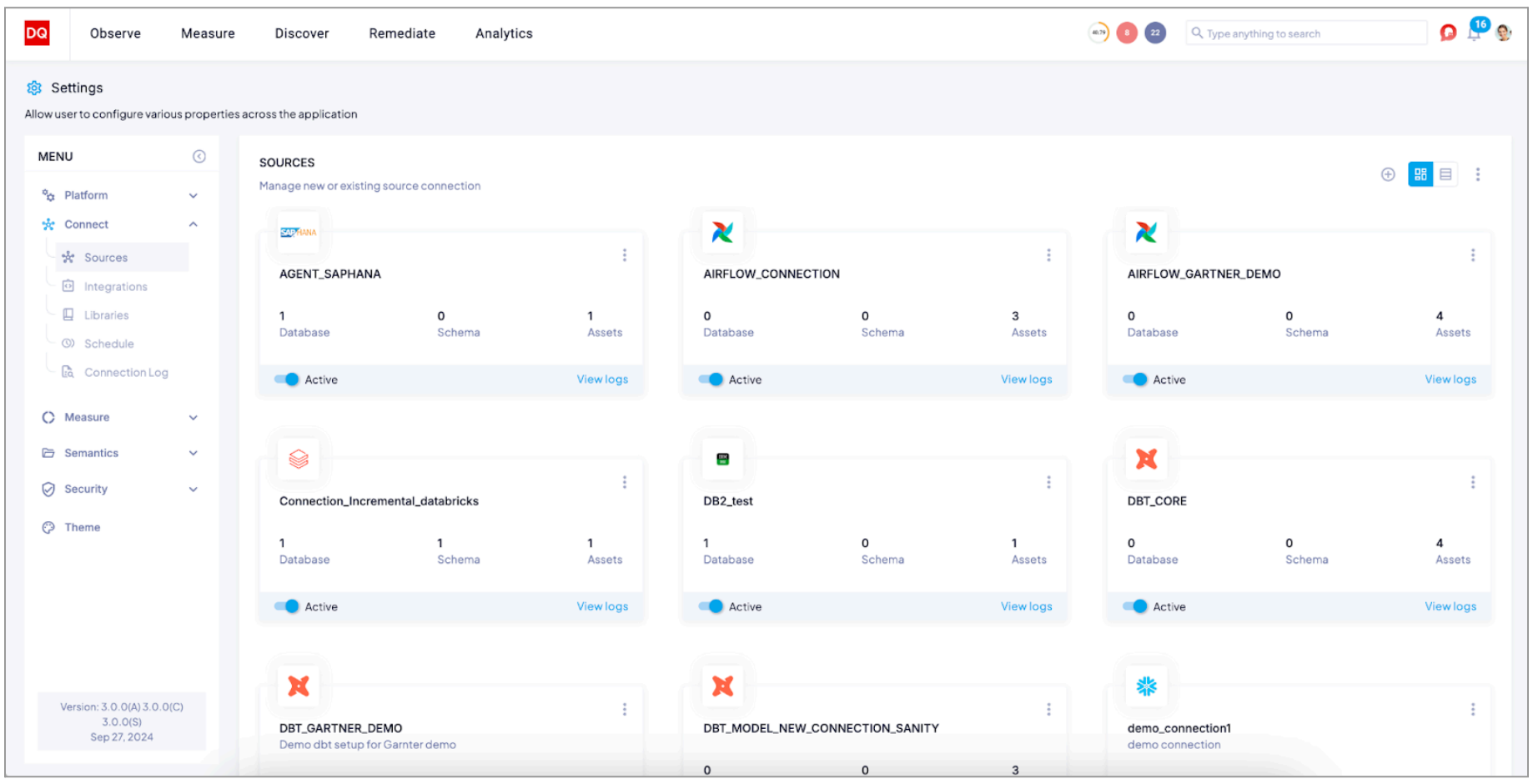
```
None
CREATE LOGIN User WITH PASSWORD = '<strong_password>';
```

- Add the new user to the db manager database role in master using the sp_addrolemember procedure

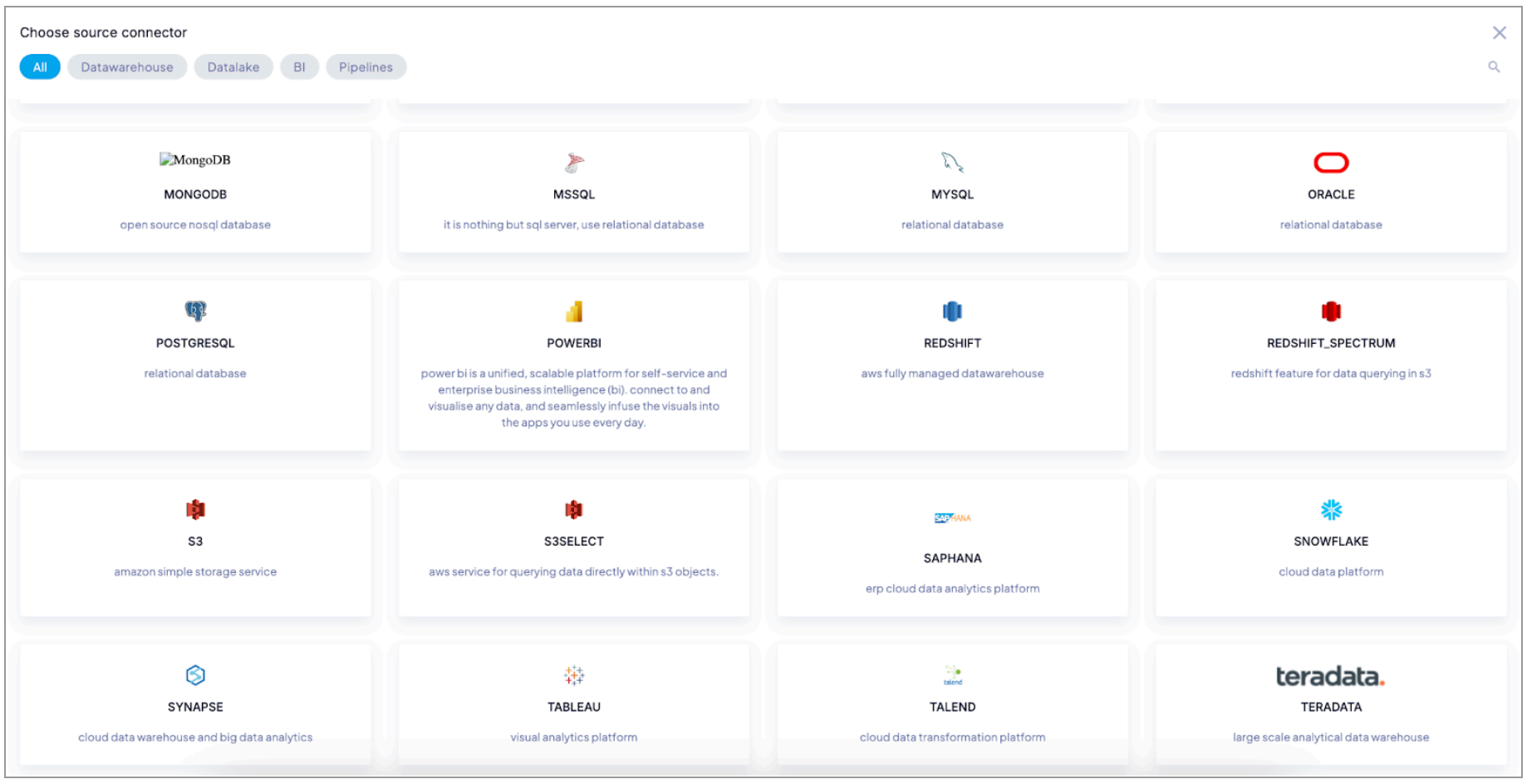
```
None
EXEC sp_addrolemember 'dbmanager', 'Mary';
EXEC sp_addrolemember 'dbmanager', 'mike@contoso.com]';
```

Connect to Synapse

Step 1: Navigate to Settings -> Sources

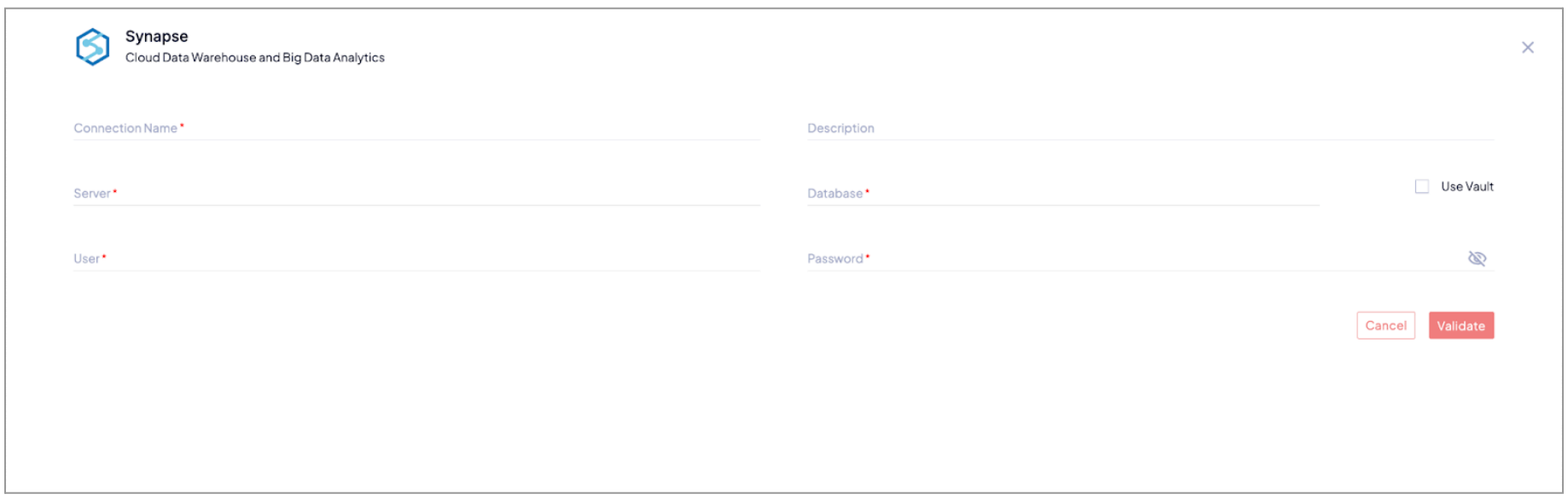


Step 2: Go to + icon in the top right-hand corner of the screen



Step 3: Click on Synapse and provide the following details

- Connection name (User Preference)
- Description (Can be used to describe the connection and its purpose)
- Server
- Database
- Username
- Password



Step 4: Validate it

Step 5: Once the connection is established, select the required schemas from the list of all available schemas and connect.

Databricks

Databricks is a unified analytics platform designed for large-scale data processing and machine learning applications. It was founded in 2013 by the original creators of Apache Spark, a widely used open-source big data processing engine.

Databricks provides a cloud-based platform that allows users to easily process large volumes of data, build machine learning models, and deploy them at scale. The platform offers a range of tools and services for data processing, data science, and machine learning, including collaborative notebooks, automated workflows, and a library of pre-built machine learning models.

Prerequisites

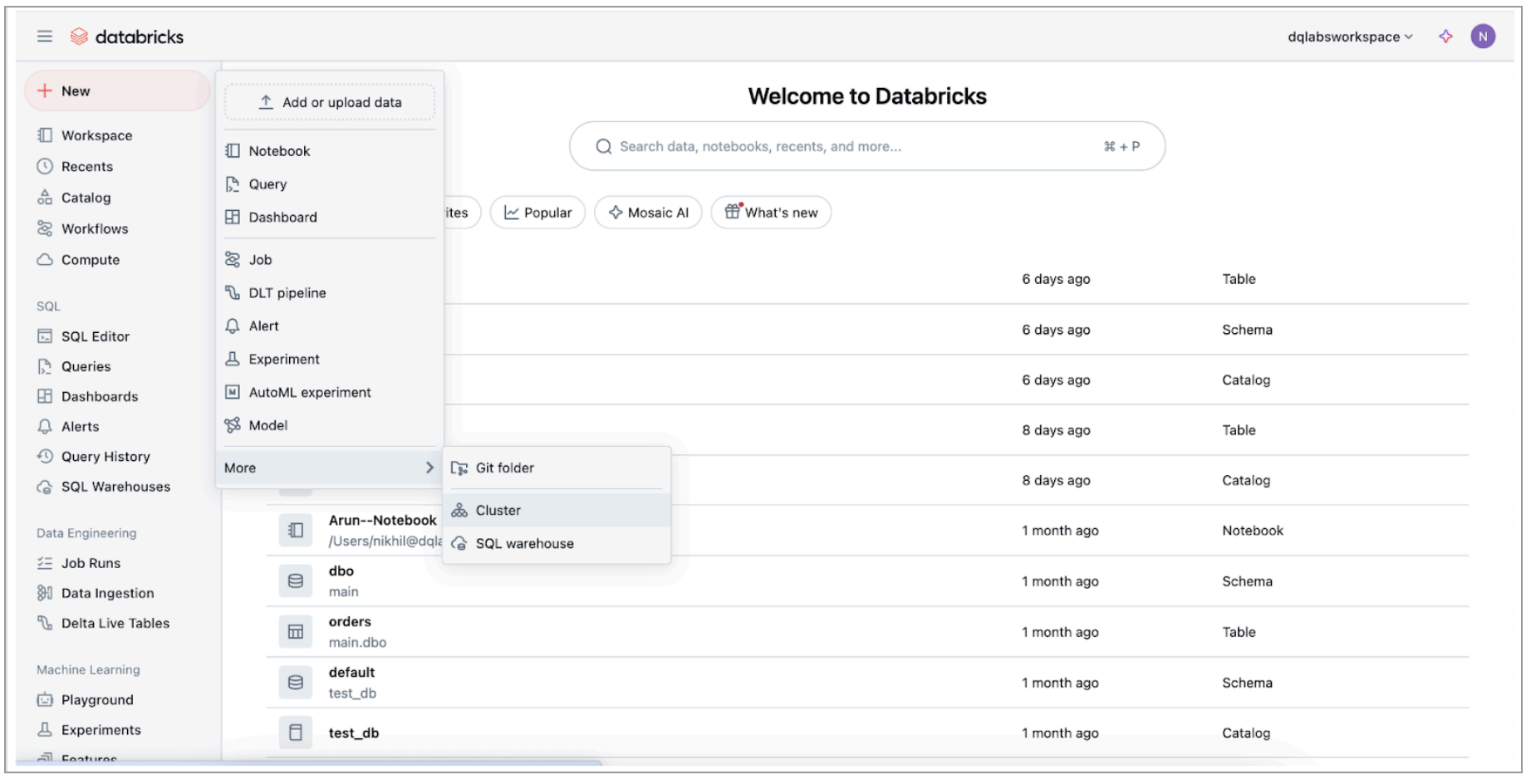
Whitelist IP

If your organization uses a whitelist to manage Databricks access, Quest DQ will only access your Databricks through IP. For assistance on whitelisting, kindly reach out to the Support team.

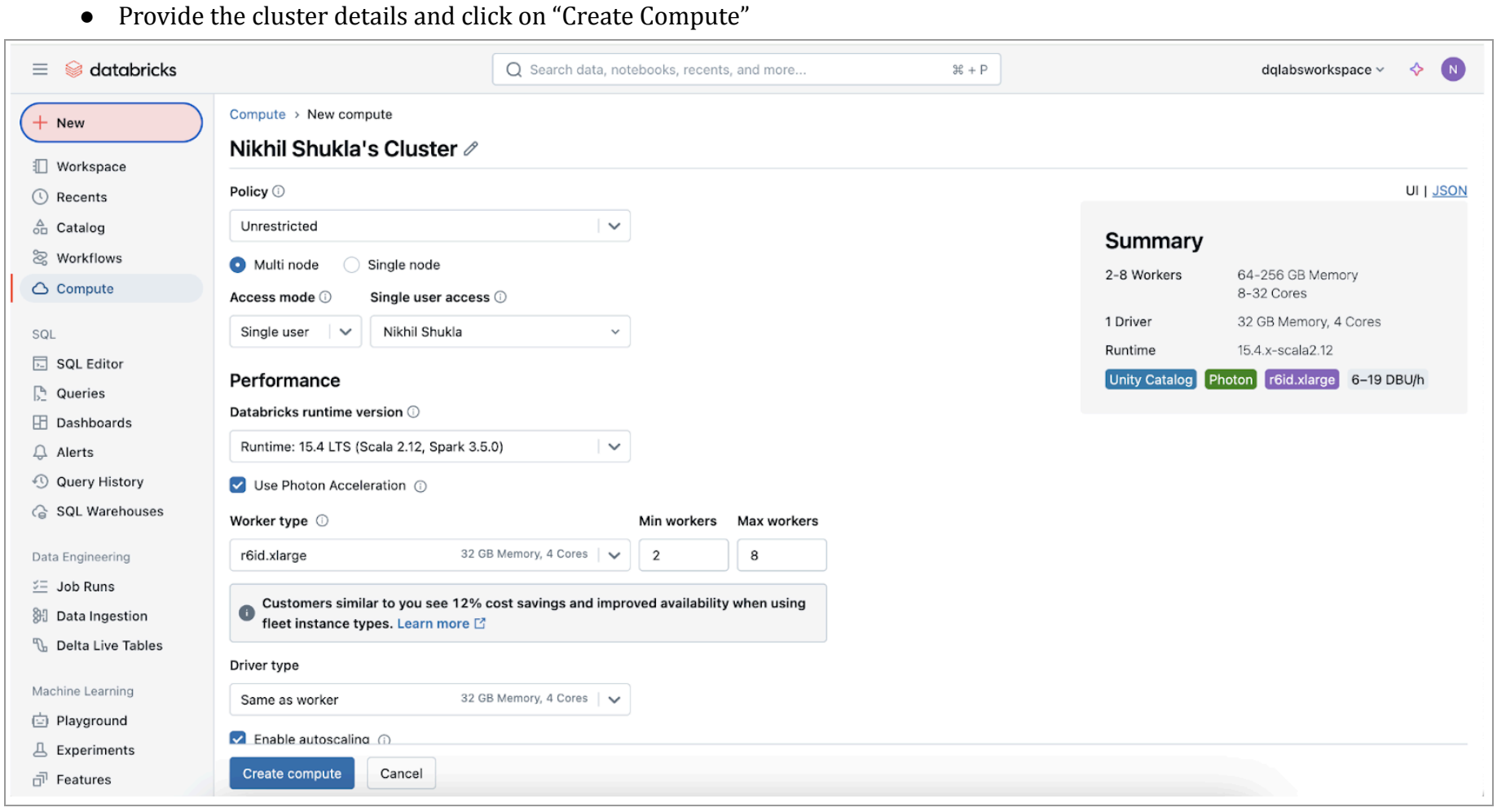
Account Setup

Follow the steps below to set up an account in Databricks:

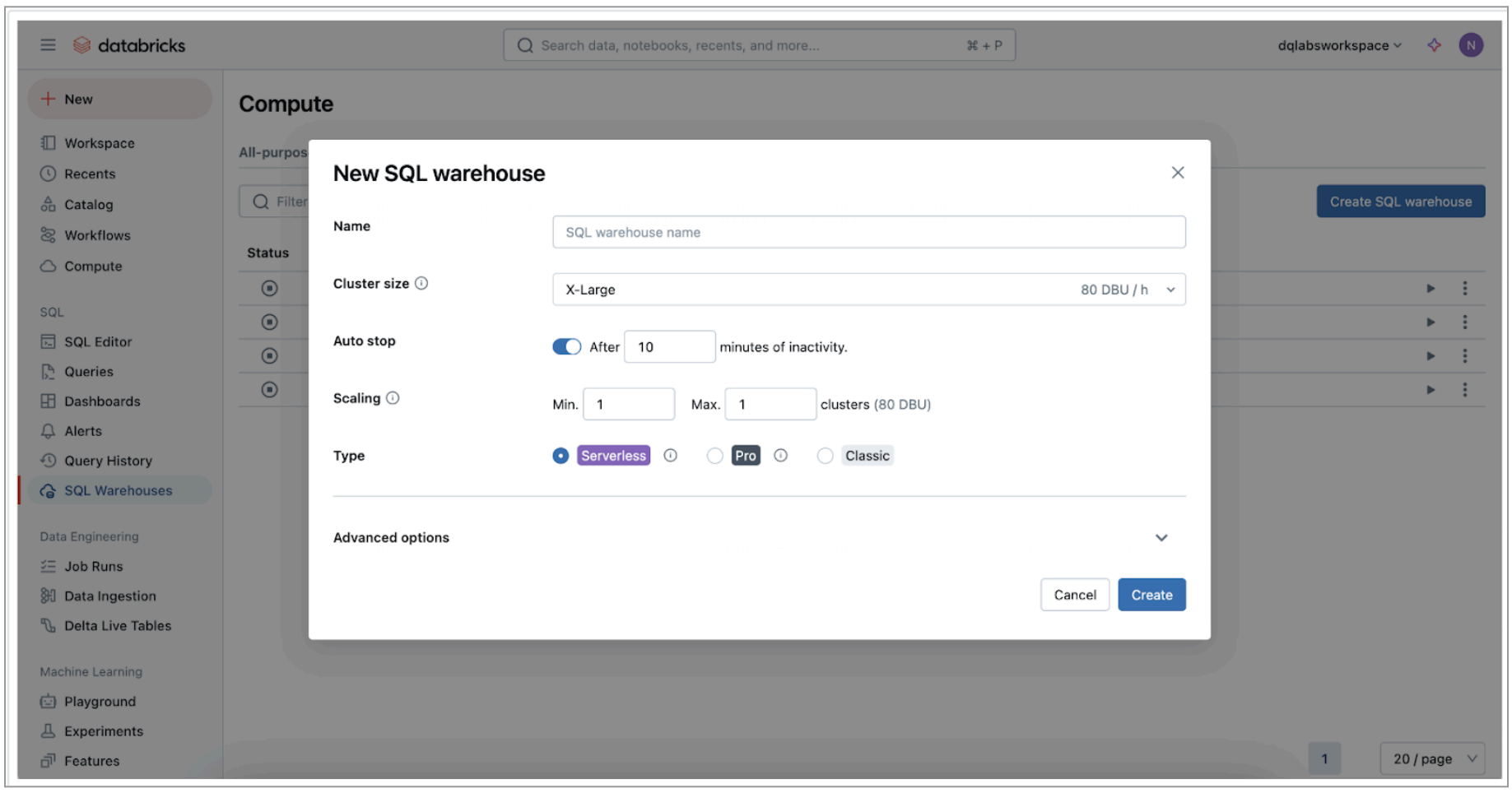
- Log in to your Databricks account.
- Click on “New” and choose “Cluster”



- Provide the cluster details and click on “Create Compute”



- Create a warehouse in SQL Warehouse by clicking on New->Create SQL Warehouse



- Click on Create and Navigate to Catalog-> Meta store and Grant Unity catalog access

Access Required for System Tables/Metadata Tables

- Grant the following access if exceptional reporting is part of the use case:

```
None
GRANT ALL PRIVILEGES ON TABLE <table_name>  To <user>
```

- Grant the following access if exceptional reporting is not part of the use case:

```
None
GRANT SELECT ON ALL TABLES IN SCHEMA <schema_name> TO <user>
```

To create an access token in Databricks, follow these steps:

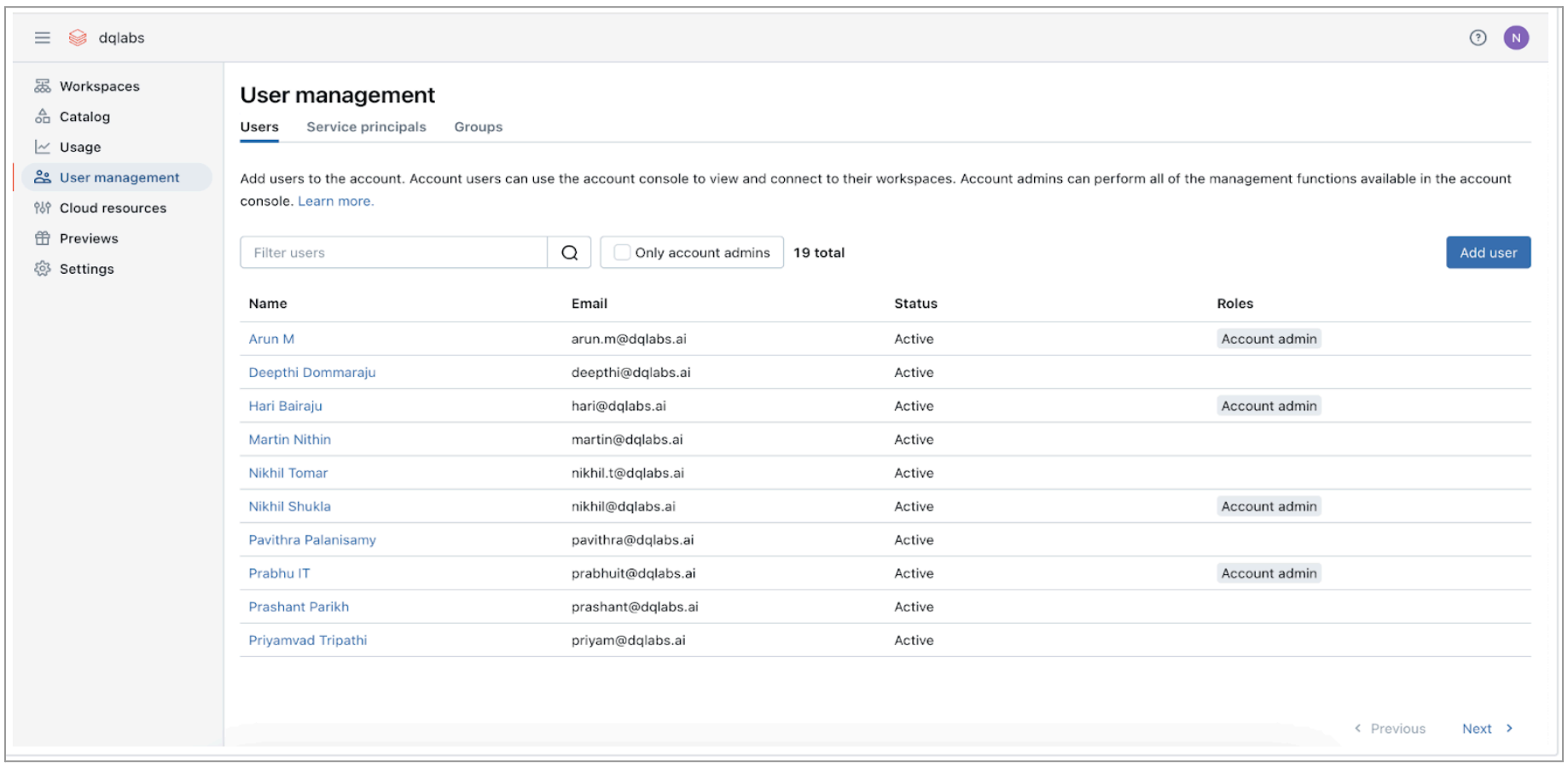
- Log in to your Databricks account and navigate to the User Settings page
- Click on the "Access Tokens" tab
- Click on the "Generate New Token" button
- Enter a name for the token and select the appropriate scopes for the token
- Click on the "Generate" button
- Copy the token value and securely store it.

The available scopes for access tokens in Databricks include"all cluster", "all org", "allow cluster creation", "allow instance pool creation", and "allow library management".

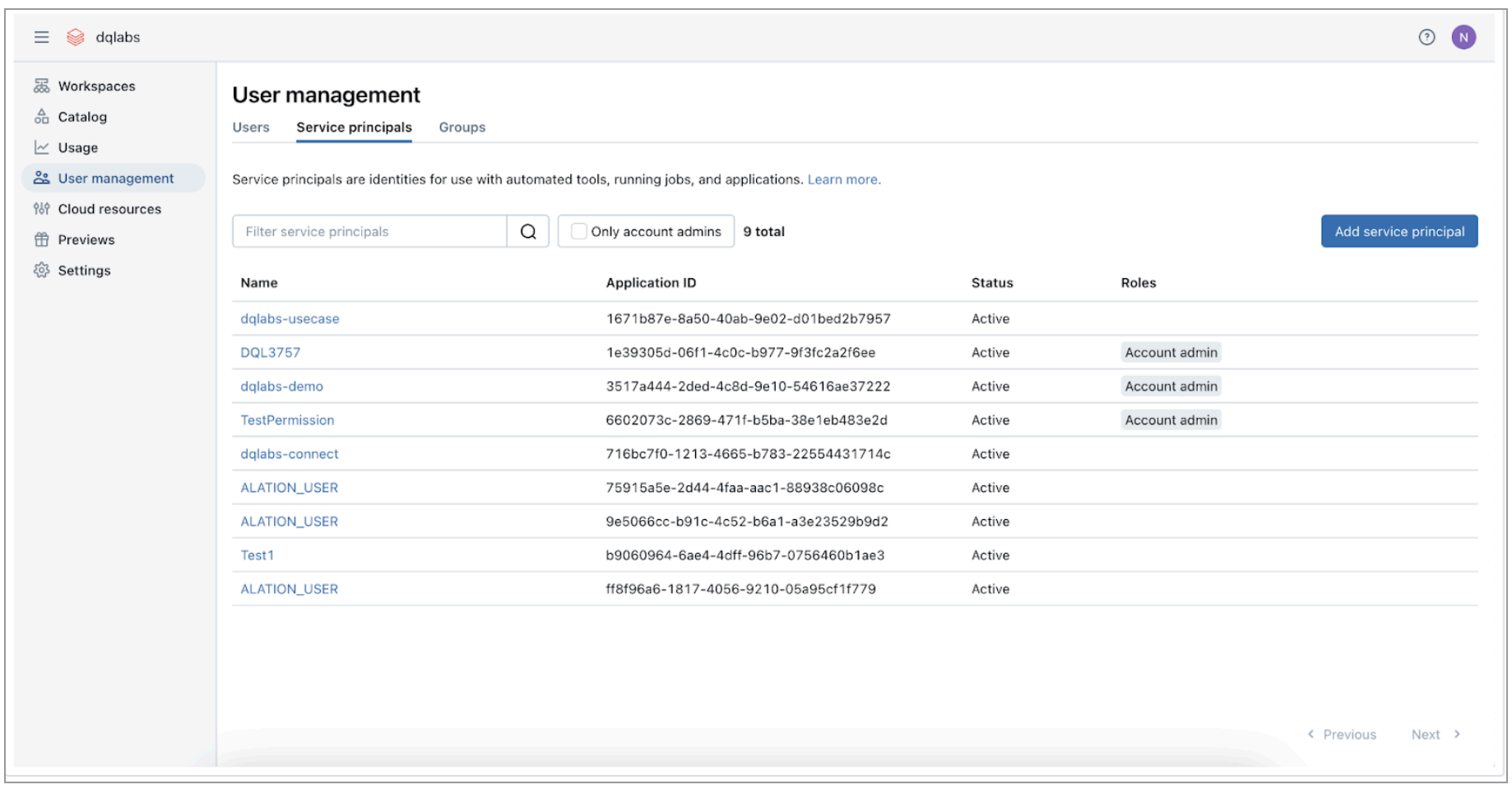
Create a service principal (For OAuth method of Authentication)

Step 1: Log in to the Databricks account console

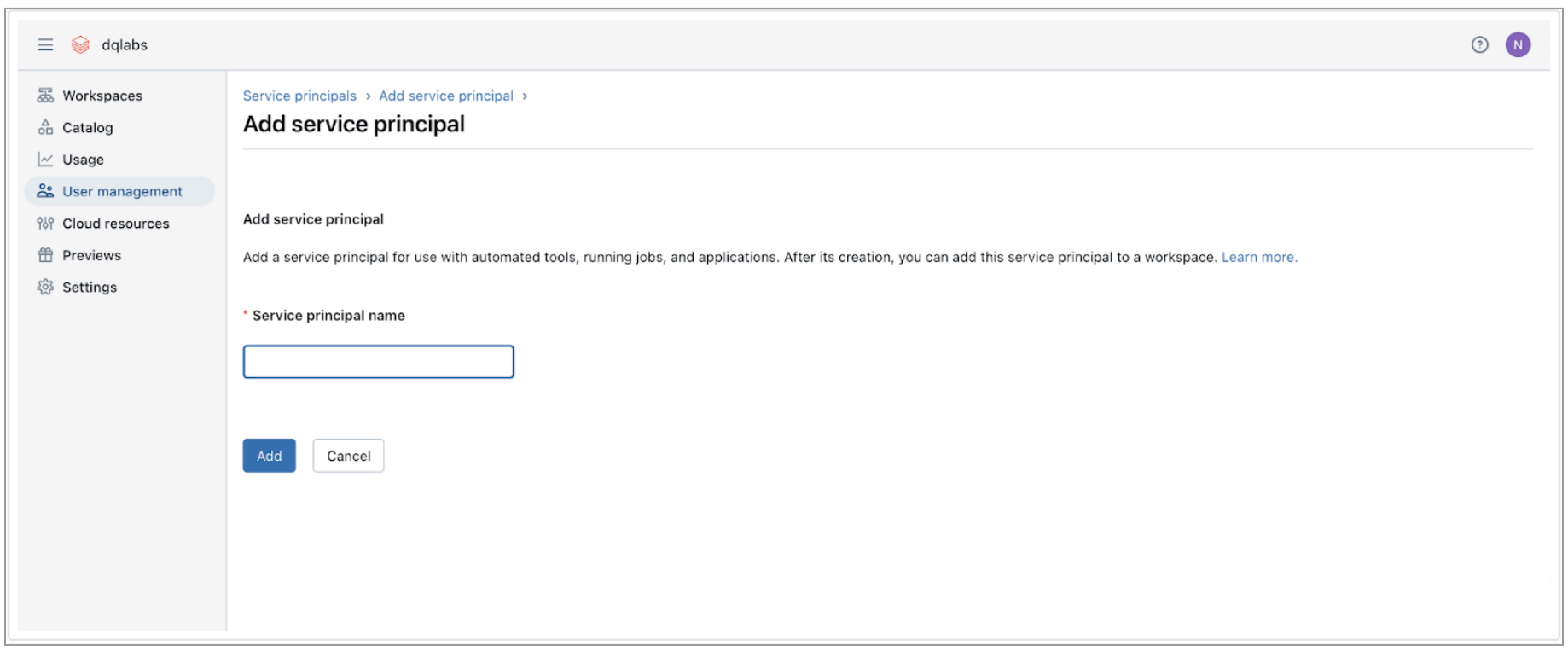
Step 2: Click on “User Management”



Step 3: Click on “Service Principal”

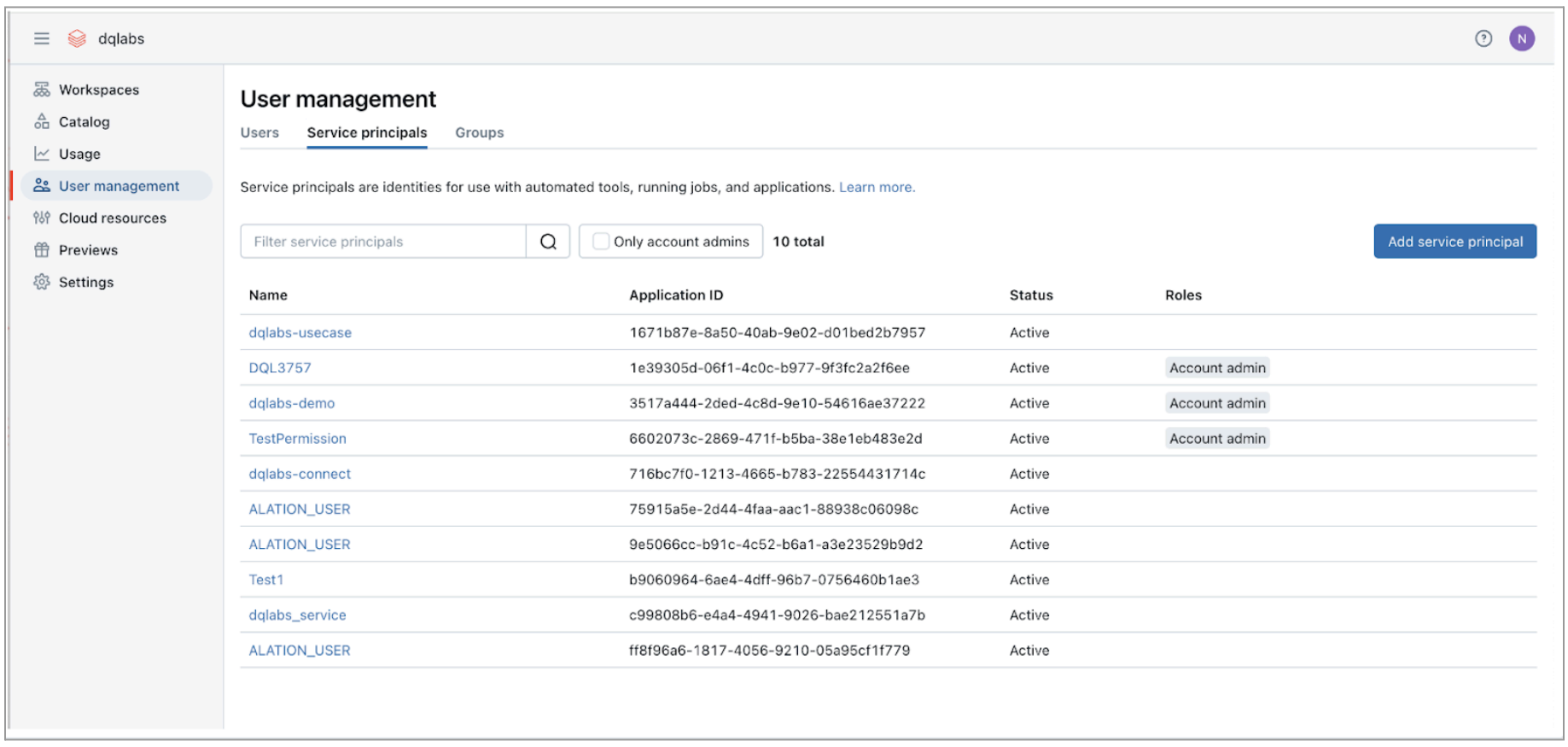


Step 4: Click on “Add Service Principal“ to create a new service principal

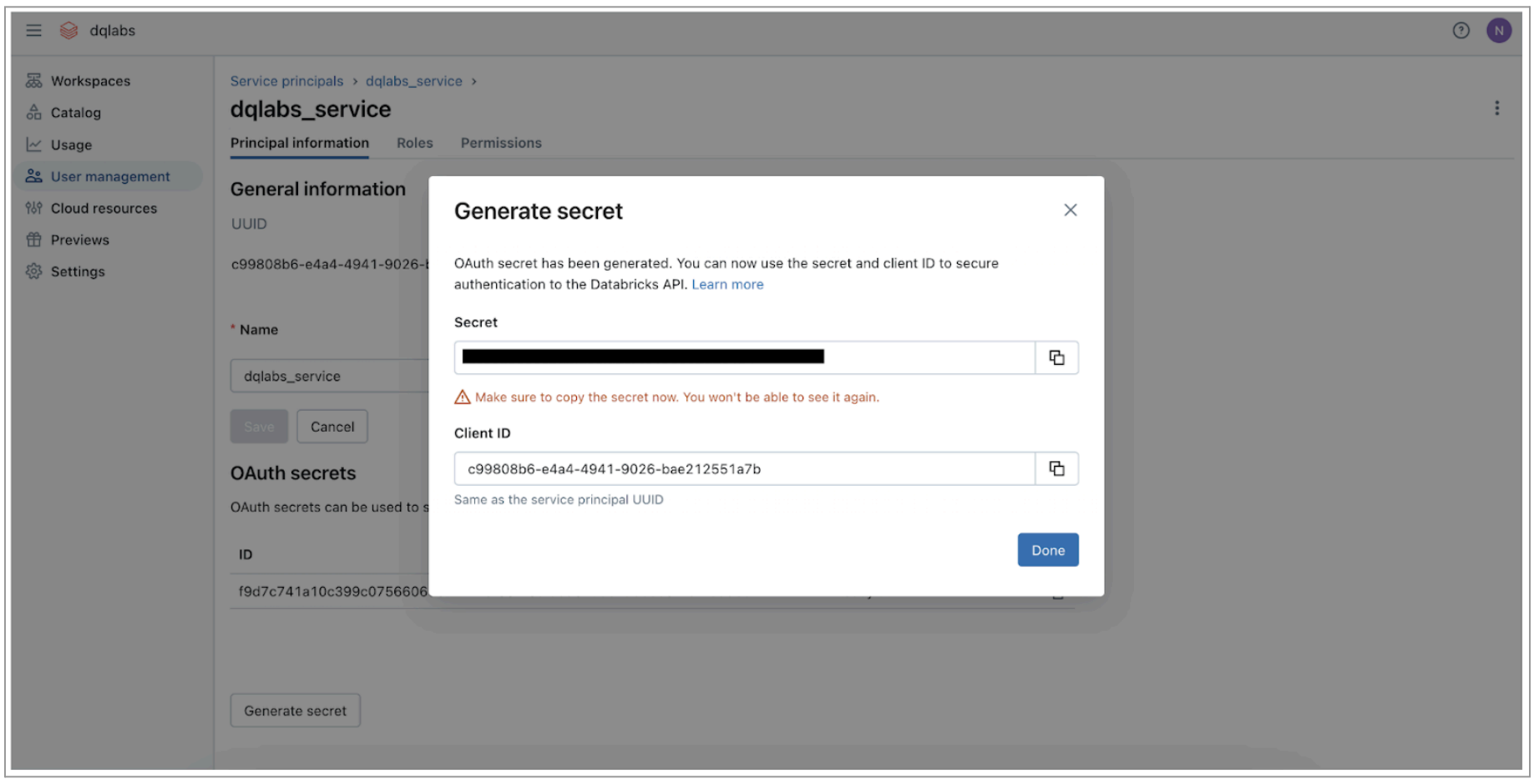


Step 5: Provide the service principal name and click on “Add”

Step 6: Once created, the service principal should be listed on the list page, click on the created service principal



Step 7: In the service principals view page, click on the “Generate Secret” button



Step 8: Copy the details to use them to authenticate to Quest DQ

Refer to the following documentation for more details: OAuth machine-to-machine (M2M) authentication

Quest DQ uses the Unity catalog to access the system tables for querying the underlying meta store of the database. Unity Catalog is available for Premium Tier Workspace only in Azure. Steps to enable the Unity Catalog for Azure Databricks are mentioned in the link below: https://learn.microsoft.com/en-us/azure/databricks/data-governance/unity-catalog/get-started

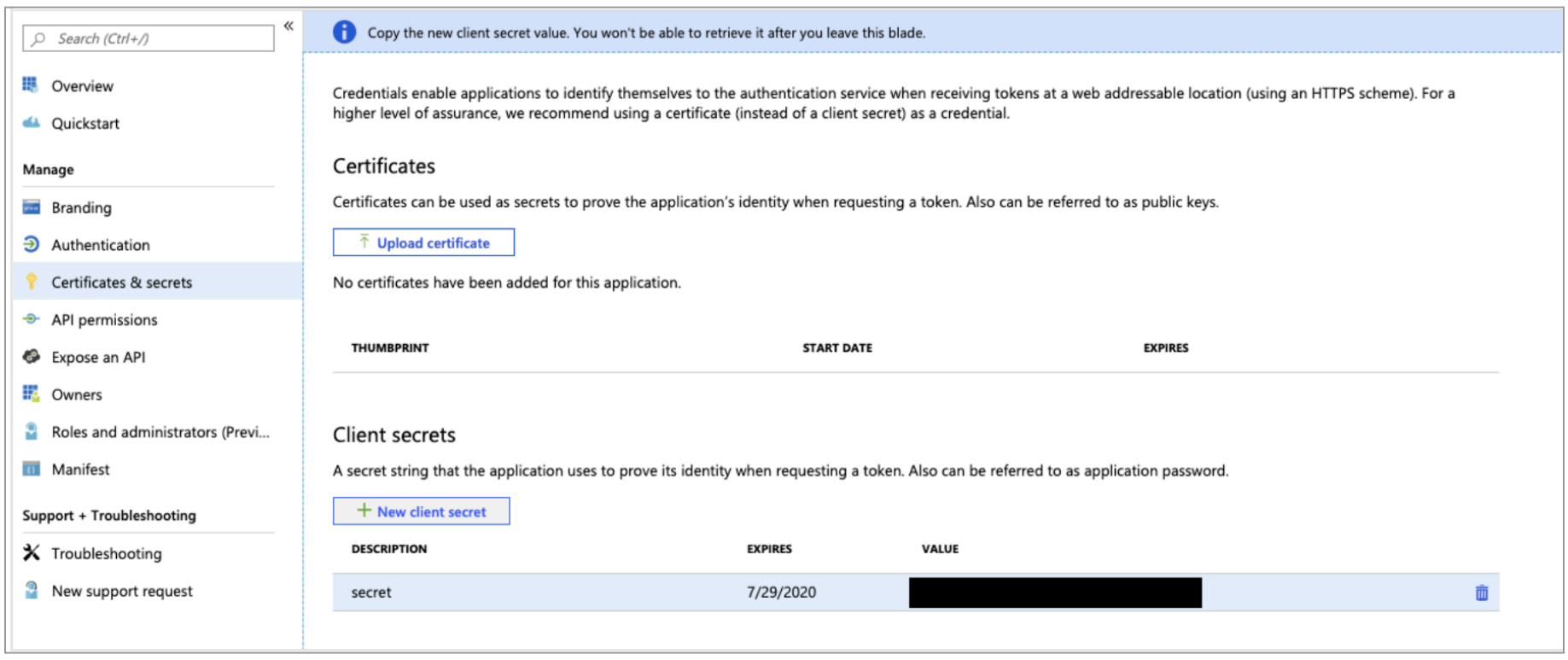
Get Entra ID tokens for service principal

- Sign in to the Azure Portal
- If you have access to multiple tenants, subscriptions, or directories, click the **Directories + subscriptions** (directory with filter) icon in the top menu to switch to the directory in which you want to provision the service principal.
- In **Search resources, services, and docs**, search for and select **Microsoft Entra ID**.
- Click **+ Add** and select **App registration**.
- For **Name**, enter a name for the application.
- In the **Supported account types** section, select **Accounts in this organizational directory only (Single tenant)**.

- Click **Register**.

- On the application page's **Overview** page, in the **Essentials** section, copy the following values:
  - **Application (client) ID**
  - **Directory (tenant) ID**

- To generate a client secret, within **Manage**, click **Certificates & secrets**.
- On the **Client secrets** tab, click **New client secret**.

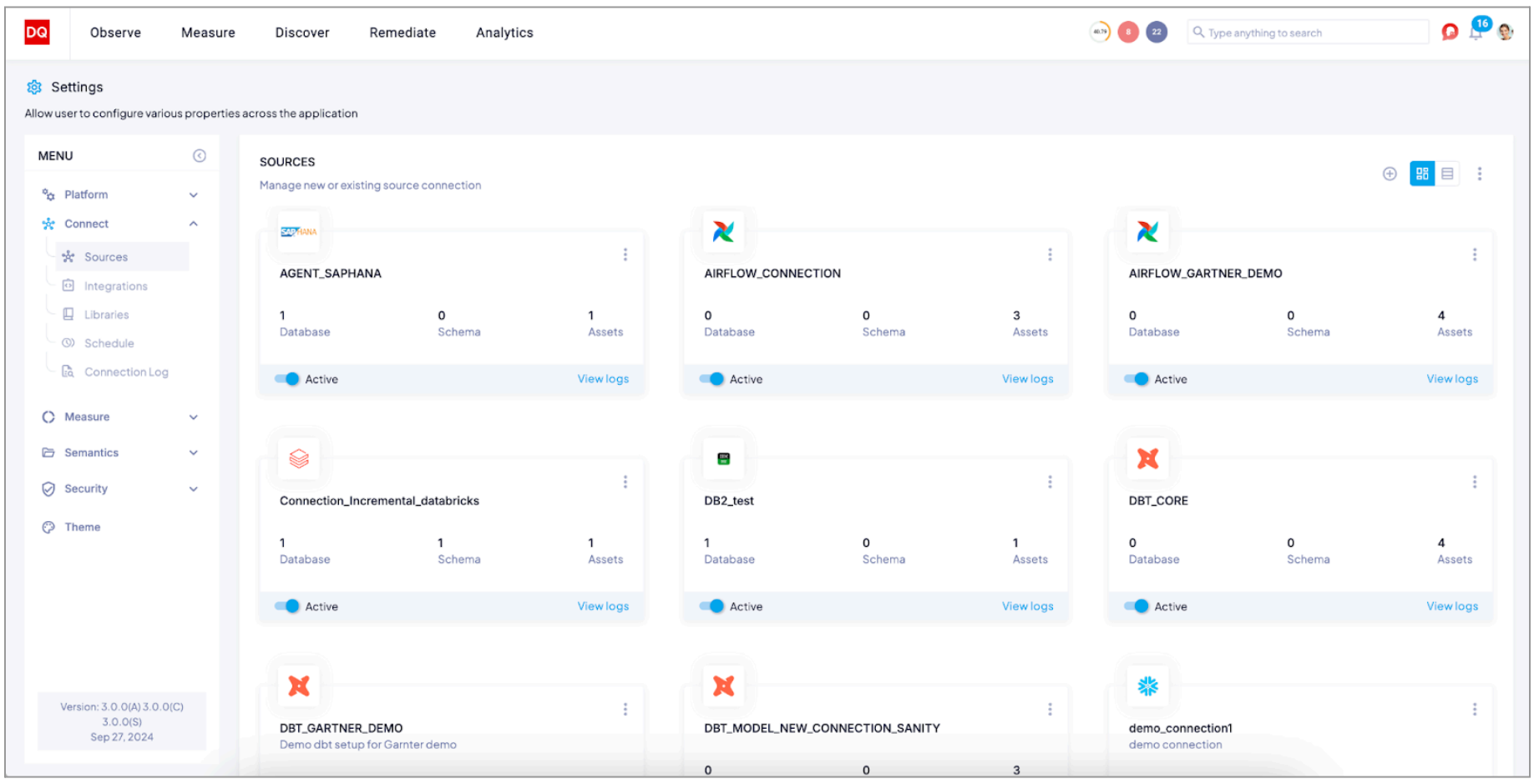


- In the **Add a client secret** pane, for **Description**, enter a description for the client secret.
- For **Expires**, select an expiry time period for the client secret, and then click **Add**.
- Copy and store the client secret's **Value** in a secure place, as this client secret is the password for your application.

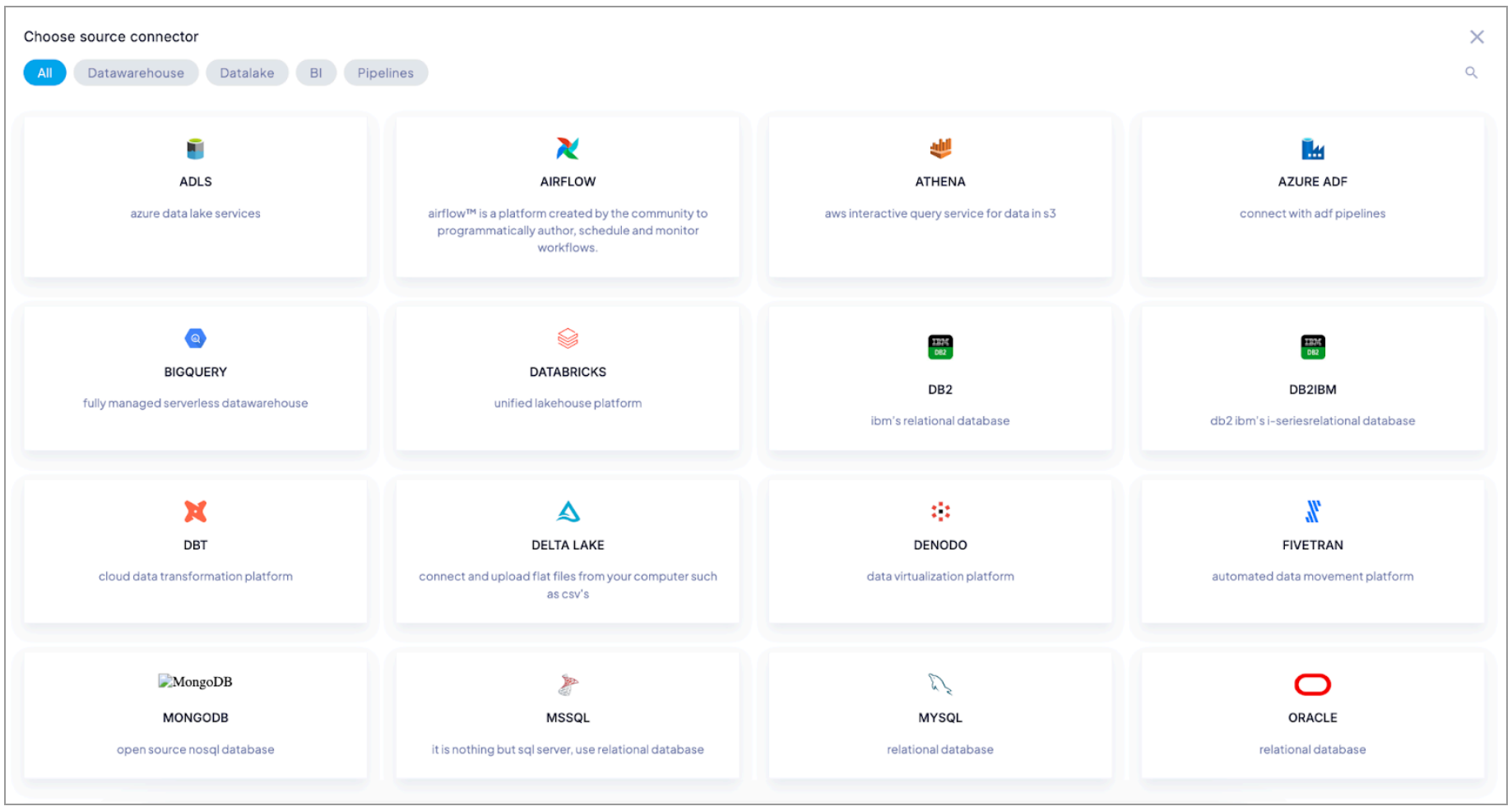
Refer to the following documentation for more details:https://learn.microsoft.com/en-us/azure/databricks/dev-tools/service-prin-aad-token?source=recommendations

Connect to Databricks

Step 1: Navigate to **Settings** > **Sources**

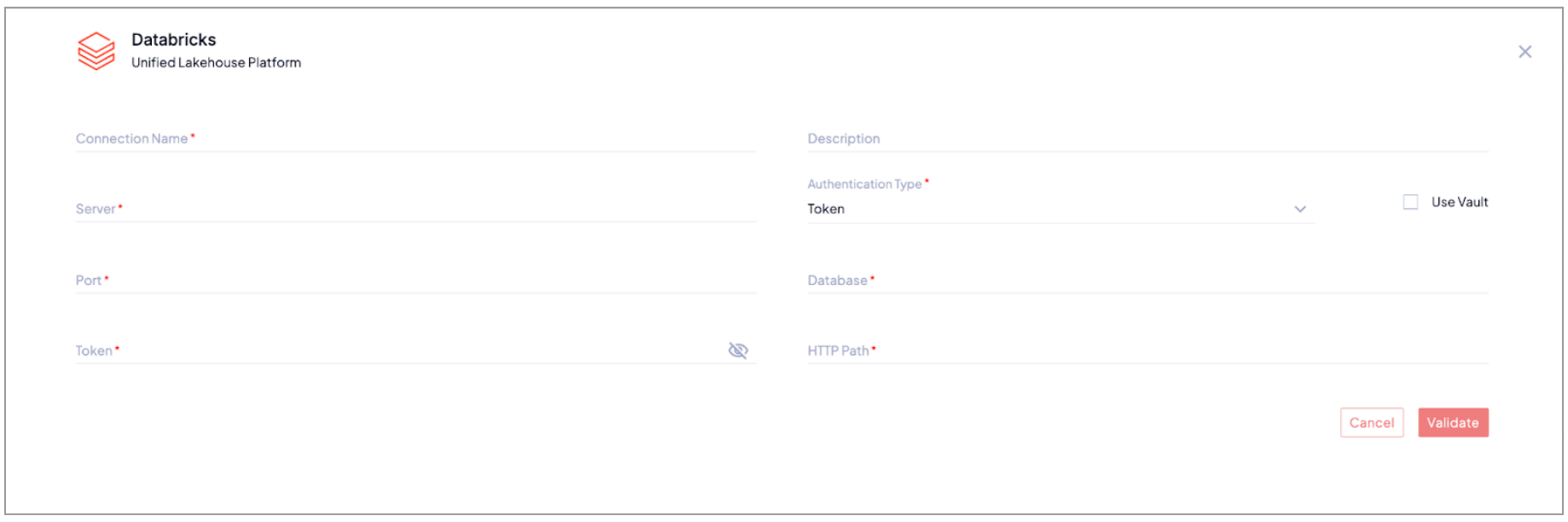


Step 2: Go to the + icon in the top right-hand corner of the screen



Step 3: Click on Databricks and provide the following details

- Connection name (User Preference)
- Description (Can be used to describe the connection and its purpose)
- Server
- Port
- Authentication Type:
    - OAuth(m2m)
    - Token
    - OAuth(Microsoft Entra ID)
- ClientID
- TenantID
- Client Secret
- Database
- Token
- HTTP path



Step 4: Validate it

Step 5: Once validated, click "Connect" to choose the desired tables and Queries

AlloyDB

AlloyDB is a fully managed, PostgreSQL-compatible database service offered by **Google Cloud**. It is designed to provide high performance, scalability, and reliability for enterprise workloads while maintaining full compatibility with PostgreSQL. Quest DQ allows users to connect to Alloy DB and profile data.

