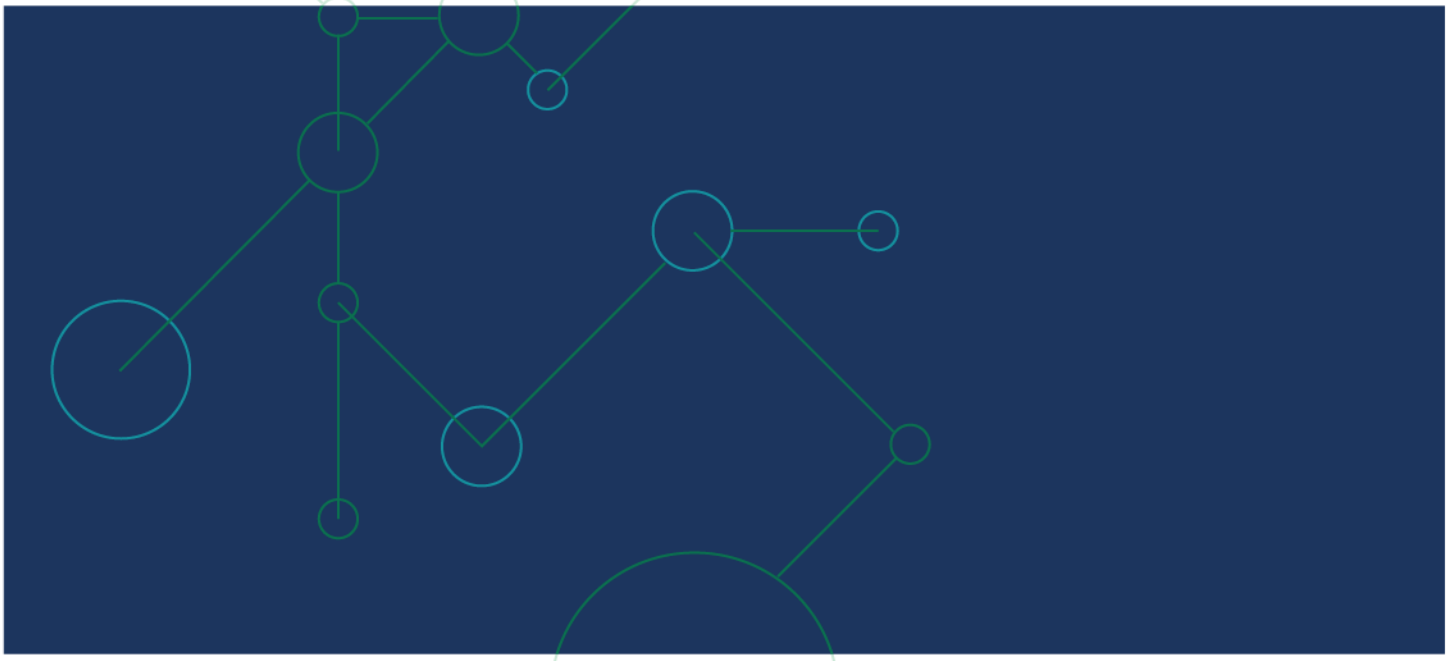
TECHNICAL BRIEF

# Configuring Qlik Replicate with Azure Databricks



## TABLE OF CONTENTS

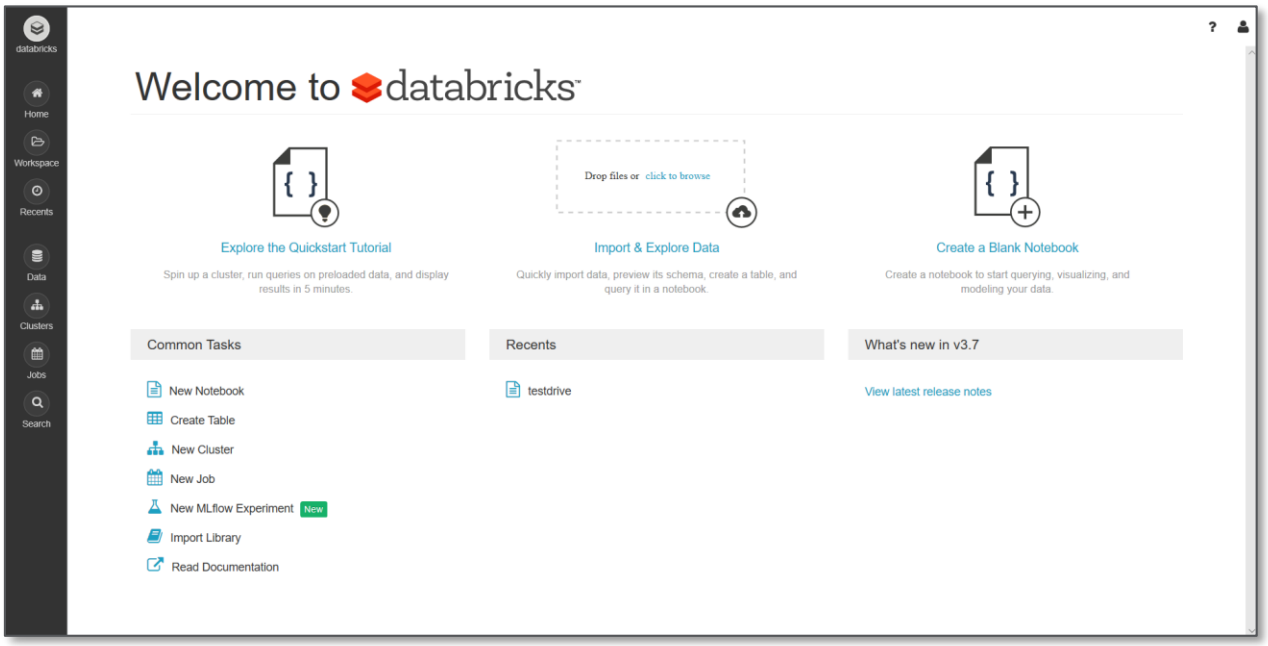
## SUMMARY

- This document was created to supplement Qlik Replicate Documentation for customers intending to Qlik Replicate and Azure Databricks. The official documentation can be found at https://help.qlik.com/en-US/replicate/Content/Replicate/Home.htm.

# A. Configure Azure Databricks Components

At this point Azure Data Lake Storage account and Active Directory settings we need