Prerequisites

Whitelisting

If your organization uses a whitelist to manage Snowflake access, Quest DQ will only access your Snowflake through IP. For assistance on whitelisting, kindly reach out to the support team.

User Access

Follow the steps below to create a user and assign permissions:

To create a service user and provide access to a database in Postgres, you can follow these steps:

- Connect to your Postgres database using an account with administrative privileges.
- Create the service user using the CREATE USER statement, specifying the desired username and password:

```
Python
CREATE USER myserviceuser WITH PASSWORD 'mypassword';
```

- Create the database that the service user will access, if it does not already exist:

```
None
CREATE DATABASE mydatabase;
```

- Grant the service user permission to access the database using the GRANT statement:

```
None
GRANT ALL PRIVILEGES ON DATABASE mydatabase TO myserviceuser;
```

This statement grants the service user all privileges on the specified database, allowing it to create tables, insert data, and perform other operations.

- Optionally, you can restrict the service user's access to specific schemas within the database by specifying the schema name in the GRANT statement:

```
None
GRANT ALL PRIVILEGES ON SCHEMA myschema TO myserviceuser;
```

This statement grants the service user all privileges on the specified schema within the database.

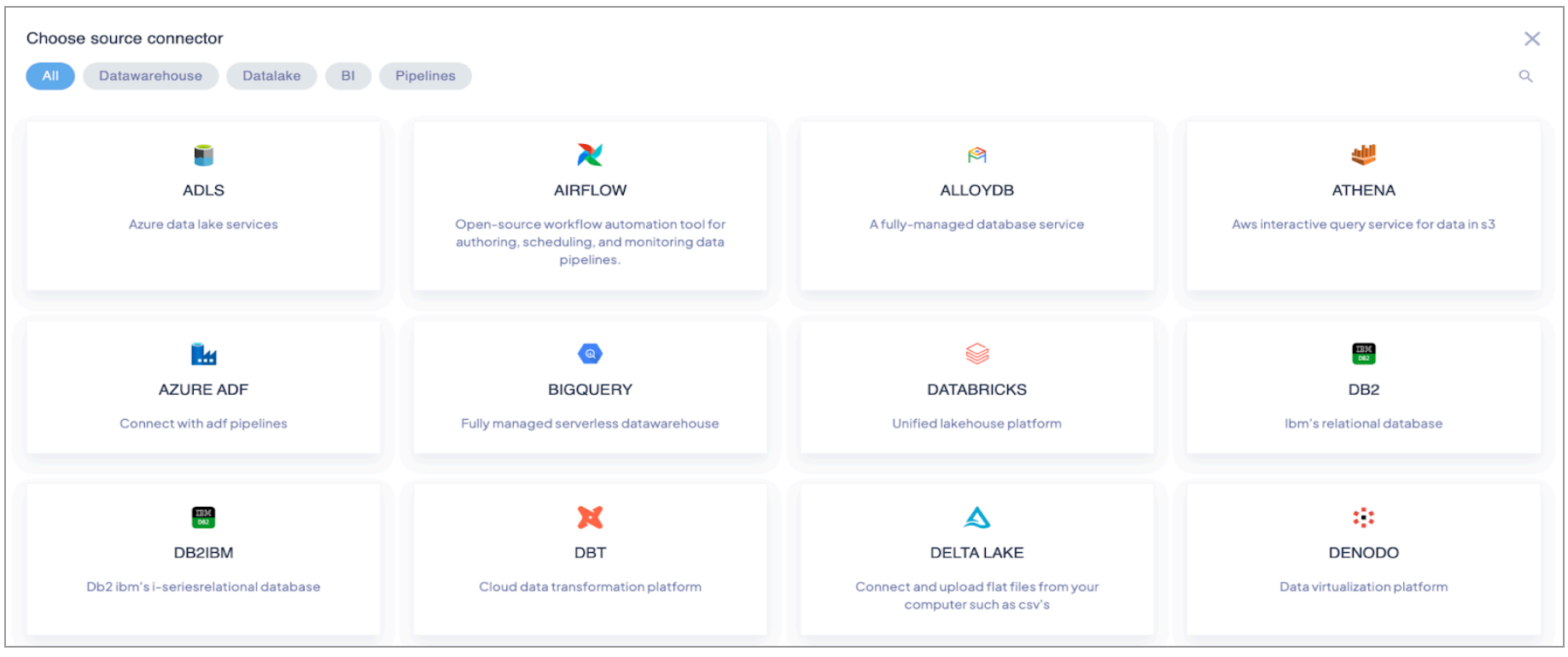
Once you have completed these steps, the service user can connect to the database and perform the authorized operations using the specified credentials.

Connect to Alloy DB

Follow the steps below to connect to Alloy DB:

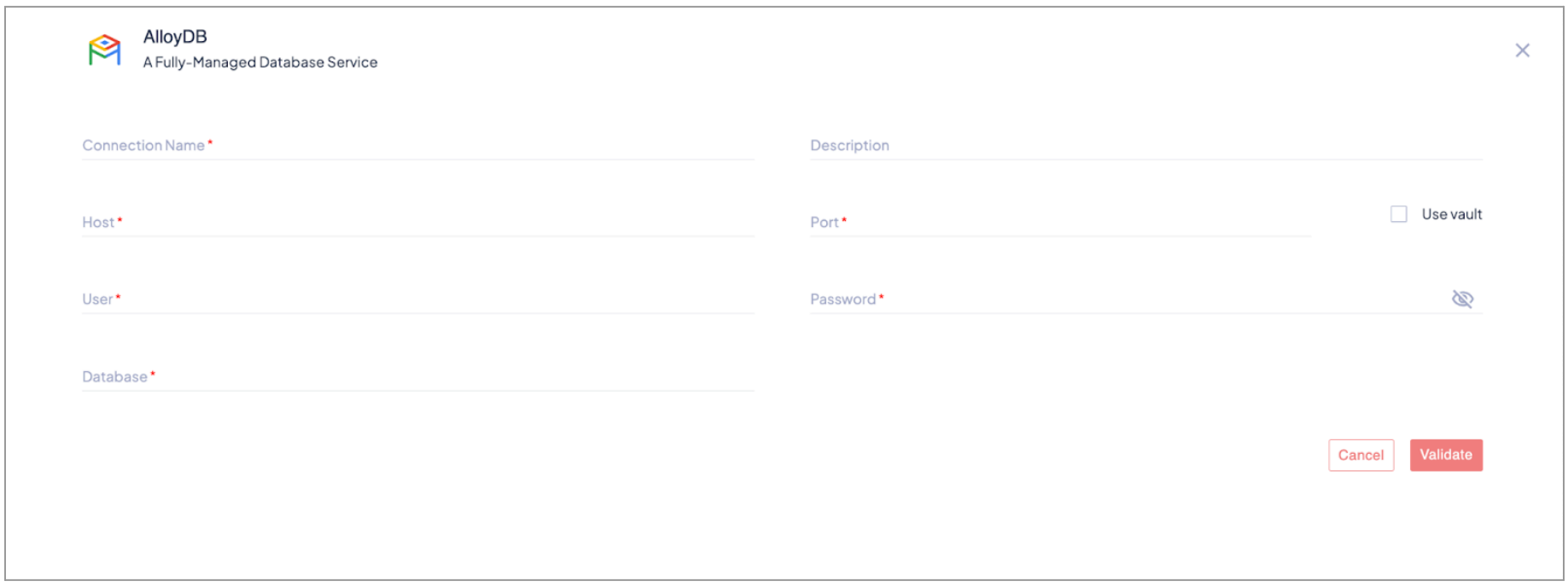
Step 1: Navigate to Settings → Connect → Sources

Step 2: Click on the “+“ icon

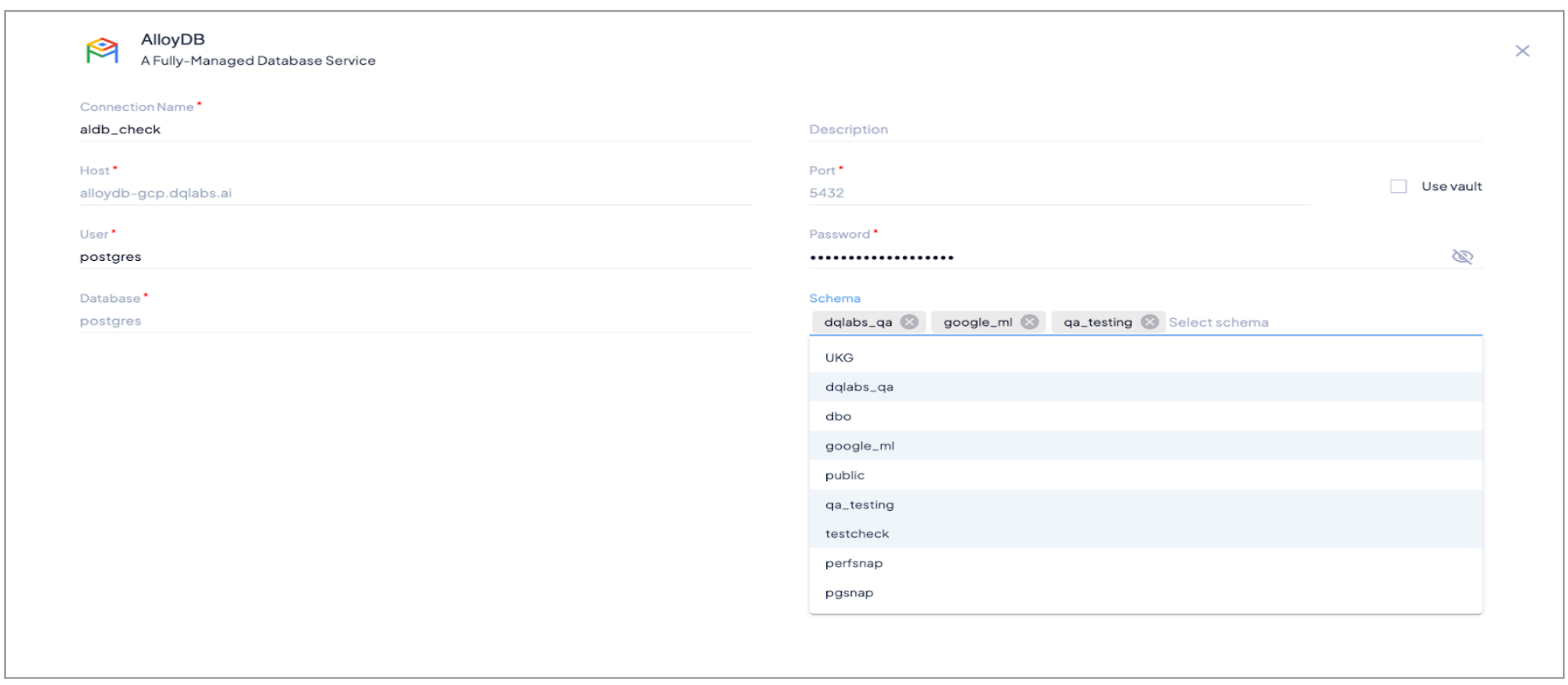


Step 3: Click on "ALLOY DB", then provide the following details and click on Validate

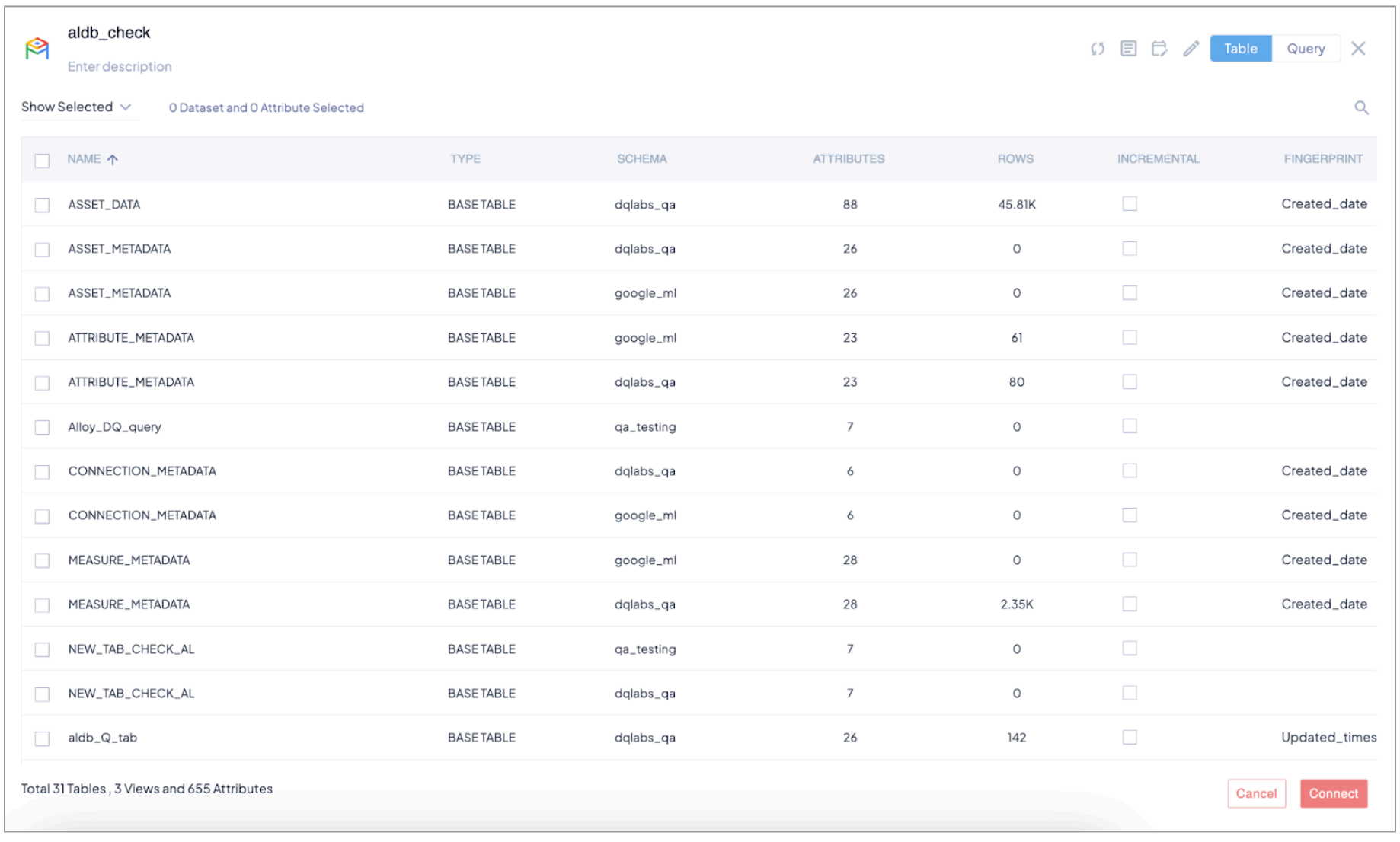
- Connection Name → Name of the connection object
- Host → The server endpoint to Alloy DB
- Port → The open port number to the server
- User → The database service username to which the connection should be established
- Password → Password to the service user
- Database → The database to which the connection should be made



Step 4: Once validated the user will be able to see the list of schemas that can be selected to fetch the objects from



Step 5: Click on Connect. Once connected the user should be able to view the list of all tables/views in the portal and configure the asset.



Step 6: Select the required tables/views and click on connect. Once connected the user will be redirected to the asset detail page of the asset.

Denodo

Denodo is a data virtualization software that allows organizations to integrate data from multiple sources, such as databases, applications, and big data platforms, without having to physically move the data into a centralized repository. Instead, Denodo creates a virtual layer that provides a unified view of the data, which can be accessed and queried in real time by applications, business intelligence tools, and other systems.

Prerequisites

Whitelist IP

If your organization uses a whitelist to manage Denodo access, Quest DQ will only access your Denodo through IP. For assistance on whitelisting, kindly reach out to the support team.

Account Setup

Create a User and Provider Access

As an admin user, follow the steps below to create a user and provide access to the assets in Denodo:

- Log in to the Denodo Virtual DataPort Administration Tool as an administrator.
- Click on the “Users and Roles” option in the navigation panel, and then click on the “Users” tab.
- Click on the “New User” button to create a new user.
- In the “General” tab, enter a username for the new user and select the “Service” option in the “User type” field.
- In the “Authentication” tab, select the authentication method you want to use for the service user. For example, you can use the “Denodo Authentication” method or an external authentication method such as LDAP.
- In the “Permissions” tab, select the “All Permissions” option to provide the service user with all permissions.
- Click on the “Save” button to create the new service user.

Once the service user is created, you can use it to access Denodo services, such as connecting to data sources or executing queries.

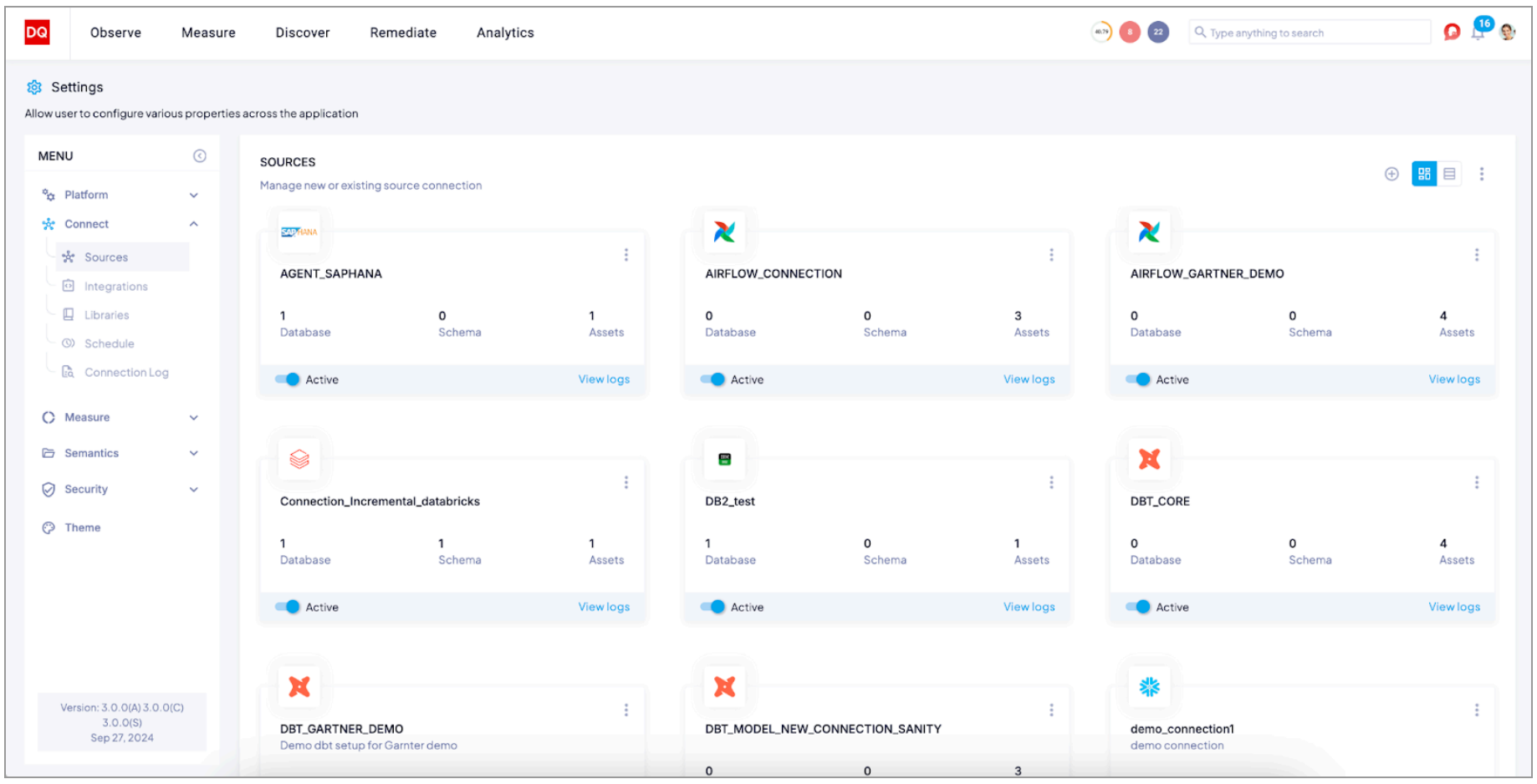
To create a service user in Denodo and provide all access using SQL, you can execute the following SQL script in the Denodo Virtual DataPort Administration Tool:

```
None
CREATE USER <username> PASSWORD '<password>' SERVICE ALL;
```

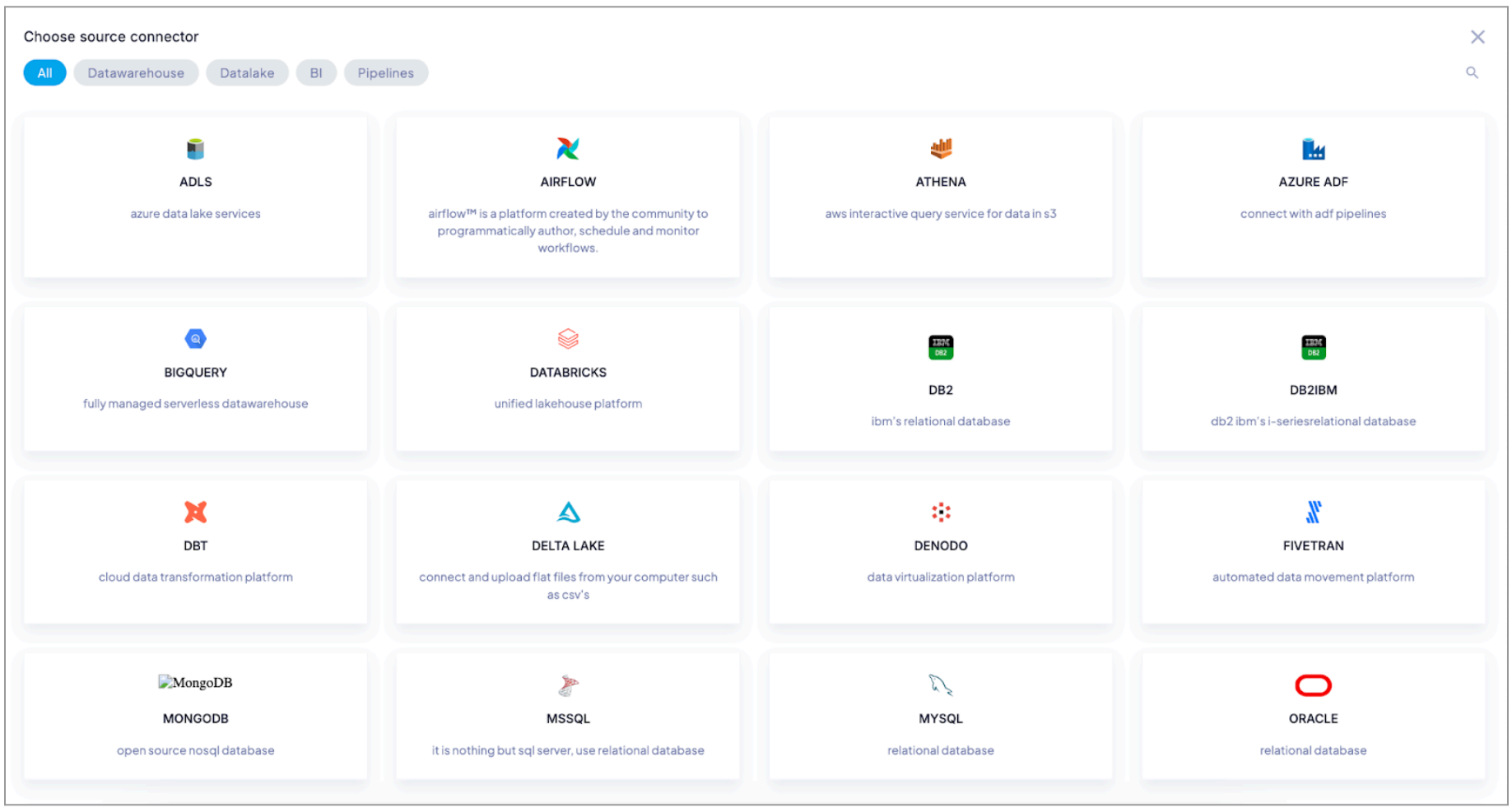
Replace <username> with the desired username for the service user and <password> with the desired password. This SQL script creates a new service user with all permissions, including the ability to connect to data sources, execute queries, and perform administrative tasks.

Connect to Denodo

Step 1: Navigate to **Settings** > **Sources**



Step 2: Go to the + icon in the top right-hand corner of the screen



Step 3: Click on Denodo and provide the following details

- Connection Name
- Description
- Server
- Port
- Database
- Driver Type - JDBC or ODBC
- User
- Password

Step 4: Validate it

Step 5: Once connected, the user will be able to select the required asset and click on connect

MYSQL

MySQL is an open-source relational database management system (RDBMS) used to store, manage, and retrieve structured data. It is widely used in web development and supports SQL (Structured Query Language) for querying and managing databases. MySQL organizes data into tables with rows and columns and allows relationships between tables.

Prerequisites

The following prerequisites must be met in order to establish the connection between MySQL and Quest DQ

Whitelist IP

If your organization uses a whitelist to manage MSSQL access, Quest DQ will only access your MySQL through IP. For assistance on whitelisting, kindly reach out to the support team.

Account Setup

Use the following script to create a user in MySQL

Step 1: Create the service account user

```
None
CREATE USER 'service_user'@'%' IDENTIFIED BY 'YourStrongPasswordHere';
```

Step 2: Grant SELECT access on the entire database

```
None
GRANT SELECT ON my_database.* TO 'service_user'@'%';
```

Step 3: Grant CREATE permissions (includes tables, views, and temporary tables)

```
None
GRANT CREATE, CREATE TEMPORARY TABLES ON my_database.* TO 'service_user'@'%';
```

Step 4: Grant specific permissions for managing views (for "Export Failed Rows Reporting")

```
None
GRANT CREATE VIEW, ALTER, SHOW VIEW ON my_database.* TO 'service_user'@'%';
```

Step 5: Grant access to metadata/system tables (equivalent to sys.schemas, sys.tables, sys.objects in MSSQL)

```
None
GRANT SELECT ON information_schema.* TO 'service_user'@'%';
GRANT SELECT ON performance_schema.* TO 'service_user'@'%';
```

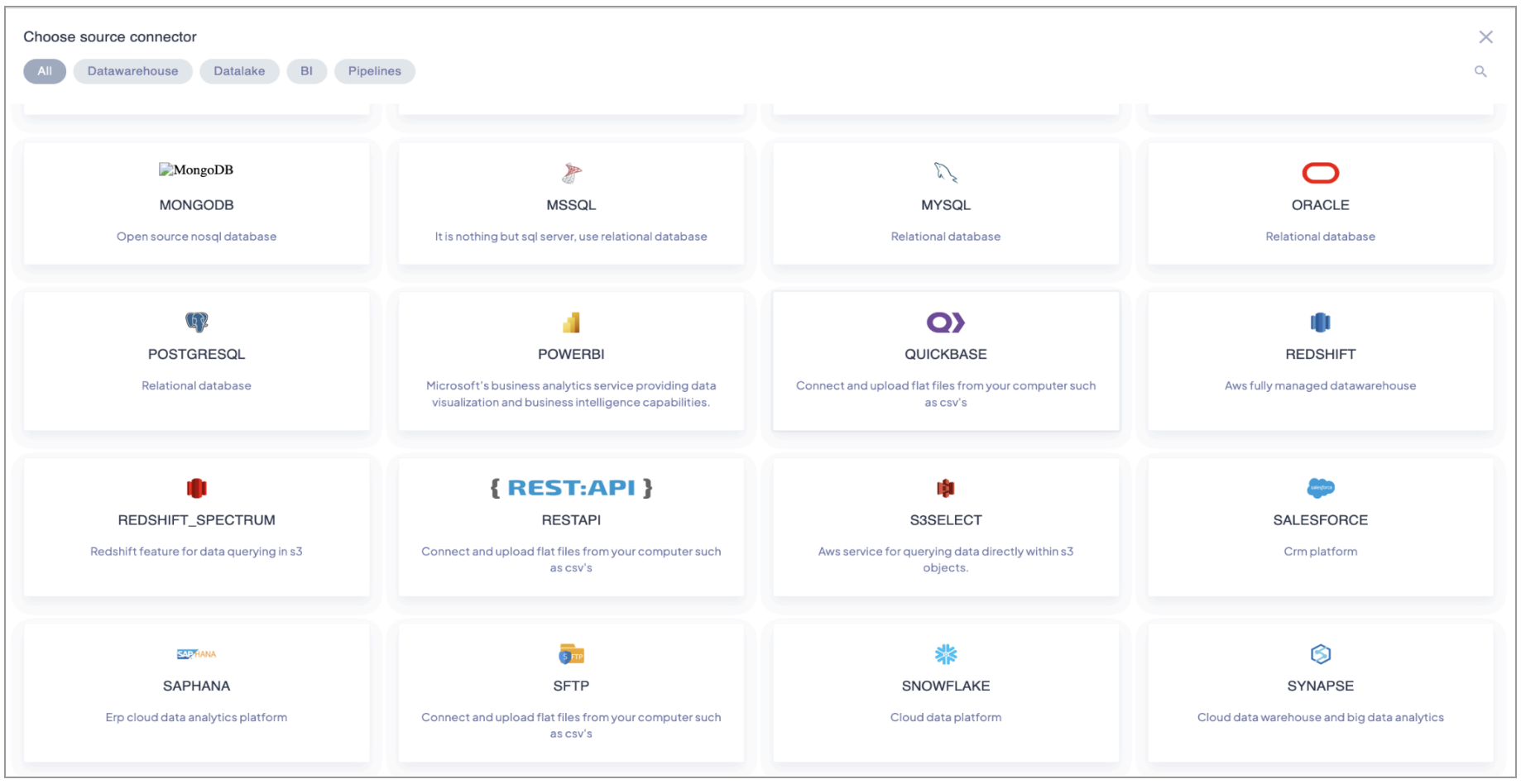
Apply the privilege changes

```
None
FLUSH PRIVILEGES;
```

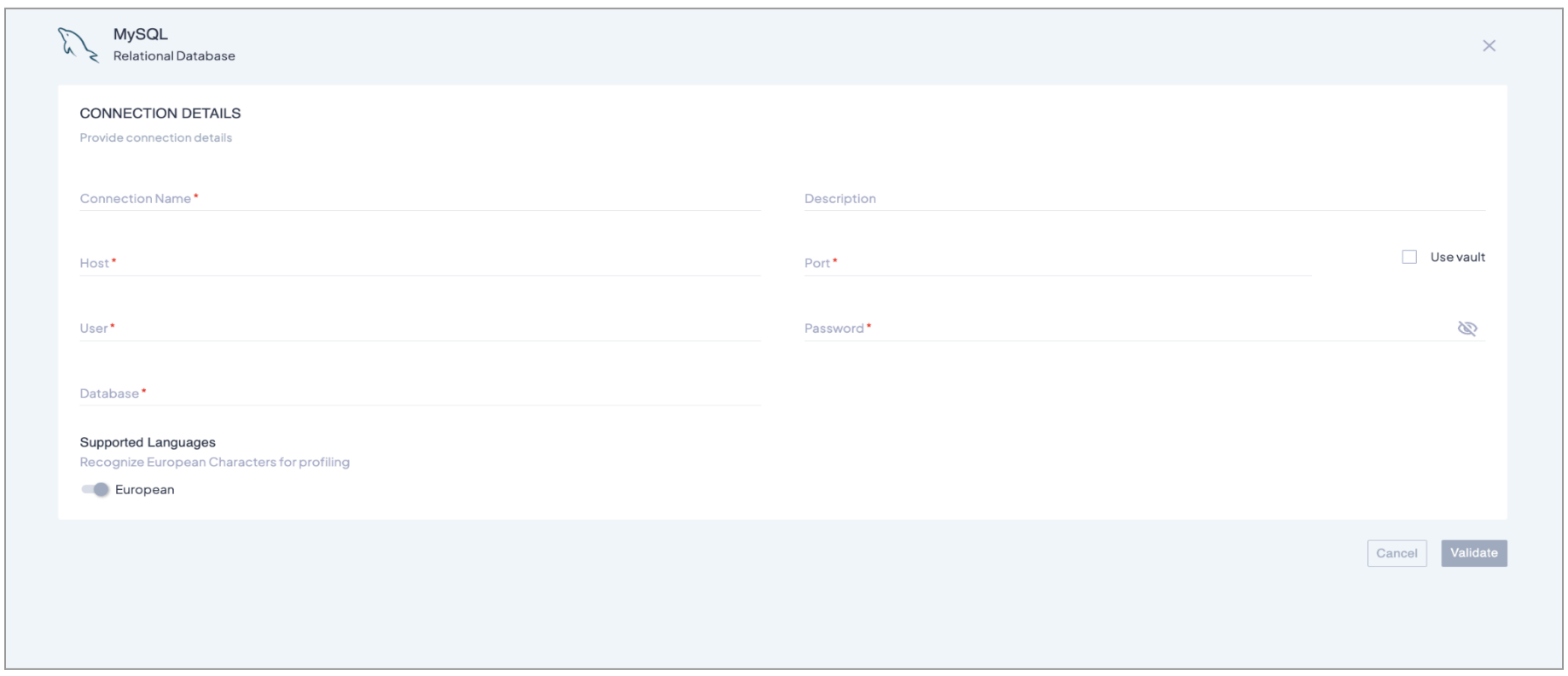
Connect to MYSQL

Step 1: Navigate to Settings -> Sources

Step 2: Go to the + icon in the top right-hand corner of the screen



Step 3: Click on MSSQL and provide the following details:



Field	Description
Connection Name	Name of the connection object
Description	Description of the connection object
Host	The IP address of the MySQL server
Port	The port number to the server
User	The username for the SQL Server
Password	The password of the provided user
Database	Select the required database/schema from the list of available schemas

Step 4: Once validated, click "Connect" to choose the desired tables and Queries

Google Big Query

Google BigQuery is a cloud-based data warehouse and analytics tool that allows users to store, query, and analyze large datasets quickly and easily. It is part of the Google Cloud Platform and is fully managed, meaning that users don't need to worry about infrastructure, scaling, or maintenance.

BigQuery uses a columnar storage format and a distributed architecture to enable fast querying of large datasets. It can process petabytes of data and is particularly suited for analyzing data in real-time. BigQuery supports standard SQL queries and can be integrated with a wide range of other tools and services.

Prerequisites

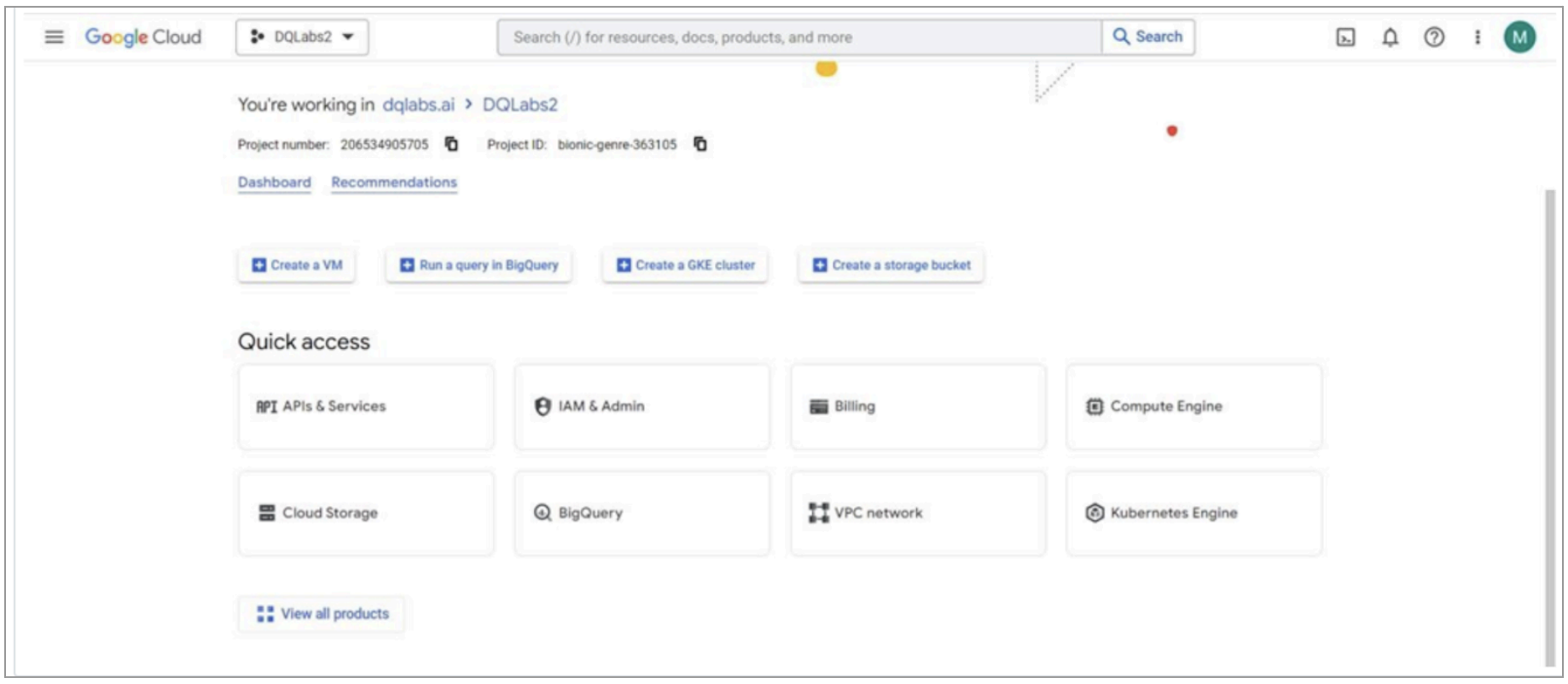
Whitelist IP

If your organization uses whitelist to manage Google Big Query access, Quest DQ will only access your BigQuery through IP. For assistance on whitelisting, kindly reach out to the Support Team.

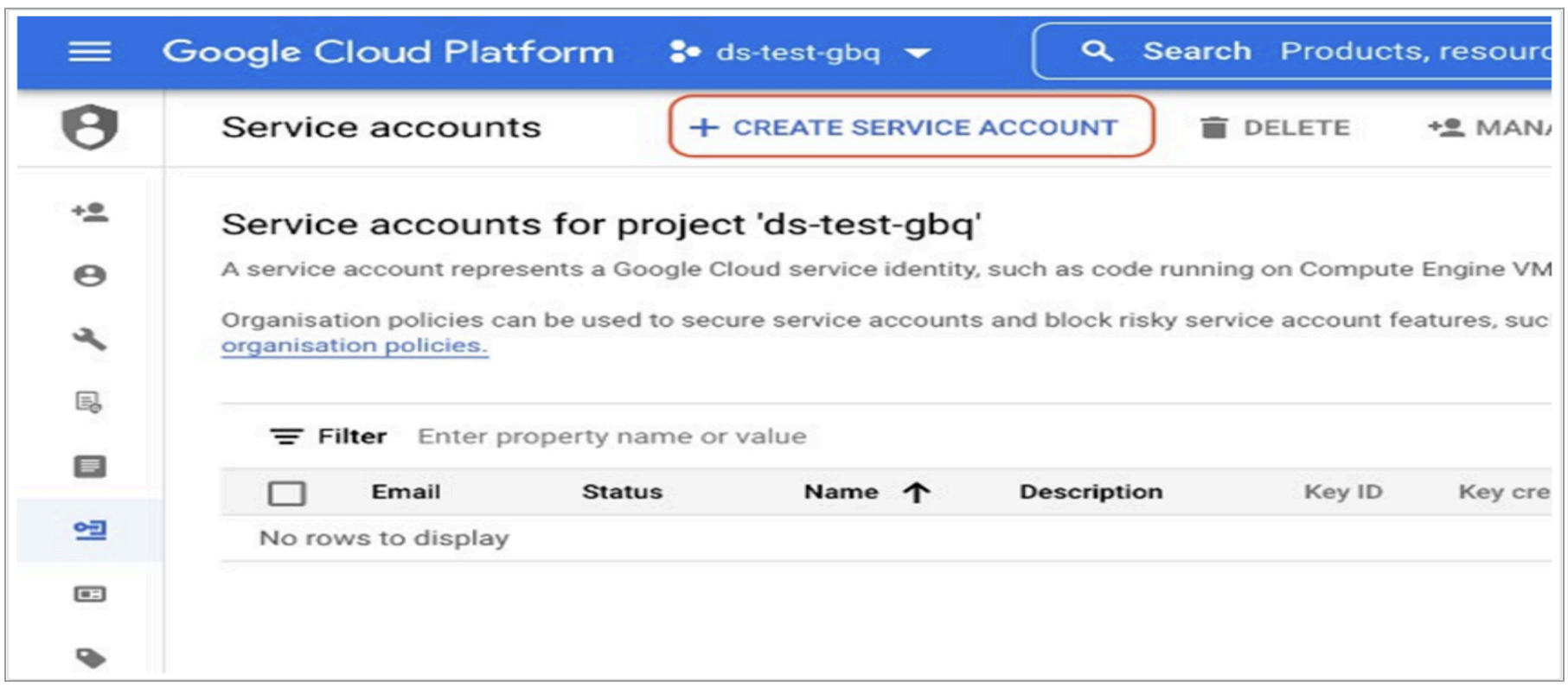
Account Setup

Create a service account and Grant access to Google BigQuery dataset and tables

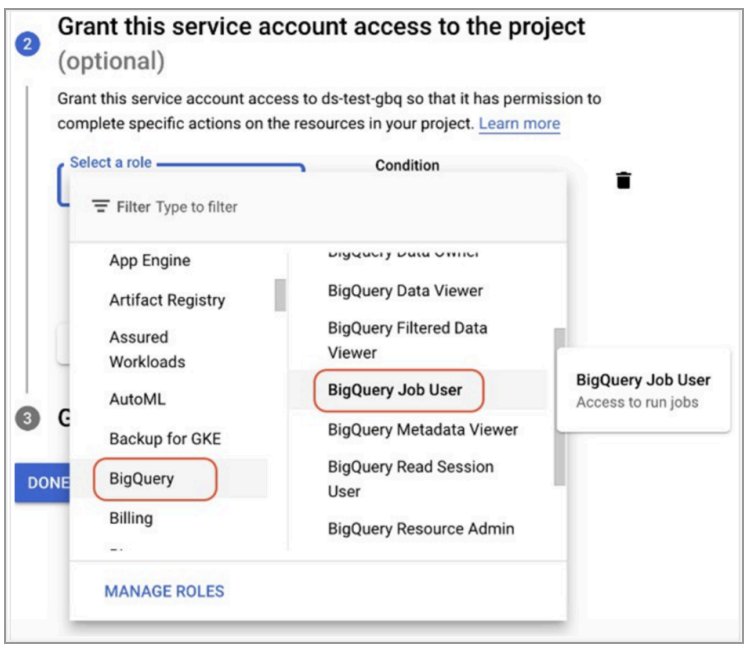
- Go to the Google Cloud Console and select your project
- In the navigation menu, select "IAM& Admin" and then "Service Accounts"



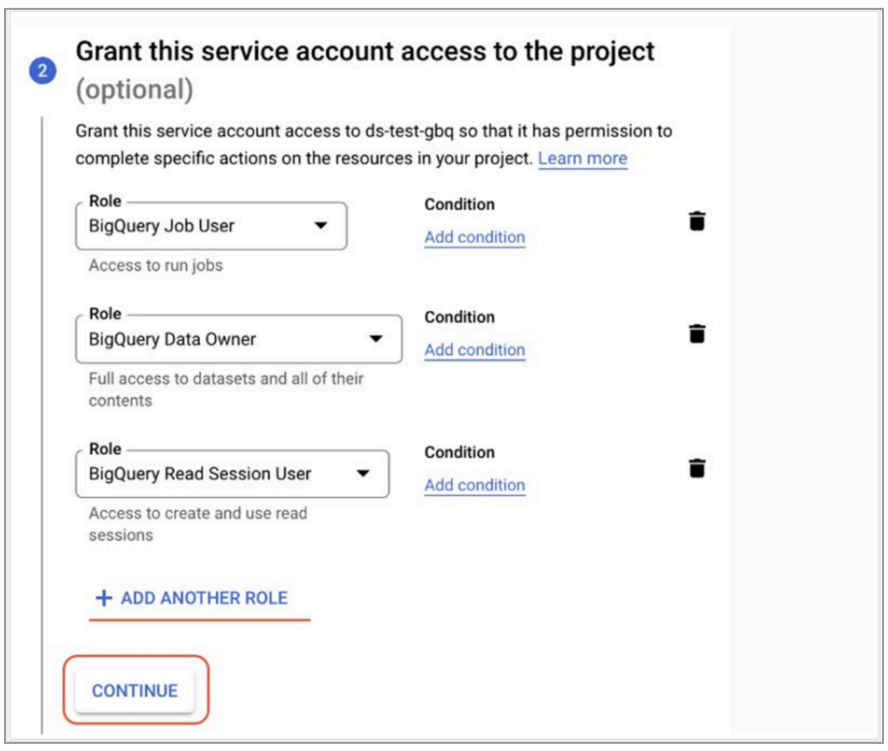
- Click "Create Service Account" and enter a name for the service account



- Select the role(s) you want to grant to the service account, such as BigQuery Data Editor

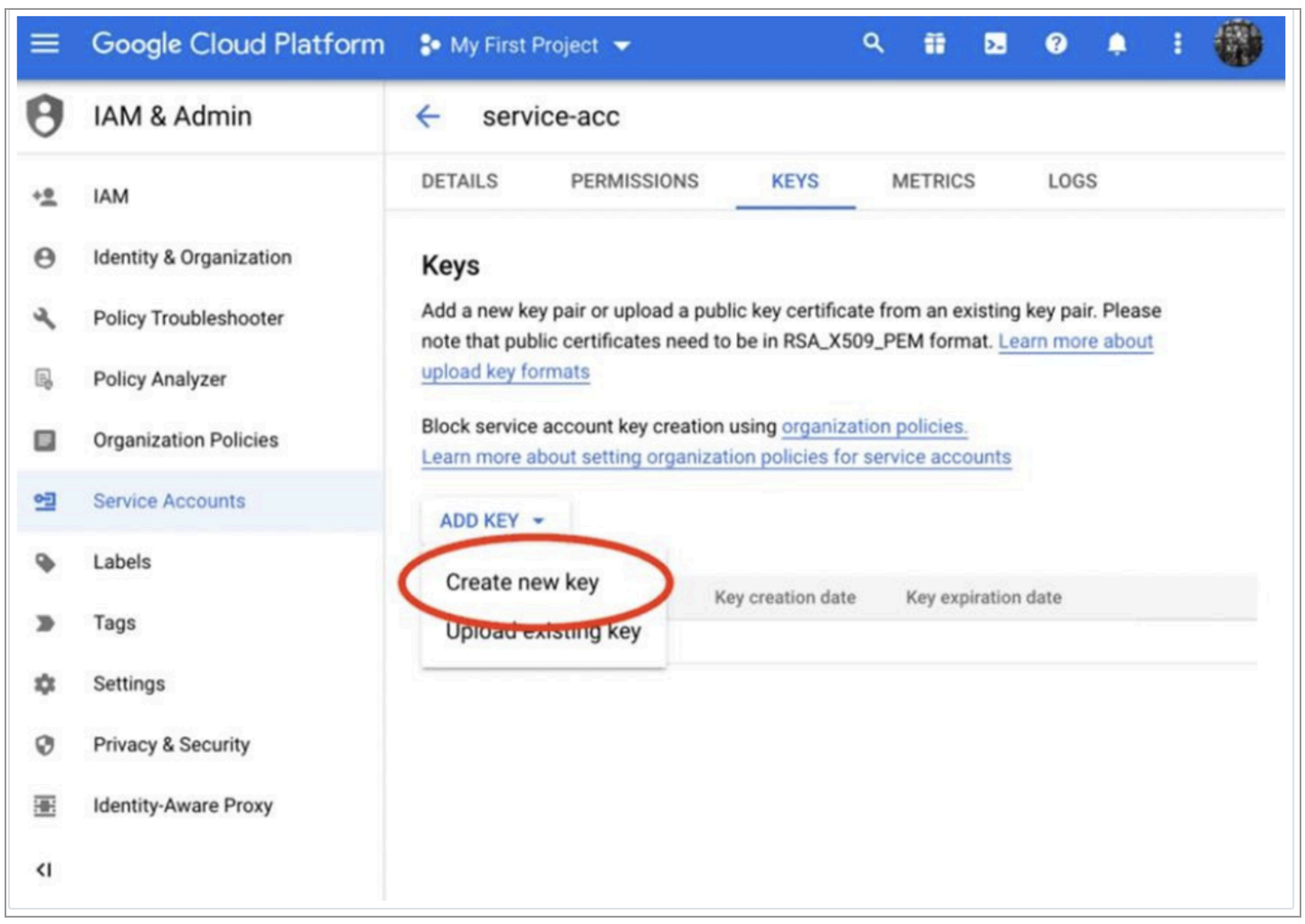


- Click "Create" and then "Done"

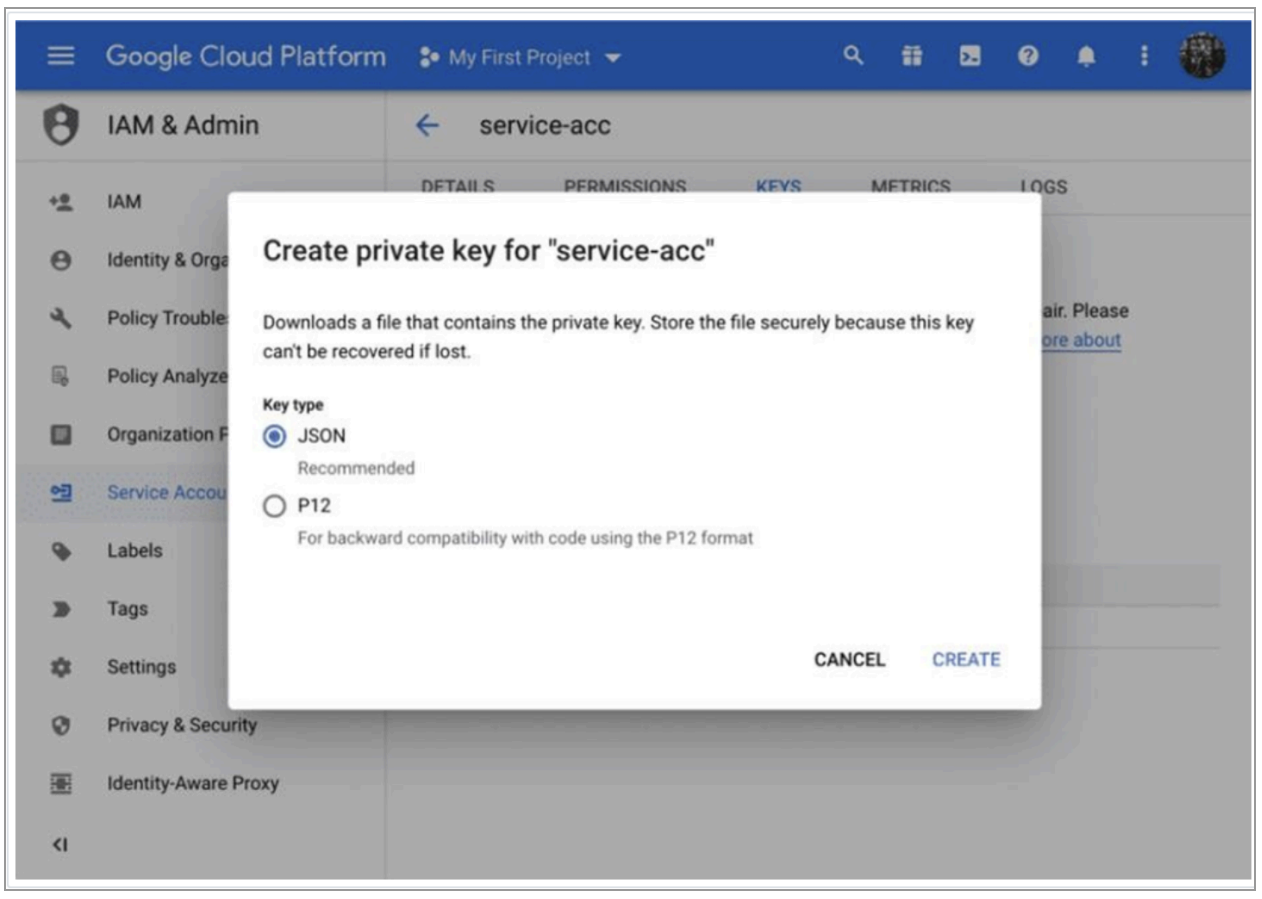


NOTE: Quest DQ requires to create and update permissions only for exception reporting

- Click on the service account just created, and then click"Add Key" or Manage Keys



- Select "JSON" as the key type and click "Create"



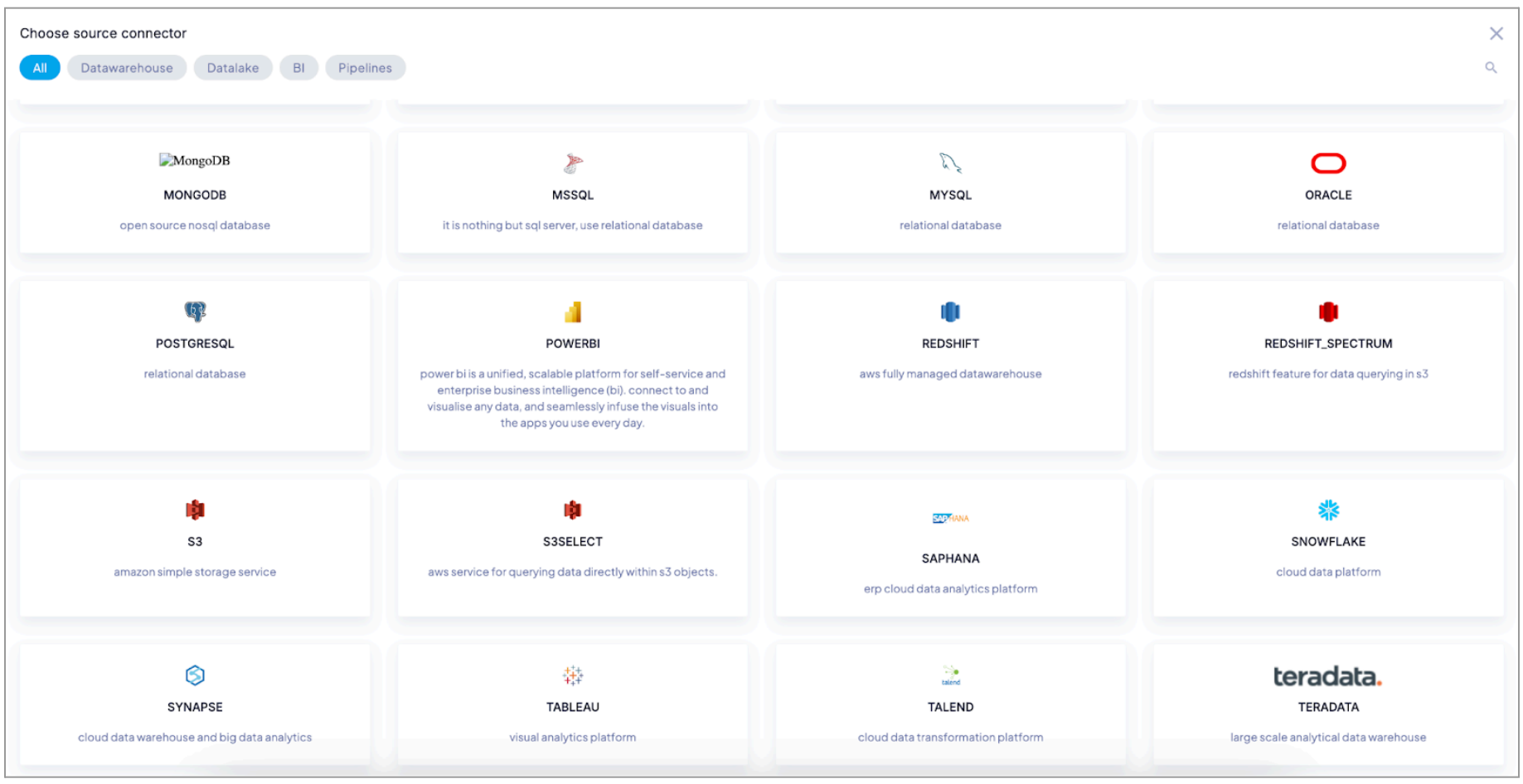
- Save the JSON file to your local machine
- In the BigQuery console, go to the dataset to which you want to grant access and click the "Share Dataset" button.
- Enter the email address associated with the service account you just created and select the appropriate role, such as "BigQuery Data Viewer" or "BigQuery Data Editor"
- Repeat steps 9 and 10 for each table you want to grant access to
- In your application, use the JSON key file to authenticate the service account and access the BigQuery dataset and tables.

Note that the exact steps may vary slightly depending on your specific use case and permissions. Note that the key file contains sensitive information and should be kept secure. You should only share it with those who need access to it

Connect to BigQuery

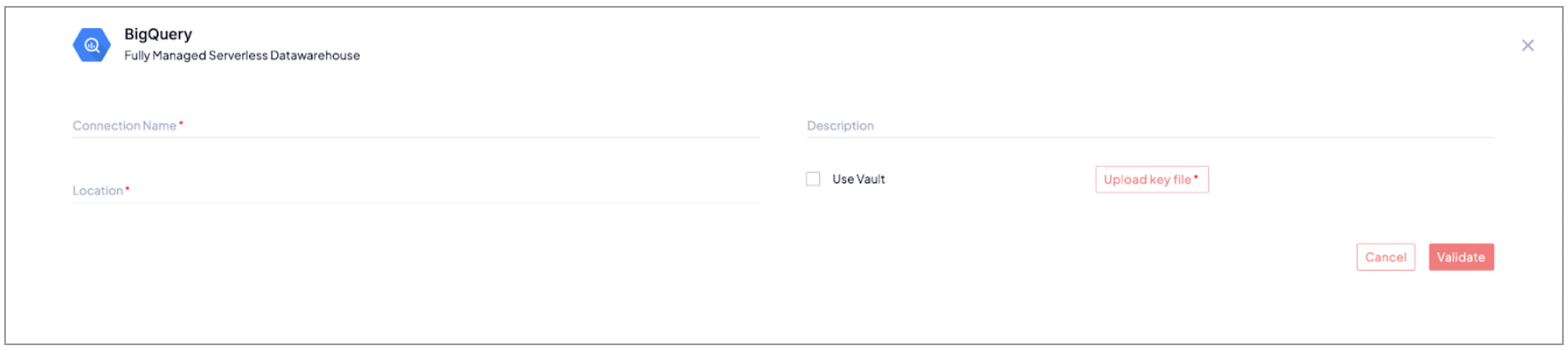
Step 1: Navigate to Settings > Sources

Step 2: Go to the + icon in the top right-hand corner of the screen



Step 3: Click on BigQuery and provide the following details

- Connection name (User Preference)
- Description (Can be used to describe the connection and its purpose)
- Location
- Upload Key file
- Schemas



Step 4: Validate it

Step 5: Once validated, click "Connect" to choose the desired tables and Queries

IBM DB2

Db2, or Database 2, is a set of relational database products built and offered by IBM. Relational databases enable enterprises to create declarative data models accessible via queries. For this purpose, IBM invented the popular and now standardized Structured Query Language (SQL)

Prerequisites

Whitelist IP

If your organization uses a whitelist to manage IBM DB2 access, Quest DQ will only access your IBM DB2 through IP. For assistance on whitelisting, kindly reach out to the Support Team.

Account Setup

To connect to Quest DQ, the service account used should have the following permissions:

- Select for profiling
- Create, write-back options

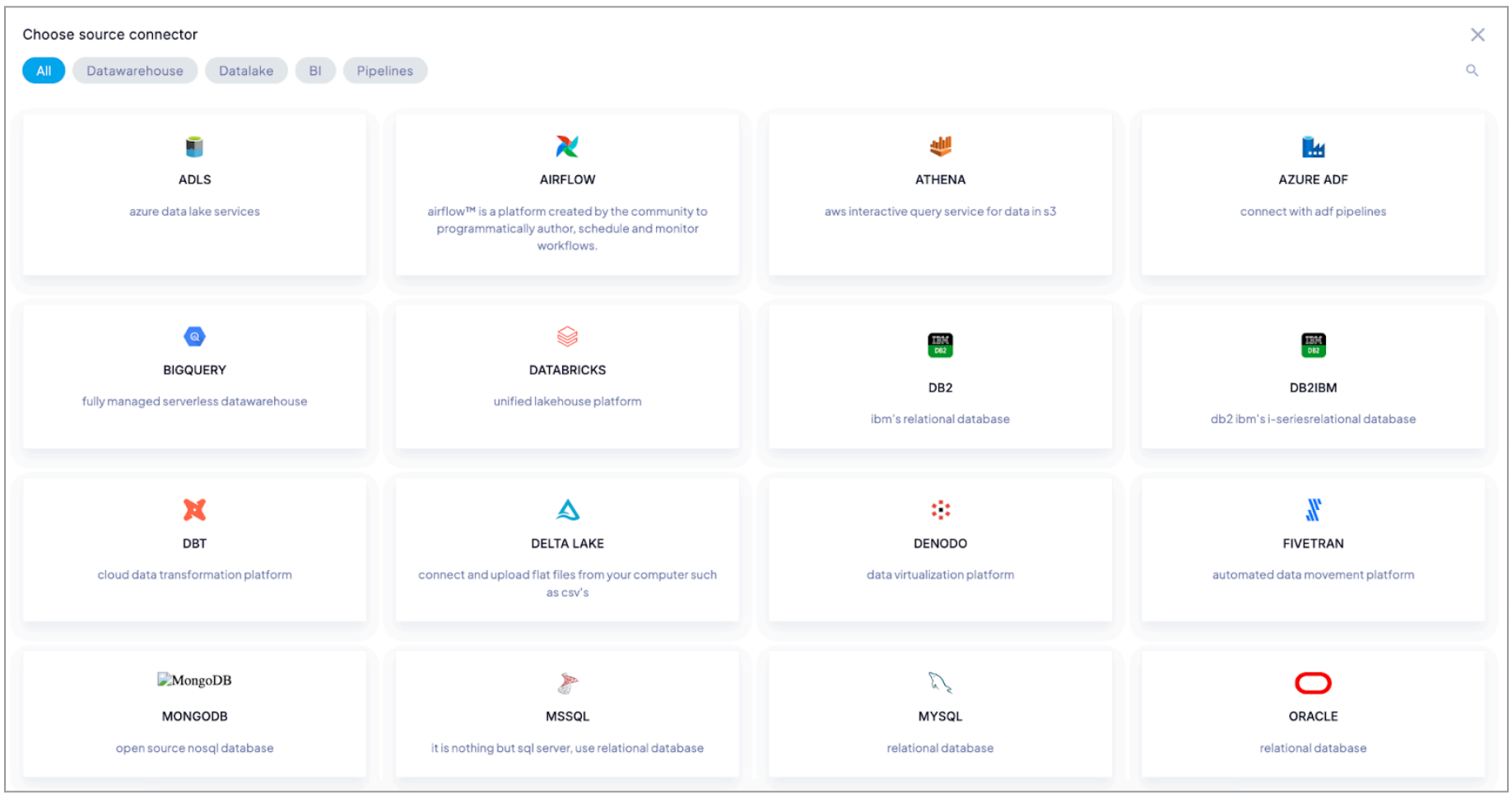
Run the following query to provide access to the account:

```
None
#Minimal permission required
GRANT SELECT ON <Table/Schema> TO <User>;

#Required in case of Exception Reporting/Push-down metrics
GRANT ALL ON TABLE <TABLE> TO <USER> WITH GRANT OPTION;
```

Connect to DB2

Step 1: Navigate to Settings > Sources

Step 2: Go to the + icon in the top right-hand corner of the screen



Step 3: Click on DB2 and provide the following details

- Connection name (User Preference)
- Description (Can be used to describe the connection and its purpose)
- Host
- Port
- Username
- Password
- Database

Step 4: Validate it

Step 5: Once the connection is established, select the required schemas from the list of all available schemas and connect.

Step6: From the list of all available base tables and views, select the required assets and click on connect

IBM DB2 - i-series

DB2 for iSeries, often simply referred to as DB2 for i or DB2/400, is a database management system designed for the IBM i (formerly known as AS/400, iSeries, and System i) platform. IBM i is an integrated operating system with a built-in database (DB2 for i), which is designed to handle enterprise-level workloads with high reliability and efficiency.

Prerequisites

Whitelist IP

If your organization uses a whitelist to manage DB2 iSeries access, Quest DQ will only access your DB2 for iSeries through IP. For assistance on whitelisting, kindly reach out to the Support team.

Account Setup

To connect to Quest DQ, the service account used should have the following permissions:

- Select for profiling
- Create write-back options

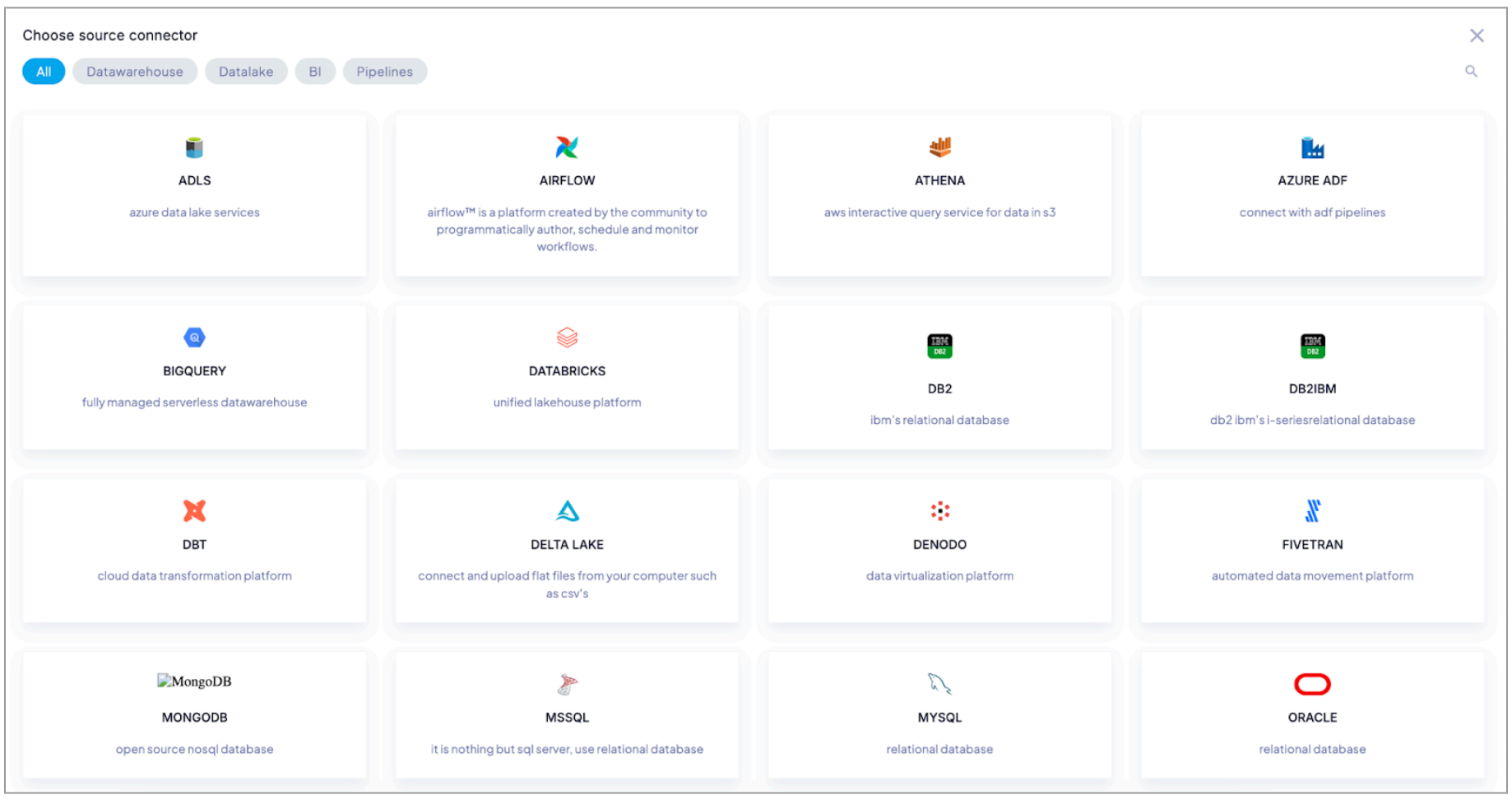
Run the following query to provide access to the account:

```
None
#Minimal permission required
GRANT SELECT ON <Table/Schema> TO <User>;

#Required in case of Exception Reporting/Push-down metrics
GRANT ALL ON TABLE <TABLE> TO <USER> WITH GRANT OPTION;
```

Connect to DB2 I-series

Step 1: Navigate to Settings > Sources

Step 2: Go to the + icon in the top right-hand corner of the screen



Step 3: Click on DB2IBM and provide the following details

- Connection name (User Preference)
- Description (Can be used to describe the connection and its purpose)
- Host
- Port
- Username
- Password

- Database

Step 4: Validate it

Step 5: Once the connection is established, select the required schemas from the list of all available schemas and connect.

Step6: From the list of all available base tables and views, select the required assets and click on connect

Redshift Spectrum

Redshift Spectrum is an extension of Amazon Redshift that enables users to query data stored in Amazon S3 buckets directly from their Redshift cluster. It allows users to use their existing SQL query skills to analyze data stored in S3 without the need to load it into Redshift first.

Redshift Spectrum leverages Redshift's massively parallel processing (MPP) architecture to parallelize and distribute queries across Redshift and S3. It supports various data formats, including CSV, JSON, Avro, and Parquet, and can handle petabytes of data stored in S3.

Prerequisites

Whitelist IP

If your organization uses a whitelist to manage Redshift Spectrum access, Quest DQ will only access your Redshift Spectrum through IP. For assistance on whitelisting, kindly reach out to the support team.

Account Setup

Create a User and Provide Access

To create a user and provide access to Redshift Spectrum, you can follow these steps:

- Create a user in Amazon Redshift by using the CREATE USER command. For example, you can create a user named "spectrum_user" by running the following command:

```
None
CREATE USER spectrum_user WITH PASSWORD 'password';
```

- Grant USAGE permission to the new user on the Redshift Spectrum schema in the Amazon Redshift database. For example, you can grant USAGE permission to the "spectrum_user" on the "spectrum_schema" schema by running the following command:

```
None
GRANT USAGE ON SCHEMA spectrum_schema TO spectrum_user;
```

- Grant SELECT permission on the external tables in Redshift Spectrum to the new user. In the case of Exceptional reporting, the user requires write permissions to the dedicated schema. For example, you can grant SELECT permission to the "spectrum_user" on the "sales" table in the "spectrum_schema" schema by running the following command:

```
None
GRANT SELECT ON TABLE spectrum_schema.sales TO spectrum_user;
```

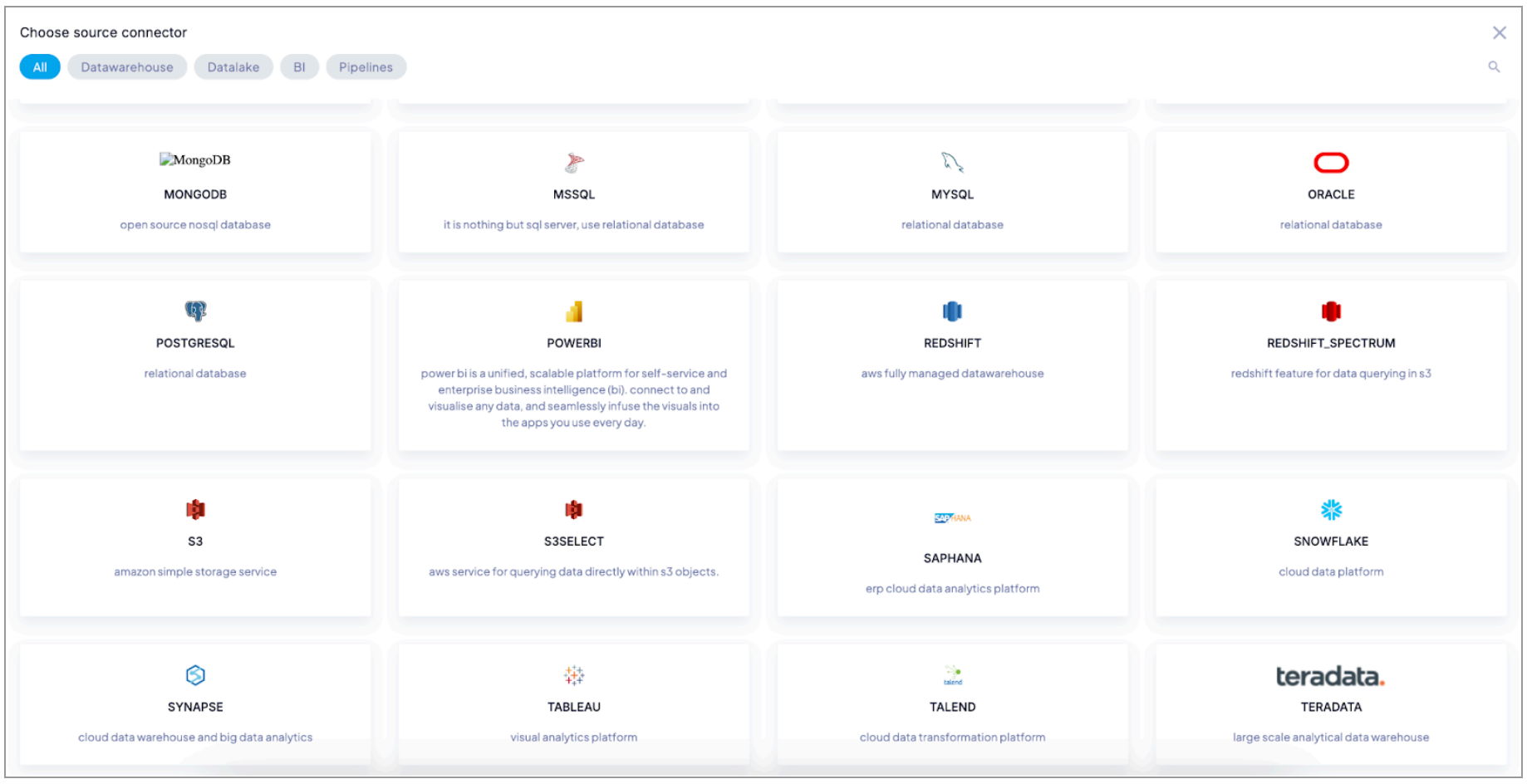
- If the external data source is in an S3 bucket, then the new user must also have permissions to access the S3 bucket. You can grant S3 bucket permissions to the new user by using an S3 bucket policy or an IAM policy that grants access to the S3 bucket.

After completing these steps, the new user will have access to the external tables in Redshift Spectrum.

Connect to Redshift Spectrum

Step 1: Navigate to **Settings** > **Sources**

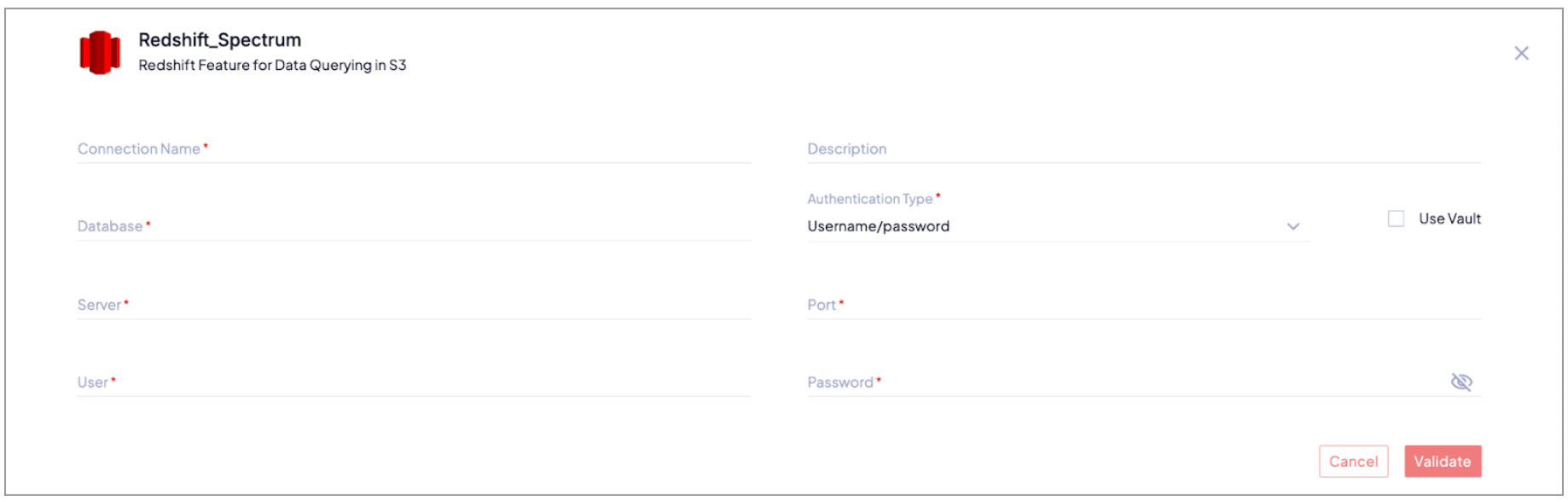
Step 2: Go to the + icon in the top right-hand corner of the screen



Quest DQ allows the user to connect to Redshift using Secret Manager or by using a username and password. Please follow the steps below to create a connection using username and password

Step 3: Click on Redshift Spectrum and provide the following details:

- Connection Name
- Description
- Server
- Port
- Database
- Authentication Type - Select Username and Password
- Username
- Password



Step 4: Validate it

Step 5: Once validated, click "Connect" to choose the desired tables and Queries

Snowflake

Snowflake is a multi-cloud data warehouse optimized for analytics workloads and requiring little maintenance. The Quest DQ platform not only observes the data within your Snowflake instance so you can be the first to know about potential data bugs but also enables your centralized data quality stewardship and discovery process seamlessly. For details on getting started with Snowflake, please refer to https://docs.snowflake.com/en/user-guide-getting-started.html

Compute Resource

In order to optimize the metadata collection better, please configure the Snowflake warehouse accordingly as below for the one assigned to the service account used for Quest DQ. The configuration, computing, and cost depend on the Warehouse Instance Size, Cluster Size, and Scaling Policy.

NEED	WAREHOUSE SIZE
Asset Level Metadata Processing for Observability Column Level Processing for Data Quality Stewardship	SMALL CREATE WAREHOUSE IF NOT EXISTS WH_DQLABS WAREHOUSE_SIZE=SMALL INITIALLY_SUSPENDED=TRUE AUTO_SUSPEND = 5 AUTO_RESUME = TRUE;
Asset Level Metadata Processing for Observability Column Level Processing for Data Quality Stewardship Data Discovery for Automated Tagging	MEDIUM CREATE WAREHOUSE IF NOT EXISTS WH_DQLABS WAREHOUSE_SIZE=MEDIUM INITIALLY_SUSPENDED=TRUE AUTO_SUSPEND = 5 AUTO_RESUME = TRUE;
Asset Level Metadata Processing for Observability Column Level Processing for Data Quality Stewardship Data Discovery for Automated Tagging	LARGE CREATE WAREHOUSE IF NOT EXISTS WH_DQLABS WAREHOUSE_SIZE=LARGE INITIALLY_SUSPENDED=TRUE AUTO_SUSPEND = 5 AUTO_RESUME = TRUE;

For more details on warehouse size and Credits/hour, Credits/second, please refer to Snowflake https://docs.snowflake.com/en/user-guide/warehouses-overview.html

Prerequisites

As a superuser, execute the following SQL commands to create a read-only role, a user (e.g. SVC_QUESTDQ) assigned to that role, and a warehouse for that role.

Permissions

Quest DQ fetches the metadata from the <database > .INFORMATION_SCHEMA, usage grant on the database should provide permissions to read INFORMATION_SCHEMA for the service account role. We also need access to Snowflake Metadata to Fetch Query Counts on the data assets, and this is pulled from SNOWFLAKE.ACCOUNT_USAGE.

Provide the following access for that “READ ONLY” service account role that is assigned to the Quest DQ service account

- USAGE Access to Database
- USAGE Access on Schema
- Usage Access on Warehouse
- Select access on all tables in the Schema/Database that is defined in Quest DQ Snowflake Connection
- Select access on Snowflake metadata for usage information

```
None
-- Read-only access to specific schemas and warehouse (CHANGE THIS)
set schema_name = 'DATABASE_NAME.SCHEMA_NAME';
set warehouse_name = 'WAREHOUSE_NAME';

grant USAGE on database identifier($database_name) to role identifier($DQ_ROLE);
grant USAGE on schema identifier($schema_name) to role identifier($DQ_ROLE);
grant USAGE on warehouse identifier($warehouse_name) to role identifier($DQ_ROLE);
grant SELECT on all tables in schema identifier($schema_name) to role identifier($DQ_ROLE);
grant SELECT on future tables in schema identifier($schema_name) to role identifier($DQ_ROLE);
grant SELECT on all views in schema identifier($schema_name) to role identifier($DQ_ROLE);
grant SELECT on future views in schema identifier($schema_name) to role identifier($DQ_ROLE);
grant imported privileges on database snowflake to $DQ_ROLE
```

Whitelist IP

If your organization uses a whitelist to manage Snowflake access, Quest DQ will only access your Snowflake through IP. For assistance on whitelisting, kindly reach out to the Support team.

Key-Based Authentication Setup

Step 1: Open a terminal window and generate a Private Key. The user can generate either an encrypted or an unencrypted version of the private key. Quest DQ suggests using an encrypted version for security.

To generate an encrypted version, use the following command:

```
None
openssl genrsa 2048 | openssl pkcs8 -topk8 -v2 des3 -inform PEM -out rsa_key.p8
```

This command generates a private key in PEM format.

```
None
-----BEGIN ENCRYPTED PRIVATE KEY-----
MIIE6T...
/-----END ENCRYPTED PRIVATE KEY-----
```

Step 2: Generate a public key
From the command line, generate the public key by referencing the private key. The following command assumes the private key is encrypted and contained in the file named rsa_key.p8.

```
None
openssl rsa -in rsa_key.p8 -pubout -out rsa_key.pub
```

This command generates a private key in PEM format.

```
None
-----BEGIN PUBLIC KEY-----
MIIBIj...
-----END PUBLIC KEY-----
```

Step 3: Store the private and public keys securely. Copy the public and private key files to a local directory for storage. Record the path to the files. Note that the private key is stored using the PKCS#8 (Public Key Cryptography Standards) format and is encrypted using the passphrase you specified in the previous step

Step 4: Assign the public key to a Snowflake user. Execute an ALTER USER command to assign the public key to a Snowflake user

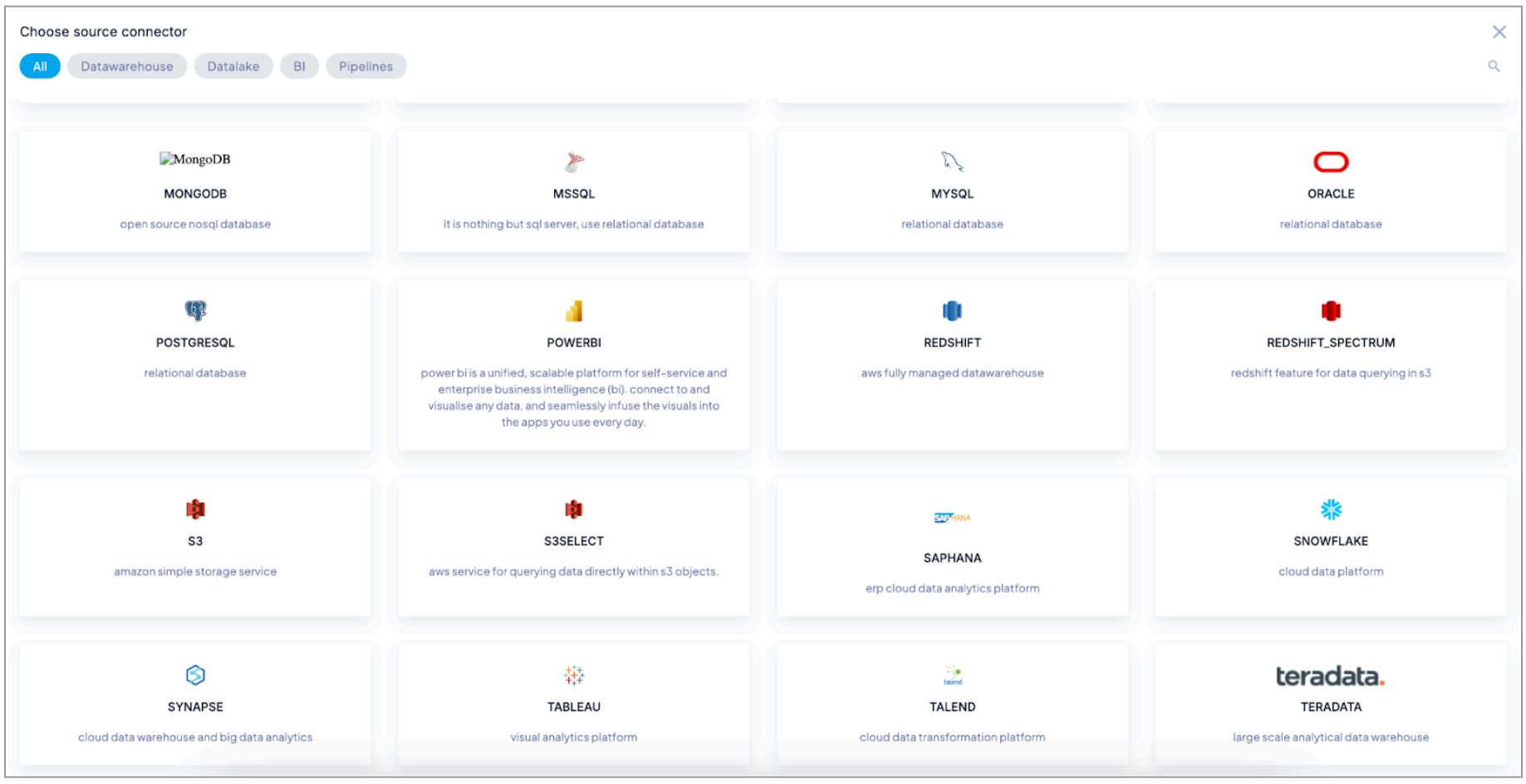
```
None
ALTER USER <user_name> SET RSA_PUBLIC_KEY='<publoc_key_in_the_rsa_key.pub_file>';
```

Step 5: Log in to the Quest DQ portal. Create a snowflake connection by selecting key-based authentication. Enter the <user_name> in the user field, and the passphrase that is used to generate the rsa_key.p8 file. Upload the rsa_key.p8 file from the local and validate

Connect to Snowflake

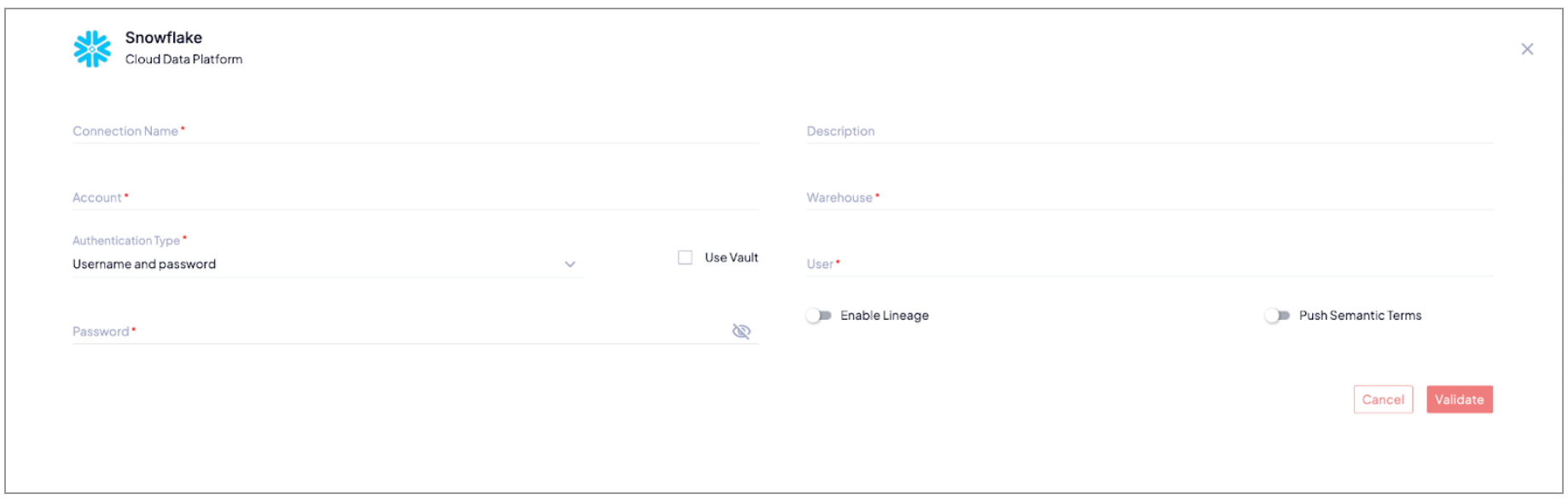
Step 1: Navigate to Settings > Sources

Step 2: Go to the + icon in the top right-hand corner of the screen



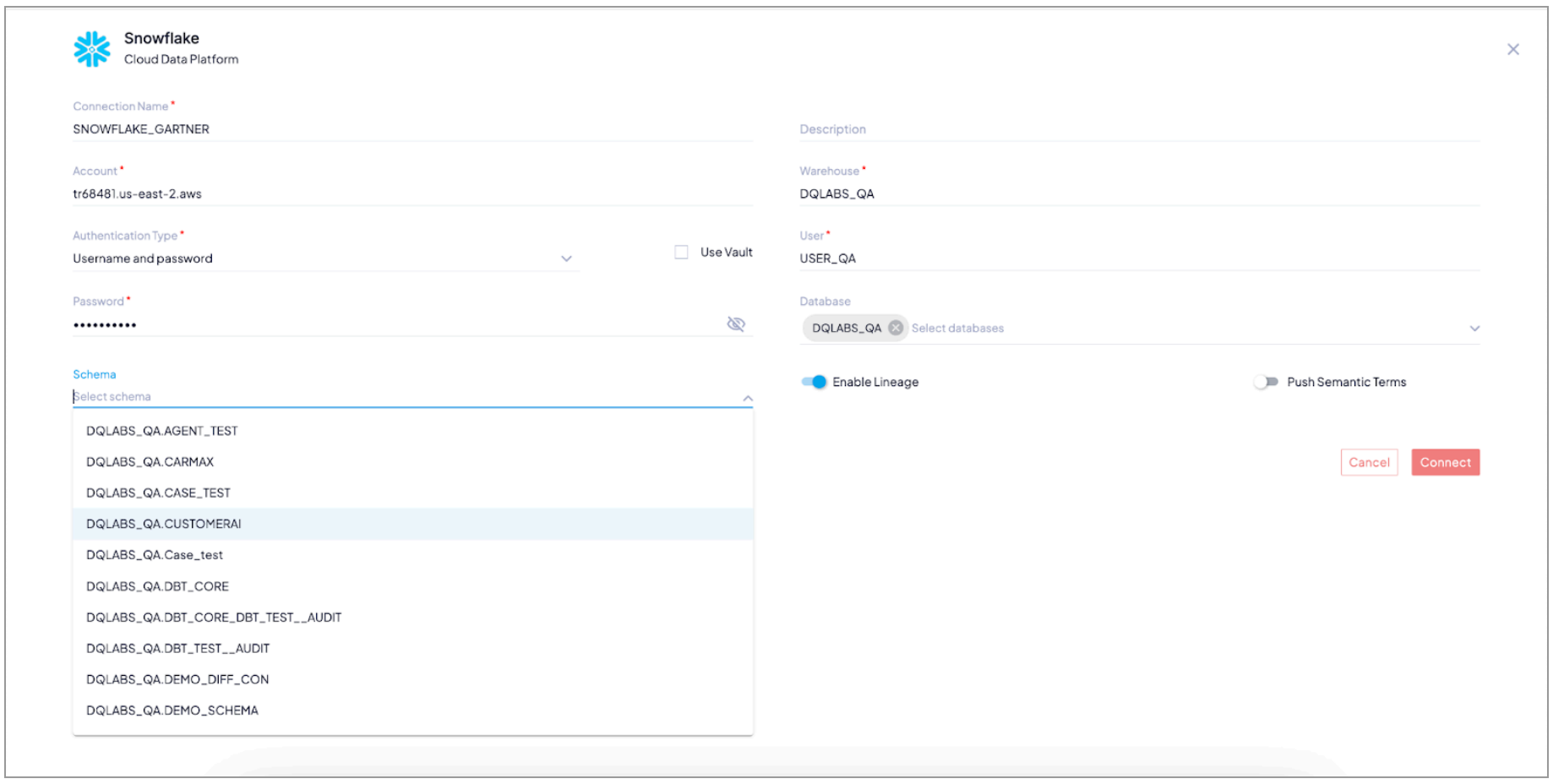
Step 3: Click on Snowflake and provide the following details

Field	**Description**
Connection Name	Name of the connection object
Description	Description of the connection object
Account	Snowflake account details
Warehouse	Warehouse to be used in Snowflake
Authentication Type	Username and Password or Key-based Authentication
Passphrase	Passphrase for key-based authentication
Upload Private Key	Upload the private key for key-based authentication
User	Snowflake user name or service user name
Password	Password for the respective user
Database	Select from the list of available databases
Schemas	Select from the list of schemas to pull the objects



Step 5: Validate it.

Step 6: Select the required database and schema from the list of available databases and schemas



Step 7: Once validated, click "Connect" to choose the desired tables and Queries

SAP Hana

SAP HANA is a multi-model database that keeps data in memory rather than on a disk. This leads to orders of magnitude faster data processing than disk-based data systems, enabling sophisticated real-time analytics.

Prerequisites

If your organization uses a whitelist to manage SAP HANA access, Quest DQ will only access your SAP HANA through IP. For assistance on whitelisting, kindly reach out to the support team.

Account Setup

To create a service user and provide access to a database in SAP HANA, you can follow these steps:

- Login to SAP HANA in the administrator account
- Run the following SQL command to create a user

```
None
CREATE USER <User> PASSWORD <Password> no force_first_password_change;
```

- Add the new user to the role that has access to the schema

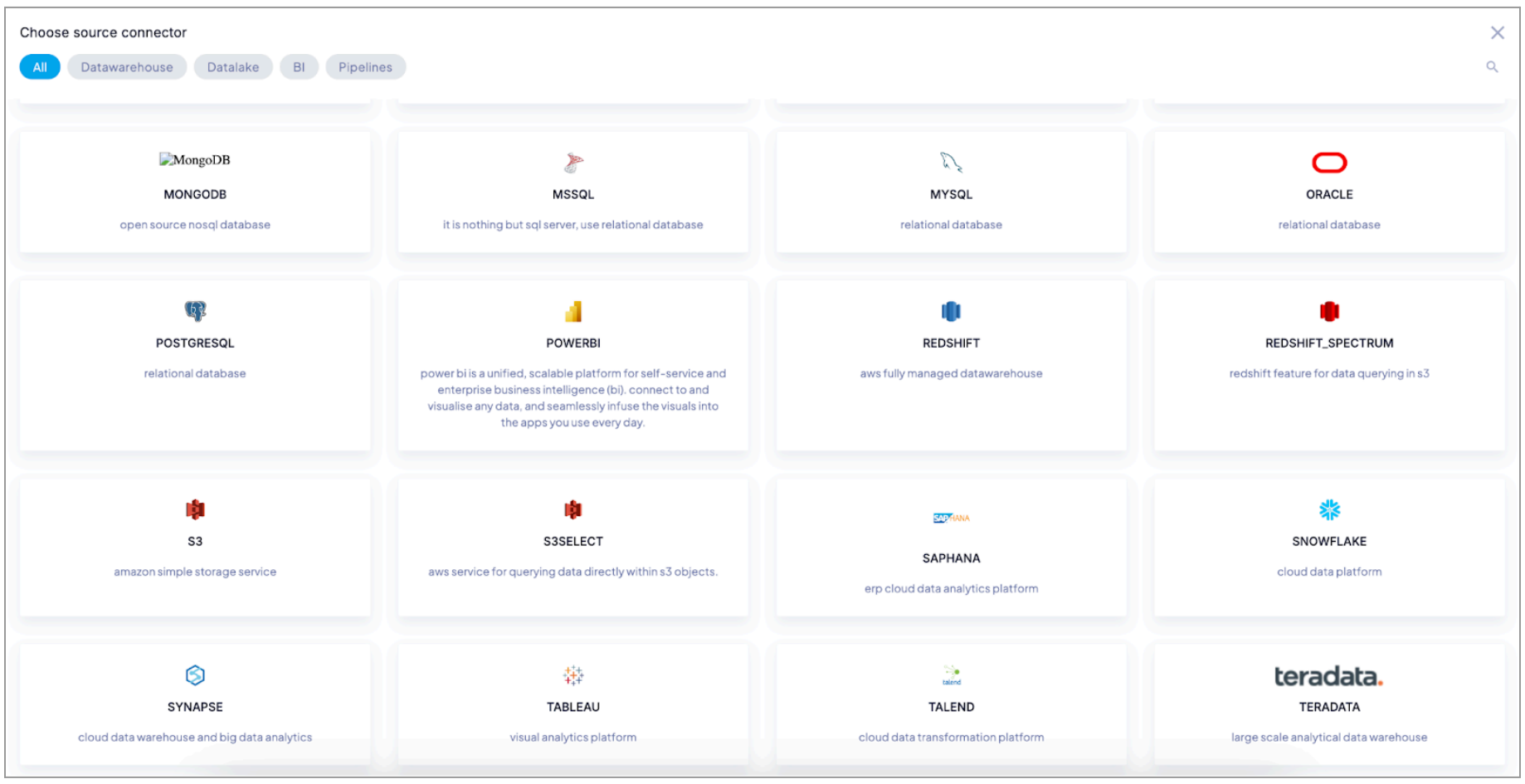
```
None
GRANT <privilege> ON <SCHEMA_or_OBJECT> <schema_or_object_name> TO <user_name>;
```

Note: The platform does not currently support parameterized and calculated views. The following are the challenges in implementing the above two:
1. Views can be constructed without default input from the SAP HANA developer, causing normal queries to fail without a parameter.
2. Parameters can only be single or multiple values, and cannot use range inclusion like "between," "in list," "greater than," or "less than." For ranges, the first parameter is the low end and the second is the high end.
3. Parameters can only be strings or numbers. While dates can be supported, they must be in a specific format and cannot be converted using the “to_date” function.
4. There is no detectable information in the information schema about the number of required parameters or their expected variables. Users must know this information beforehand.
5. Queries require manual input of parameters at runtime.

Connect to SAP Hana

Step 1: Navigate to Settings -> Sources

Step 2: Go to the + icon in the top right-hand corner of the screen



Step 3: Click on SAP HANA and provide the following details

- Connection name (User Preference)
- Description (Can be used to describe the connection and its purpose)
- Host
- Port
- Username
- Password
- Database

Step 4: Validate it

Step 5: Once the connection is established, select the required schemas from the list of all available schemas and connect.

Teradata

Teradata is a relational database management system (RDBMS) designed for data warehousing and big data analytics. It is known for handling large-scale data and complex queries efficiently. Teradata uses parallel processing to improve performance, making it a strong choice for enterprises that need to analyze vast amounts of data. Quest DQ allows users to connect to Teradata and profile data.

Prerequisites

Whitelisting

If your organization uses a whitelist to manage Snowflake access, Quest DQ will only access your Snowflake through IP. For assistance on whitelisting, kindly reach out to the Support team.

User Access

Follow the steps below to create a user and assign permissions:

- Create a user in Teradata

```
None
CREATE USER new_user AS
PASSWORD = 'your_password',
PERM = 10000000,  -- Allocate space in bytes
TEMP = 1000000,   -- Temporary space for queries
SPOOL = 20000000; -- Spool space for query execution
```

Note: Replace new_user with the actual username.

- Create a role, or an existing role can be used

```
None
CREATE ROLE read_write_role;
```

Here, read_write_role is the role name.

- Grant permission to the role

```
None
SELECT 'GRANT SELECT, UPDATE, DELETE ON ' || DatabaseName || ' TO read_write_role;'
FROM DBC.Databases
```

- Assign the role to the new user

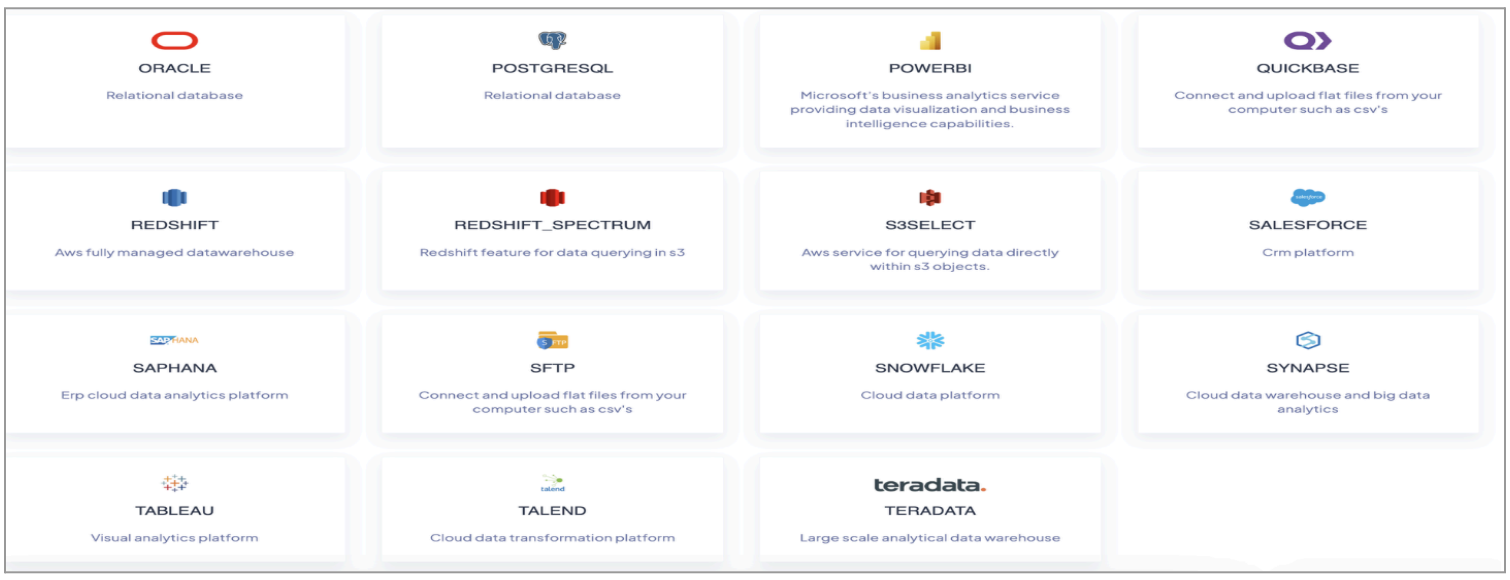
```
None
GRANT read_write_role TO new_user;
```

Connect to Teradata

Follow the steps below to connect to Teradata:

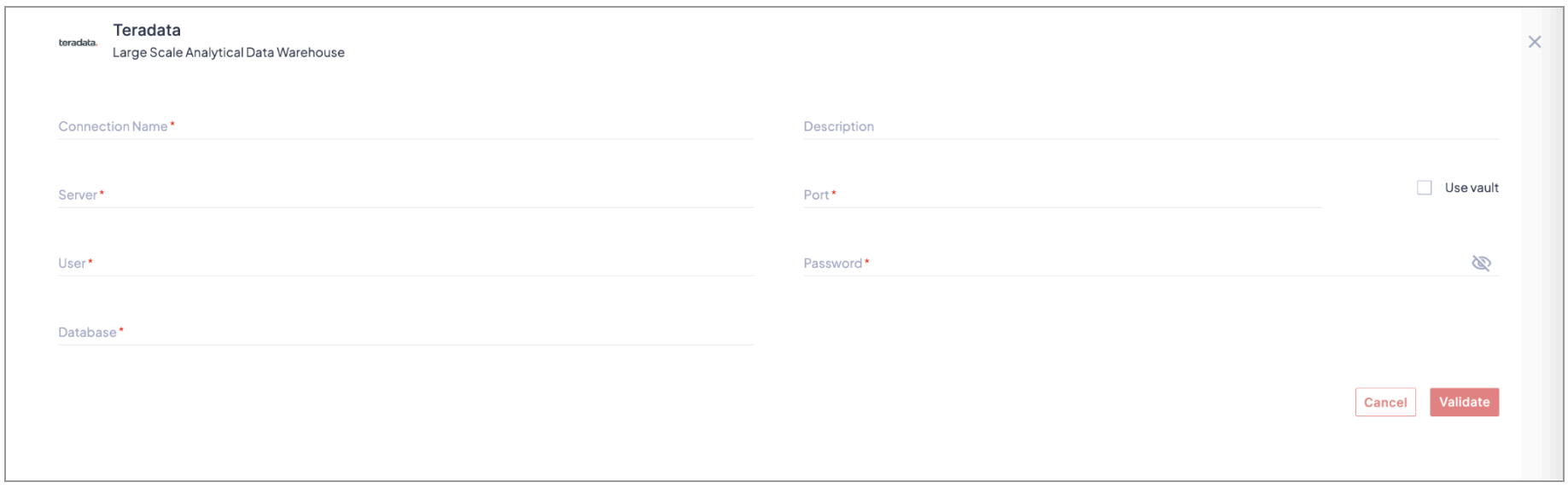
Step 1: Navigate to Settings → Connect → Source

Step 2: Click on the “+“ icon



Step 3: Choose Teradata and provide the following details:

- Connection Name
- Server - Server/host URL of the Teradata instance
- Port - Port number of the Teradata instance
- Username - username in the database
- Password - password for the username
- Database - Database Name to be connected to



Note: Teradata only has a database and does not have a schema

Step 4: Click on Validate, and once validated, click on “Save“

Step 5: Select the required assets and click on “Connect“. Once connected the user will be taken to the asset details page.

Redshift

Amazon Redshift is a fully managed, cloud-based data warehousing solution that provides fast query performance on large datasets and is designed to handle petabyte-scale workloads. Quest DQ allows users to connect to Amazon Redshift and monitor and observe data quality across Redshift assets. Here is a link to the Amazon Redshift Documentation: https://docs.aws.amazon.com/redshift/index.html

Prerequisites

Whitelist IP

If your organization uses a whitelist to manage Redshift access, Quest DQ will only access your Redshift through IP. For assistance on whitelisting, kindly reach out to the Support team.

Account Setup

Create a User and Provide Access

To connect to Amazon Redshift, create a user and provide SELECT access to the required schemas.

To create a user and provide access to tables in Amazon Redshift, you can follow these steps:

- Log in to the AWS Management Console and navigate to the Amazon Redshift dashboard.
- Click on the "Clusters" tab and select the cluster for which you want to create a user.
- In the cluster details page, click on the "Query editor" button to open the query editor.
- In the query editor, execute the following SQL command to create a new user:

```
None
CREATE USER username PASSWORD 'password';
```

- Replace username with the name of the new user and password with a strong password
- Grant the necessary permissions to the new user. For example, to grant SELECT access to a table called "my_table", execute the following SQL command:

```
None
GRANT SELECT ON my_table TO username;
```

- Replace the username with the name of the new user and the password with a strong password.
- Save the changes and close the query editor.

The new user is now created and has access to the specified tables in Amazon Redshift.

To create a user and provide necessary access to the user in Amazon Redshift, you can execute the following SQL script:

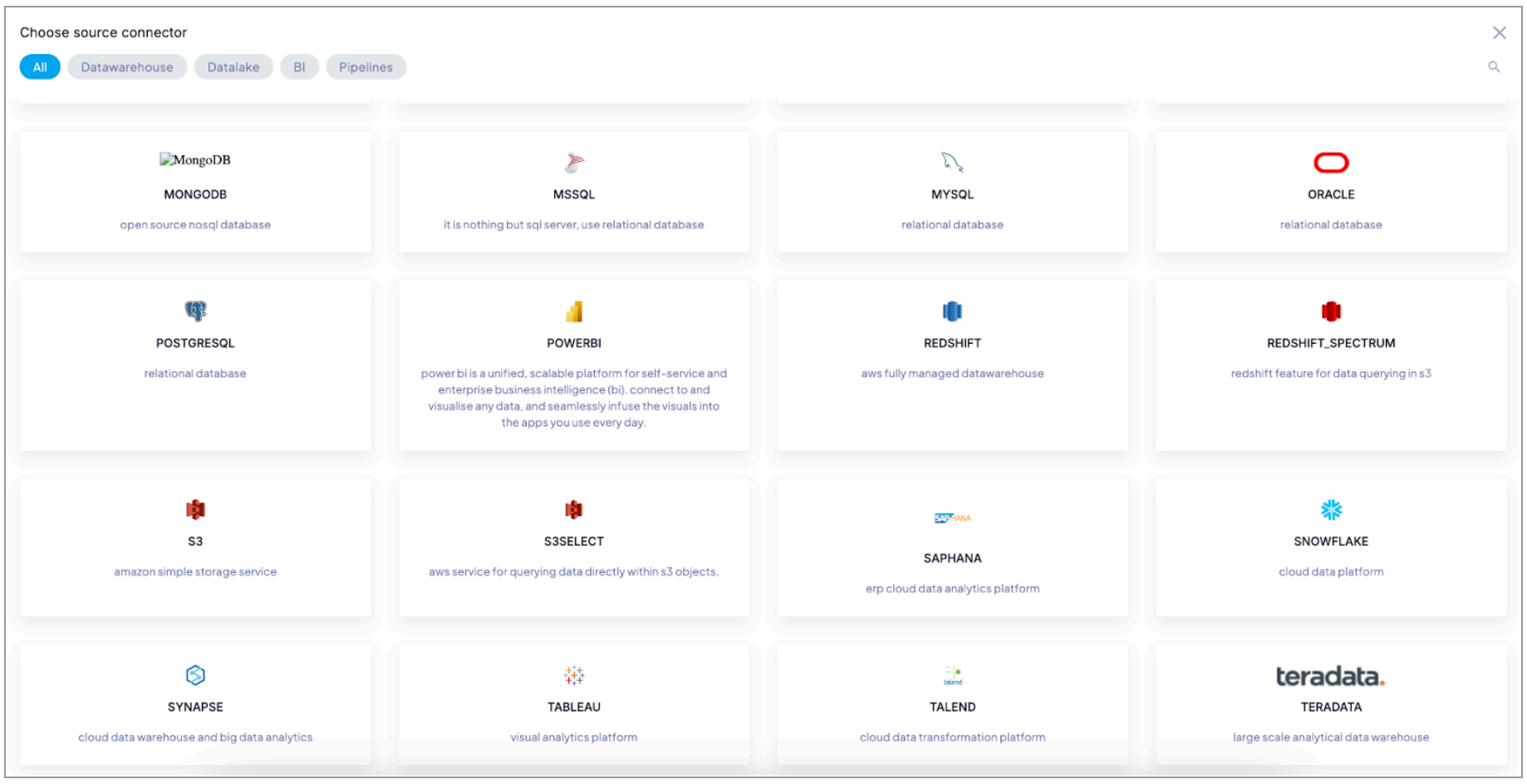
```
None
CREATE USER username PASSWORD 'password';
GRANT USAGE ON SCHEMA public TO username;
GRANT SELECT ON ALL TABLES IN SCHEMA public TO username;

#In case of views/temp tables
GRANT CREATE ON SCHEMA your_schema TO your_user;
```

This script creates a new user, grants usage privileges to the "public" schema, and grants all privileges to all tables and sequences in the "public" schema. Update the schema information to provide the user access to different schemas

Connect to Amazon Redshift

Step 1: Navigate to **Settings** > **Sources**

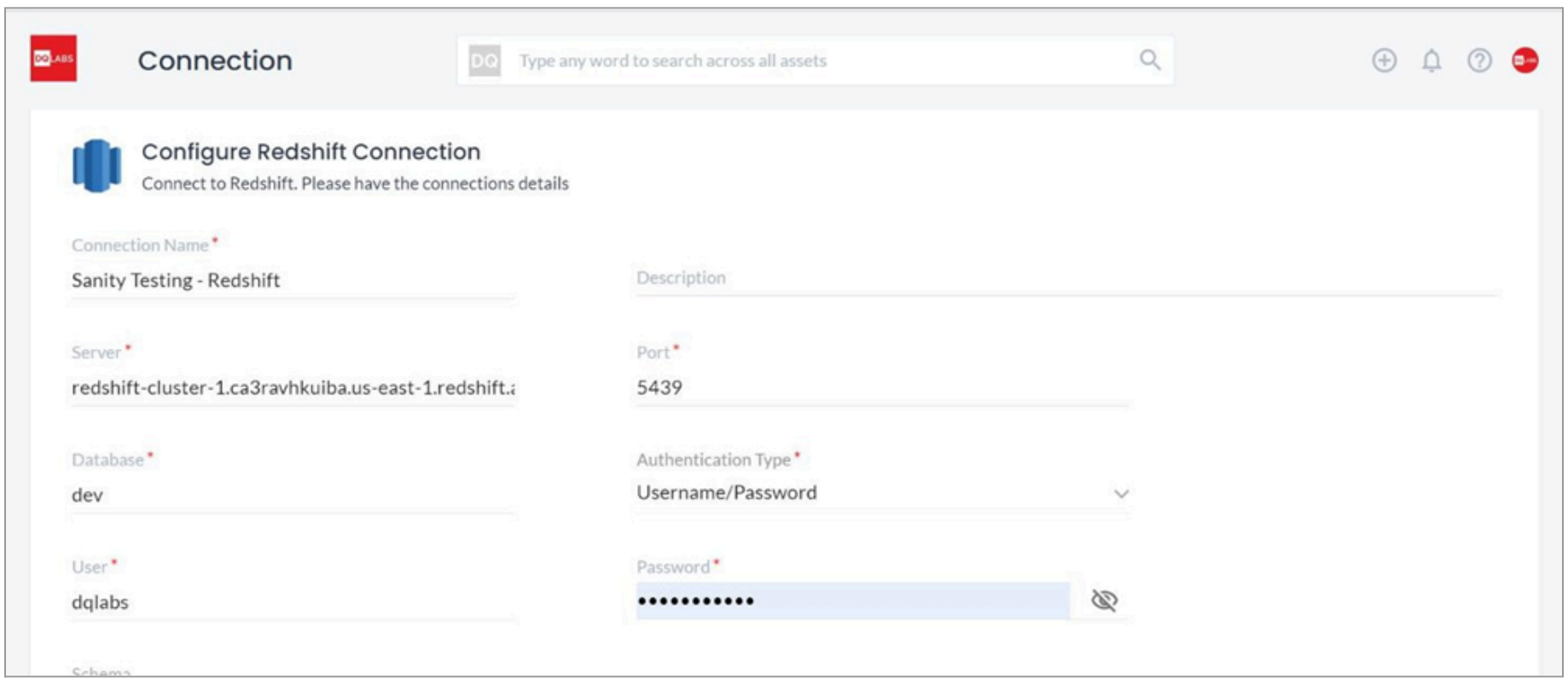
Step 2: Go to the + icon in the top right-hand corner of the screen



Quest DQ allows the user to connect to Redshift using Secret Manager or by using a username and password. Please follow the steps below to create a connection using username and password:

Step 3: Click on Redshift and provide the following details:

- Connection Name
- Description
- Server
- Port
- Database
- Authentication Type - Select Username and Password
- Username
- Password



Step 4: Validate it

Step 5: Once validated, click "Connect" to choose the desired tables and Queries

PostgreSQL

Postgres, also known as PostgreSQL, is a powerful, open-source object-relational database management system. It is one of the most popular and widely used databases in the world, known for its robustness, flexibility, and reliability.

Postgres supports a wide range of features, including transactions, concurrency, and replication. It also provides advanced data types such as arrays, JSON, and XML. Postgres supports SQL, the standard language for accessing and manipulating relational databases, as well as stored procedures and triggers, which allow developers to create custom logic within the database.

Prerequisites

Whitelist IP

If your organization uses a whitelist to manage Postgresql access, Quest DQ will only access your Postgresql through IP. For assistance on whitelisting, kindly reach out to the Support team.

Account Setup

Following are the steps to create a service user and provide access to a database in Postgres:

1. Connect to your Postgres database using an account with administrative privileges.

2. Create the service user using the CREATE USER statement, specifying the desired username and password:

```
None
CREATE USER myserviceuser WITH PASSWORD 'mypassword';
```

3. Create the database that the service user will access, if it does not already exist:

```
None
CREATE DATABASE mydatabase;
```

4. Grant the service user permission to access the database using the GRANT statement:

```
None
GRANT ALL PRIVILEGES ON DATABASE mydatabase TO myserviceuser;
```

This statement grants the service user all privileges on the specified database, allowing it to create tables, insert data, and perform other operations.

Optionally, you can restrict the service user's access to specific schemas within the database, by specifying the schema name in the GRANT statement:

```
None
GRANT ALL PRIVILEGES ON SCHEMA myschema TO myserviceuser;
```

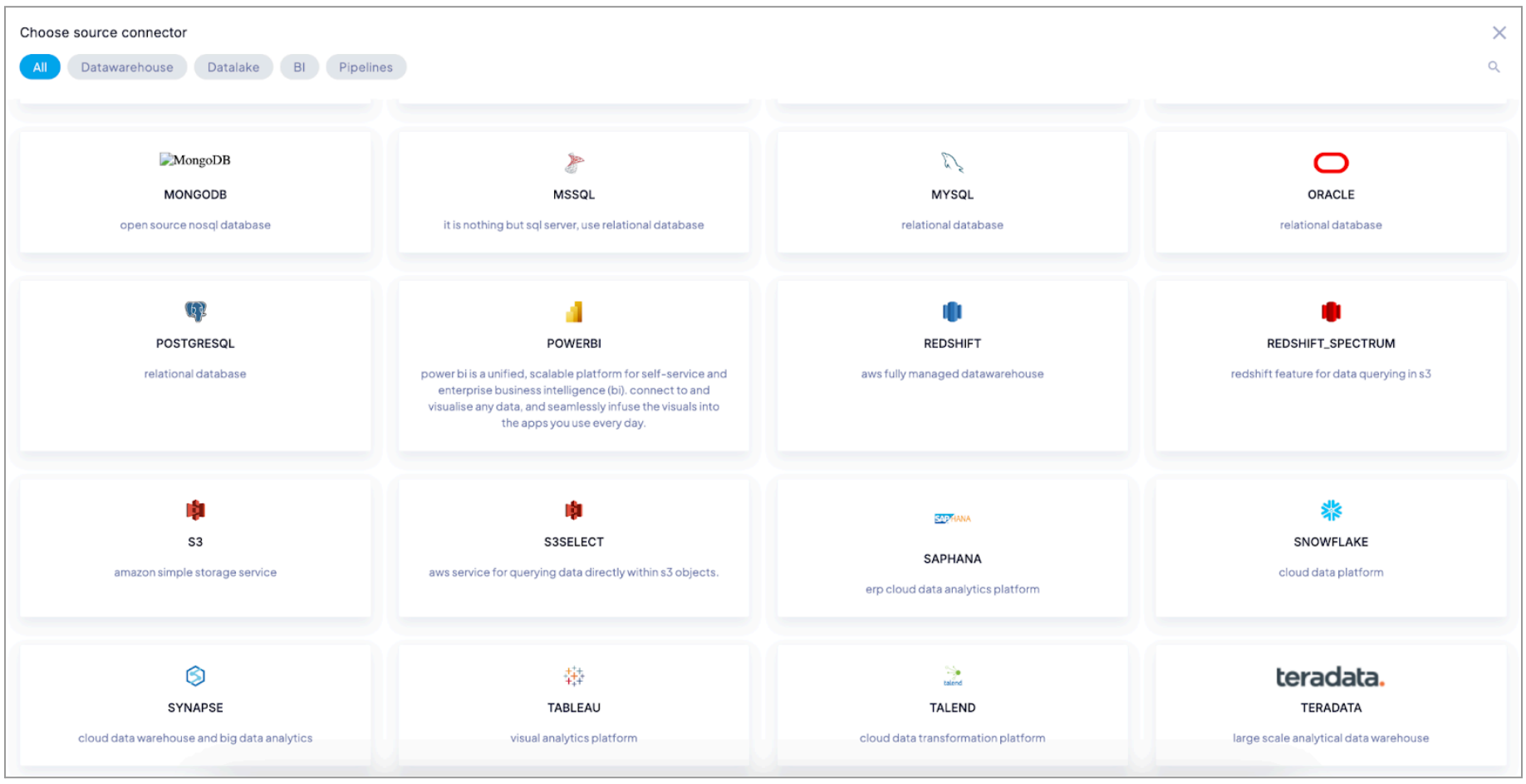
This statement grants the service user all privileges on the specified schema within the database.

Once you have completed these steps, the service user should be able to connect to the database and perform the authorized operations using the specified credentials.

Connect to Postgres

Step 1: Navigate to **Settings** > **Sources**

Step 2: Go to the + icon in the top right-hand corner of the screen



Step 3: Click on Postgresql and provide the following details

- Connection Name
- Description
- Server
- Port
- Server name
- Password
- Database
- Schema

Step 4: Validate it

Step 5: Once the connection is established, select the required schemas from the list of all available schemas and connect.

Step 6: From the list of assets on the asset list page, select the asset that has to be configured.

Oracle

Oracle Database is a relational database management system (RDBMS) developed by Oracle Corporation. It is one of the most widely used enterprise-level database systems in the world, designed to provide efficient, reliable, and secure storage and management of large amounts of data.

Oracle Database offers a variety of features and tools for data management, including support for SQL and PL/SQL programming languages, advanced security features, backup and recovery capabilities, high availability options, and scalability for handling growing data volumes.

Prerequisites

Whitelist IP

If your organization uses a whitelist to manage Oracle access, Quest DQ will only access your Oracle through IP. For assistance on whitelisting, kindly reach out to the Support team.

Account Setup

To create a service user in Oracle and provide access to all databases using SQL, you can follow these steps as an admin user:

- Connect to the Oracle database as a user with administrative privileges, such as the SYSTEM user
- Create a new user account using the CREATE USER statement, specifying a username and password for the new user. For example:

```
None
CREATE USER myserviceuser IDENTIFIED BY mypassword;
```

- Grant any necessary privileges to the new user, such as the ability to connect to the database, create objects, and execute procedures. To grant access to all databases, you can grant the CREATE SESSION privilege to the user at the system level. For example:

```
None
GRANT CREATE SESSION TO myserviceuser;
```

- To grant the user access to all databases, you can grant the CREATE ANY CONTEXT privilege. This privilege allows the user to create a context for any database in the system. For example:

```
None
GRANT CREATE ANY CONTEXT TO myserviceuser;
```

- You may also need to grant the user additional privileges, such as the ability to create and modify objects, particularly if they plan to use the reporting functionality. This permission is necessary for a dedicated schema where failed records are pushed. This permission can be skipped if Push down metrics are not required.

```
None
GRANT CREATE TABLE, ALTER ANY TABLE on <schema> TO myserviceuser;
```

- Finally, you can test the new user account by connecting to any database using the new username and password. For example, using SQL*Plus:

```
None
myserviceuser/mypassword@anydatabase
```

This will allow you to create a service user in Oracle and provide access to all databases using SQL

Create a Service name (If Authentication type is Service name)

Step 1: Use SQL*Plus or any other Oracle database client to log in as a user with the necessary privileges.

Step 2: Use the DBMS_SERVICE package to create a service. For example:

```
None
EXEC DBMS_SERVICE.CREATE_SERVICE(
  service_name => 'my_service',
```

```
  network_name => 'my_service'
);
```

Step 3: After creating the service, you need to start it. This can be done using the DBMS_SERVICE package as well:

```
None
EXEC DBMS_SERVICE.START_SERVICE( service_name => 'my_service' );
```

Step 4: Update the listener.ora file to include the new service. This file is usually located in the $ORACLE_HOME/network/admin directory. Add a description for your new service:

```
None
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (GLOBAL_DBNAME = my_service)
      (SID_NAME = ORCL)
      (ORACLE_HOME = /u01/app/oracle/product/12.2.0/dbhome_1)
       )
  )
```

Step 5: Restart the listener to apply the changes:

```
None
lsnrctl reload
```

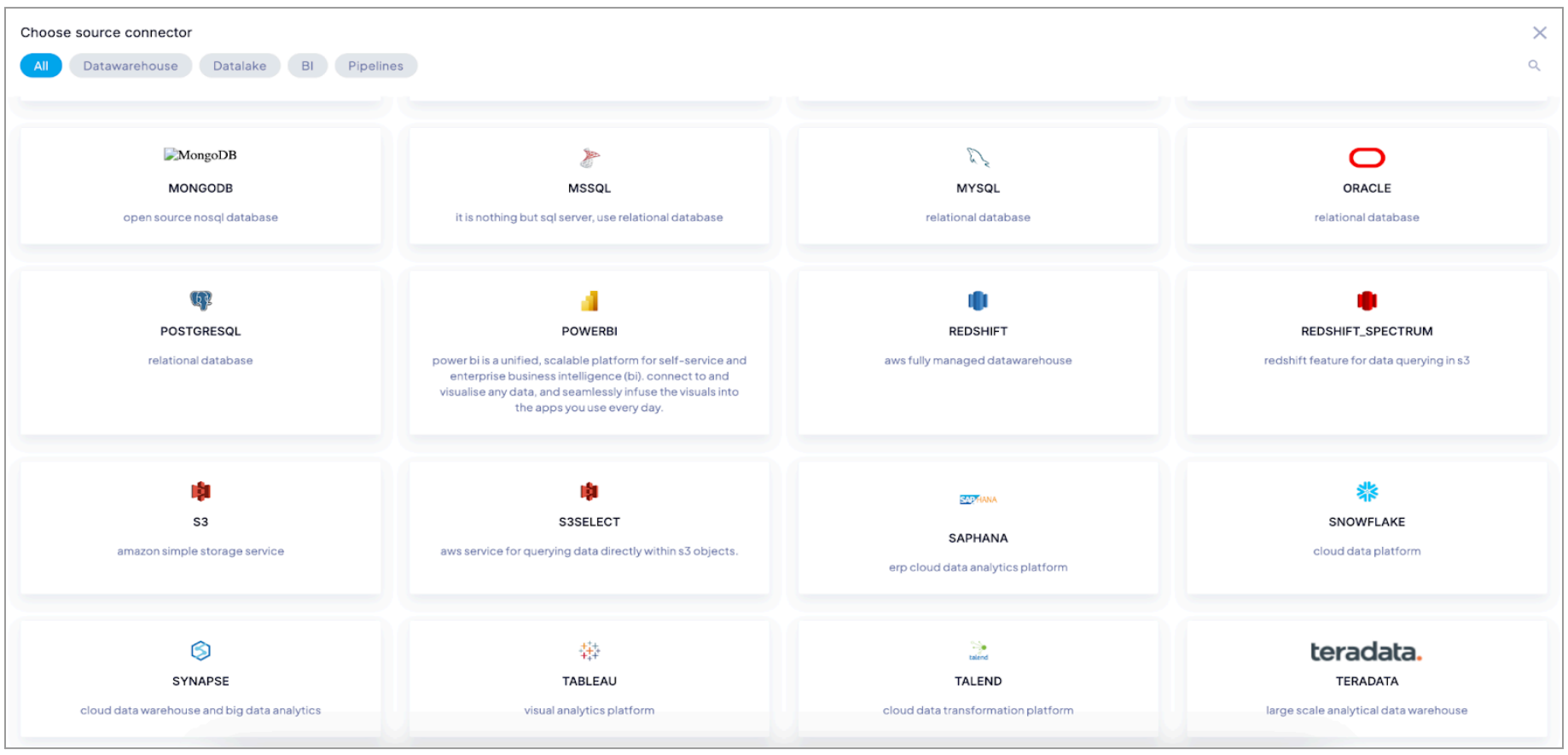
Step 6: Update the tnsnames.ora file on the client side to include the new service name:

```
None
MY_SERVICE =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = your_host_name)(PORT = 1521))
       (CONNECT_DATA =
      (SERVICE_NAME = my_service)
       )
  )
```

Connect to Oracle

Step 1: Navigate to **Settings** > **Sources**

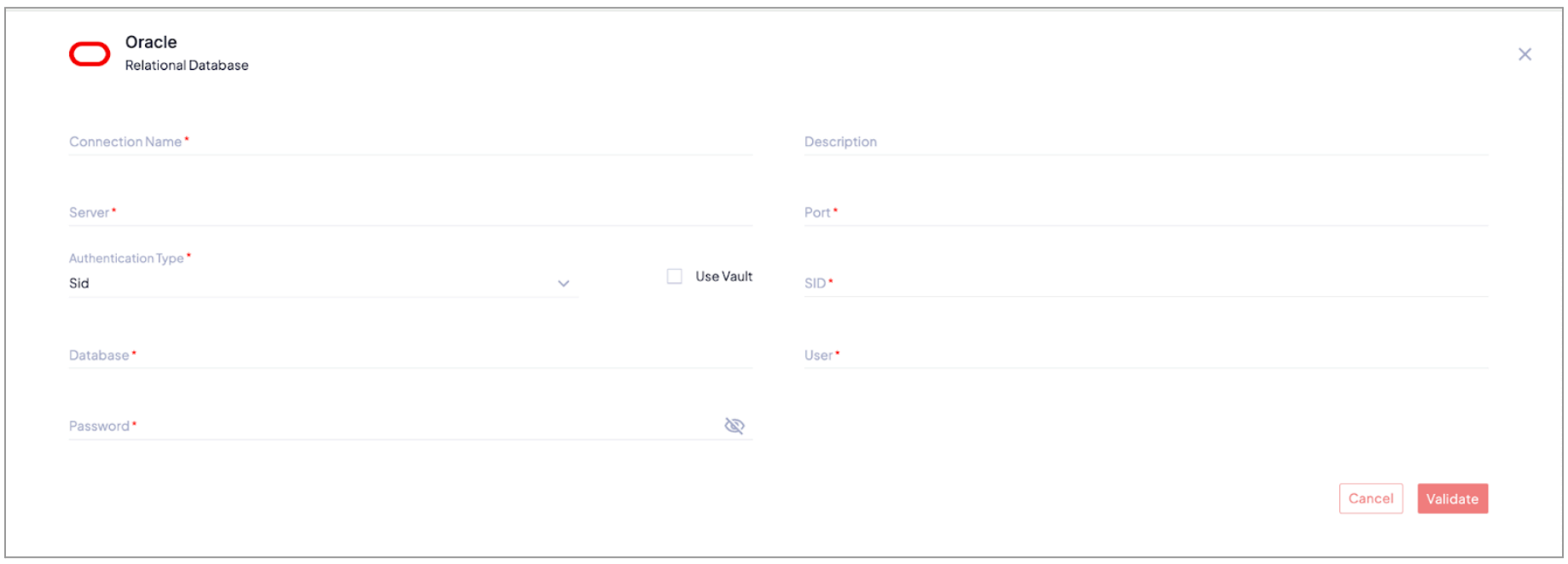
Step 2: Go to the + icon in the top right-hand corner of the screen



Step 3: Click on Oracle and provide the following details

- Connection name (User Preference)

- Description (Can be used to describe the connection and its purpose)
- Authentication type: Sid or Service name
- Server
- Port
- Database
- SID/Service name
- User
- Password



Step 4: Validate it

Step 5: Once the connection is established, select the required schemas from the list of all available schemas and connect.

Step 6: From the list of assets on the asset list page, select the asset that has to be configured.

MSSQL

Microsoft SQL Server (MSSQL Server) is a relational database management system (RDBMS) developed by Microsoft. It is designed to store, manage, and retrieve data as requested by other software applications. MSSQL Server supports a wide range of applications in various environments, including enterprise-level data processing, e-commerce platforms, and business intelligence applications.

Prerequisites

The following prerequisites must be met in order to establish the connection between MSSQL and Quest DQ

Whitelist IP

If your organization uses a whitelist to manage MSSQL access, Quest DQ will only access your MSSQL through IP. For assistance on whitelisting, kindly reach out to the support team.

Account Setup

1. Service Account Set-up
2. Service Account Access to the following:
    a. SELECT ACCESS ON DATABASE
    b. CREATE SCHEMA access to create temp tables for Query-based assets and Views
    c. Adding the Server Account to the group using SP_ADDROLEMEMBER
    d. SELECT ACCESS ON THE SCHEMA
    e. CREATE AND ALTER PERMISSIONS ON VIEWS IN SCHEMA (Only for Export failed Rows Reporting)
    f. SELECT ACCESS ON SYS TABLES:
        i. SCHEMAS
        ii. TABLES
        iii. OBJECTS

Script for Execution of the grants listed in the above section

```
None
use [master]
GO
CREATE LOGIN <login_name> WITH PASSWORD = '<enterStrongPasswordHere>';
GoUSE <MY_DATABASE>
GO
CREATE USER <user_name> FOR LOGIN <login_name>
GOuse [master]
GO
GRANT VIEW SERVER STATE TO [<user_name>]USE <MY_DATABASE>
GO
EXEC sp_addrolemember N'db_datareader', N'<user_name>'
GoUSE <MY_DATABASE>
GO

#Used in the events of creating a default schema where temp tables will be created

GRANT CREATE SCHEMA ON DATABASE :: [DatabaseName]   TO [user_name]
GoUSE <MY_DATABASE>
GO
GRANT SELECT ON SCHEMA :: [YourSchema] TO <user_name>
GoUSE <MY_DATABASE>
GO
#Required for Push Down Metrics
GRANT CREATE VIEW TO <user_name>
GRANT ALTER,VIEW DEFINITION ON SCHEMA::[YourSchema] TO [<user_name>]
Go
```

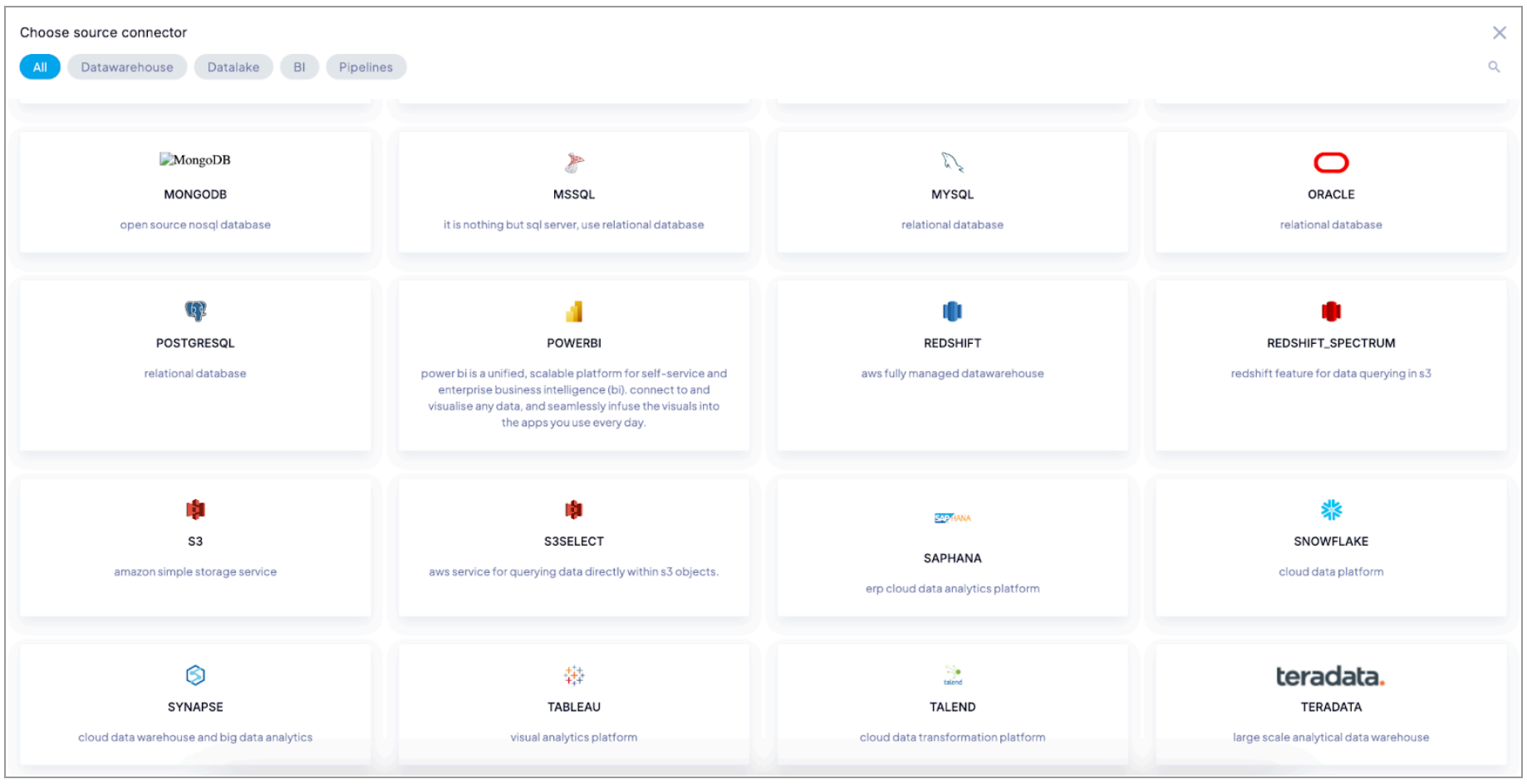
Support for Deep Profiling :

- Create a new schema “DQLABS” in the same database and same connection.
- Create the following function in the DQLABS schema, and the account used in the connection should have access to execute it dqlabs_deep_profile_pattern.sql

Connect to MSSQL

Step 1: Navigate to Settings > Sources

Step 2: Go to the + icon in the top right-hand corner of the screen



Step 3: Click on MSSQL and provide the following details

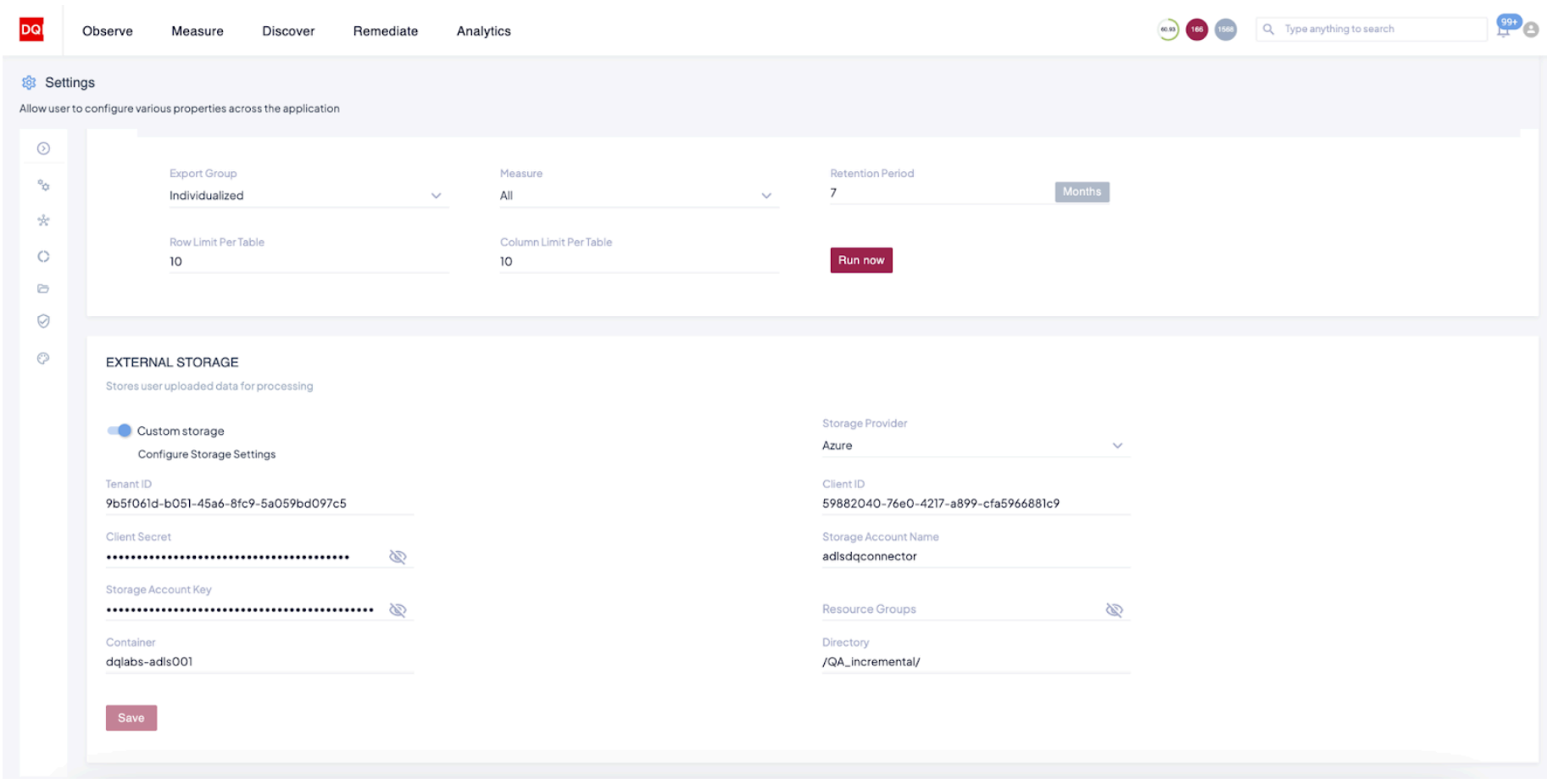
Field	**Description**
Connection Name	Name of the connection object
Description	Description of the connection object
Server	The IP address of the MSSQL server
Port	The port number to the server
Authentication Type	Select between username and password or Windows authentication
User	The username for the SQL Server
Password	The password of the provided user
Schema	Select the required schemas from the list of available schemas

Step 4: Once validated, click "Connect" to choose the desired tables and Queries

File

Quest DQ allows the user to connect to the file stored in local storage to be configured as assets in the Quest DQ platform. Once configured, the OOB measures can be applied to the asset, and profiling can be run on top of it.

The users can setup external storage under settings → Configuration → External Storage. The user can use either AWS or Azure as external storage to process the file. The files uploaded in the UI are stored in the configured external storage and then used for processing



Quest DQ leverages the Spark clusters to create iceberg tables for the connected files, and the measure queries will be executed on the iceberg table created to get the metadata information. Once all the measures are executed and the metadata is extracted, then the iceberg table will be dropped from the database.

Currently, the following measures are supported in File connectors

- OOB Measures

A user can provide the folder or file path, which creates the File connection, and each connection can create only one asset in Quest DQ with the user-specified column names and datatypes. Currently, the following file types are supported

- CSV

Prerequisites

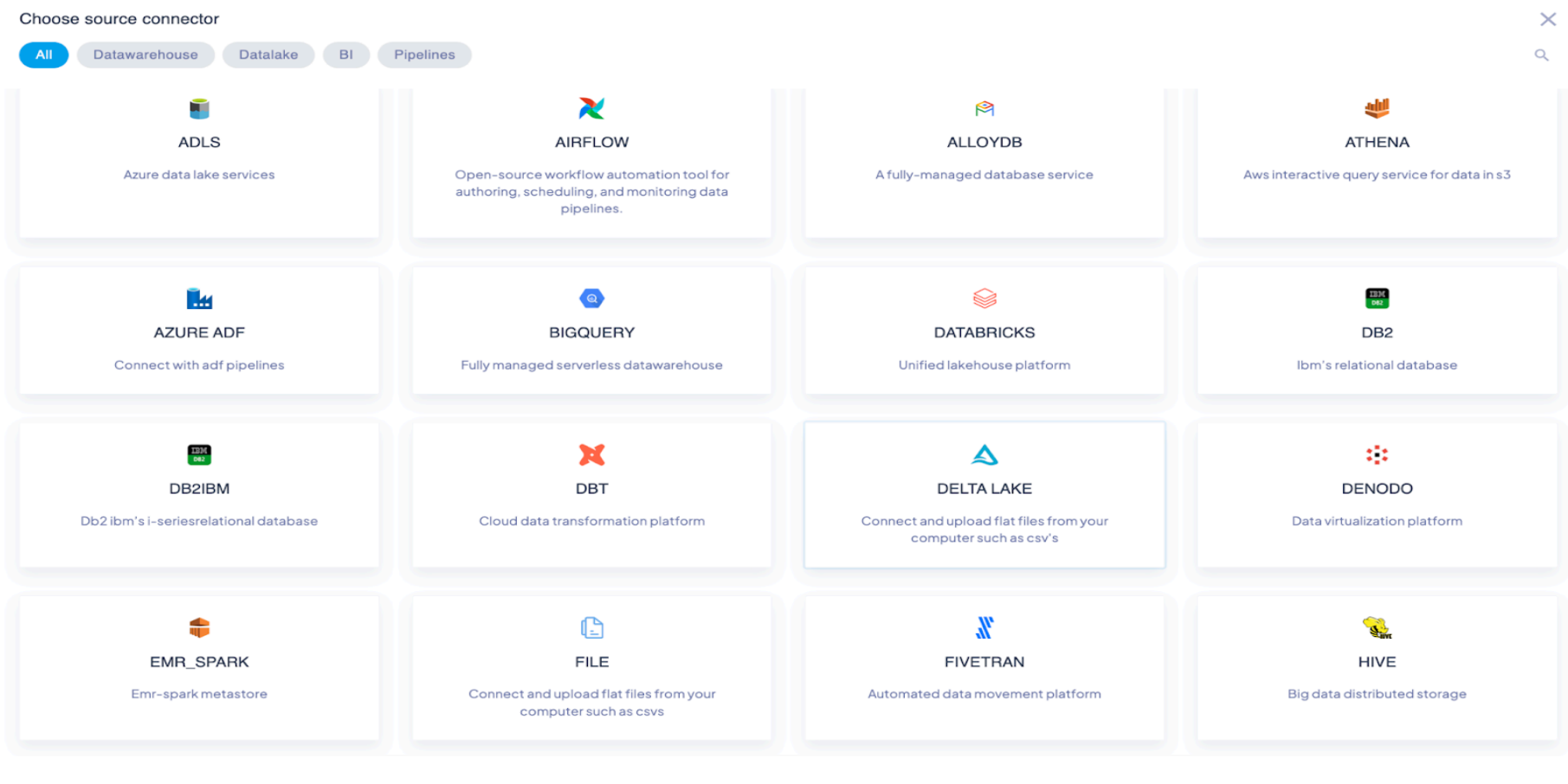
Whitelist IP External Source

If your organization uses a whitelist to manage Azure ADLS/AWS, Quest DQ will only access your ADLS through IP. For assistance on whitelisting, kindly write to customersupport@dqlabs.ai

Connect to File Connector

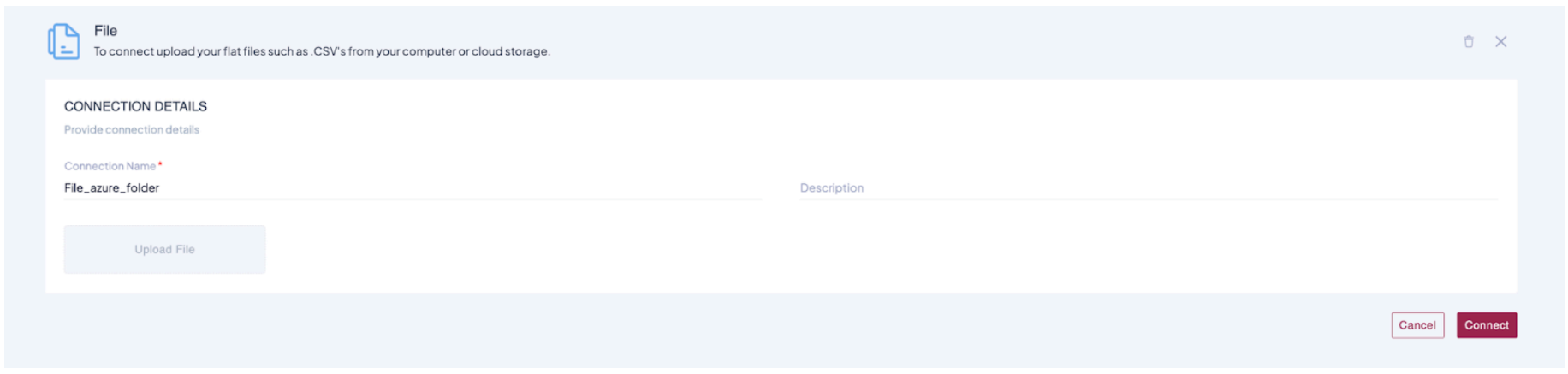
Follow the steps below to connect to File from Quest DQ and create assets.

Step 1: Navigate to Settings → Connect → Source and click on the “+“ icon

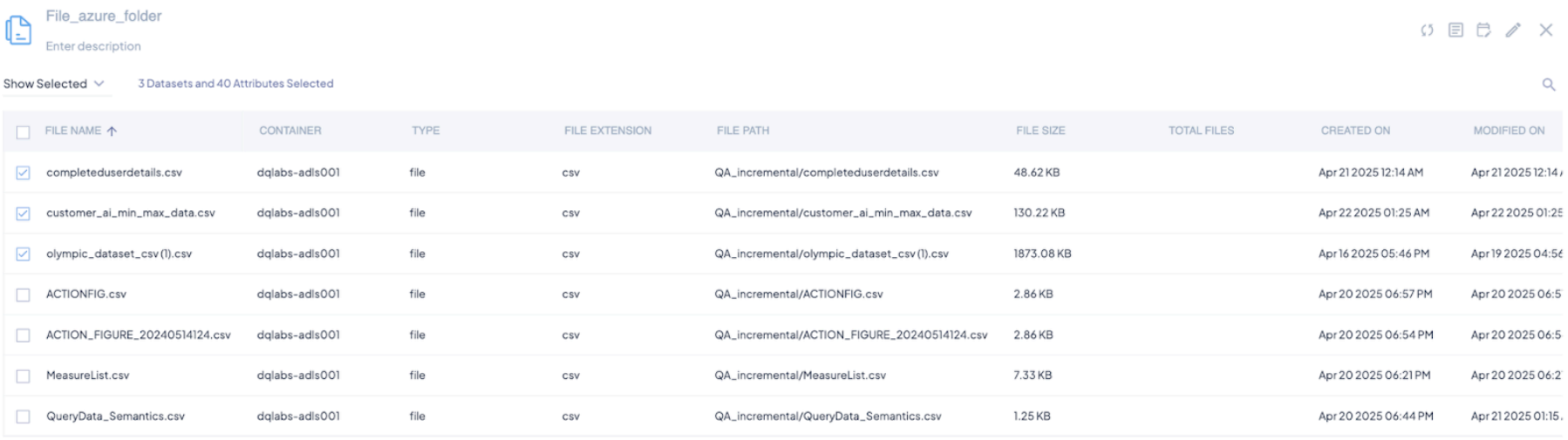


Step 2: Click on File and provide the following details

- Connection Details
  - Connection Name
  - Description
- Upload File
- The admin/privileged user will be able to add multiple assets based on the file uploaded using the above configuration



Step 3: After providing the details, click on connect, and once validation is complete, the user can connect to the assets.



The connected assets will be listed on the Quest DQ platform.

Limitations/Constraints:

- Incremental is not supported
- Partitioning of the file is not supported
- Change of datatype is not supported

Salesforce Data Cloud

Salesforce Data Cloud (previously known as Salesforce CDP – Customer Data Platform) is Salesforce's platform for unifying, managing, and activating customer data across multiple systems in real time. It's designed to help organizations create a single, 360-degree view of each customer to enable personalized marketing, sales, and service experiences. Quest DQ allows users to connect to Salesforce data cloud and bring in data lake objects, data models, and transformations into Quest DQ for observability

Pre-requisites

Whitelisting

If your organization uses a whitelist to manage Salesforce Data Cloud access, reach out to customersupport@dqlabs.ai to set up the whitelisting.

Account Access

To create a Client ID (Consumer Key) and Client Secret (Consumer Secret) for Salesforce Data Cloud, follow these steps:

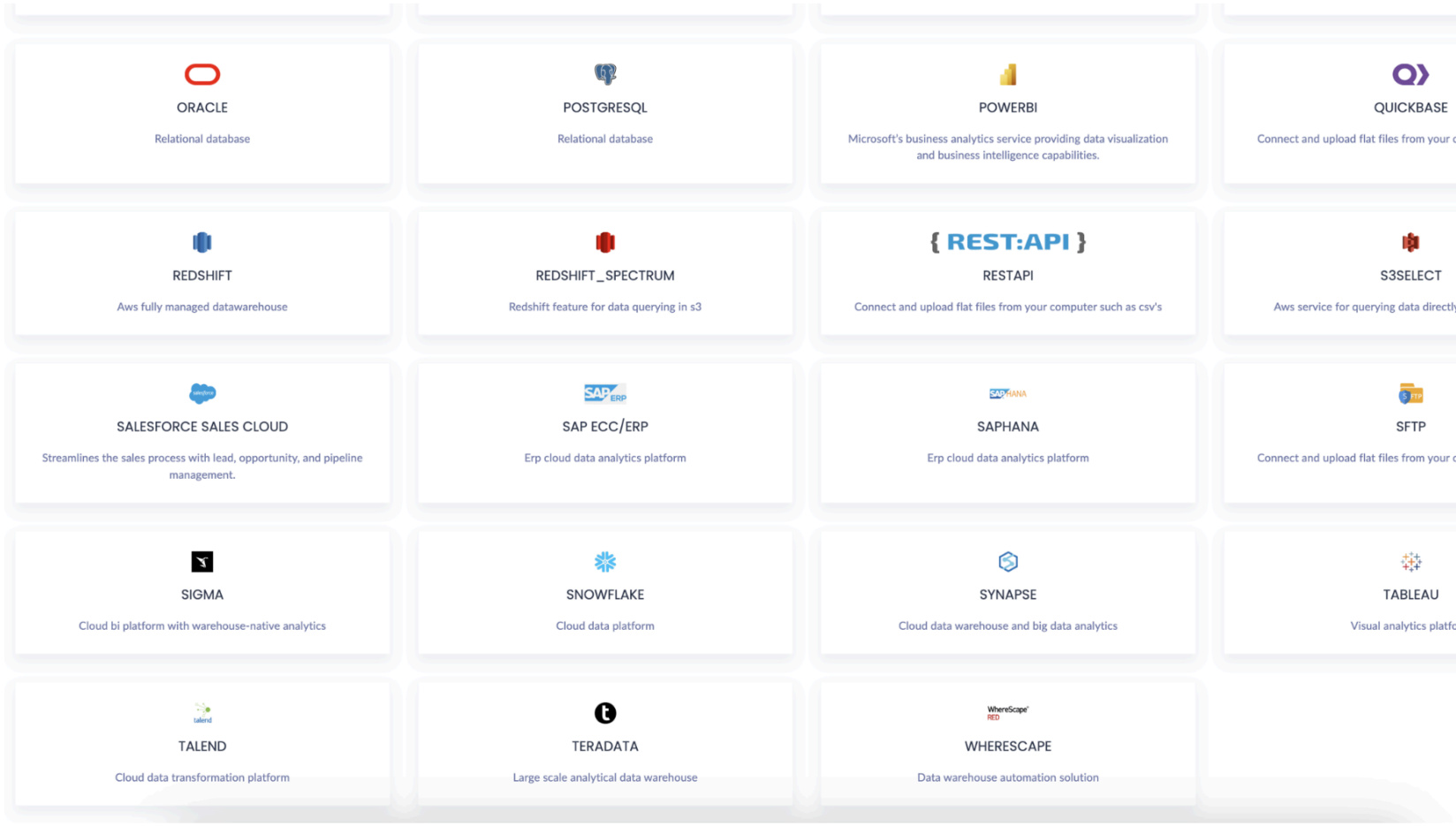
- **Log in to Salesforce:** Access your Salesforce instance with an administrator account.
- **Navigate to Setup:** Click the gear icon in the top right corner and select "Setup."
- **Go to App Manager:** In the Quick Find box, type "App Manager" and select it under "Apps."
- **Create a New Connected App:** Click "New Connected App" in the App Manager.
- **Configure Basic Information:**
  - Provide a "Connected App Name," "API Name," and "Contact Email."
- **Enable OAuth Settings:**
  - Under the "API (Enable OAuth Settings)" section, check "Enable OAuth Settings."
  - Specify a "Callback URL." This URL is where the user's browser is redirected after successful authorization.
  - Select the necessary "OAuth Scopes." For Data Cloud integration, you will likely need scopes such as "Access and manage your data (api)" and "Perform requests on your behalf at any time (refresh_token, offline_access)." You might also need Data Cloud-specific scopes, depending on your integration needs (e.g., cdp_query_api, cdp_profile_api).
- **Save the Connected App:** Click "Save."
- **Obtain Consumer Key and Secret:**
  - After saving, you will be redirected to the Connected App detail page.
  - Click "Manage Consumer Details" to reveal the "Consumer Key" (Client ID) and "Consumer Secret" (Client Secret).
- **Important**: Copy these values and store them securely, as the Consumer Secret will only be displayed once.
  - These credentials (Consumer Key and Consumer Secret) can then be used to authenticate with Salesforce and subsequently with Salesforce Data Cloud APIs using OAuth 2.0 flows.
  - Provide permission for the client credential flow in the settings.
  - For tables, every data lake object needs to run needs to be added to any one data space; if it is not mapped to a data space, queries will not run, so it must map to a data space

Connect to Salesforce Data Cloud

Follow the steps below to connect to Salesforce Data Cloud:

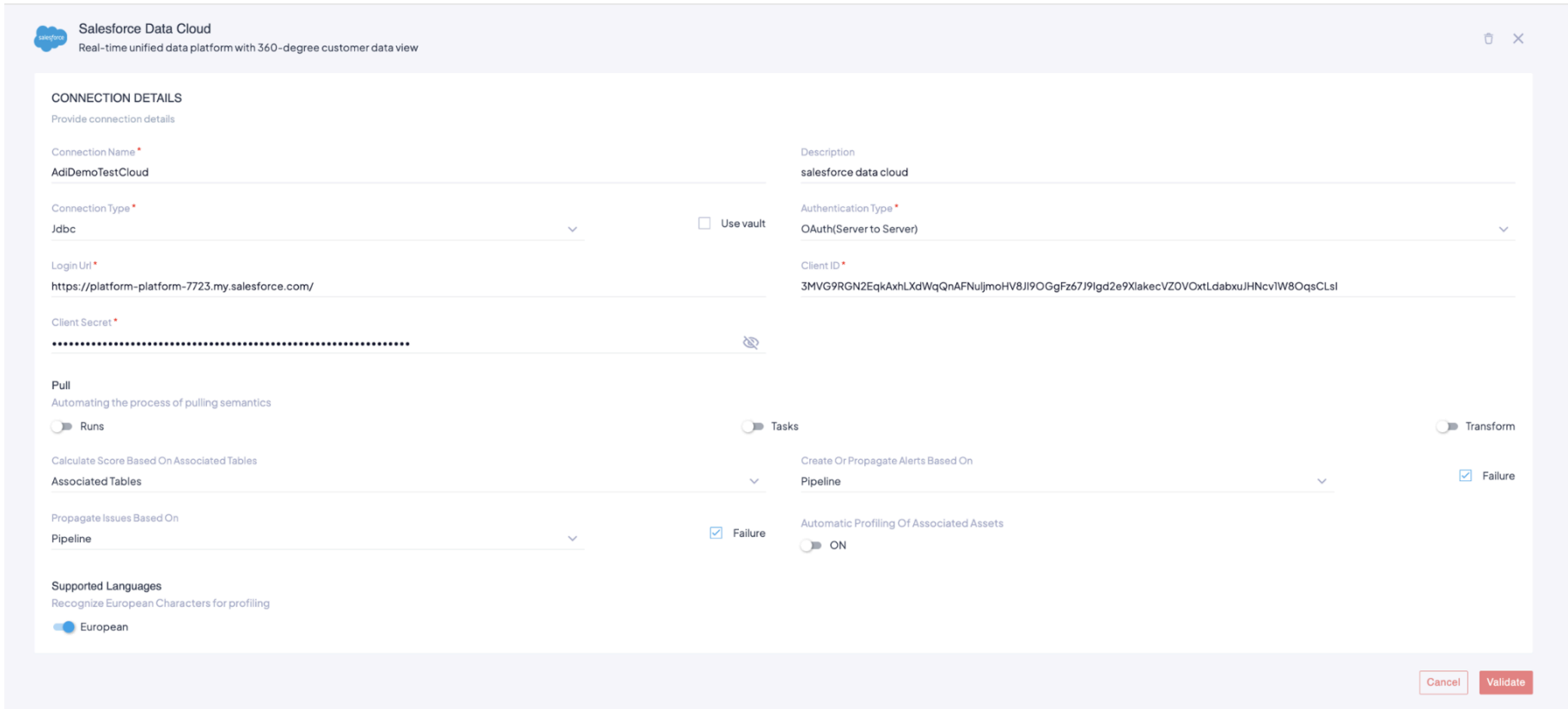
Step 1: Navigate to Settings → Connect → Source

Step 2: Click on the "+" icon



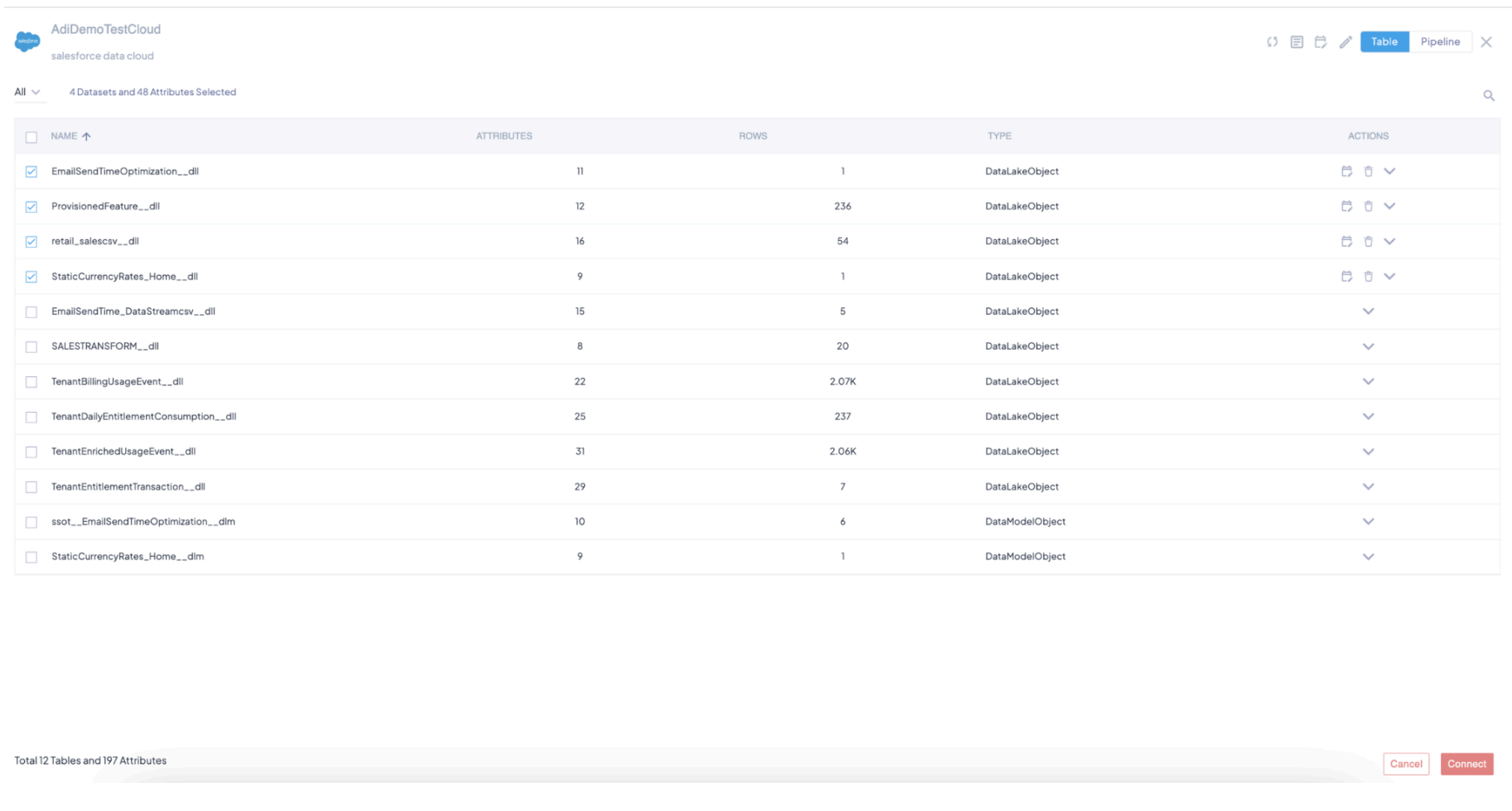
Step 3: Click on Salesforce data cloud and provide the following details:

Field / Option	Description
Connection Name*	A required name for identifying your connection (e.g., AdiDemoTestCloud)
Connection Type*	Type of connection protocol (e.g., Jdbc)
Description	Optional text describing the connection purpose or context (e.g., Salesforce data cloud)
Login Url*	URL to authenticate against the Salesforce platform
Authentication Type*	Specifies the OAuth type used (e.g., OAuth(Server to Server))
Client ID*	OAuth client ID provided for authentication
Client Secret	Secret key/password used for authentication (hidden by default)
Use Vault	Option to store and retrieve sensitive credentials securely via a vault
Pull - Runs	Toggle to automate pulling pipeline runs
Pull - Tasks	Toggle to automate pulling pipeline tasks
Pull - Transform	Toggle to automate pulling pipeline transformations
Calculate Score Based On	Select what scoring calculation is based on
Propagate Issues Based On	Select on which criteria issues should propagate
Create or Propagate Alerts Based On	Select on which criteria alerts should be created
Failure Checkbox	When checked, failures will be tracked and propagated as issues
Automatic Profiling Of Associated Assets	Toggle to enable or disable automatic data profiling of linked assets
Supported Languages	Option to recognize specific character sets, e.g., European, for profiling



Step 4: Once connected, the user will be able to view the list of transformations in Salesforce Data Cloud

Step 5: Select the required transformations and click on connect.



Once connected, the admin/privileged user will be able to select Table and Pipelines. Tables include the following objects in the Salesforce data cloud:

- Data lake objects
- Data Models

Pipelines include the following objects:

- Data Transformations

The user can select the objects and click on connect, and the user will be redirected to the asset detail page

Salesforce Marketing CRM

Salesforce Marketing CRM refers to Salesforce Marketing Cloud, which is Salesforce's platform specifically built for marketing automation, customer engagement, and personalized marketing campaigns across multiple channels like email, SMS, social media, web, and advertising.

Quest DQ allows users to connect to Salesforce Marketing CRM and bring objects into Quest DQ and then apply profiling on top of it for data quality.

Prerequisites

1. **SFMC Account** – A valid Marketing Cloud account with API access.
2. **Authentication Method**

- **OAuth 2.0** (Preferred) – Uses Client ID + Secret

3. **User Permissions**

- **API User Role** – The user must have API permissions in SFMC.
- **Data Access Permissions** – Ensure access to:
    - Data Extensions (if reading/writing to DEs)
    - Email Studio (if sending emails)
    - Automation Studio (if triggering automations)

4. **Network & Security Requirements**

- **IP Whitelisting** – If SFMC has IP restrictions, allow the connector's IP.
- **TLS 1.2+** – Required for secure API calls.

Steps to set up

- Log in to your Salesforce Marketing Cloud account.
- In the top right of the page, click your username and select **Setup**.
- On the navigation menu, go to **Platform Tools > Apps > Installed Packages**.
- Click **New** to create a new package.
- Enter a name.
- Click **Save**.
- Select your new package from the list of packages to go to its detail page.
- In the **Components** section, select **Add Component**.
- Select **API Integration** as your component type and click **Next**.
- Select **Server-to-Server** as your integration type.
- Give **Read** permissions to the following:
    - Campaign
    - Documents and Images
    - Email
    - Journeys
    - List and Subscribers
    - OTT
    - Push
    - Saved Content
    - SMS
    - Social
    - Web
- Give **Read** and **Write** permissions to the following:
    - Data Extensions
- Give **Read**, **Write**, and **Execute** permissions to the following:
    - Automations

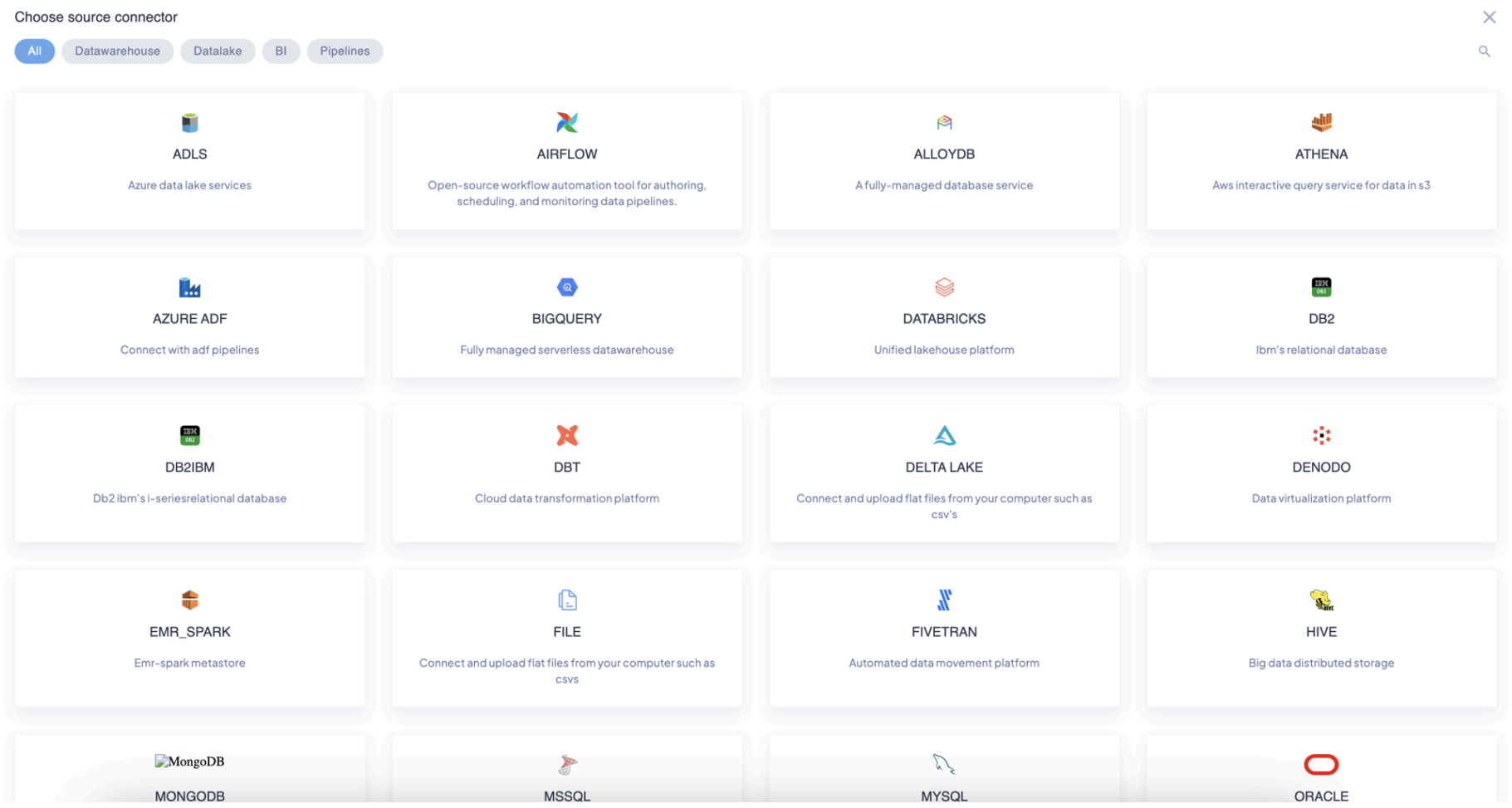
The following is the summary of supported features in Salesforce Marketing CRM

Category	Status	Notes
Reliability	✔ Supported	All standard reliability metrics are implemented
Profiling	✔ Supported	Exceptions noted for certain Text data type metrics
Statistics	✔ Supported	Includes core distribution and aggregation measures
Custom Measures	✔ Supported	Lookup-based checks are not available for file sources
Usage Query	❌ Not Supported	Salesforce does not return required query usage data
Export Failed Rows	❌ Not Supported	Requires table creation, which Salesforce connector does not support

Connect to Salesforce Marketing Cloud

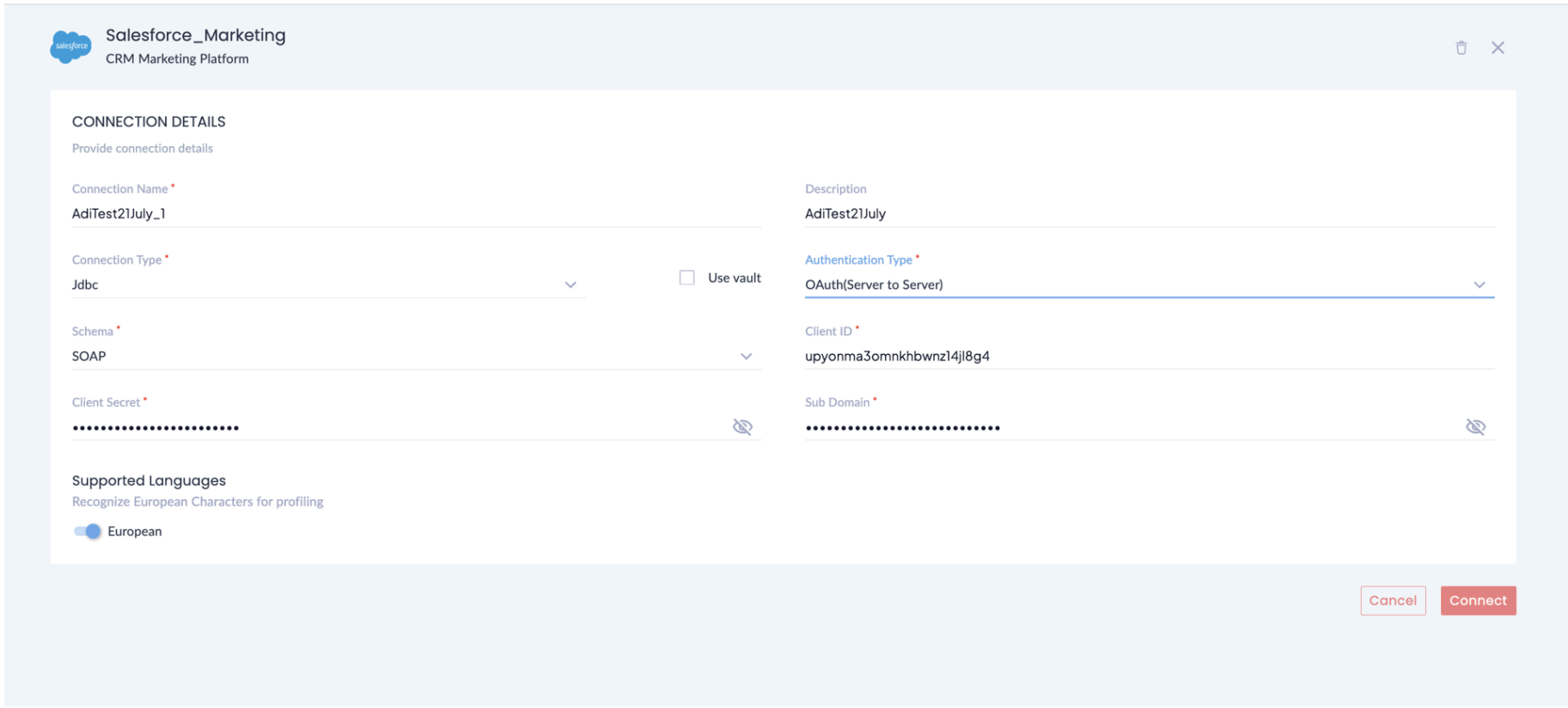
Follow the steps below to connect to the objects in Salesforce Sales Cloud

STEP 1: Navigate to Settings → Connect → Sources and click on the "+" icon

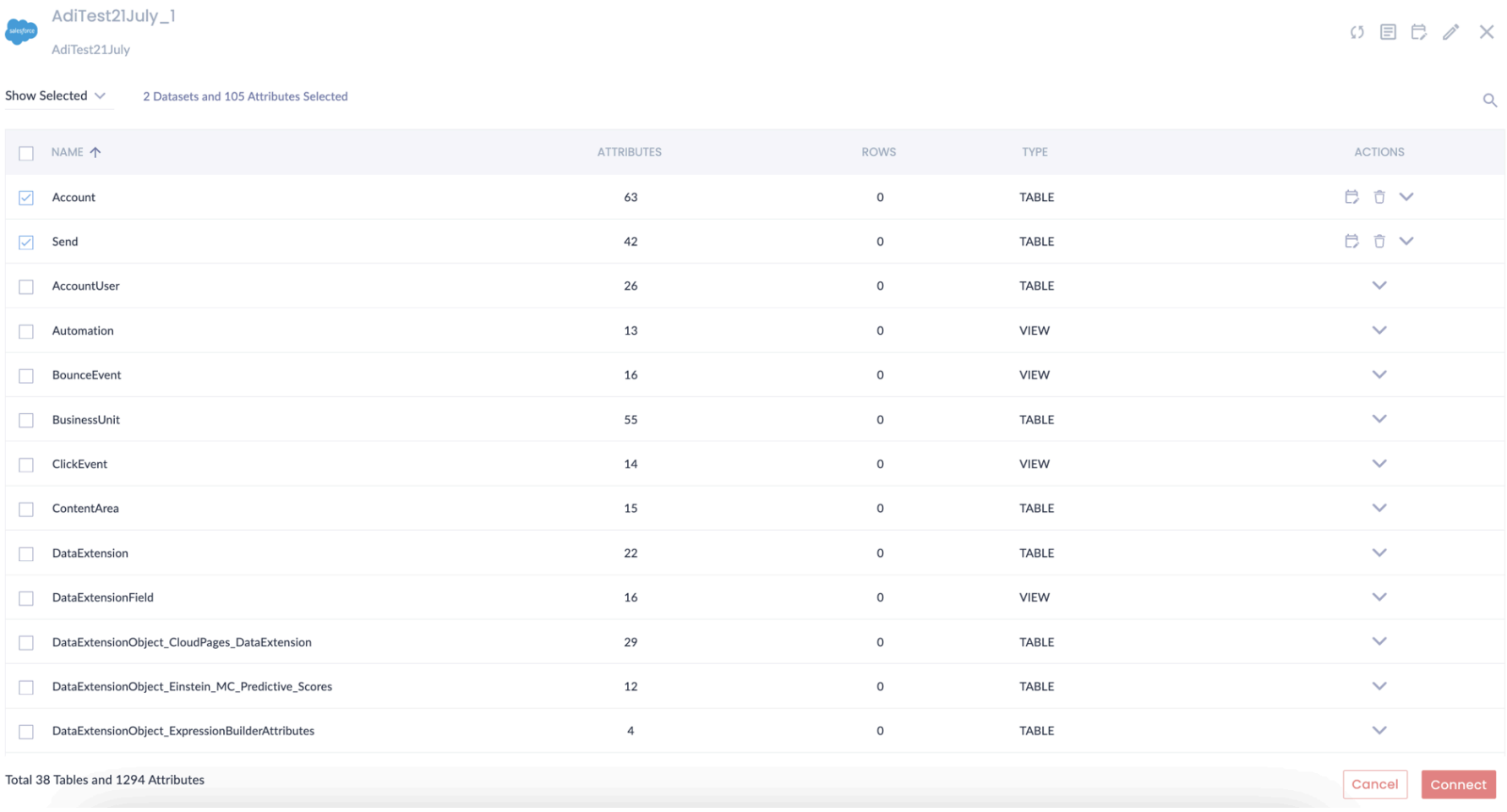


STEP 2: Select "Salesforce Marketing" and provide the following details:

Field / Option	Description
Connection Name*	A required name for identifying your connection (e.g., API_Marketing_Test)
Connection Type*	Specifies the protocol or service type , Select API for Pipeline Assets and JDBC for Data Assets
Description	Optional description about the connection (e.g., API_Marketing_Test)
Client ID*	Required OAuth client identifier for authenticating API calls
Sub Domain*	The Salesforce sub-domain value, typically a masked/secret input
Authentication Type*	Specifies the authentication protocol (e.g., OAuth(Server to Server))
Client Secret*	Secret key or password for authorization
Account ID*	Required account identifier for API operations (
Use Vault	Option to store sensitive credentials securely in a vault
Pull – Runs	Toggle ON to automate the collection of pipeline run data
Pull – Tasks	Toggle ON to automate the collection of pipeline task data
No Of Runs (Days)*	Specify how many days' worth of pipeline run history to pull (e.g., 30)
Status	Specify which run status to include
Calculate Score Based On	Select basis for scoring calculation
Propagate Issues Based On	Choose the source for propagating issues
Failure Checkbox	When enabled, failures are tracked and propagated as operational issues
Create or Propagate Alerts Based On	Determine how alerts are created or propagated (e.g., Pipeline)
Supported Languages	Recognize and process specific character sets, such as European, for profiling



STEP 3: Click on Validate after providing the above details. Once validated, click on connect to view the list of all available tables.



Step 4: Once connected, the user will be able to view the list of pipelines in Salesforce Marketing

Step 5: Select the required pipelines and click on connect.

Once connected, the admin/privileged user will be redirected to the asset detail page. The following objects are mapped in Salesforce marketing pipelines

- Jobs - Automations in Salesforce marketing pipelines
- Task - Steps in Salesforce marketing pipelines
- Runs - Runs for the jobs

COLLABORATION INTEGRATION

Email - MS Graph

Quest DQ will now support MS Graph API for Outlook integration. The email integration functionality can now use MS Graph API endpoints to send notifications. Follow the steps below to configure Outlook using Microsoft Graph for email integration.

Prerequisites

To integrate Microsoft Teams into the Quest DQ **application** using **Microsoft Graph API**, follow these steps:

Set Up an Azure AD App Registration

To authenticate and interact with Microsoft Teams data, register your app in **Azure Active Directory (Azure AD)**. Follow the steps below:

1. **Go to** Azure Portal → **Azure Active Directory**.
2. Navigate to **App registrations** → Click **New registration**.
3. Enter Quest DQ
4. Choose Single-tenant
5. Set **Redirect URI** (if using OAuth)
6. Click **Register**.

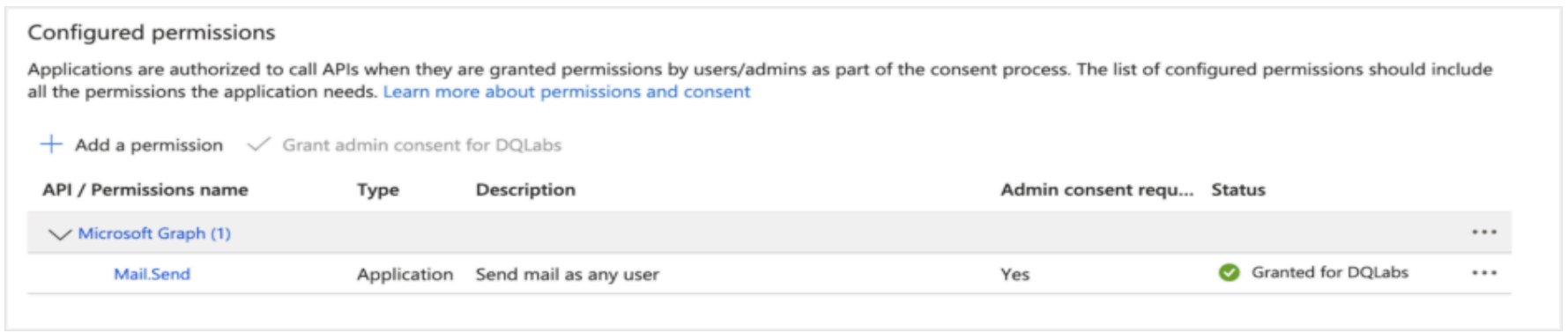
Obtain Credentials:

- Copy **Application (Client) ID**.
- Go to **Certificates & secrets** → **New client secret** → Save the generated secret.

API Permissions:

1. Navigate to **API permissions** → **Add a permission**.
2. Select **Microsoft Graph**.
3. Add the required **delegated** or **application** permissions (see below).
4. Click **Grant admin consent**.

Common Permissions:

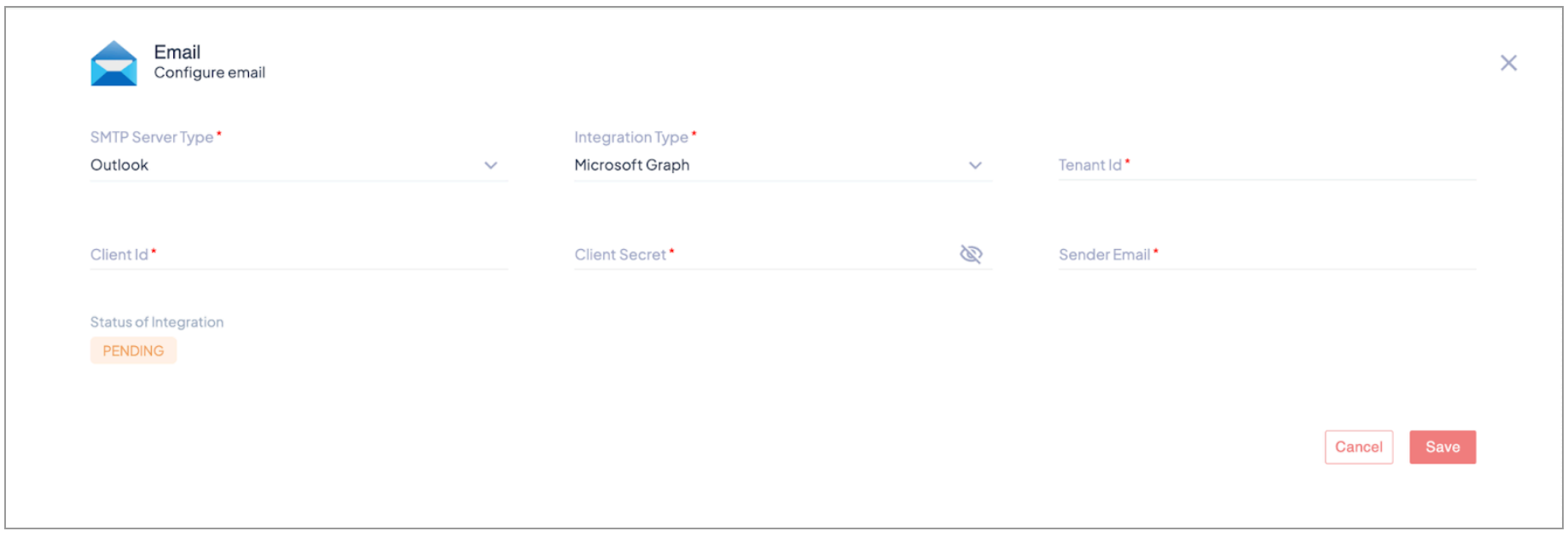


Set up in Quest DQ

Step 1: Navigate to Settings → Connect → Integrations

Step 2: Click on Email and select “Outlook” as SMTP Server Type and provide the following details

- Integration Type - Microsoft Graph
- Tenant ID
- Client ID
- Client Secret
- Sender Email



Step 3: Click on Save.

Once the integration is complete, the email notification will be sent through the configured channel.

AWS SES

Amazon Simple Email Service (AWS SES) is a cloud-based email service that allows businesses to send and receive emails securely, scalably, and cost-effectively. Quest DQ will now allow users to integrate with AWS SES and send notification emails. Follow the steps below to configure AWS SES for email integration:

Prerequisites

The following setup is required in AWS SES for email integration in Quest DQ:

Verify an Email Address

1. Sign in to the **AWS SES Console**: AWS SES
2. Navigate to **Email Addresses** → Click **Verify a New Email Address**.
3. Enter the email address you want to verify and click **Verify This Email Address**.
4. AWS will send a verification email—click the link inside to confirm.

Move SES Out of Sandbox Mode (Production Mode)

By default, AWS SES is in **sandbox mode**, meaning you can only send emails to verified addresses.

1. Go to the **AWS Support Center**.
2. Click **Create Case** → Select **Service Limit Increase**.
3. Choose **SES Sending Limits** and select the desired region.
4. Fill out the form, including:
    a. Type of emails you will send.
    b. Expected email volume.
    c. Compliance with AWS email policies.
5. Submit the request—AWS may take up to 24 hours to approve.

Configure SMTP for Sending Emails

If you want to send emails using SMTP (instead of AWS SDK), you need SMTP credentials.

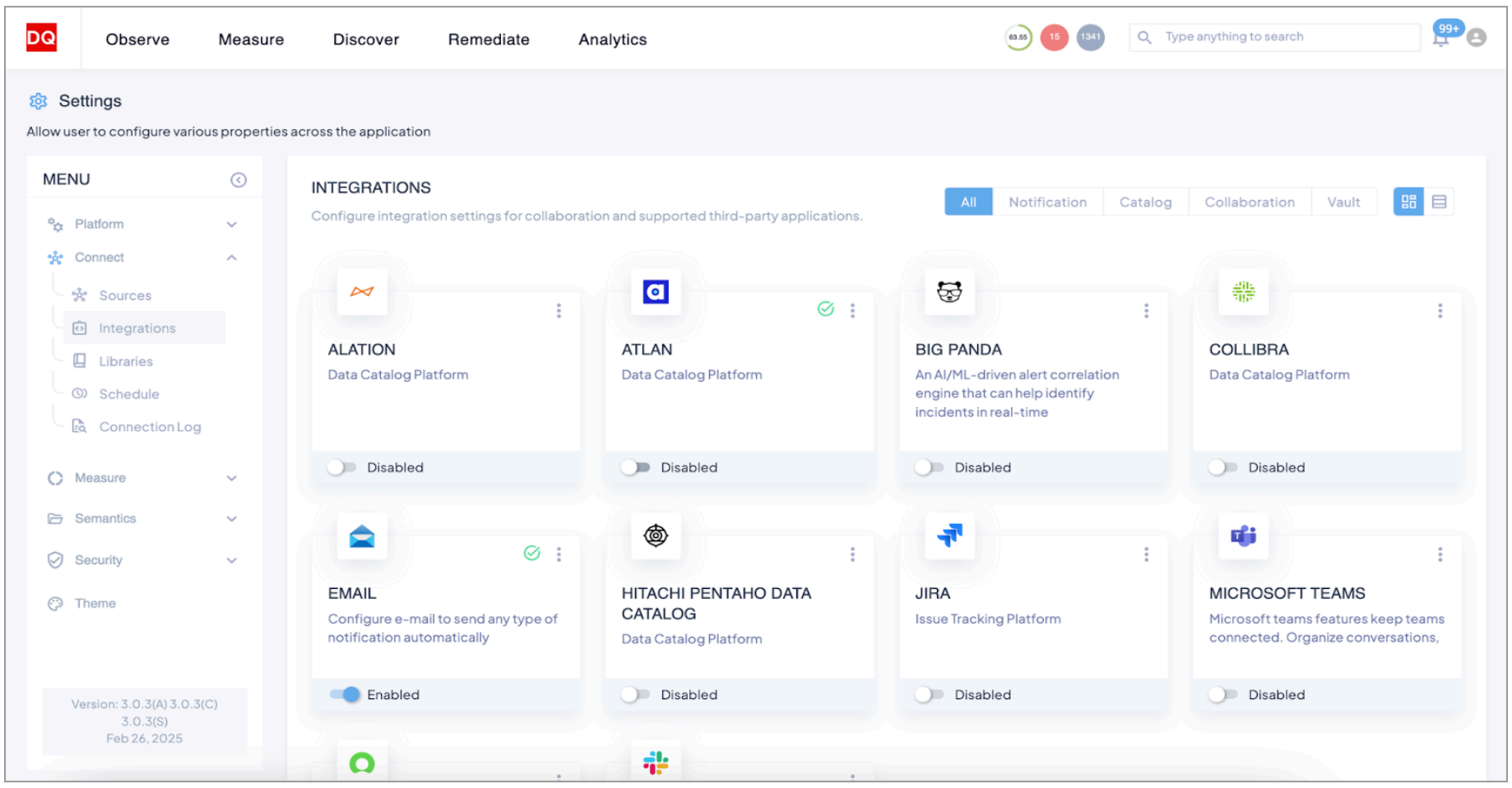
1. In the **SES Console**, go to **SMTP Settings** → Click **Create My SMTP Credentials**.
2. Follow the instructions to generate an **SMTP username and password**.
3. Store the credentials securely.

SMTP Settings:

- SMTP Server: email-smtp.<region>.amazonaws.com
- Port: 587 (TLS), 465 (SSL), or 25 (unencrypted)
- Authentication: **Yes** (Use the SMTP username/password)

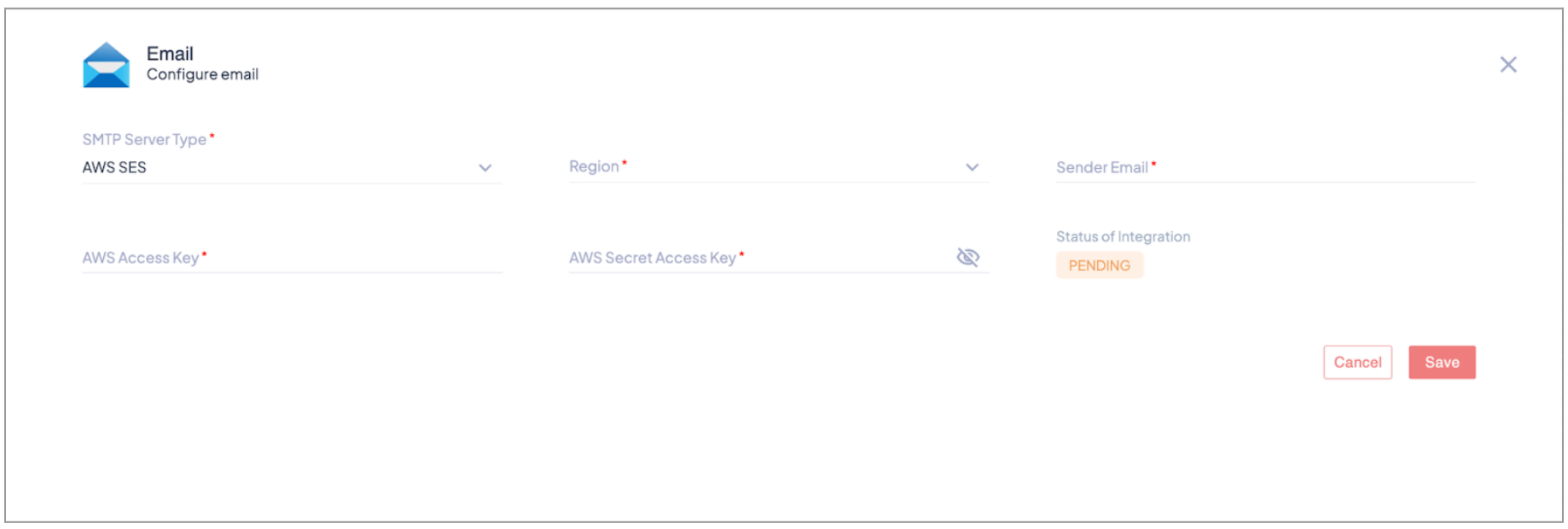
Setup in Quest DQ

Step 1: Navigate to Settings → Connect → Integrations



Step 2: Click on Email and select “AWS SES“ as SMTP Server Type and provide the following details

- Region
- Sender Email
- AWS Access Key
- AWS Secret Access Key



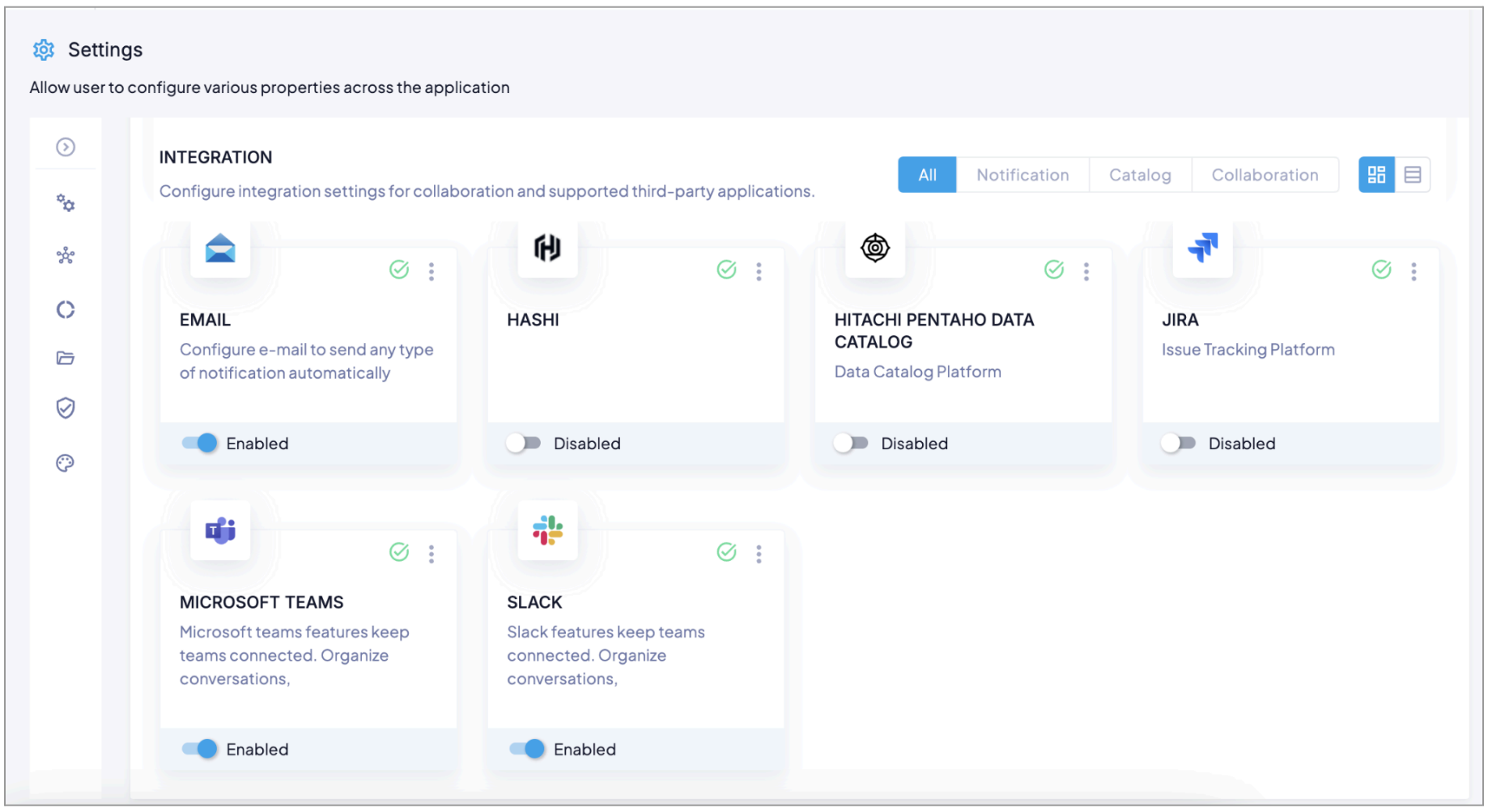
Step 3: Click on Save.

Once the integration is complete, the email notification will be sent through the configured channel.

Slack

Quest DQ allows users to receive alerts and issue notifications through Slack integration

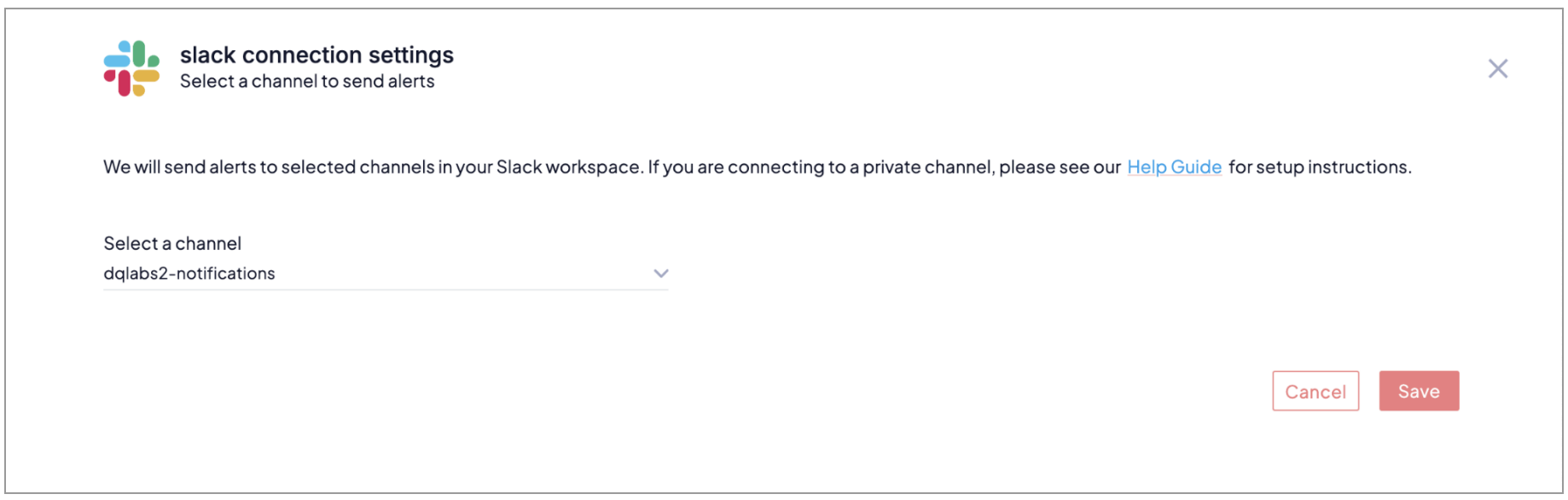
Step 1: Navigate to Settings > Connect > Integrations.



Step 2: Click on Slack. Clicking on Slack will prompt a pop-up, which must be allowed to continue the configuration.

Step 3: Select the Slack workspace from the drop-down at the top right, where the channel is configured, and click on "**Allow"**

Step 4: Once on the next page, the admin will then be able to select a channel to determine where the notifications from Quest DQ should be sent. Once the integration is complete, the notifications for Alerts and Issues will be sent to the Slack channel



Microsoft Teams

Quest DQ allows users to integrate with MS Teams to receive notifications on alerts and issues in the portal

Prerequisites

The following configurations must be set up in Teams before integrating with Quest DQ

1. A team for notifications from Quest DQ
2. A channel for notifications from Quest DQ
3. Webhook for the respective channel

Generate WebHook

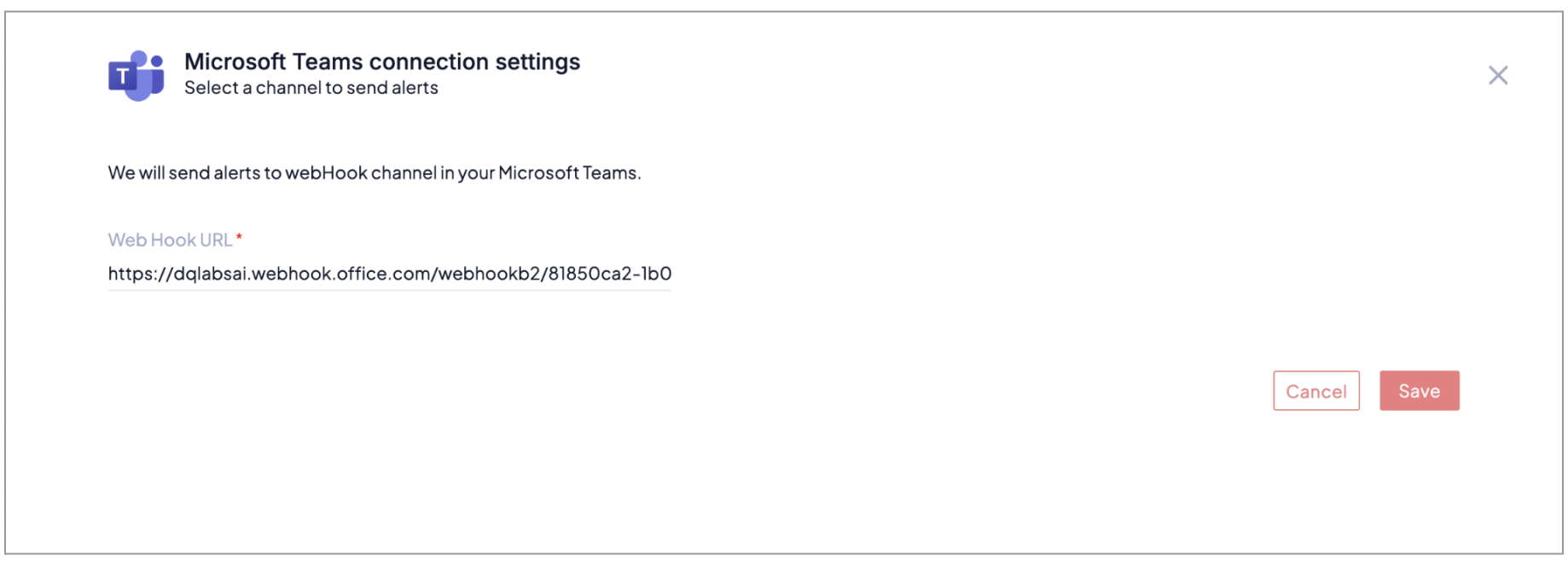
Follow the steps below to generate a webhook in Teams:

1. Open MS Teams, select "**Channel,"** and then click on **More options** (…) and choose **Connectors**.
2. A pop-up window will come up. Select All from the Category section in the left pane, find Incoming Webhook, and click the Add button to add Incoming Webhook.
3. Another window will pop up. Click on the Install button.
4. Give it a Name, and you may change the image (Optional) and click on the Create button.
5. The final step is to copy the URL and click on the Done button.

Follow the steps below to integrate with Teams:

Step 1: Navigate to Settings > Connect > Integrations.

Step 2: Click on Teams, and provide the webhook URL to the team channel



Step 3: Click on the "Save" Button

Email - Outlook, Gmail and Sendgrid

Quest DQ currently supports the following email providers:

1. Gmail
2. Outlook
3. Send Grid

Prerequisites

- A service mail account has to be created by the Organization - dedicated to Quest DQ alerts and notifications. Reach out to your internal IT team for assistance on the same
- Once the account is set up, Emails can be integrated into Quest DQ by providing the required credentials
- We support TLS model SMTP server configuration with username and password on port 587. If a user is using port 25 on their on-premises, the user needs to confirm that port 25 is opened in the machine's outbound rule.
- Quest DQ also supports all SSL modes, but the Quest DQ installed machine should also have a subdomain of DNS, and it should be in 443; otherwise, the user will get the mail as Spam. (Please ignore this step if the user received the mail in your inbox)
- The username used must not contain 2-factor authentication.
- For the email account to be functional in Quest DQ, it is necessary to whitelist the required IPs. Reach out to the support team for assistance on IP Whitelisting

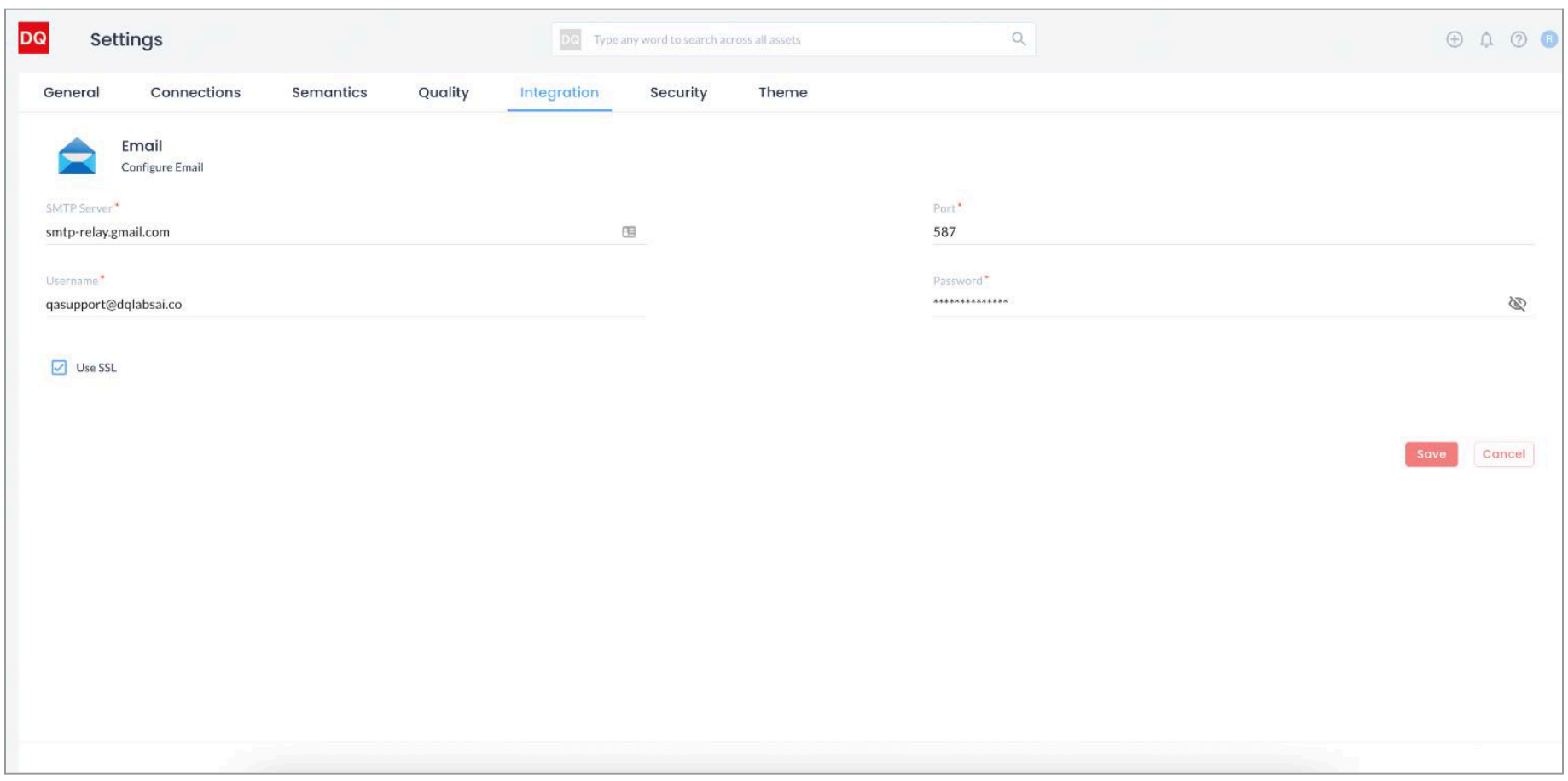
Integrate into Quest DQ

Step 1: Log in to Quest DQ and navigate to Integrations in the settings page

Step 2: Click on Email and provide the following details:

- SMTP server type
- SMTP Server (Example: smtp.gmail.com)
- Port
- Username
- Password

Step 3: Click on Save, and now the alerts and notifications can be shared via email



WORKFLOW INTEGRATION

Jira

JIRA is a popular project management tool developed by Atlassian. It is primarily used to track and manage software development projects, but it can be adapted to other types of projects as well. JIRA allows users to create and organize tasks, assign them to team members, set deadlines, and track progress.

Quest DQ provides the ability to create issues automatically in Jira based on the data quality issues created in Quest DQ. To integrate with Jira, you have to create an API Key and then configure it in Quest DQ for the integration

Create an API key in JIRA

Quest DQ connects with JIRA using API keys. The JIRA admin can create API keys and use them to integrate with Quest DQ. API keys are used to create Issues and update status, Priority, and comments in JIRA.

1. Log in to your JIRA account.
2. Click on your profile picture or initials in the top right corner of the screen.
3. Select "Account settings" from the dropdown menu.
4. In the "Security" section, click on "Create and manage API tokens".
5. Click on "Create API token" and enter your password when prompted.
6. A new API key will be generated. Be sure to copy it to a secure location, as it will only be displayed once.
7. Use the API key in your API requests by including it as a bearer token in the Authorization header.

Required Permissions:

Jira Global Administrator - sync b/w Jira and Quest DQ (Using webhook)

Jira Administrator - One-way update from Quest DQ (without web hook)

Note that some versions of JIRA may have slightly different steps for creating an API key, but the general process should be similar

Integrate with JIRA

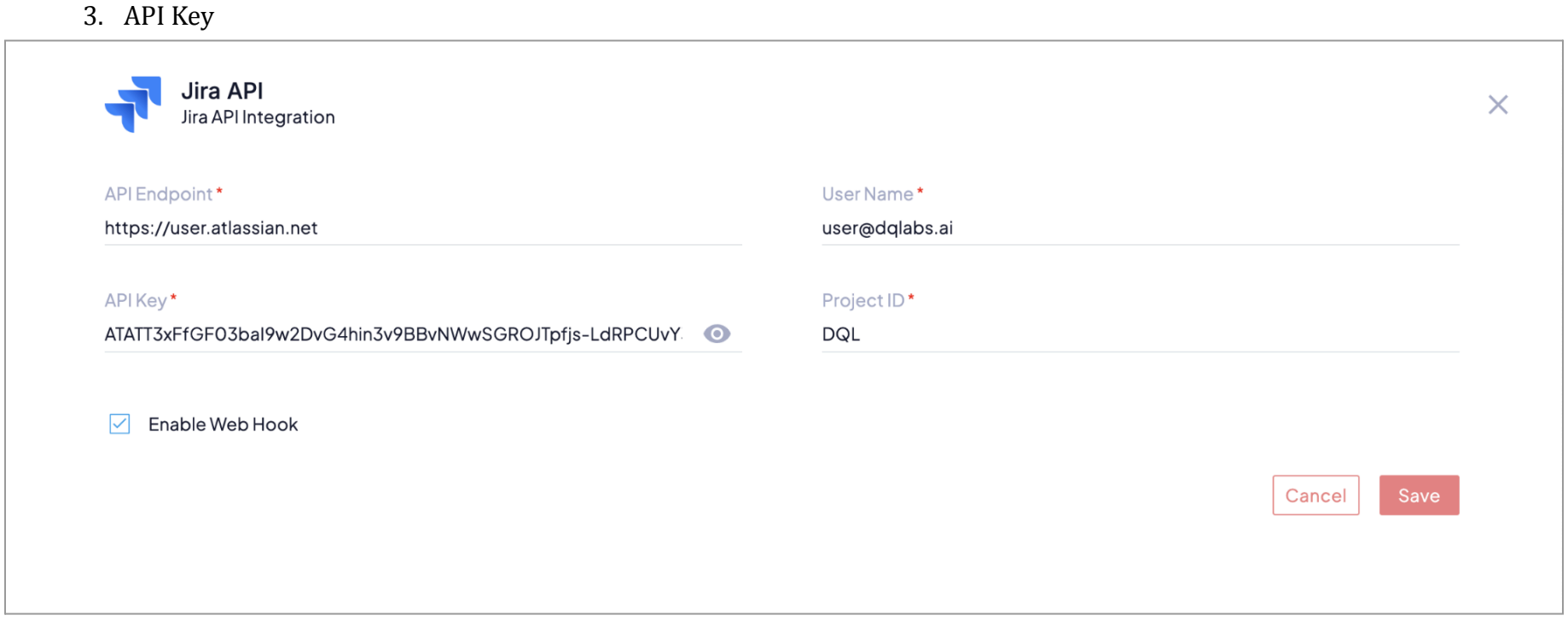
Follow the steps below to integrate with Jira:

Step 1: Navigate to Settings > Connect > Integration

Step 2: Click on Jira API

Step 3: On the Jira API page, provide the following details

1. API Endpoint
2. Username
3. API Key



Step 4: Click on Save

Once the integration is saved, the data quality issues identified in Quest DQ will be automatically created as an issue in the respective Jira Project

ServiceNow

ServiceNow is a cloud-based platform that provides IT Service Management (ITSM), IT Operations Management (ITOM), and IT Business Management (ITBM). It helps organizations automate workflows, manage IT services, and improve operational efficiency. Quest DQ allows users to integrate with ServiceNow to create incidents and alerts in ServiceNow automatically based on the alerts and issues created in Quest DQ. This allows users to centralize all alerts and issues in one place.

The users will be able to perform the following actions with ServiceNow integration:

- Push Alerts to ServiceNow
- Push Issues to ServiceNow
- Update Alerts and Issues in Quest DQ based on the updates in ServiceNow

Pre-requisites

The following pre-requisites should be met before integrating with ServiceNow:

Authentication:

- ServiceNow Instance
- Instance Detail: Instance URL, Instance Name, Username, Password
- The user must have the **Incident Manager** and **Rest API Explorer** privilege role

Install Plugins:

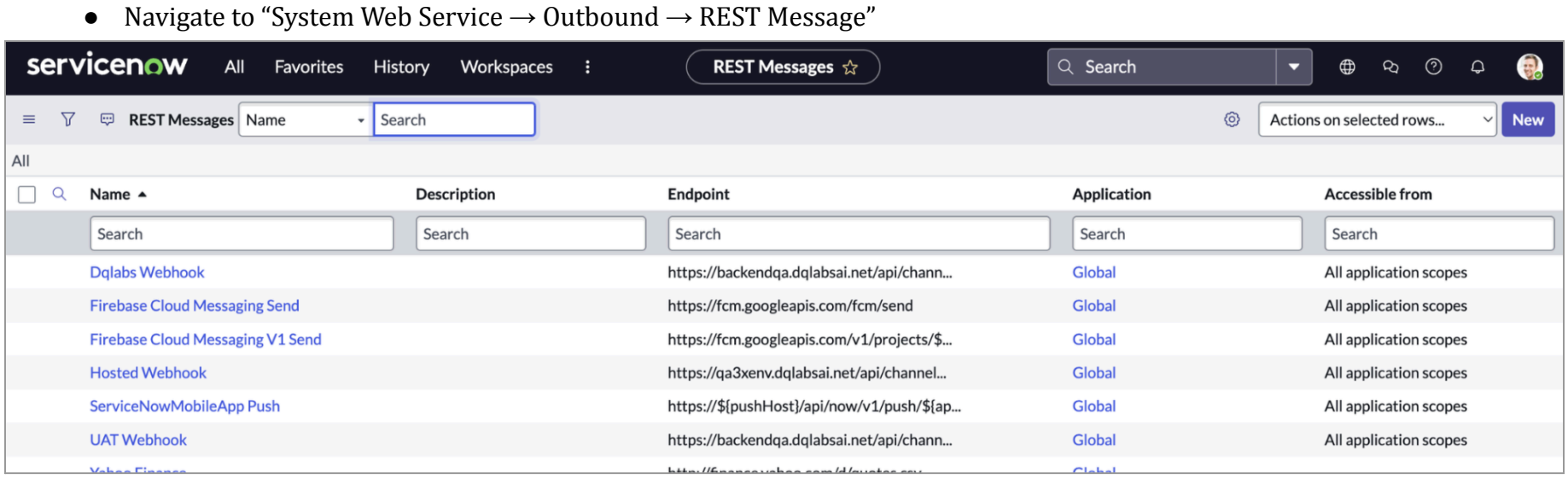
- The following plugins should be installed in service now to integrate with Quest DQ:
  - Install the event-management Plugin for the em_alert table
  - Install IntegrationHub Plugin for outbound_rest_message

Configuration in Service Now Instance for Webhook

The following configurations have to be set up for bidirectional updates in alerts and issues:

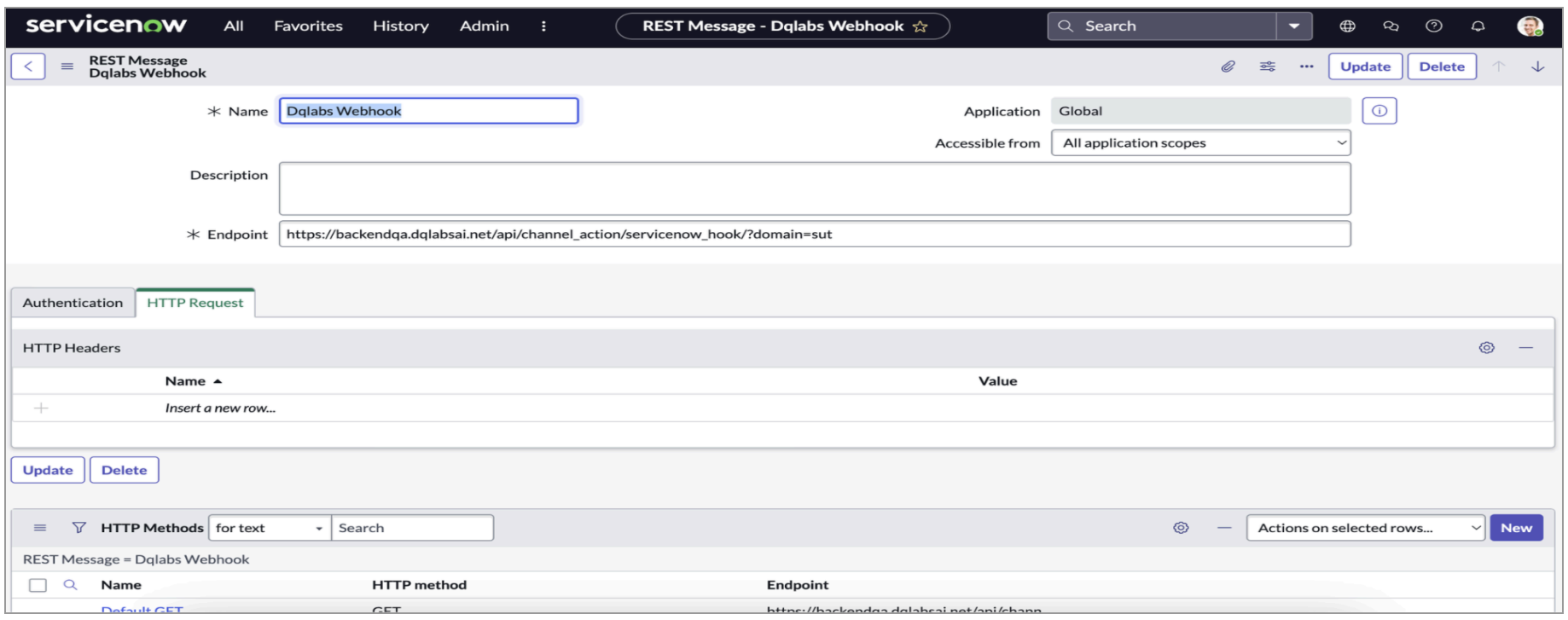
Create Rest Message

- Navigate to “System Web Service → Outbound → REST Message”



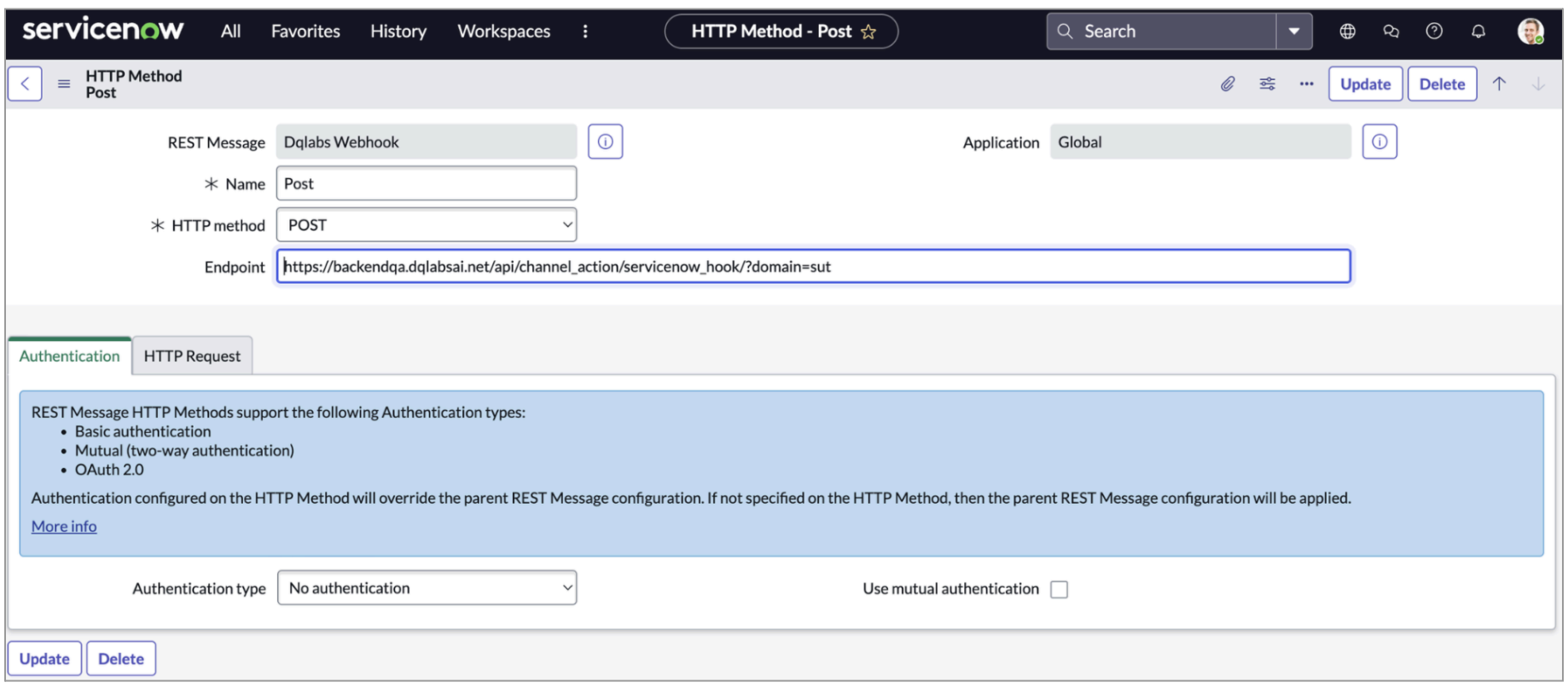
- Click on “NEW“ and provide the following details
  - Name
  - Endpoint → The API endpoint from Quest DQ with the domain(Refer to Screenshot)

```
None
htpps://<API endpoint>/api/channel_action/servicenow_hook/?domain=<domain_name>
```



- Click on “New“ under the HTTP request tab and create a POST Method.

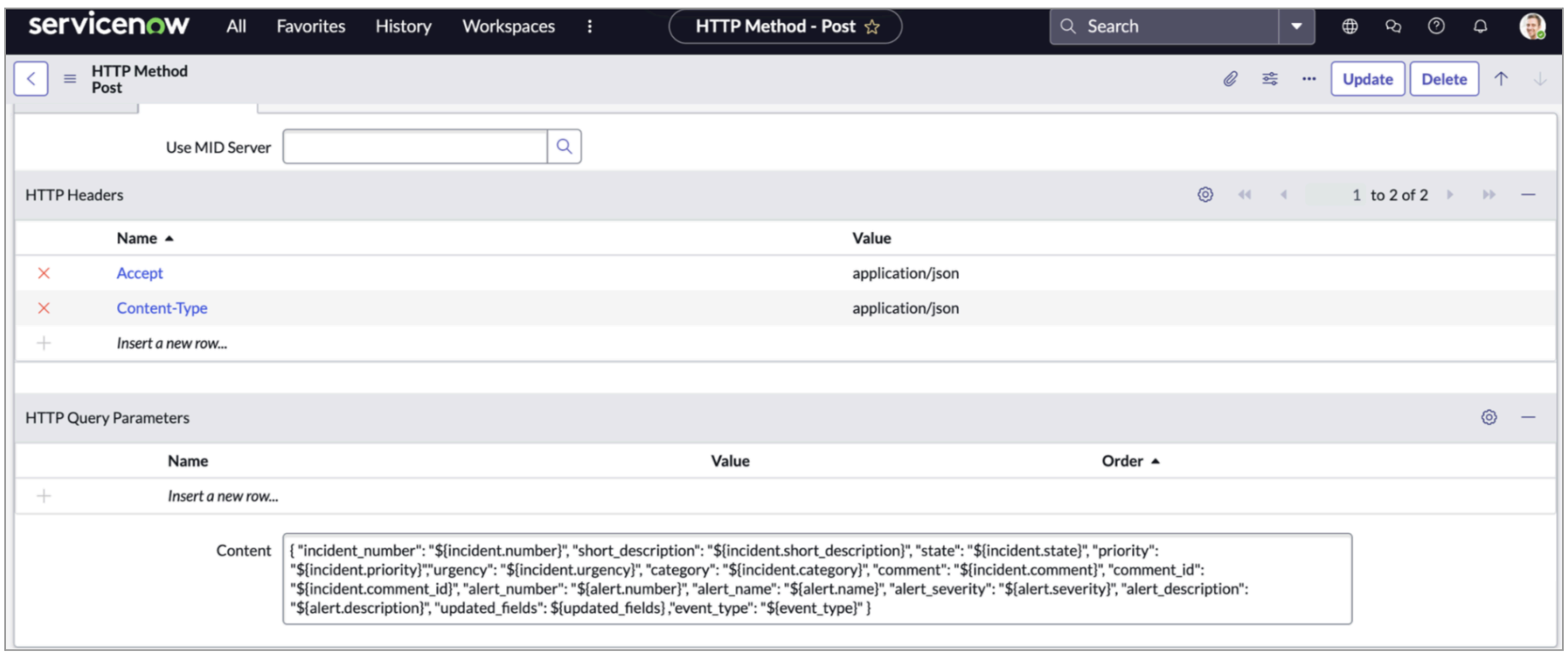
- Provide the following details on the page:
  - Name
  - HTTP Method
  - Endpoint



- Create the HTTP Headers as shown in the screenshot and paste the following under the Content tab of HTTP Query Parameters > Save

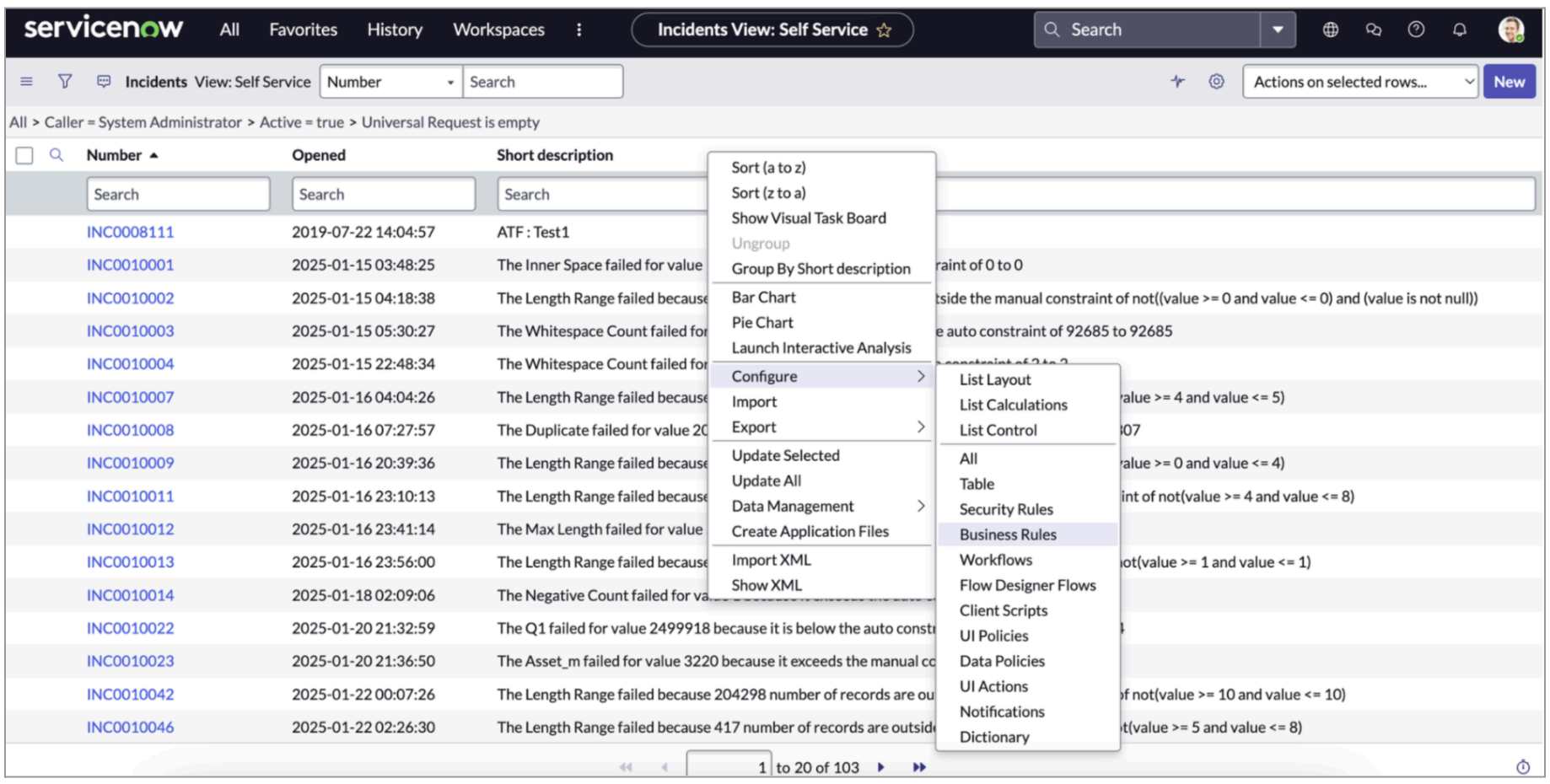
None

```
{
"incident": {
"number": "${incident.number}",
"short_description": "${incident.short_description}",
"state": "${incident.state}",
"priority": "${incident.priority}",
"urgency": "${incident.urgency}",
"category": "${incident.category}",
"comment": "${incident.comment}",
"comment_id": "${incident.comment_id}"
},
"alert": {
"number": "${alert.number}",
"name": "${alert.name}",
"severity": "${alert.severity}",
"description": "${alert.description}"
},
"updated_fields": ${updated_fields},
"event_type": "${event_type}"
}
```

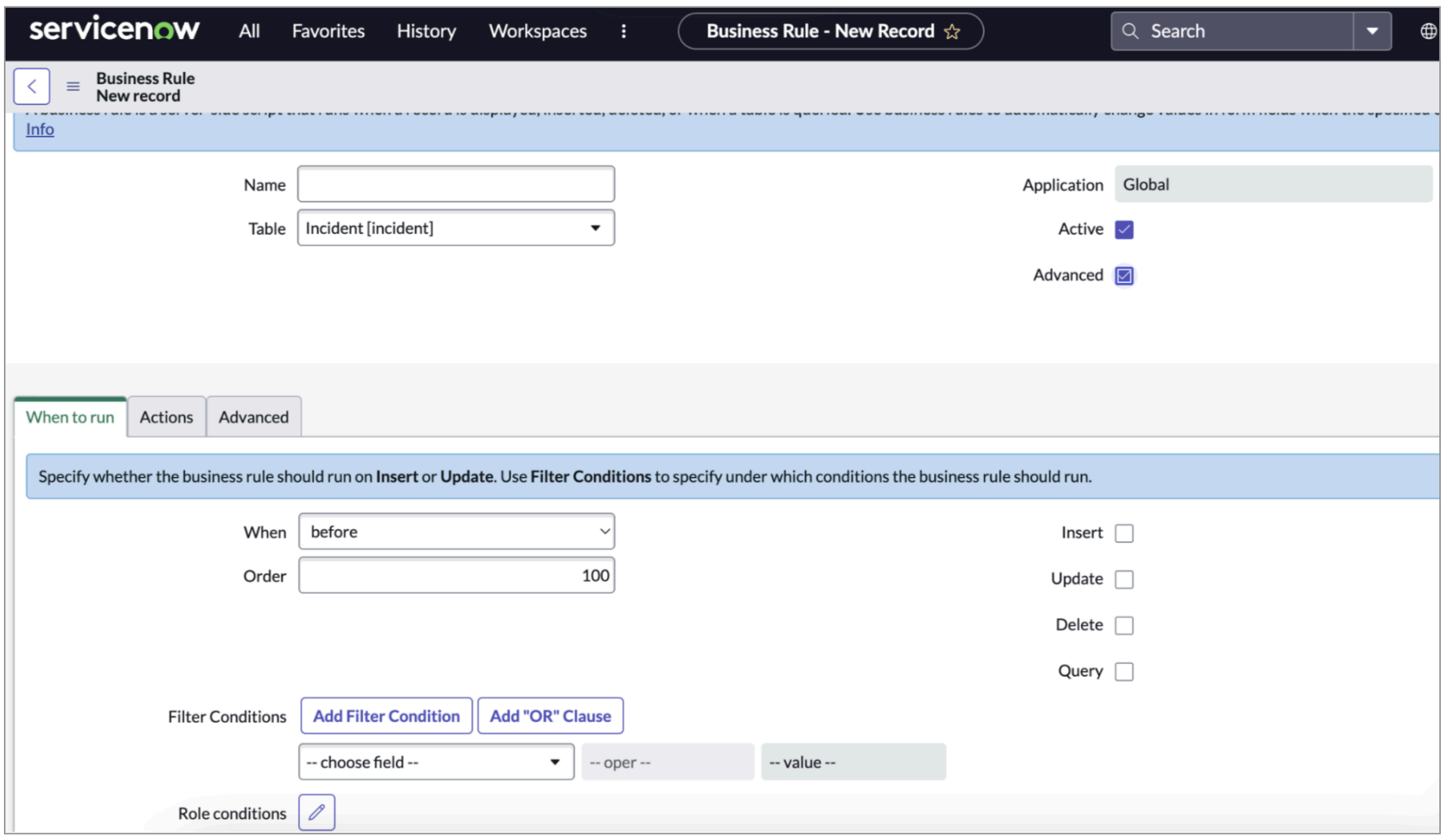


Create Business Rules for the incident

- Navigate to the incidents page and right-click on the screen and click on the business rule



- Click on “New“ and provide the following details
  - Name
  - Check Advanced
  - Under the “When to Run” tab, set "When" to "After" and check the following:
    - Insert
    - Update



- Paste the following in the Advanced Tab with changes to the red highlighted text:
- Replace Quest DQ Web Test with the name of the Rest Message
- Replace Send Webhook with the name of the Post method
- Replace the Endpoint with the endpoint configured on the Rest Message Page

None

```
(function executeRule(current, previous /*null when async*/) {
// Create GlideRecord to fetch incident details   var gr = new GlideRecord('incident');
   gr.get(current.sys_id);


   // Initialize the REST message
   var restMessage = new sn_ws.RESTMessageV2('Quest DQ Web Test', 'Send Webhook');
   restMessage.setEndpoint('https://7924-223-185-27-130.ngrok-free.app/api/channel_action/servicenow_hook/');


// Add the event type parameter based on operation
var eventType = 'incident_' + current.operation();


var commentContent = "";
var commentID = null;
if (current.comments.changes()) {
    var gr2 = new GlideRecord('sys_journal_field');
    gr2.addQuery('element_id', current.sys_id);
    gr2.addQuery('element', 'comments');
    gr2.orderByDesc('sys_created_on');
    gr2.setLimit(1);
    gr2.query();


    // Handle when a comment is added or updated
    if (gr2.next()) {
           commentContent = gr2.value.toString();
           commentID = gr2.sys_id.toString();
           eventType = previous.comments ? "comment_updated" : "comment_added";
           gs.log('Exact Comment: ' + commentContent);
           gs.log('Comment ID: ' + commentID);
```

```
    } else {
          gs.log('No comments found for this record.');
    }
}


// Handle Comment Deletion if the current comments field is empty
if (current.comments == "") {
    var gr3 = new GlideRecord('sys_journal_field');
    gr3.addQuery('element_id', current.sys_id);
    gr3.addQuery('element', 'comments');
    gr3.query();


    while (gr3.next()) {
          if (gr3.value == "") {
          commentID = gr3.sys_id.toString();
          eventType = "comment_deleted";
          gs.log('Deleted Comment ID: ' + commentID);
          break;
          }
    }
}
```

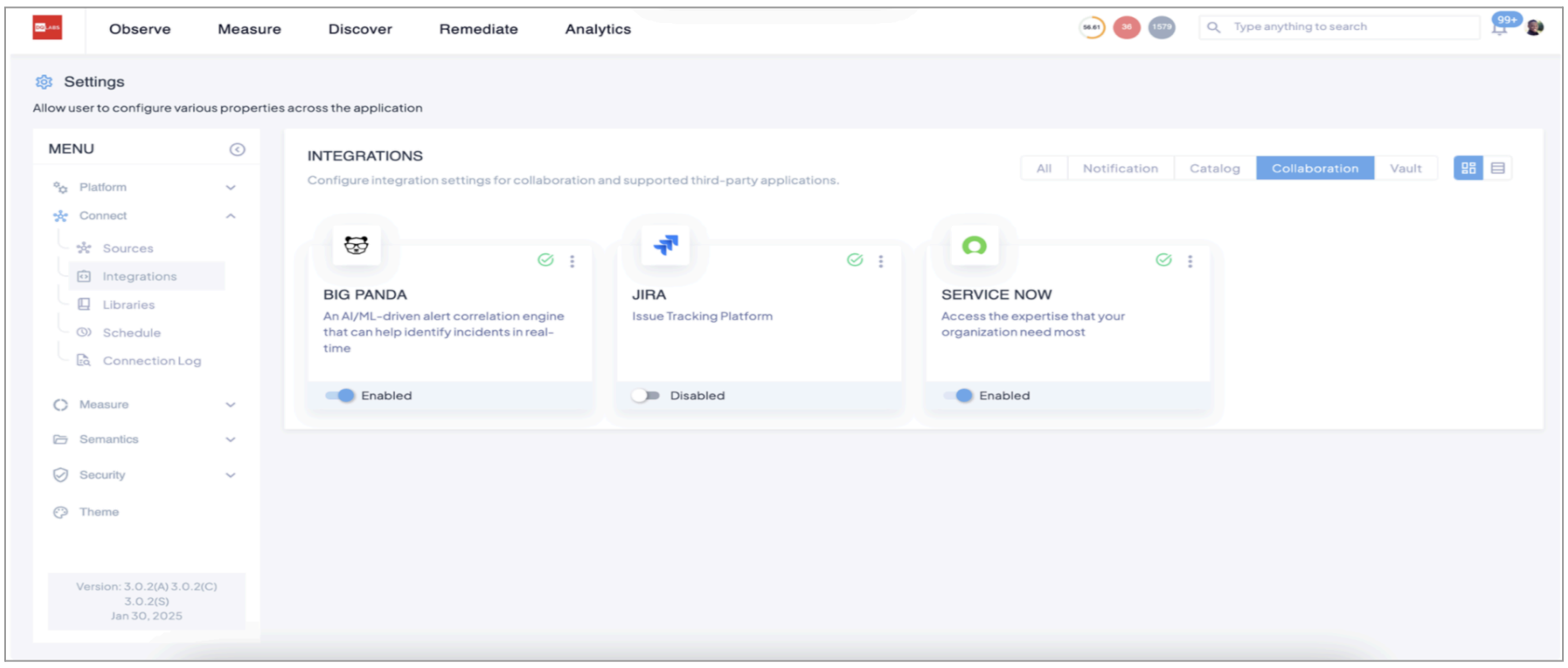
```
// Helper function to safely convert any value to a string
function safeToString(value) {
    if (value === null || value === undefined) {
          return "";
    } else if (typeof value === "object") {
          return JSON.stringify(value);
    } else {
          return String(value);
    }
}


// Track updated fields and push into an array
var updatedFields = [];
var fieldsToCheck = ['short_description', 'state', 'priority', 'urgency', 'comments'];


fieldsToCheck.forEach(function(fieldName) {
    var oldValue = previous.getValue(fieldName);
    var newValue = current.getValue(fieldName);


    if (oldValue !== newValue) {
    updatedFields.push({
          field: fieldName,
          old_value: safeToString(oldValue),
          new_value: safeToString(newValue)
    });
    }
});


// Send each field individually using setStringParameterNoEscape
restMessage.setStringParameterNoEscape('incident.number', gr.number);
restMessage.setStringParameterNoEscape('incident.short_description', gr.short_description);
restMessage.setStringParameterNoEscape('incident.state', gr.state);
restMessage.setStringParameterNoEscape('incident.priority', gr.priority);
restMessage.setStringParameterNoEscape('incident.urgency', gr.urgency);
restMessage.setStringParameterNoEscape('incident.category', gr.category);
restMessage.setStringParameterNoEscape('incident.comment', commentContent);

// Include the comment ID if available
if (commentID) {
    restMessage.setStringParameterNoEscape('incident.comment_id', commentID);
}


restMessage.setStringParameterNoEscape('event_type', eventType);


// Convert updatedFields to a string and set it as a parameter
```

```
    if (updatedFields.length > 0) {
        restMessage.setStringParameterNoEscape('updated_fields', JSON.stringify(updatedFields));
    } else {
        restMessage.setStringParameterNoEscape('updated_fields', JSON.stringify({}));
    }


    // Send the request and capture the response
    var response = restMessage.execute();
    var httpResponseStatus = response.getStatusCode();
    gs.log('Webhook response status: ' + httpResponseStatus);


})(current, previous);
```

Integrating ServiceNow in Quest DQ

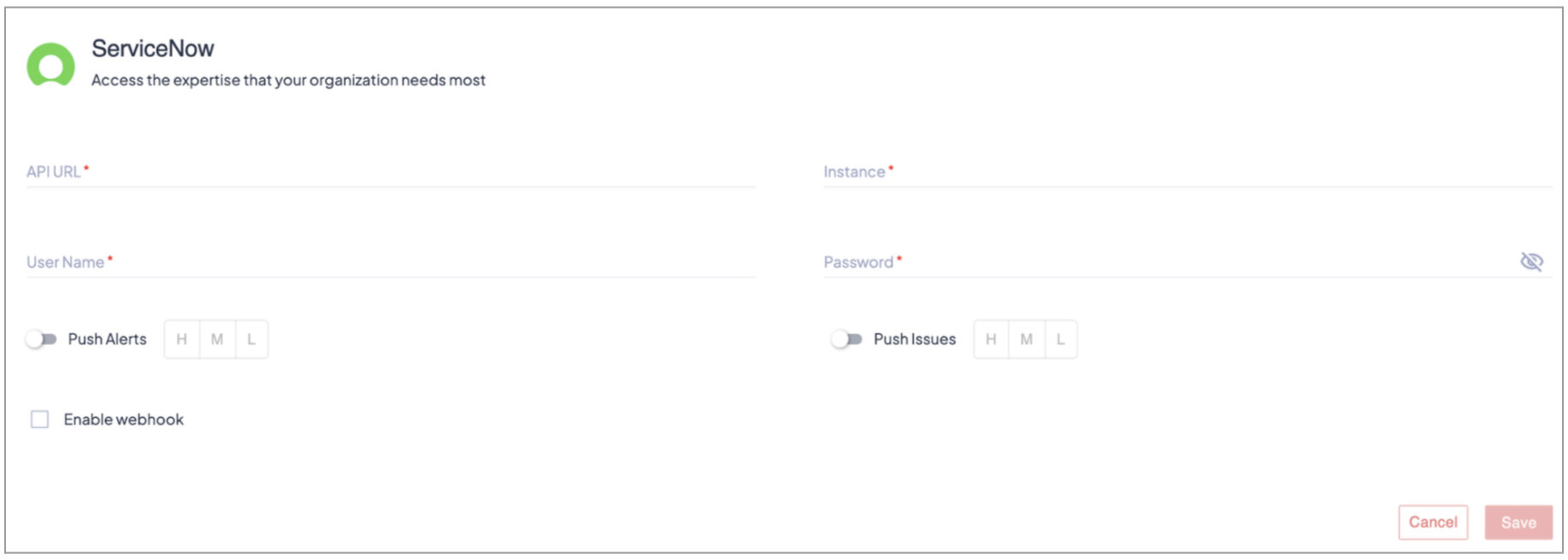
Follow the steps below to integrate Big Panda in Quest DQ:

Step 1: Log in to Quest DQ, and Navigate to Settings → Connect → Integrations

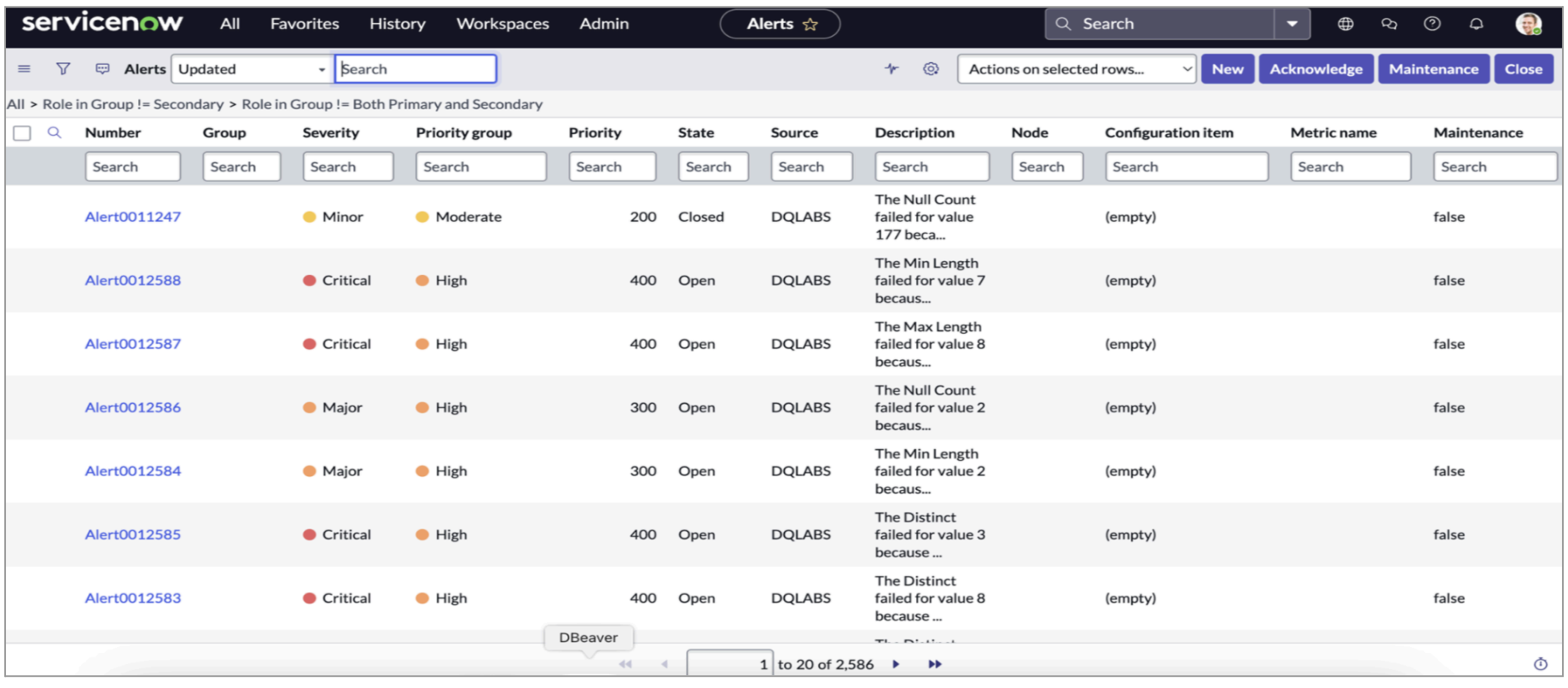


Step 2: Click on ServiceNow and provide the following details

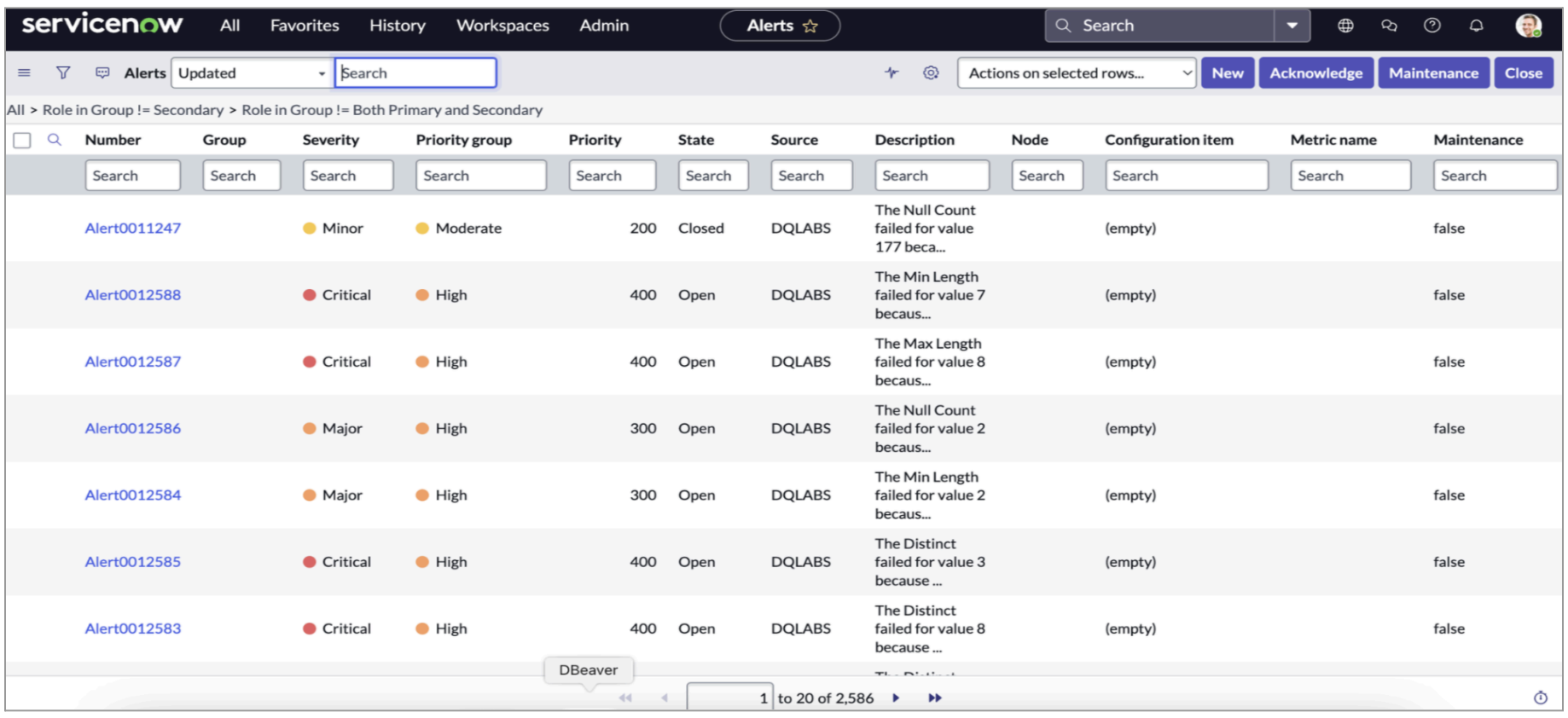
- API URL - ServiceNow API Endpoint
- Instance
- Username
- Password
- Push Alerts → Select the priority of alerts to be pushed to ServiceNow
- Push Issues → Select the priority of issues to be pushed to ServiceNow
- Enable Webhook → For bi-directional Update



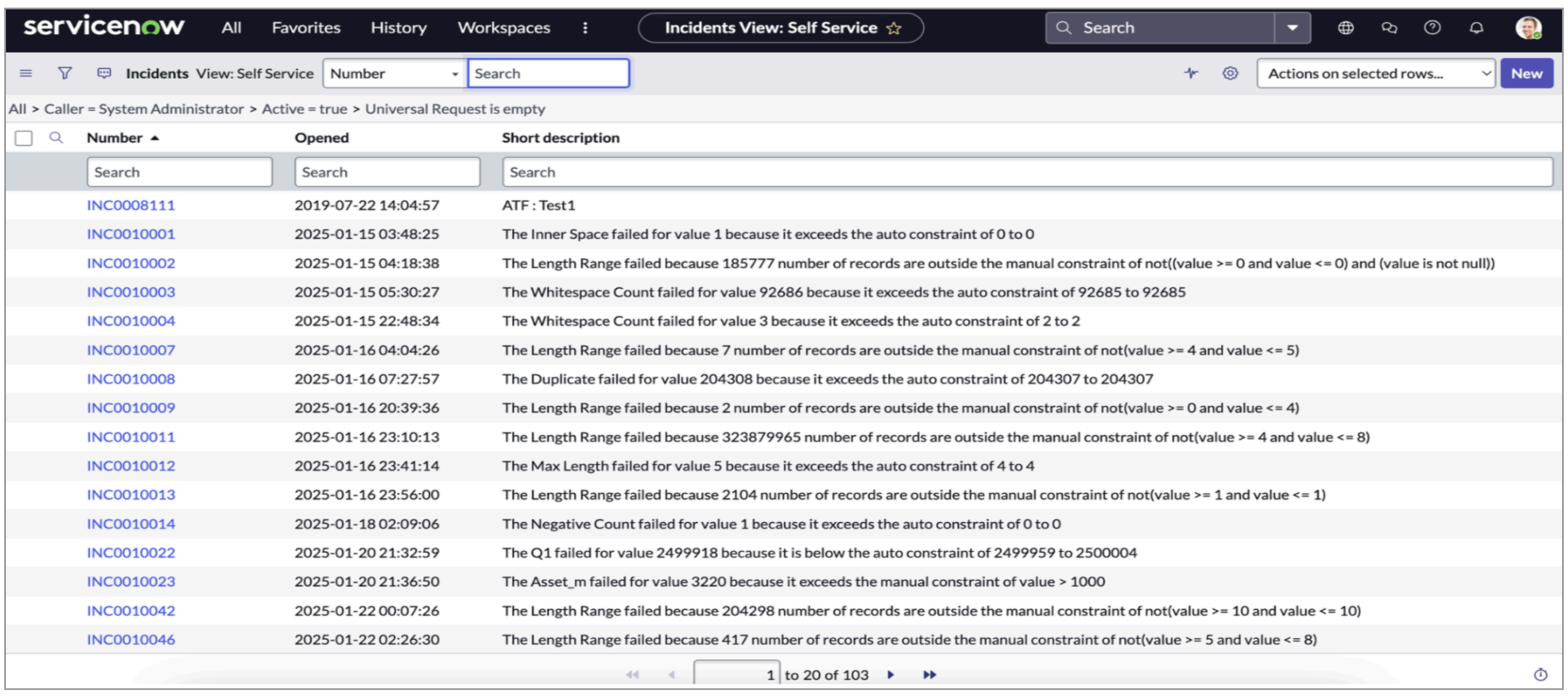
Step 3: Once saved, the created issues and alerts will be pushed to ServiceNow



Alerts in Quest DQ will be pushed to alerts in ServiceNow with the following details:



Issues in Quest DQ will be pushed to incidents in ServiceNow with the following details:



Big Panda

BigPanda's robust AI/ML-driven alert correlation engine can expedite triage by providing business context and business logic, enabling real-time problem identification. BigPanda builds a comprehensive picture of the problems in your infrastructure by correlating high-quality warnings.

Prerequisites

The user should have the following details pre-configured in order to integrate with Big Panda

- Whitelisting Quest DQ IP
- Generate APP Key
- Get Organization Token

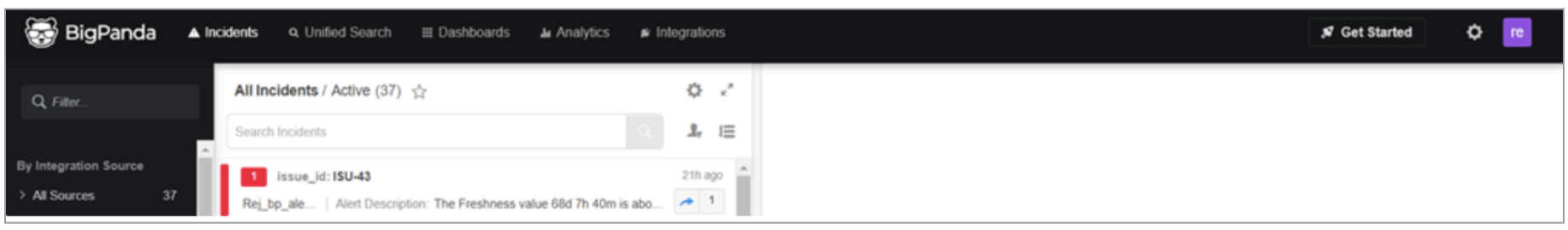
Whitelist IP

If your organization uses a whitelist to manage Big Panda, Quest DQ will only access the tool through the specific IP range.

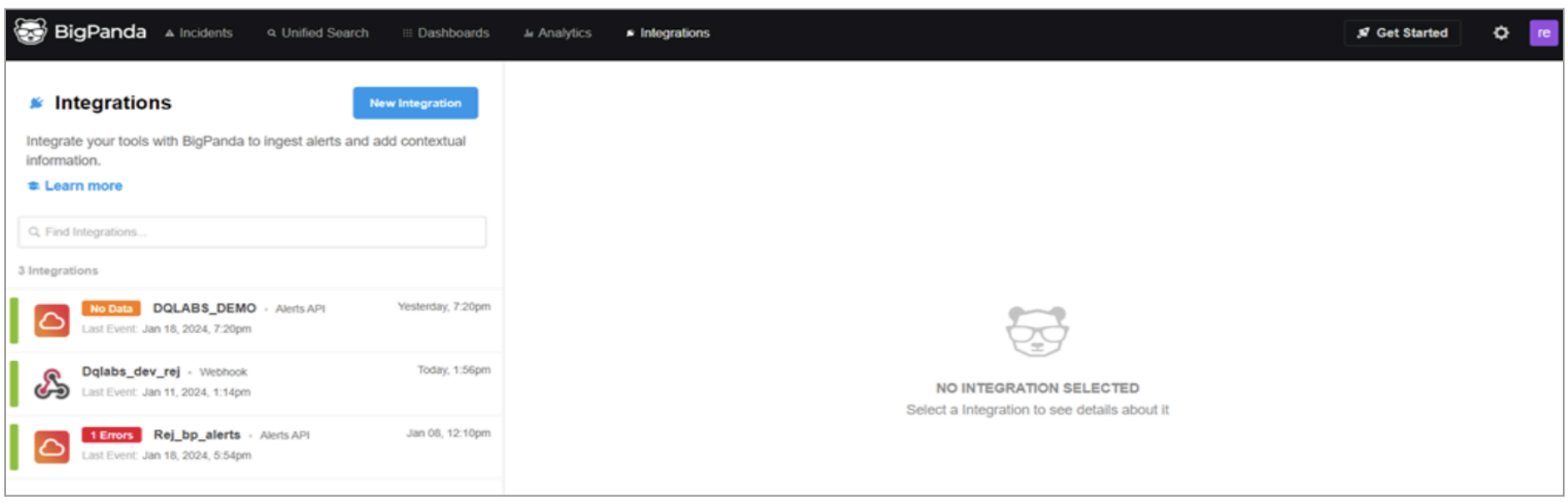
Generate APP Key

Quest DQ integrates with Big Panda by using APP keys. To generate APP keys, follow the steps given below in Big Panda

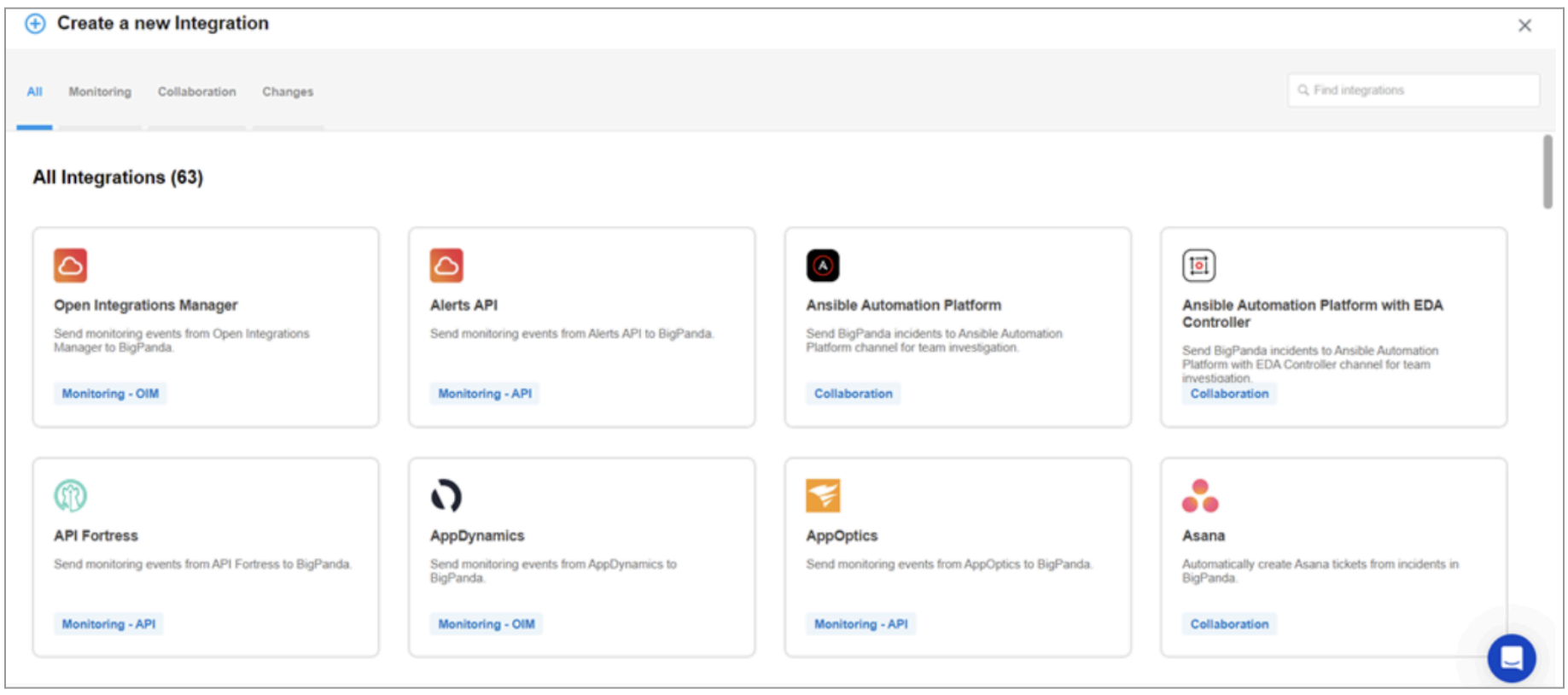
Step 1: Log in to Big Panda and Click on “Integrations.”



Step 2: On the integrations page, click on “New Integration”

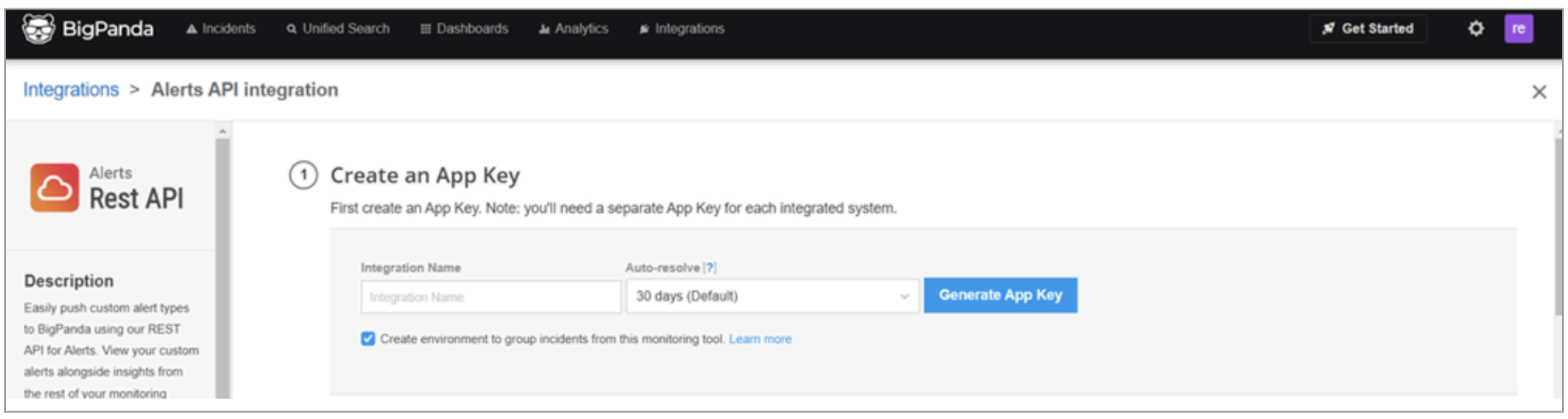


Step 3: On the Create a new Integration page, click on Alerts API



Step 4: Provide the following details and click on “Generate APP Key”

- Integration Name
- Auto-resolve time

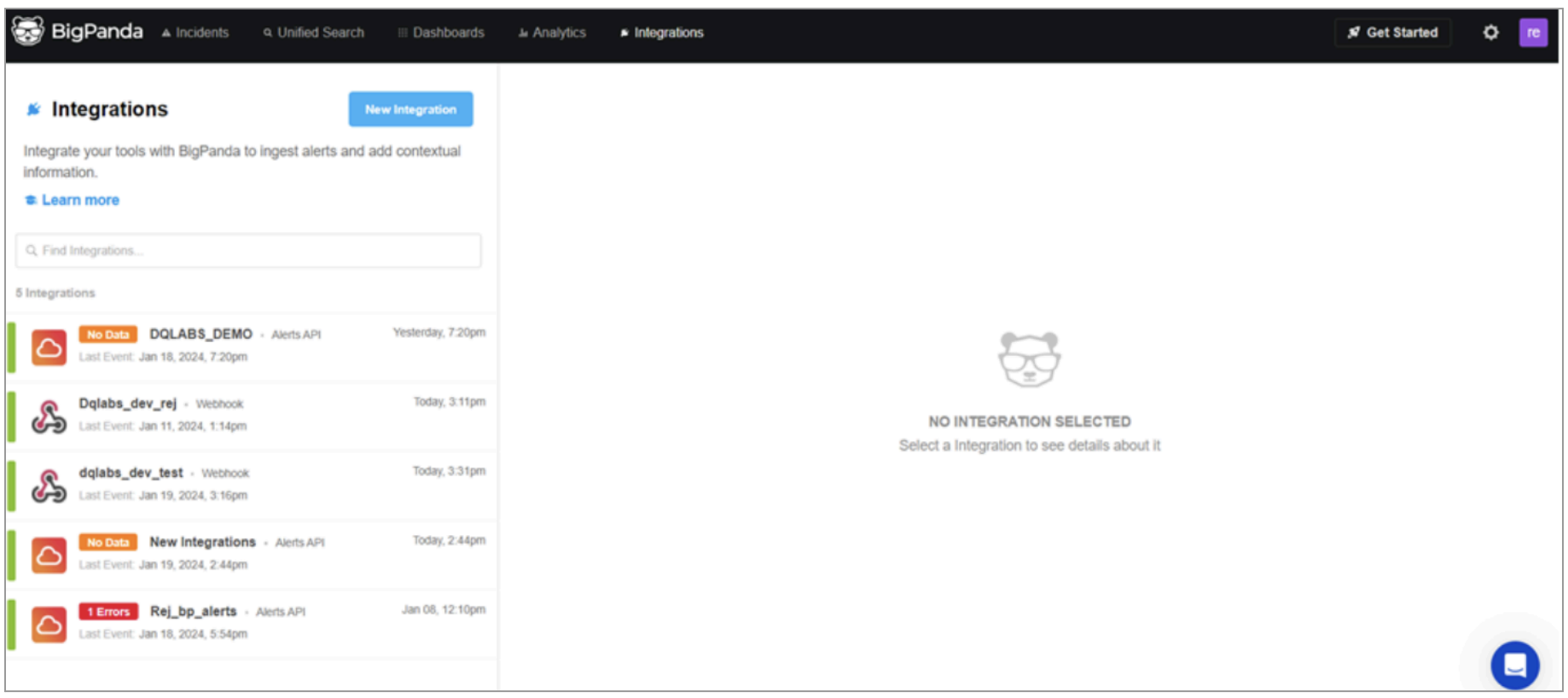


Once a key is generated, the user will be able to view the APP key on the integrations page, copy the APP key to use in the integration in Quest DQ

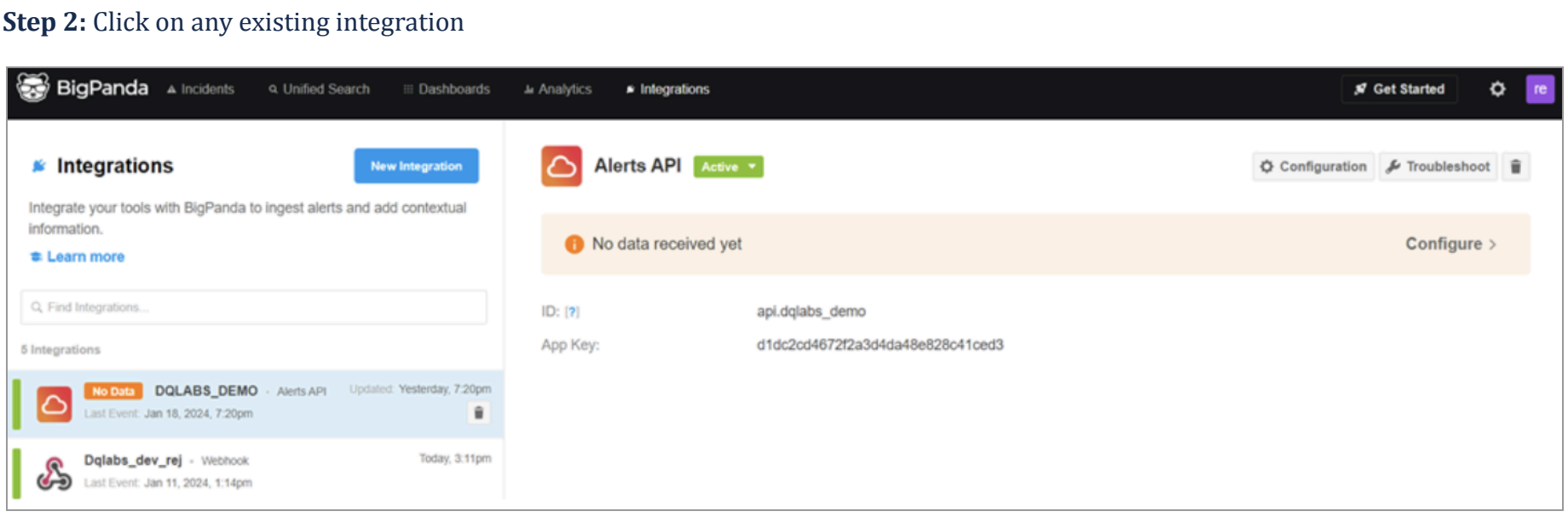
Get Organization Token

Quest DQ requires an organization token to authenticate to Big Panda to push alerts and issues. Follow the below given steps below to generate an organization token

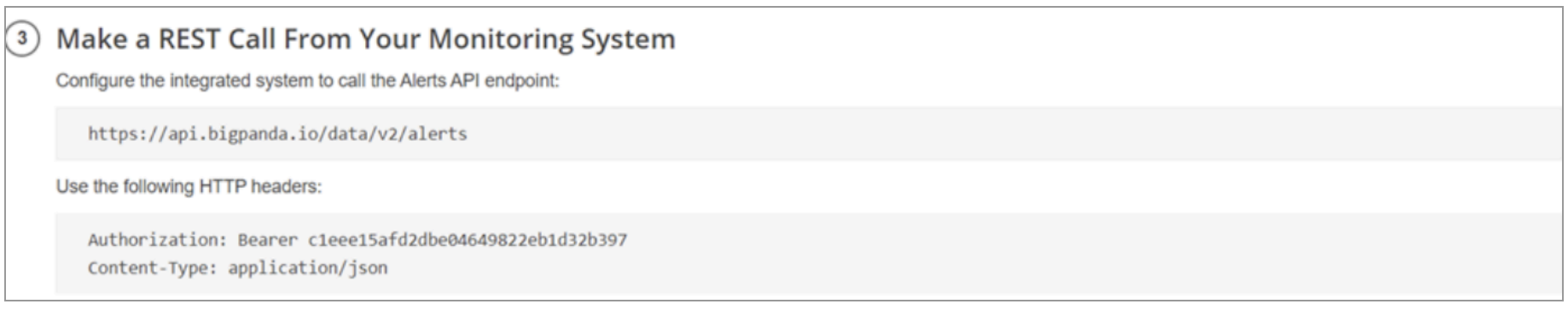
Step 1: Navigate to the Integrations page in Big Panda



Step 2: Click on any existing integration



Step 3: Click on the “configuration” option



Step 4: On this page, copy the Bearer token from Section 3

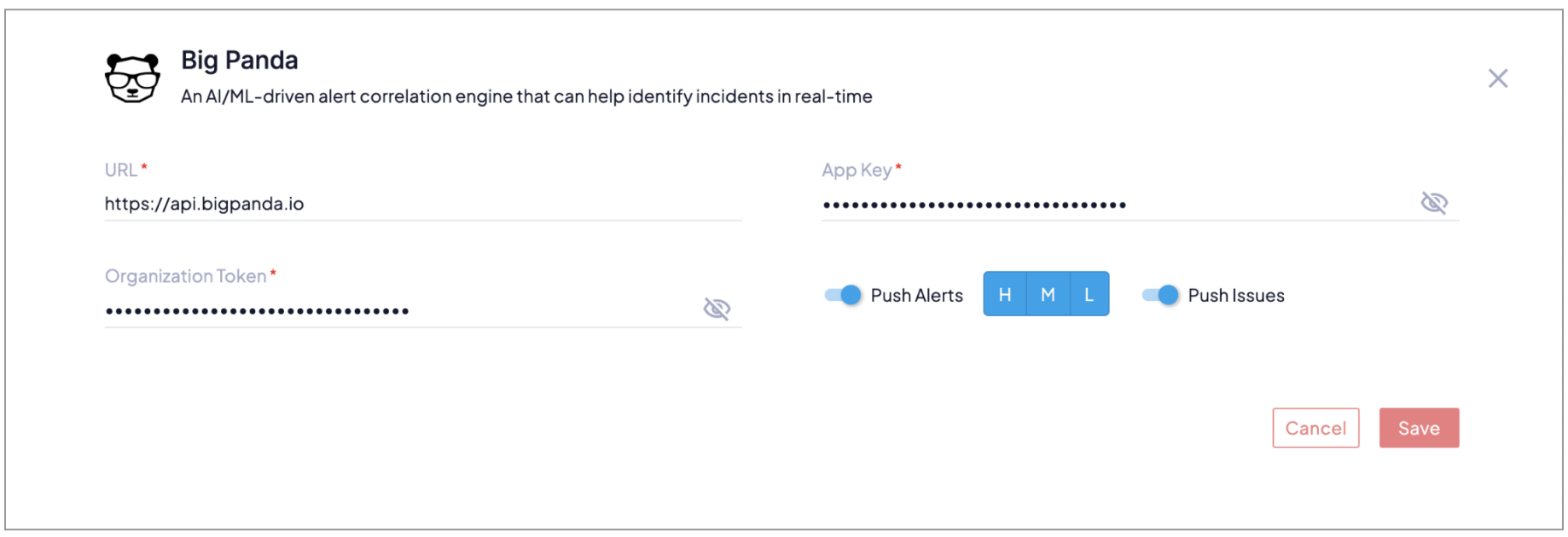
Save this bearer token to use as an organization token on the integration page.

Integration in Quest DQ

Follow the steps below to integrate Big Panda in Quest DQ.

Step 1: Log in to Quest DQ, and Navigate to Settings -> Connect -> Integrations
Step 2: Click on Big Panda
Step 3: In the Big Panda, Integration Page provide the following details:

- URL
- App KEY (Generated from Big Panda)
- Organization Token (Generated from Big Panda)
- Enable Push Alerts to create alerts as incidents in Big Panda – Select the priority of alerts to be pushed to Big Panda
- Enable Push Issues to create issues as incidents in Big Panda
- Enable Webhook for bi-directional sync



Step 4: Click on save after providing the above-mentioned details

Once connected, the alerts and issues should be automatically created as incidents in Big Panda

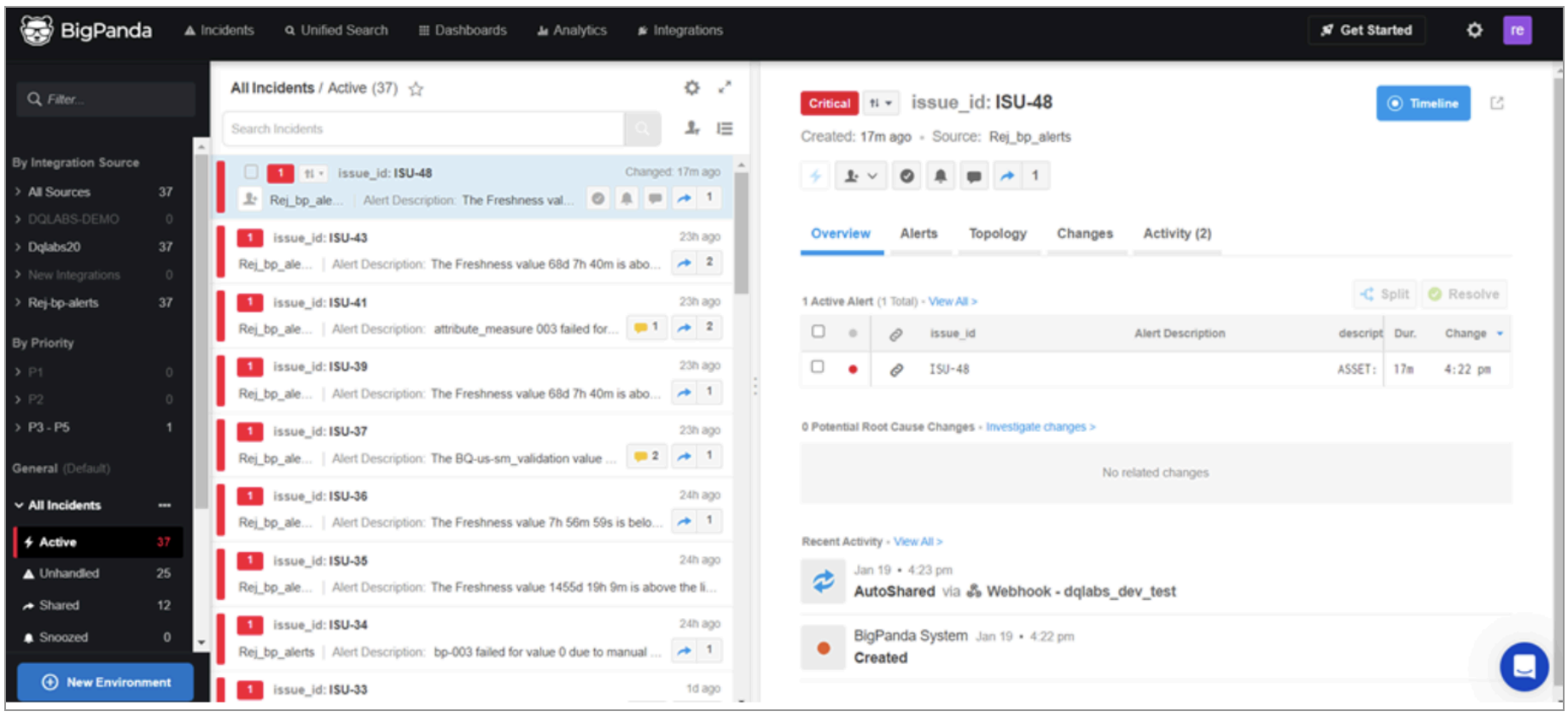
Workflow:

Once the integration is complete, the asset/issue created in Quest DQ will be created as incidents in Big Panda automatically. The change in the column in Quest DQ will automatically update the column for the respective incidents in Big Panda and vice versa. Each incident in Big Panda will be distinguished by a unique issue ID from Quest DQ for incidents created for the issue and a use measures ID for incidents created for alerts.

The following details will be pushed to Big Panda to create an incident for an alert in Quest DQ

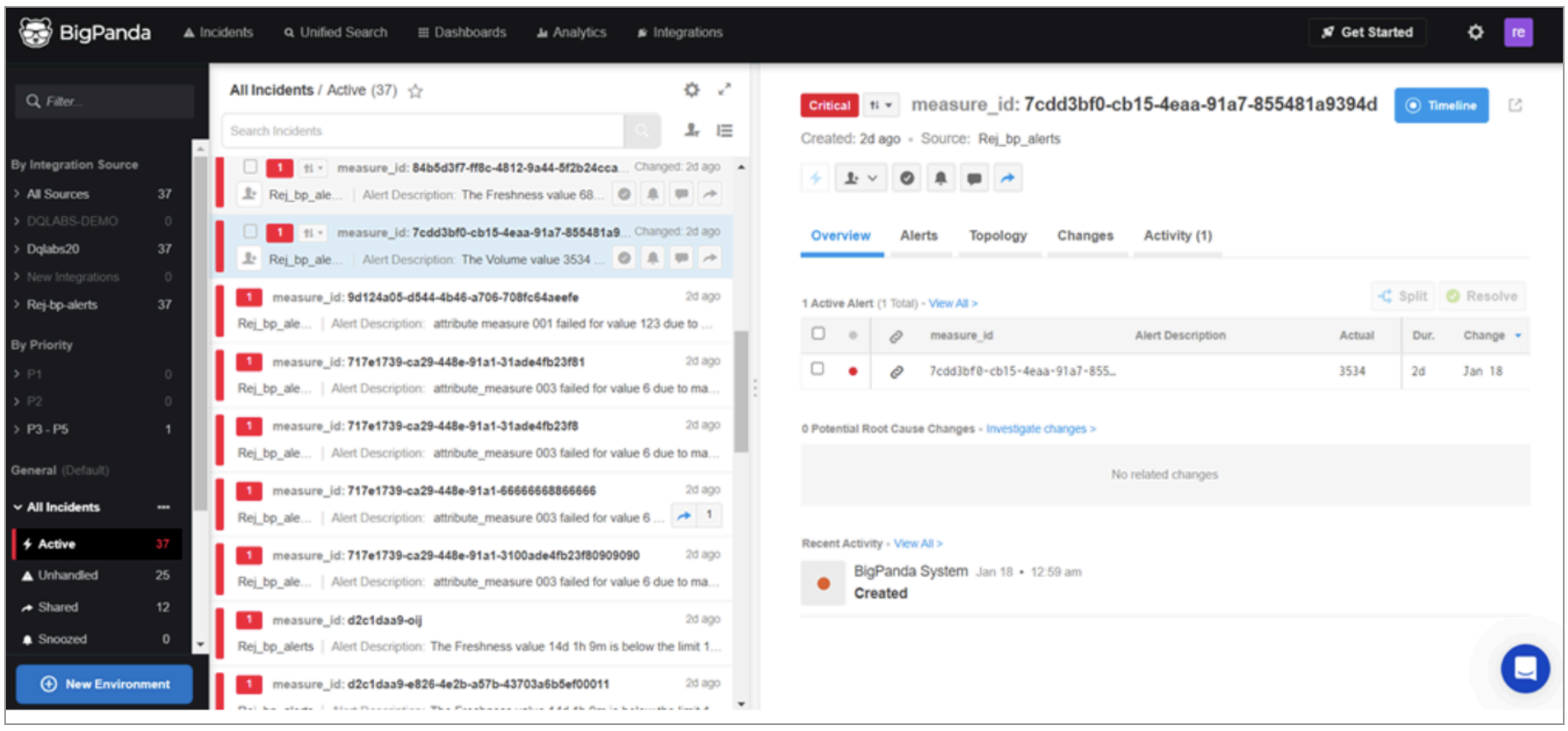
- Source Quest DQ
- Issue ID
- Type - Alert
- Connection Name
- Asset Name
- Attribute Name
- Application
- Domain
- Measure Name
- Measure Type
- Measure Level
- Alert Priority
- Alert Description

- Lower threshold
- Upper threshold
- Expected
- Actual
- Deviation
- Created Date time
- Alert Link



The following details will be pushed to Big Panda for creating an incident for an issue in Quest DQ:

- Source Quest DQ
- Measure ID
- Type - Issues
- Connection Name
- Asset Name
- Attribute Name
- Measure Name
- Measure Type
- Measure Level
- Alert Priority
- Alert Description
- Issue ID
- Issue Name
- Issue Description
- Issue Status
- Issue Priority
- Assignees
- Reported By
- Application
- Domain
- Lower threshold
- Upper threshold
- Expected
- Actual
- Deviation
- Created Date time
- Issue Link



The user can go to the respective alert/issue in Quest DQ by clicking on the asset/issue link in an incident in Big Panda.

SSO INTEGRATION

Pre-Requisites for SAML SSO Setup

1. Domain & HTTPS Configuration

- A **domain name** (e.g., app.mycompany.com) for your application
- HTTPS enabled with a **valid SSL certificate** (self-signed only for internal/dev use)
- The app should be **accessible over the network** (public or internal, depending on use)

2. SAML-Capable Application (Service Provider)

- The application must support the **SAML 2.0** protocol
- It should allow:
  - Uploading or entering IdP metadatA
  - Configuring Assertion Consumer Service (ACS) URL
  - Configuring Entity ID (SP Identifier)
  - Optionally, providing SP metadata

3. Identity Provider (IdP) Configuration

- An IdP that supports SAML 2.0 (e.g., Azure AD, Okta, Ping Identity)
- An **SSO Application** set up on the IdP side
- Ability to export or access:
  - **IdP metadata XML**, or
  - **SSO URL**, **Entity ID**

Quest DQ allows you to integrate your existing Azure Active Directory identity provider and access the platform using Single Sign On. Using SAML, all users in the domain will be able to log in to the sign-in page for Quest DQ

Quest DQ uses email as the claim information, and you need to create a federation.xml file in your SAML provider and then update it in the Quest DQ platform. The following section provides the steps involved in configuring Azure AD in Quest DQ for single sign-on.

- Creation of Federation file in Microsoft Azure AD.
- Configuration of the Federation file and Private key in Quest DQ.
- Logging in using SSO

IBM SAML

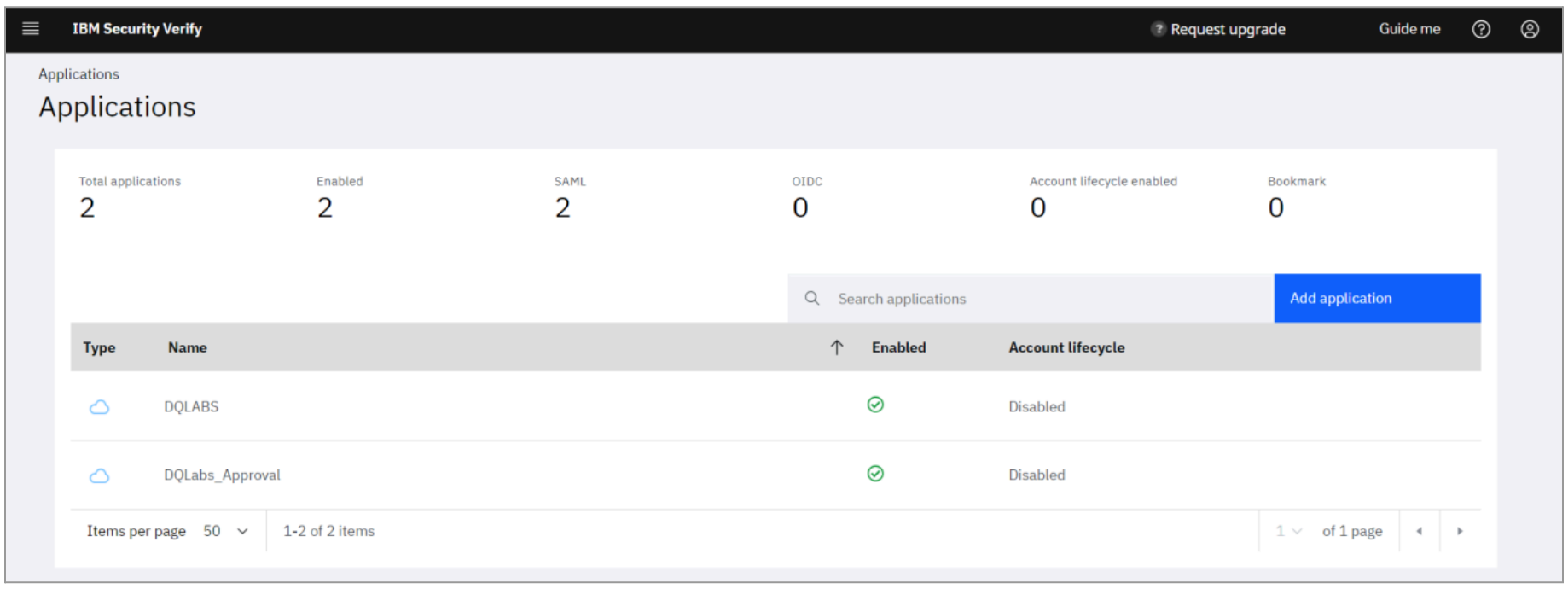
Quest DQ allows you to integrate your existing IBM SAML provider and access the platform using Single Sign On. Using SAML, all users in the domain can log to the sign-in page into Quest DQ. Quest DQ uses email as the claim information, and you need to create a federation.xml file in your SAML provider and then update it in the Quest DQ platform. The following section provides the steps in configuring IBM SAML in Quest DQ for single sign-on.

- Creation of Federation file in IBM Saml
- Configuration of Federation file in Quest DQ.
- Logging in using SSO.

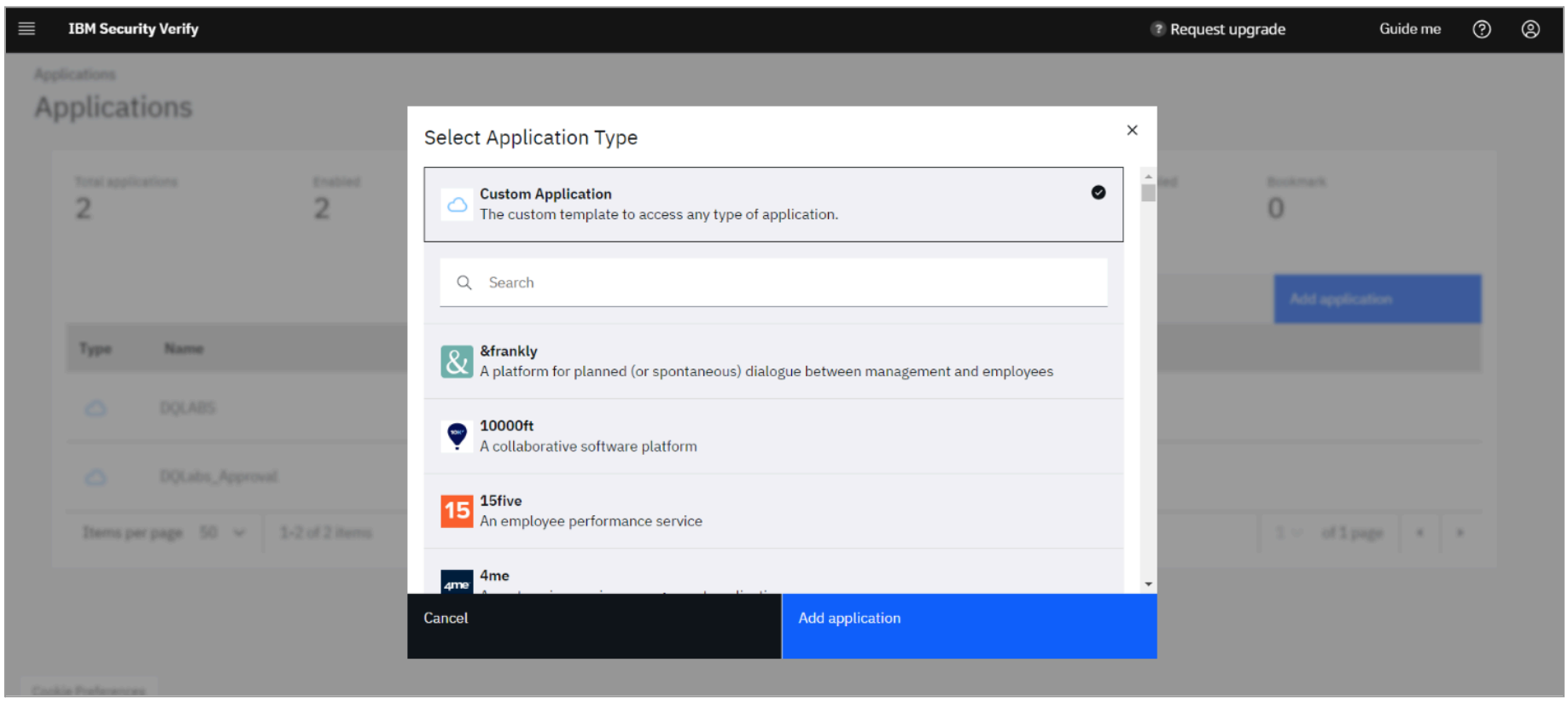
Creation of Federation file in IBM SAML

Step 1: Login into the IBM Security platform

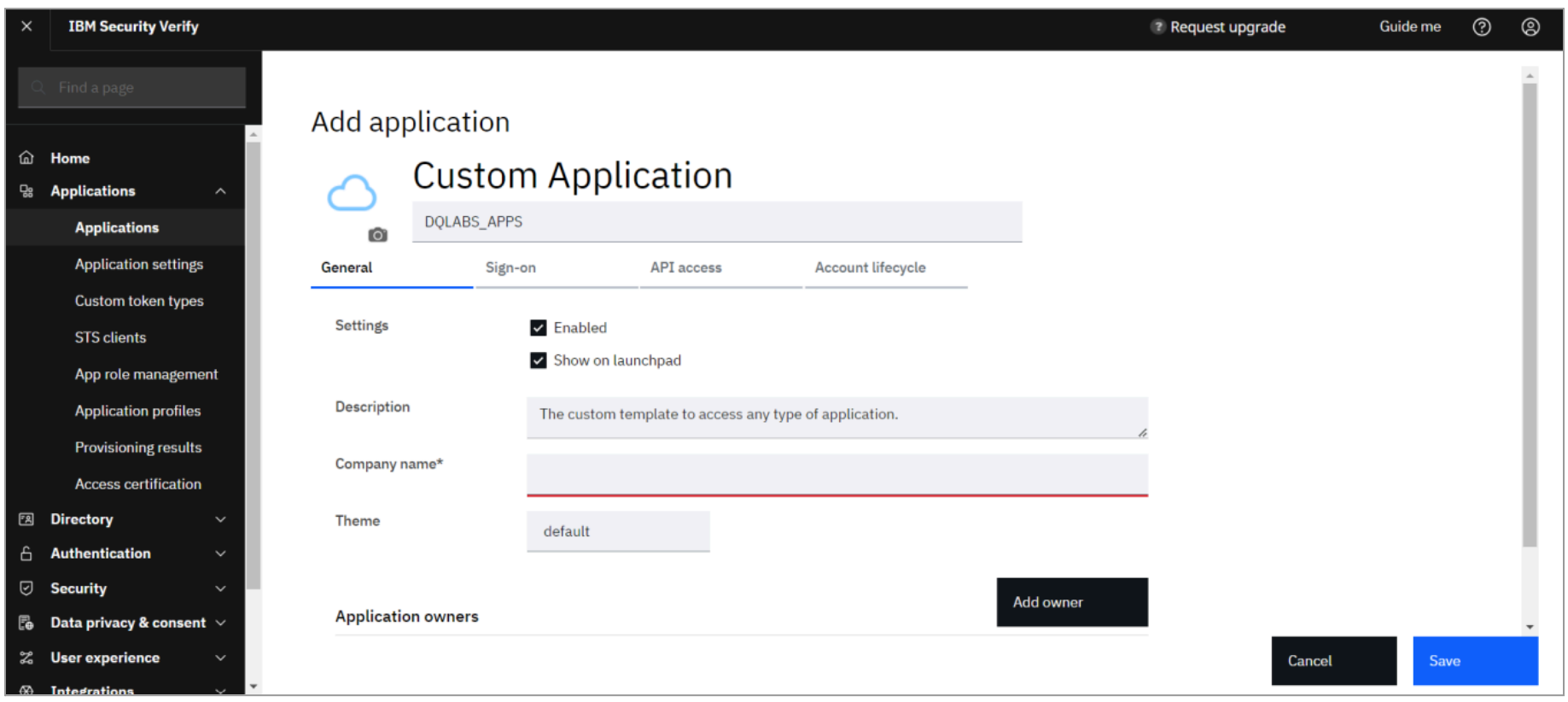
Step 2: Navigate to Applications and click on Add Application



Step 3: Select “Custom Application” and select click Add Application



Step 4: On the Add Application Page, provide the application name, Company Name, and Owner and click Save

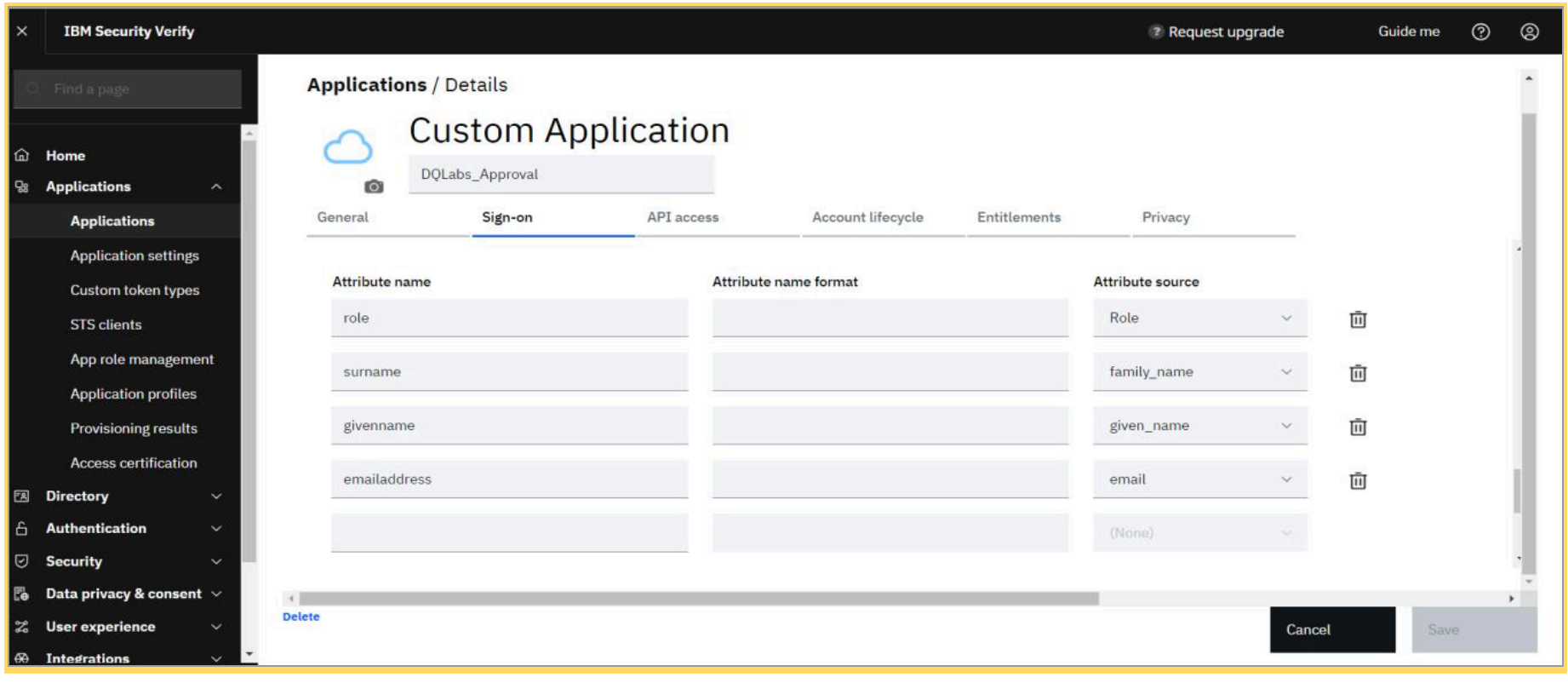


Step 5: On the Sign On Page, provide the following details and click on Save

- Uncheck the “Use Metadata“ field
- Provider ID - Entity ID URL
- Assertion consumer service URL (HTTP-POST) - ACS URL
- Target URL - Sign In URL
- Service provider SSO URL -Sign-In URL
- Map the following attributes

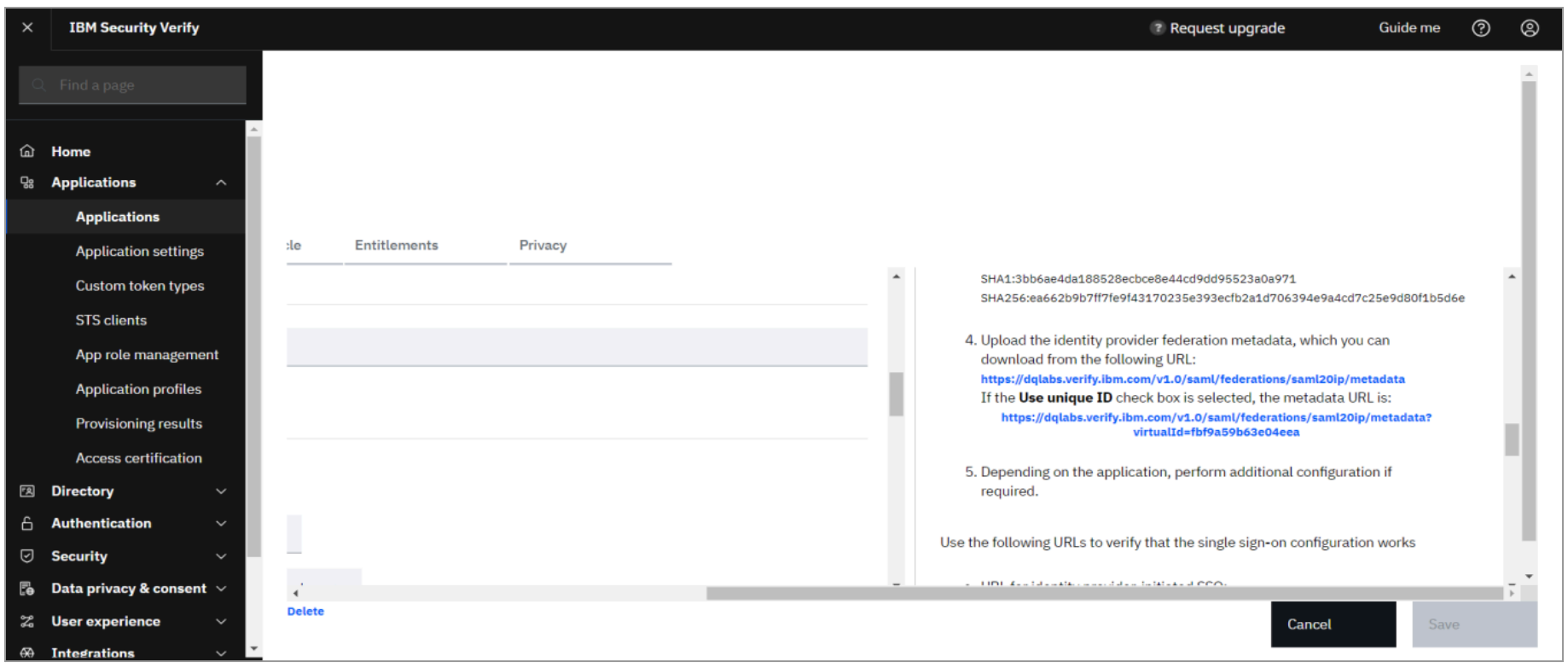
Quest DQ 2.0	**IBM**
role	Role
emailaddress	Email Address
givenname	Given Name
surname	Family Name

NOTE: Role attribute should be added as a custom attribute



Step 6: On the Entitlement tab, add the users and click on save

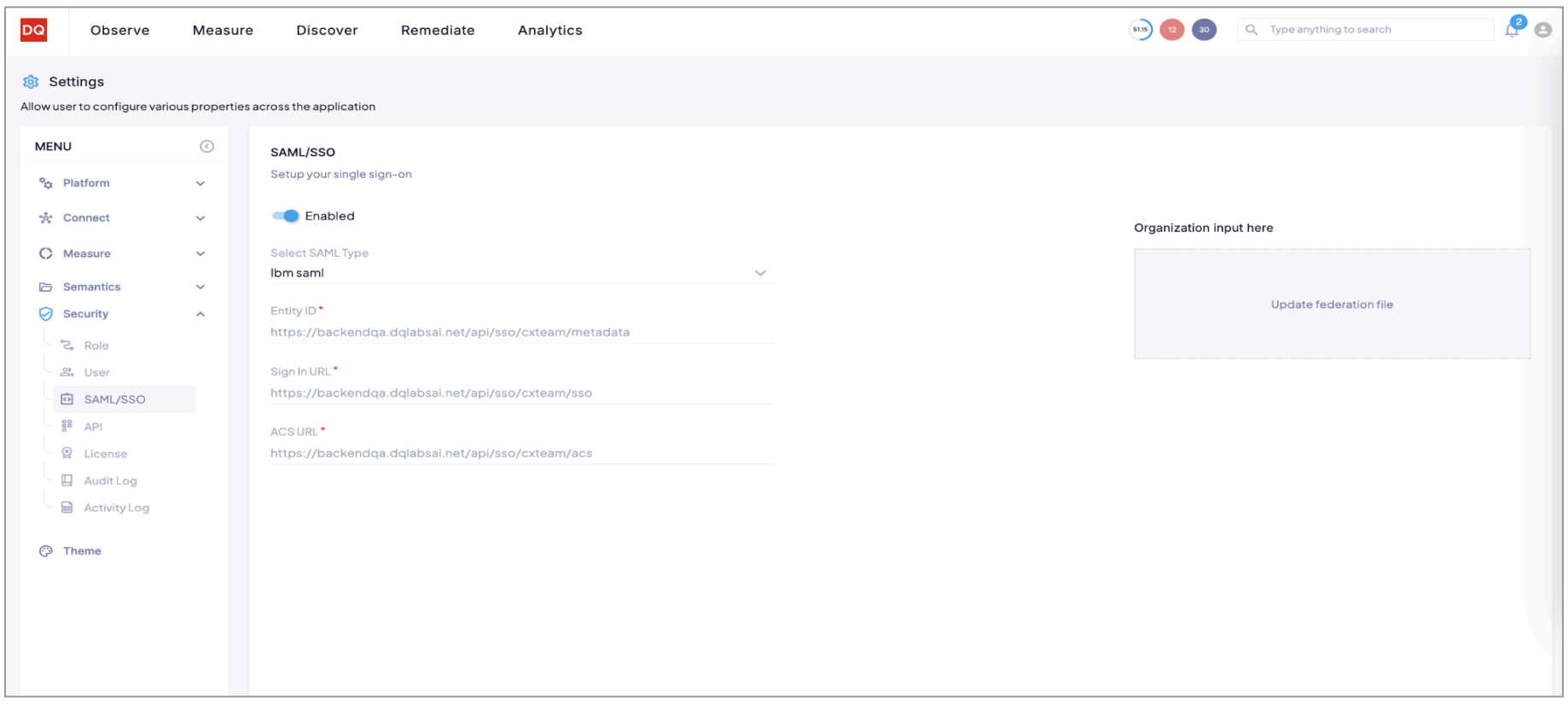
Step 7: Once created, click on edit for the application, scroll to the right, and download the metadata file



Configuration of Federation file in Quest DQ.

Step 1: Log in to the Quest DQ platform > navigate to Integrations in settings and choose IBM

Step 2: Upload the Federation file and Click on Save.



Login into Quest DQ using SSO

- Go to the Quest DQ login page and click on SSO.
- Now the user will get navigated to the corresponding SSO login page.
- Provide the valid credentials and the user will be logged into the Quest DQ portal

User Provisioning

- Authorized users provisioned in OKTA, can log into the Quest DQ Portal using the SSO button on the login screen.
- Quest DQ will automatically provision the user in the Portal with the “USER” role.
- Users who have Admin access in the Quest DQ Portal can modify the role that is assigned to the user based on their persona.

Okta

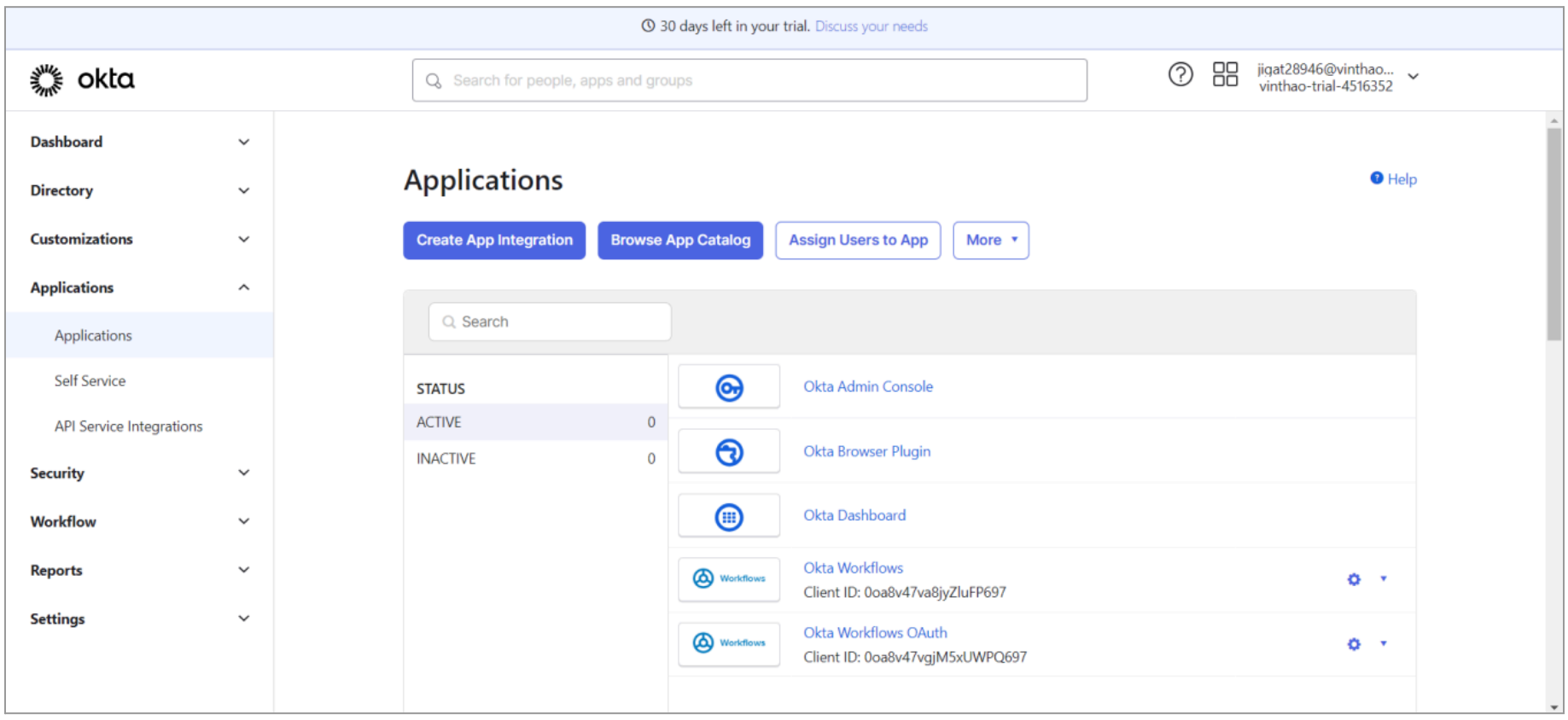
Quest DQ allows you to integrate your OKTA and access the platform using Single Sign On. Using SAML all users in the domain will be able to login to the sign-in page into Quest DQ.

Quest DQ uses email as the claim information, and you need to create a federation.xml file in your SAML provider and then update it in the Quest DQ platform. The following section provides the steps involved in configuring Okta in Quest DQ for single sign-on.

- Creation of Federation file in OKTA
- Configuration of Federation file in Quest DQ.
- Logging in using SSO.

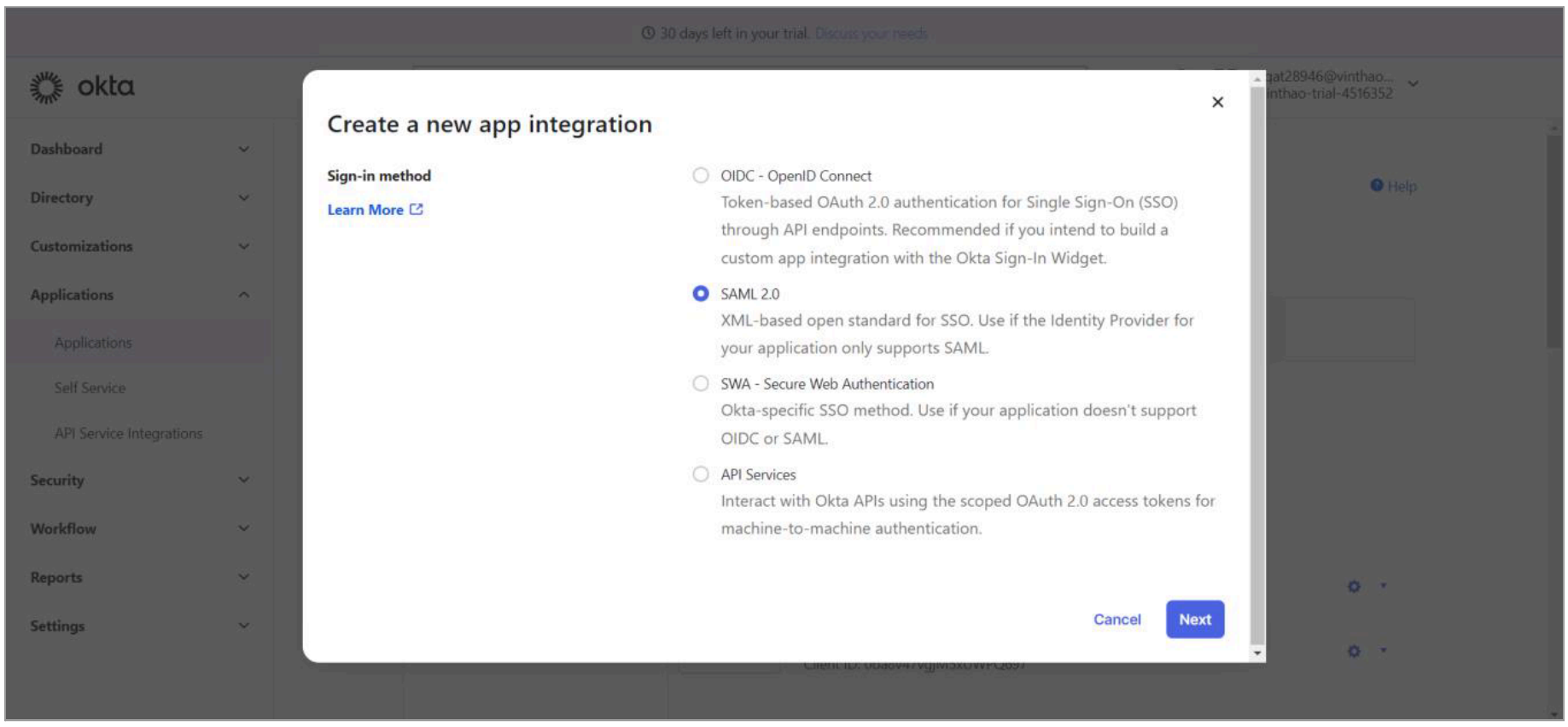
Creation of Federation file in OKTA

Step 1: Navigate to Applications



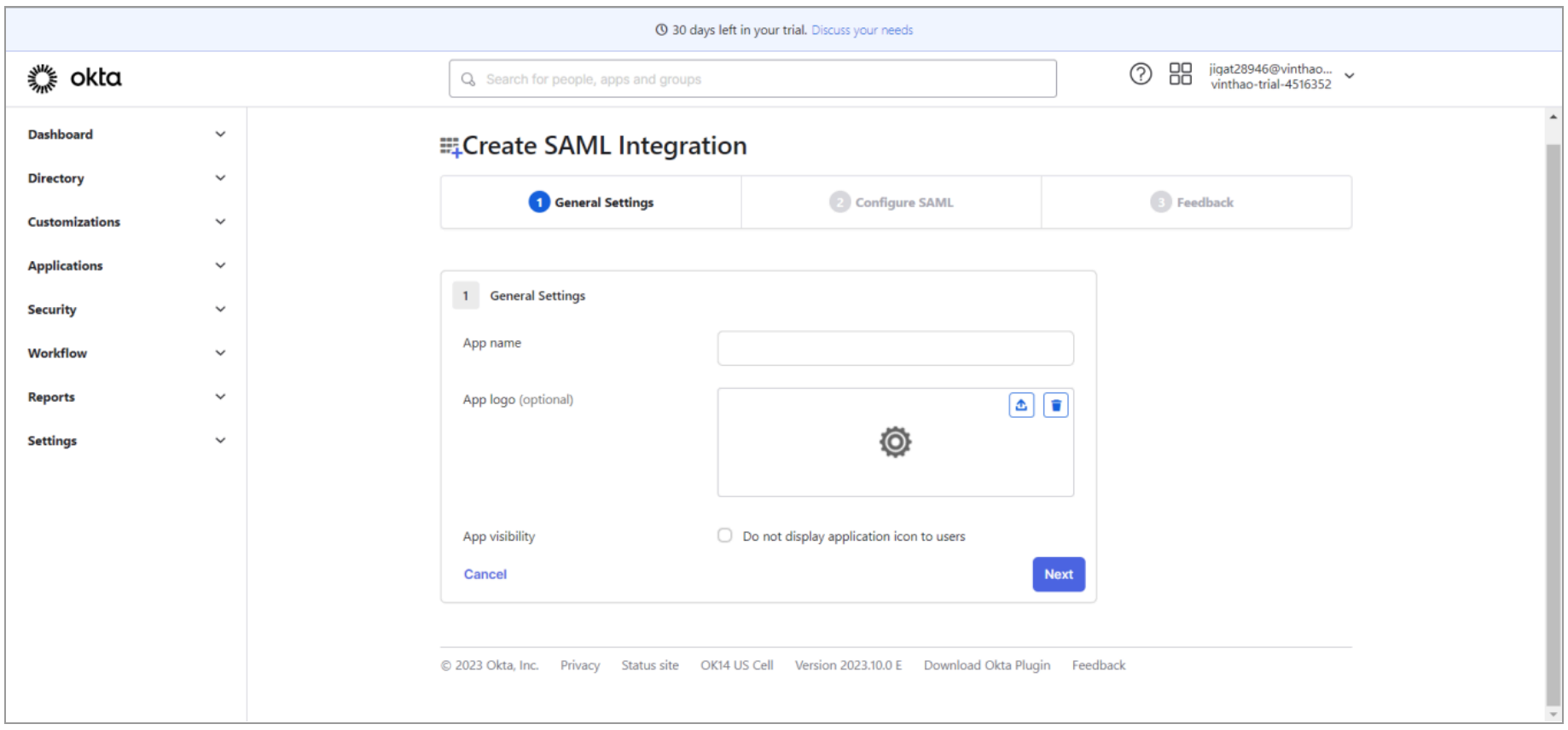
Step 2: Click on Create APP Integrations

Step 3: Select SAML 2.0 and click on Next.



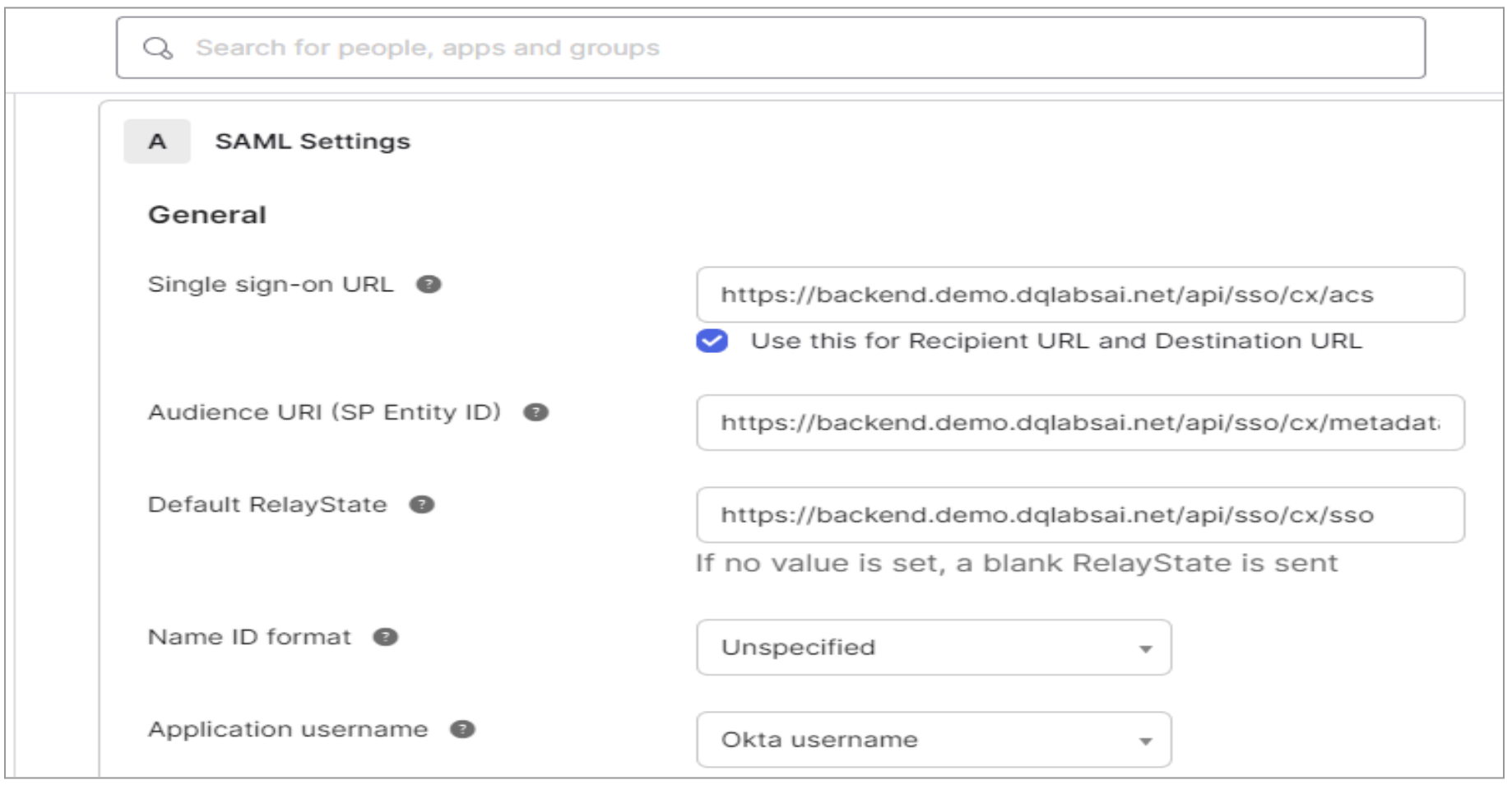
Step 4: Provide the following details and click on next

- App Name
- App Logo
- App Visibility



Step 5: On the "Configure SAML" Page, provide the following details and click on next

- Single sign-on URL - **ACS URL** from Quest DQ(Single Sign-on URL is a term used by OKTA for the Assertion Consumer Service URL, which is the ACS URL)
- Audience URI (SP Entity ID) - **Entity ID URL** from Quest DQ.
- Default RelayState - Here we have to input the **SSO URL** from Quest DQ.

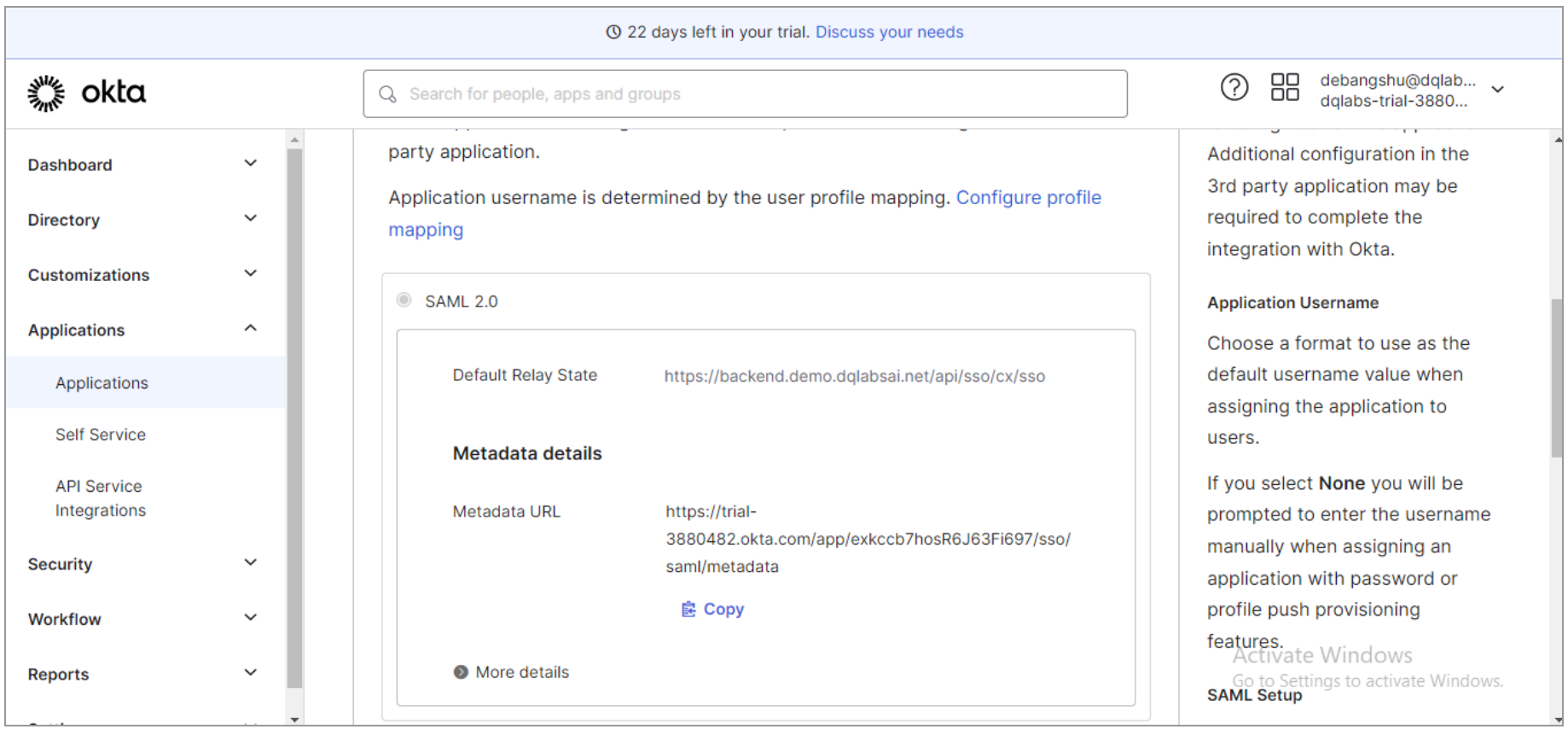


- Attribute Mapping as per the below table

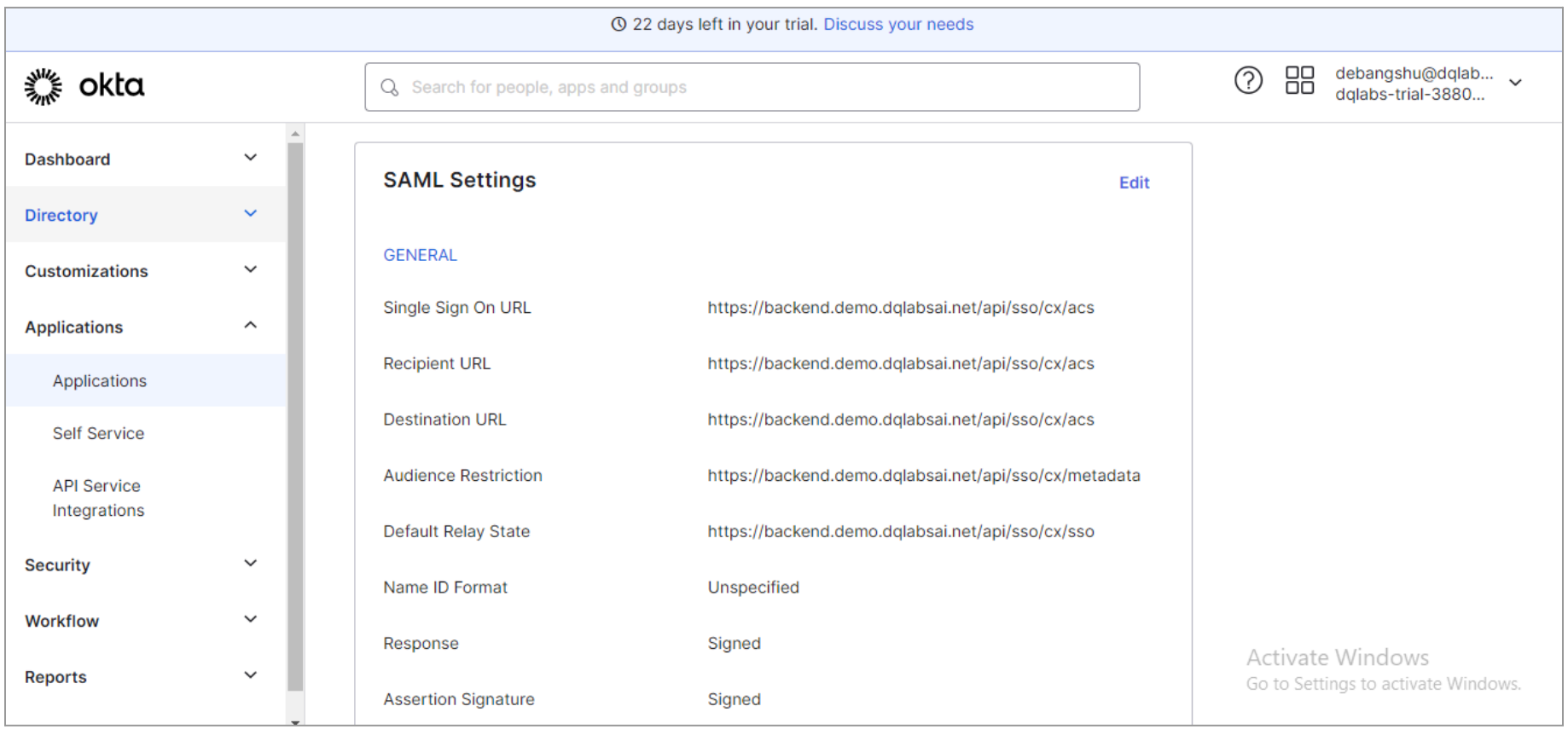
Quest DQ 2.0	**OKTA**
emailaddress	user.email
givenname	user.firstname
surname	user.lastname

Step 6: After mapping the attributes we can scroll down and Click on Next and then Click on FINISH to complete the Integration configuration in OKTA.

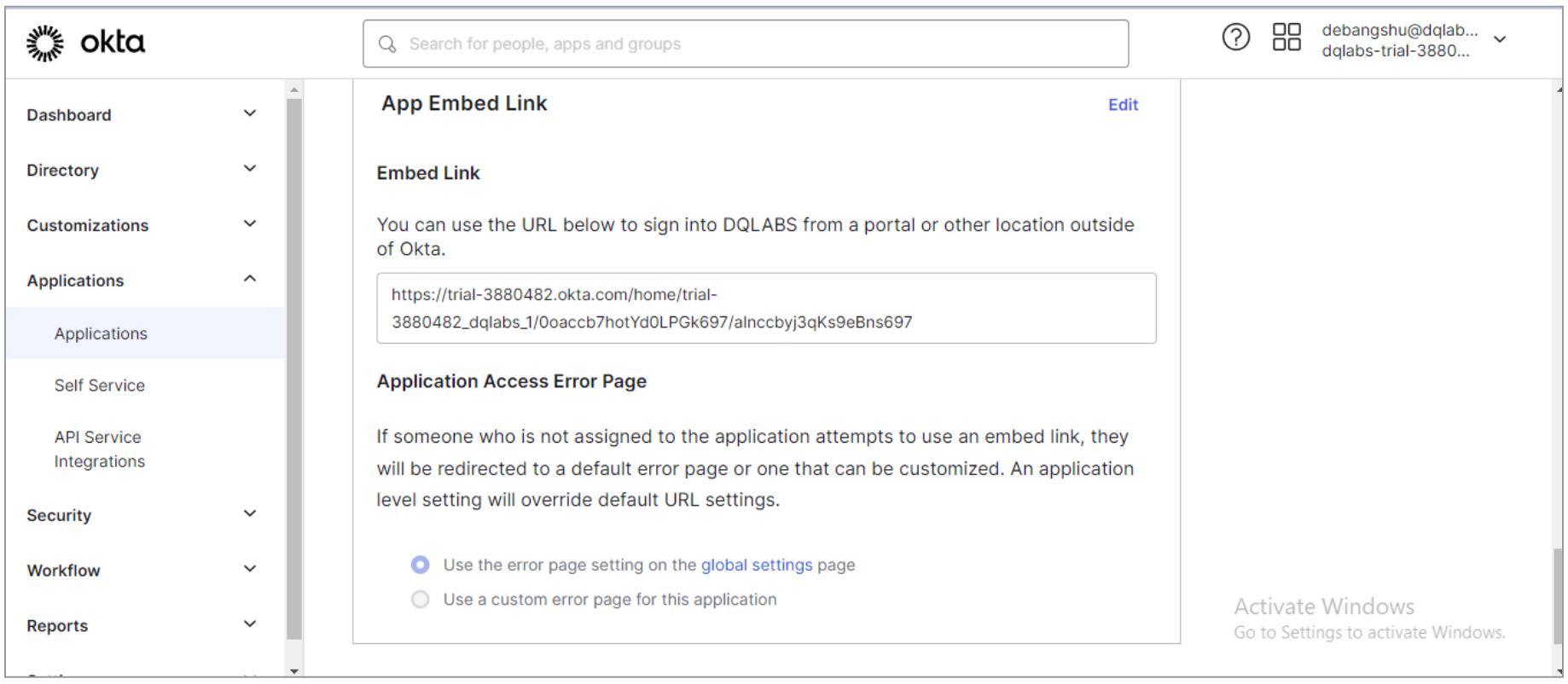
Step 7: On the sign-on page, copy the metadata URL, paste it into the browser, and save it as a federation.xml file, this will be the Federation File that we upload in the Quest DQ application, under OKTA Integration.



Step 8: Make sure that the **SAML Settings** look like the below screenshot once you have finished configuring your application in OKTA



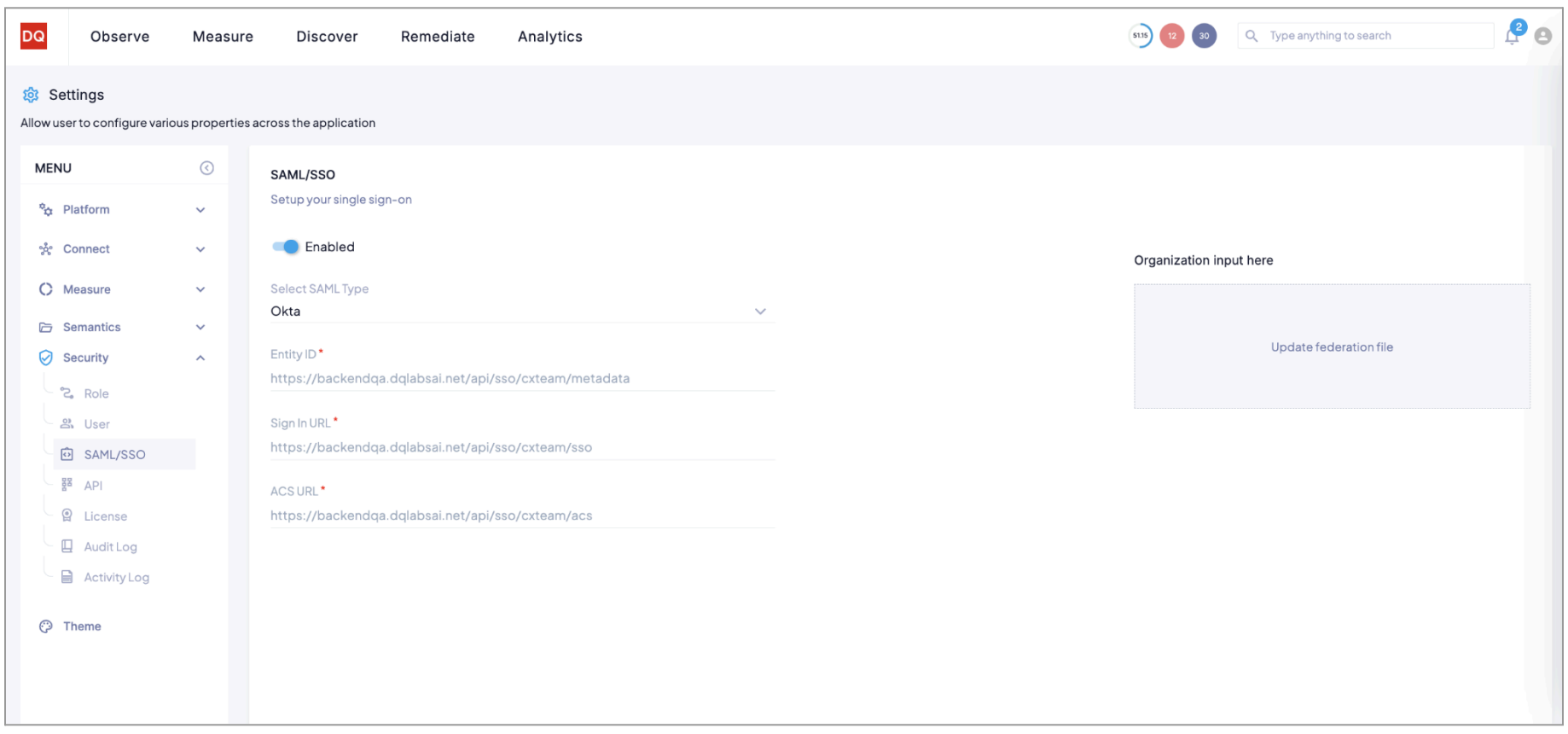
Step 9: Make sure to try out the Integration by using the URL which you get below under **App Embed Link**



Configuration of Federation file in Quest DQ

Step 1: Login to the Quest DQ platform, navigate to Integrations in settings, and choose OKTA

Step 2: Upload the Federation file and Click on Save.



Login into Quest DQ using SSO

- Go to the Quest DQ login page and click on SSO.
- Now the user will get navigated to the corresponding SSO login page.
- Provide the valid credentials and the user will be logged into the Quest DQ portal.

User Provisioning

- Authorized users provisioned in OKTA, can log into the Quest DQ Portal using the SSO button on the login screen.
- Quest DQ will automatically provision the user in the Portal with the “USER” role.
- Users who have Admin access in the Quest DQ Portal can modify the role that is assigned to the user based on their persona.

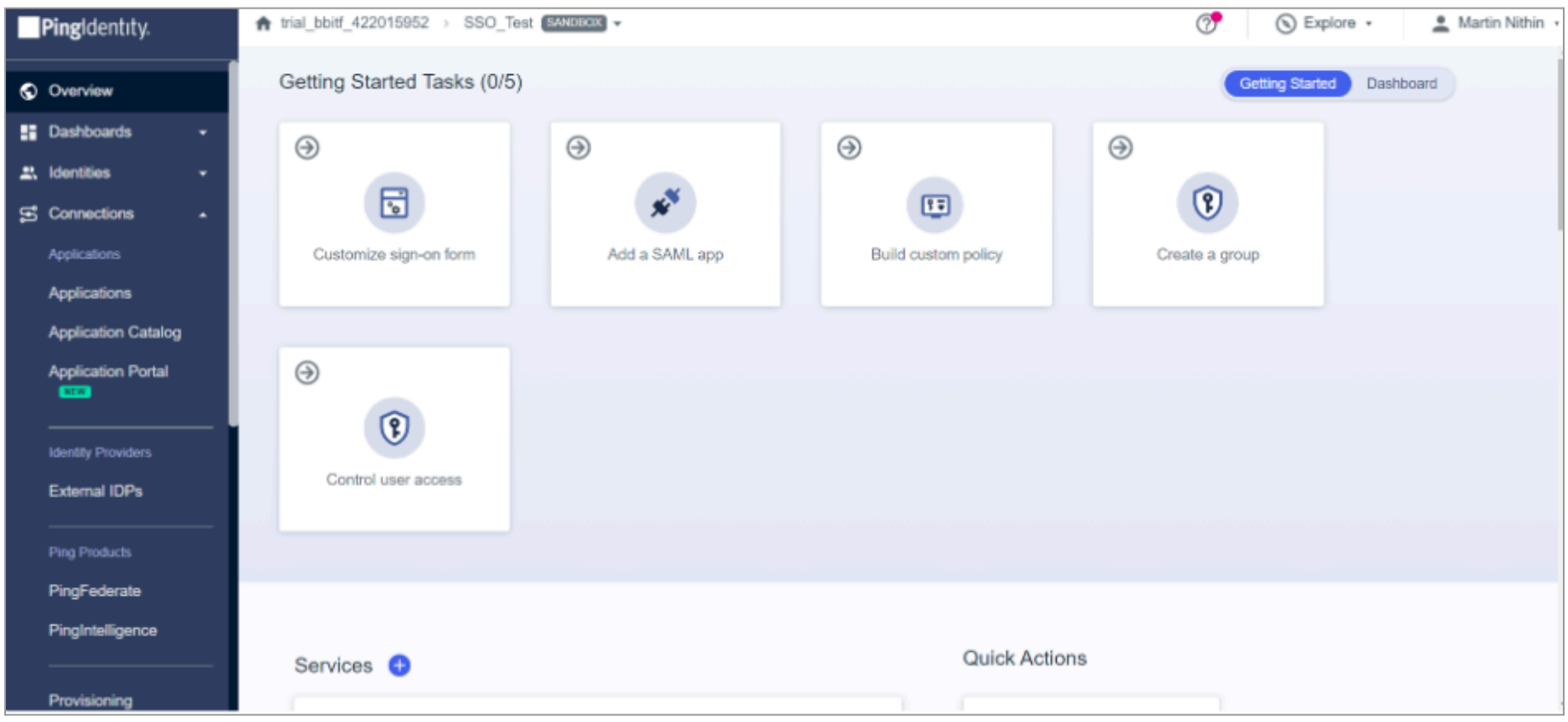
Ping Federate

Quest DQ allows you to integrate your existing Ping Federate identity provider and access the platform using Single Sign On. Using SAML all users in the domain will be able to login to the sign-in page into Quest DQ. Quest DQ uses email as the claim information, and you need to create a federation.xml file in your SAML provider and then update them in the Quest DQ platform. The following section provides the steps involved in configuring PingFederate in Quest DQ for single sign-on.

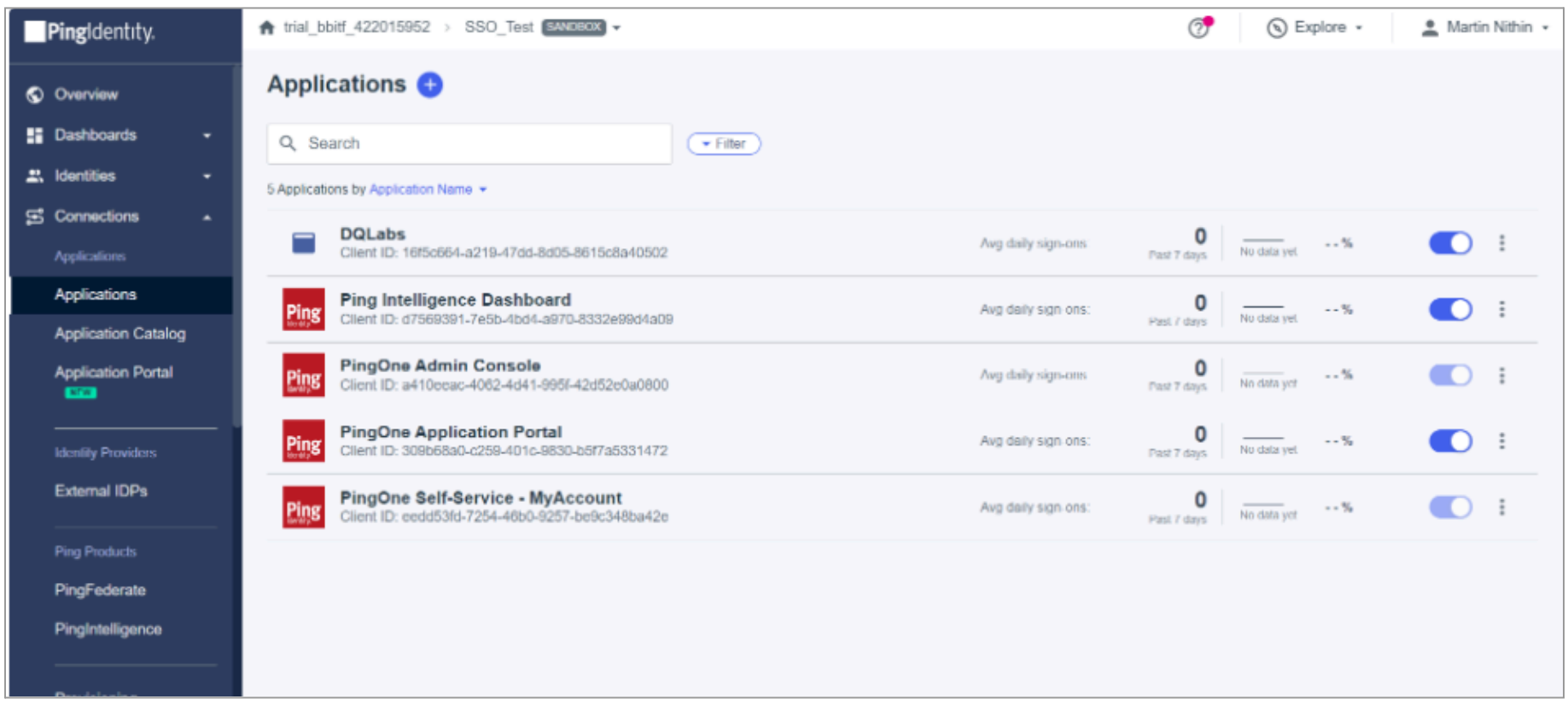
Creating Federation File in Ping Federate

Step 1: Login into the Ping Identity platform

Step 2: Navigate and click on the "Add a SAML app" icon

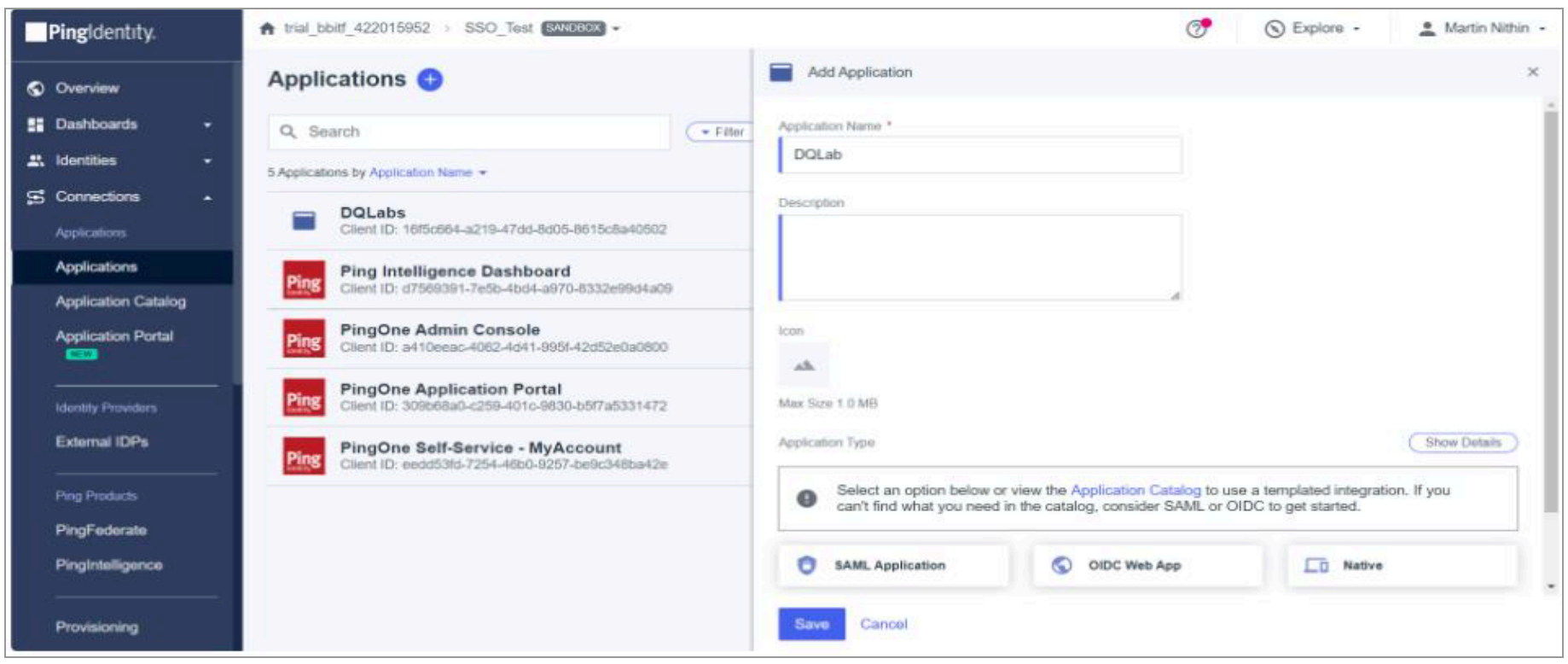


Step 3: Click on the "+" icon to add a new application

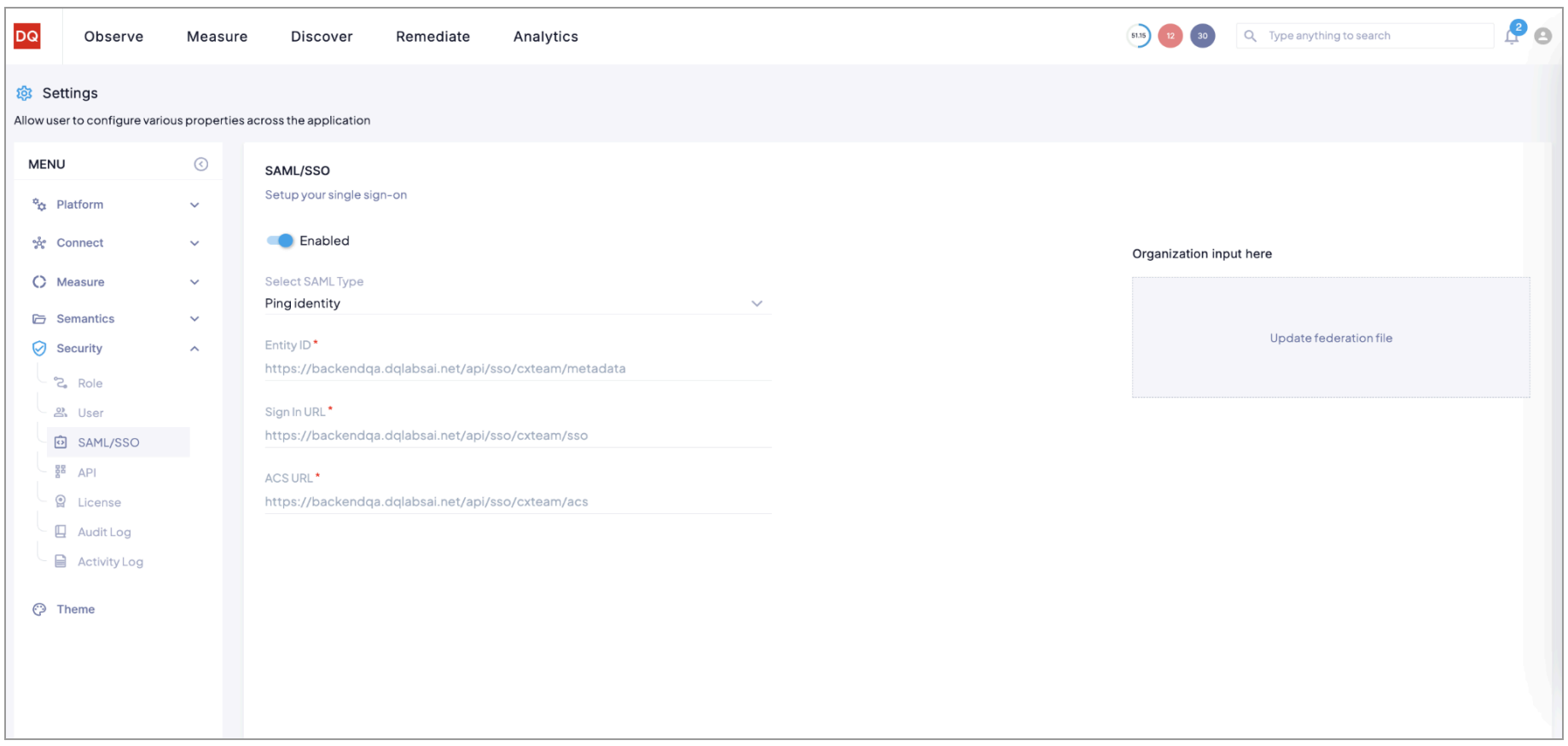


Step 4: Provide the following information on the Add application screen and click on Save

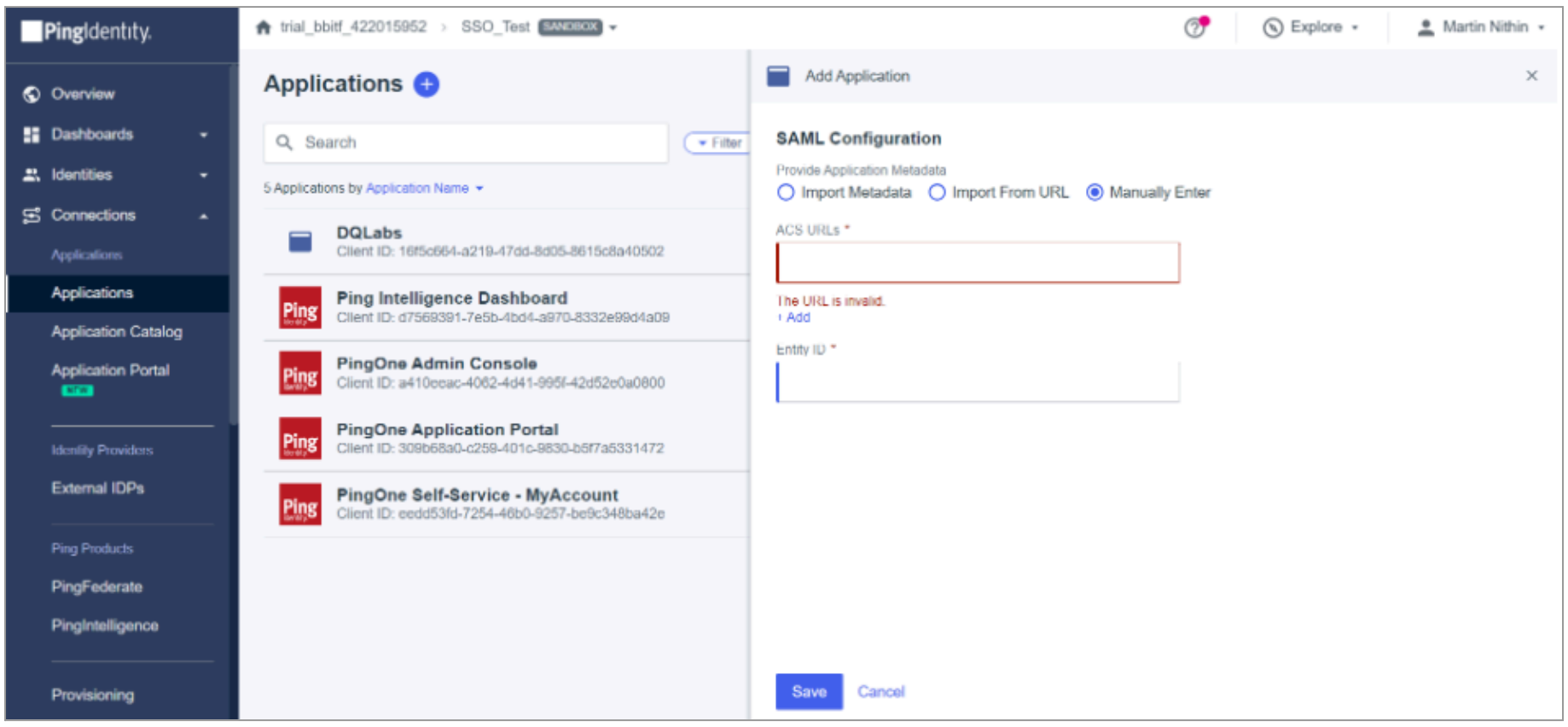
- Application Name
- Description
- Icon
- Application Type - Select SAML Application



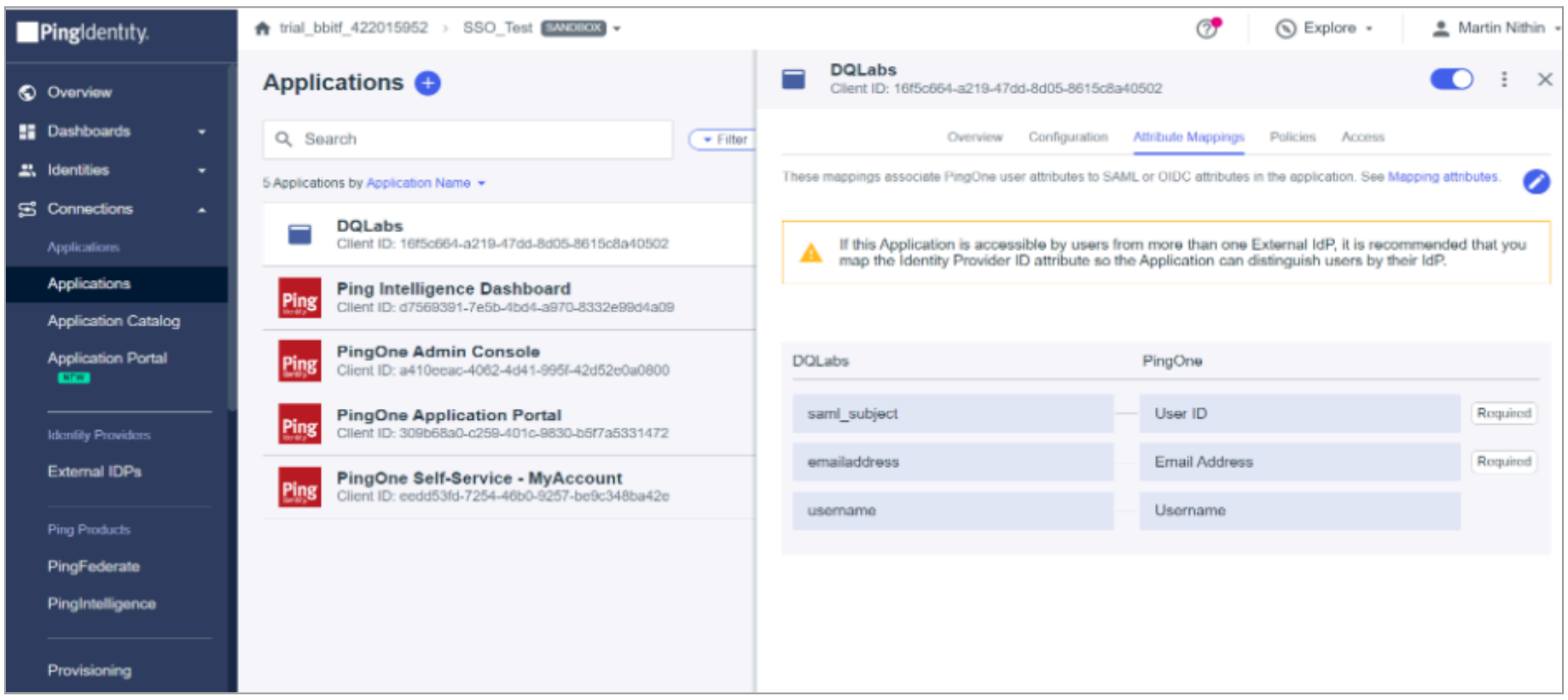
Step 5: Copy the ACS URL and Entity ID from Quest DQ, which can be fetched from the PingFederate integration page



Step 6: On the next screen in the SAML configuration click on the “Manually Enter” radio button and provide the ACS URL and Entity ID from Quest DQ and click on “Save”.



Step 7: On the created SAML application, click on Attribute Mappings



Step 8: On the Attribute Mapping screen, Click on Add and provide the following details, and click on Save

Quest DQ	**PingOne**
emailaddress	Email Address
givenname	Given Name
surname	Last Name

NOTE: To show first name and last name, map the Quest DQ and Ping one attributes that match the results needed

Step 9: Navigate to Configuration and click on Download Metadata, to download the federation.xml file

After downloading configure the federation file in Quest DQ to establish the connection

Step 10: Log in to the Quest DQ platform and navigate to Integrations in settings and choose Ping Federate

Step 11: Upload the Federation file and Click on Save.

Login into Quest DQ using SSO

- Go to the Quest DQ login page and click on SSO.
- Now the user will get navigated to the corresponding SSO login page.
- Provide the valid credentials and the user will be logged into the Quest DQ portal.

User Provisioning

- Authorized users provisioned in Ping Federate, can log in into the Quest DQ Portal using the SSO button on the login screen.
- Quest DQ will automatically provision the user in the Portal with the “USER” role.
- Users who have Admin access in the Quest DQ Portal can modify the role that is assigned to the user
- based on their persona.
- Once the role attribute is added, the roles in the AD will be mapped to the roles in Quest DQ for the users.
- When a new user login to Quest DQ via SSO, if the role is already present in the Quest DQ then that role will be mapped based on the role name.
- If the role is not present, then a new role will be created based on default settings and mapped to the user.
- If the user has already logged in and has a role, then the new roles will be mapped when logged in using SSO.
- When the existing user in Quest DQ who has been assigned a role in Quest DQ, logins with SSO, the role that is already there in the Quest DQ platform should be maintained when there are no roles in AD
- If the users log in for the first time the role in the AD should be mapped or created newly in Quest DQ based on the roles in the Quest DQ platform.
- The role-based functionality should work only if the role attribute is mapped while generating the Federation file.

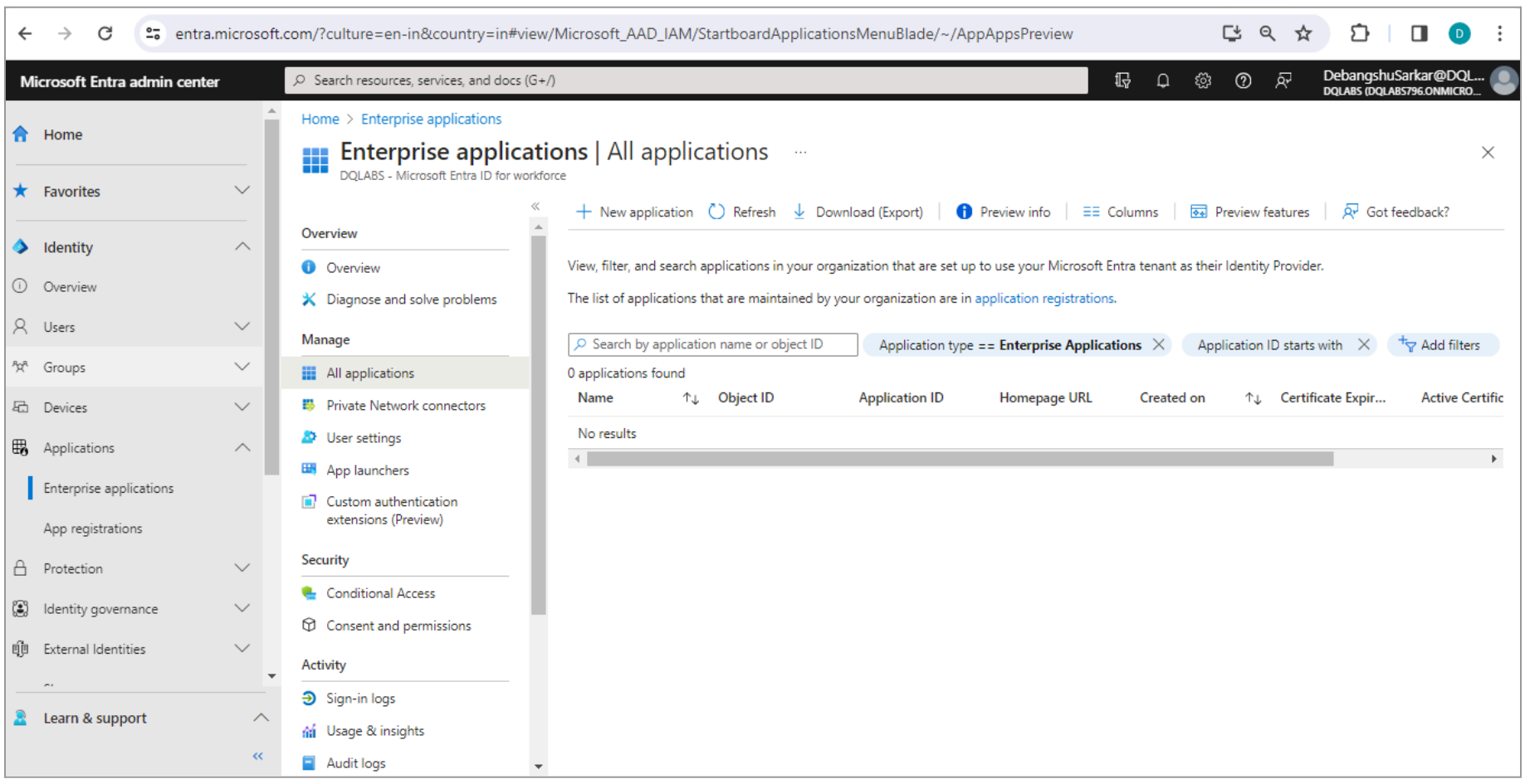
Azure Active Directory

Quest DQ allows you to integrate your existing Azure Active Directory also known as Microsoft ENTRA ID provider and access the platform using Single Sign On. Using SAML all users in the domain will be able to login to the sign-in page into Quest DQ. Quest DQ uses email as the claim information, and you need to create a federation.xml file in your SAML provider and then update it in the Quest DQ platform. The following section provides the steps involved in configuring Azure Active Directory in Quest DQ for single sign-on.

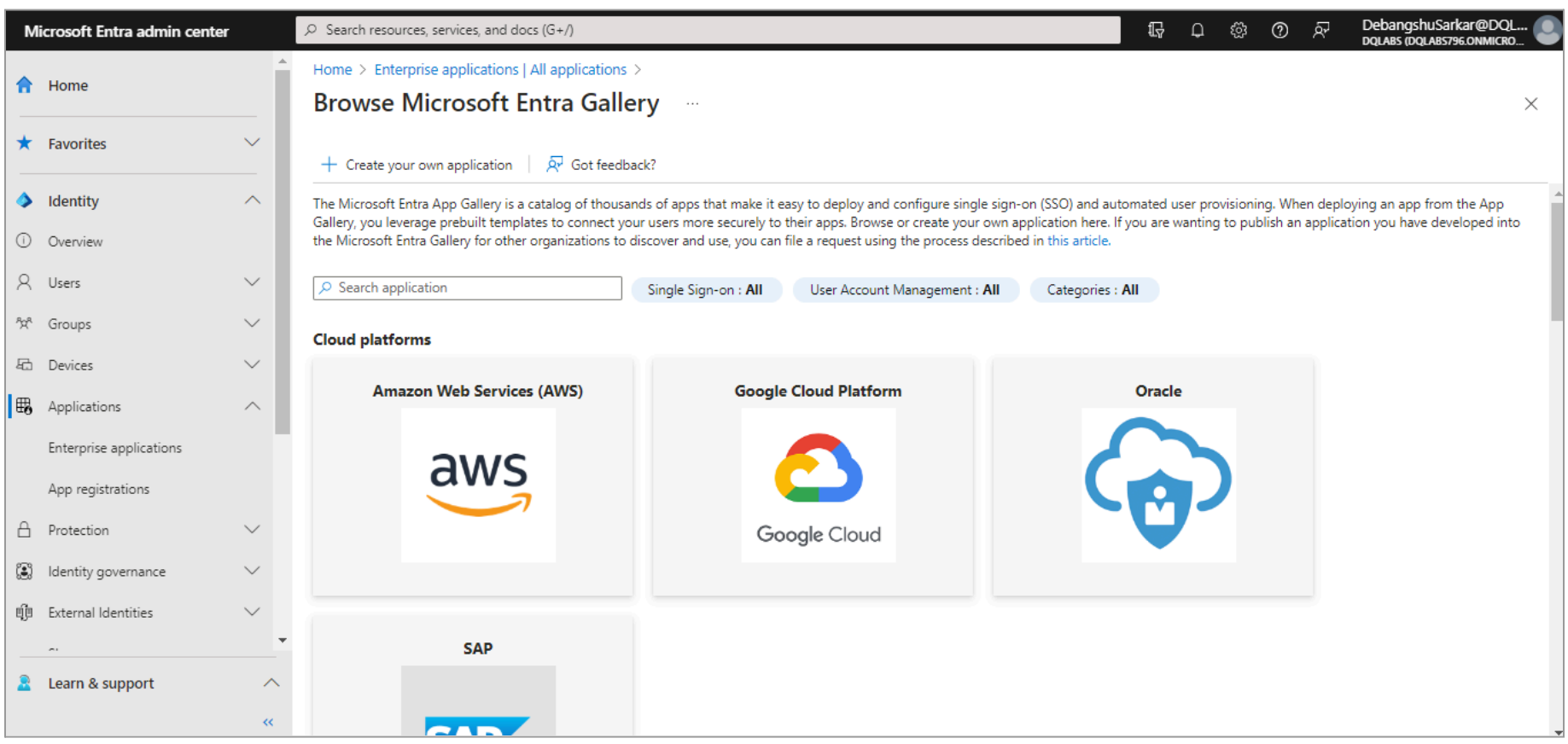
Creating Federation File in Azure Active Directory

STEP 1: Login into the Azure Active Directory platform or Microsoft Entra Admin Center

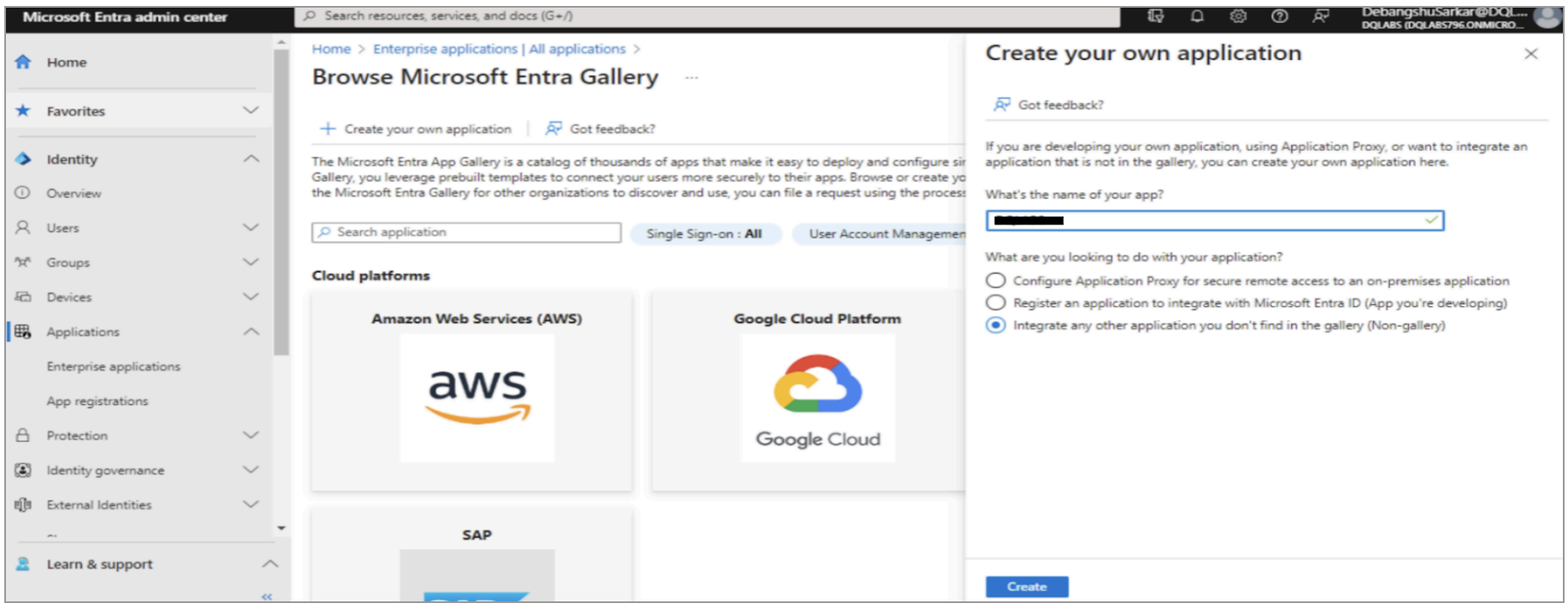
STEP 2: Navigate and click on the Drop down under **Applications** and click on **Enterprise applications**



STEP 3: Click on the "**+ Create your own application**" on the top to create Quest DQ in the Entra ID portal.

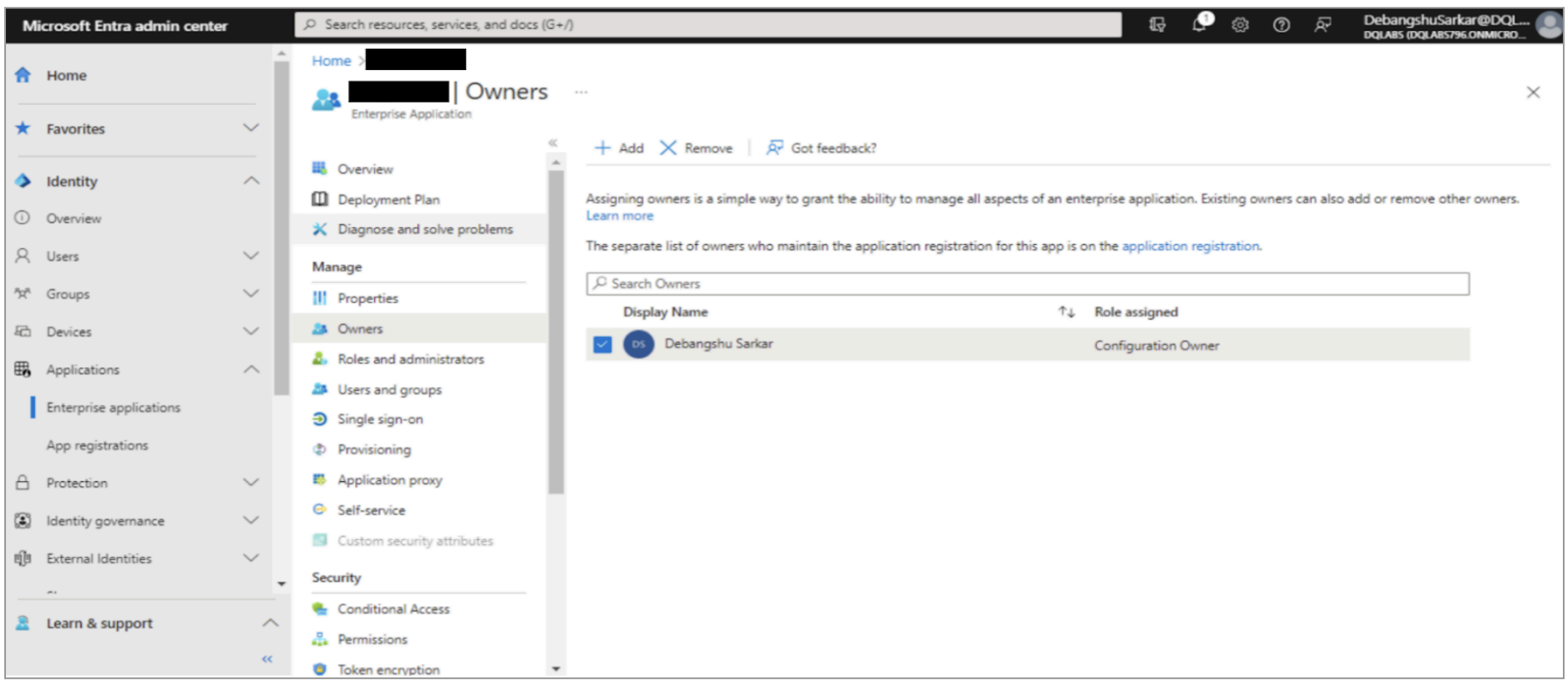


STEP 4: Give the Application a name and click on the **Create** button down below.



STEP 5: The ENTRA ID platform will now direct you to your newly created application

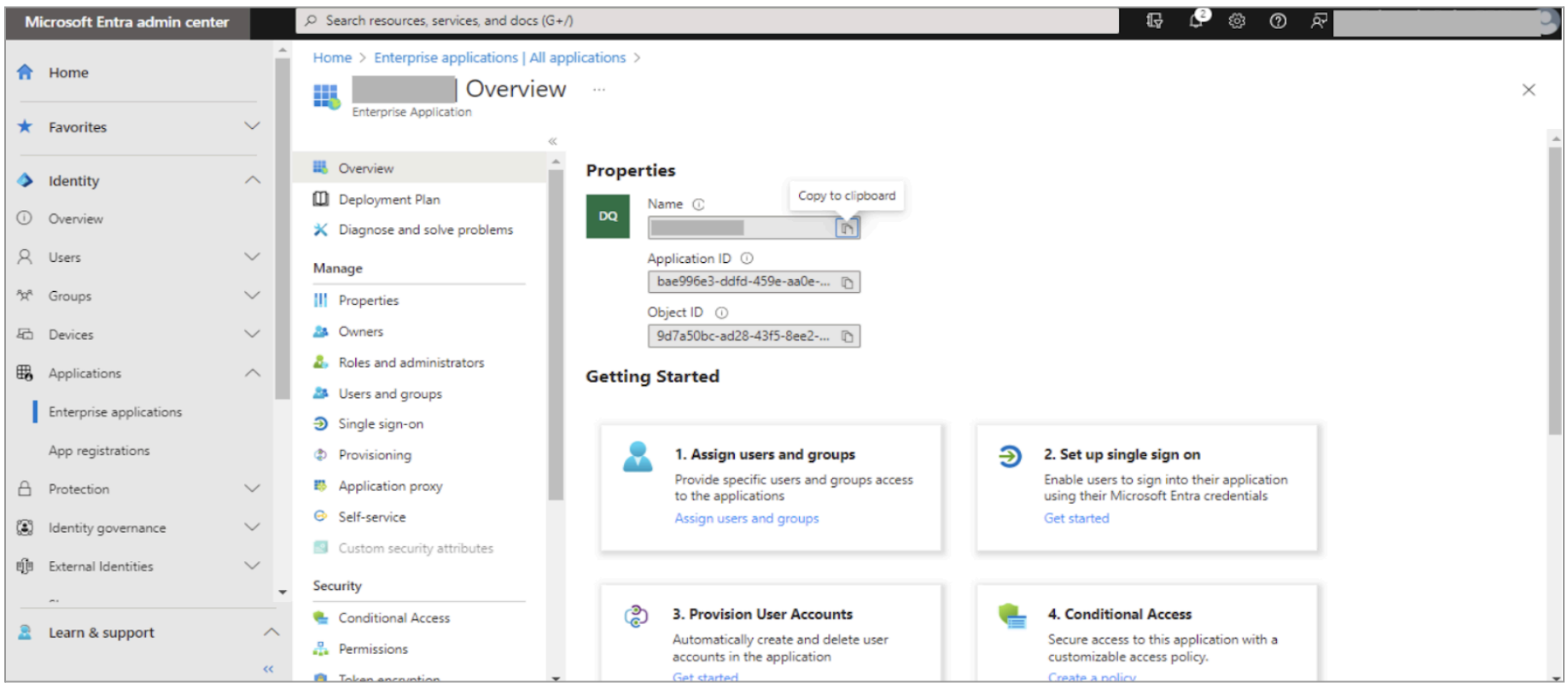
STEP 6(Optional): Add yourself as an owner if you are creating the application by navigating to **Owners** under **Manage** and clicking on **Add Owners**



STEP 7(Optional): Clicking on **Add Owners** will redirect the user to a page where the name and other information will be displayed. Check the box and click on the **Select** button down below

STEP 8: Click on **Enterprise applications** on the left pane under **Applications** to view the application created

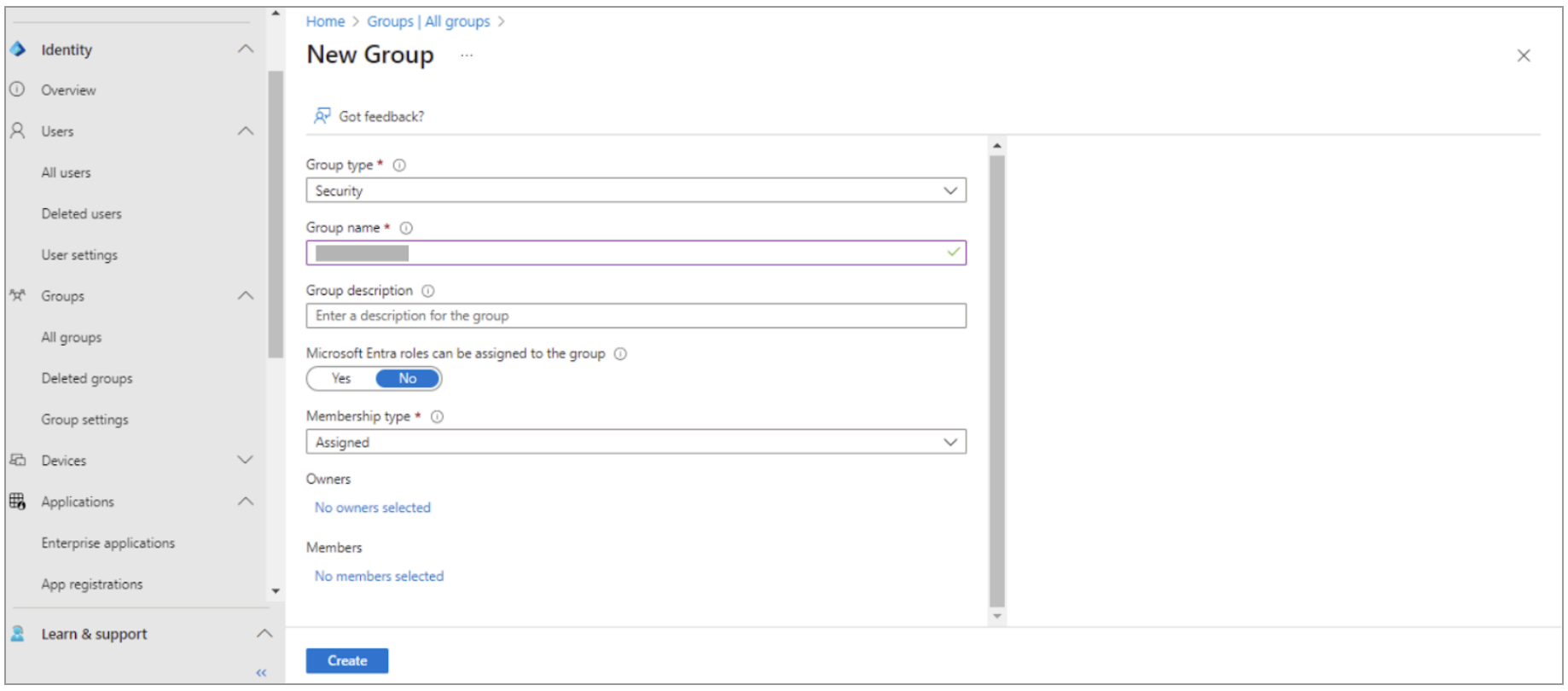
STEP 9: Now click on your application which is **Quest DQ**, that will take you to the overview page of the application



STEP 10: Click on the **Users** dropdown to add users, under the Users dropdown

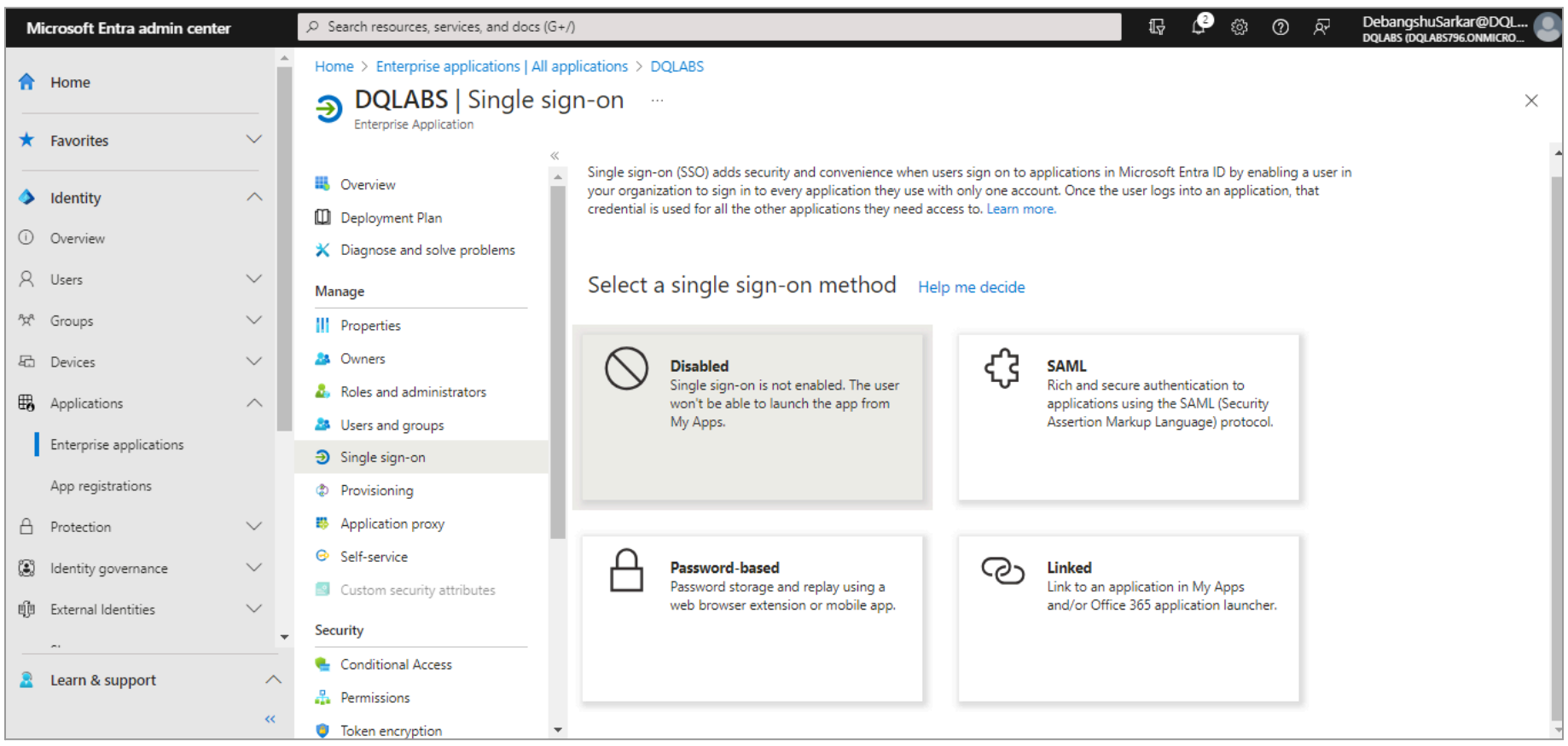
STEP 11: Select the required users or group and click on Assign. Now click on the **Single sign-on** option just below **Users and Groups** to configure the ENTRA ID single sign-on with Quest DQ and to get the Federation file. This below page will pop up once you click on the Single sign-on option and to configure the SSO click on **SAML**.

STEP 12: Click on the 3 horizontal dots and click on Edit to configure the SSO and a page will pop up where we need to enter the Entity ID, and ACS URLs from the Quest DQ platform into the ENTRA ID portal.



STEP 13: Copy the Entity ID and ACS URLs from the Quest DQ platform and paste them accordingly in the Correct places by clicking on **Add identifier** and **Add reply to URL** links respectively below, click on the Save.

STEP 14: Cross-verify our configuration we can click on the **Test sign-in** button to log in to the Quest DQ platform.



STEP 15: After testing scroll down on the same page, download the **Federation Metadata XML** file, and add it to the Quest DQ platform.

Login into Quest DQ using SSO

- Go to the Quest DQ login page and click on SSO.
- Now the user will get navigated to the corresponding SSO login page.
- Provide the valid credentials and the user will be logged into the Quest DQ portal.

User Provisioning

- Authorized users provisioned in Azure Active Directory, can log in into the Quest DQ Portal using the SSO button on the login screen.

    - Quest DQ will automatically provision the user in the Portal with the “USER” role.
- Users who have Admin access in the Quest DQ Portal can modify the role that is assigned to the user based on their persona.
- Once the role attribute is added, the roles in the Azure AD will be mapped to the roles in Quest DQ for the users.
- When a new user logs in to Quest DQ via SSO, if the role is already present in the Quest DQ then that role will be mapped based on the role name.
- If the role is not present, then a new role will be created based on default settings and mapped to the user.
- If the user has already logged in and has a role, then the new roles will be mapped when logged in using SSO.
- When the existing user in Quest DQ who has been assigned a role in Quest DQ, logins with SSO, the role that is already there in the Quest DQ platform should be maintained when there are no roles in Azure AD.
- If the users log in for the first time the role in the AD should be mapped or created newly in Quest DQ based on the roles in the Quest DQ platform.
- The role-based functionality should work only if the role attribute is mapped while generating the Federation file.