should be configured. We now need to configure Azure Databricks so that it can make use of that storage. We also need to do configure few things to prepare Azure Databricks to accept data loaded by Qlik Replicate. Please refer to "Configuring Azure ADLS Gen2 for Qlik Data Integration" guide for ADLS Gen2 Instructions.

Everything we do in this section of the setup will be done from your Databricks workspace, so go ahead and log in to Databricks from Azure Portal.

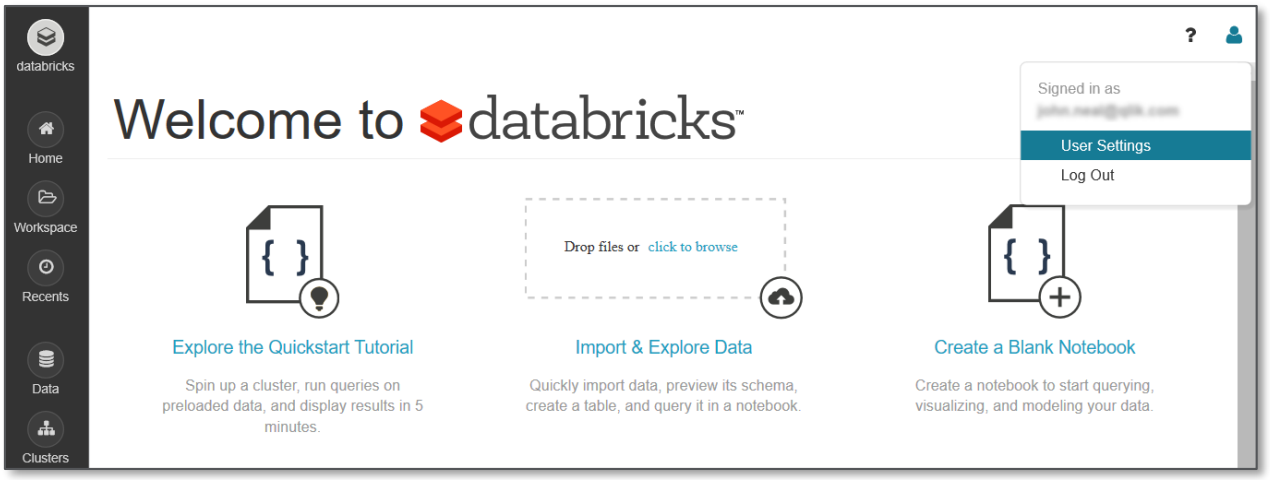


## 1. Download and Install Databricks ODBC Driver

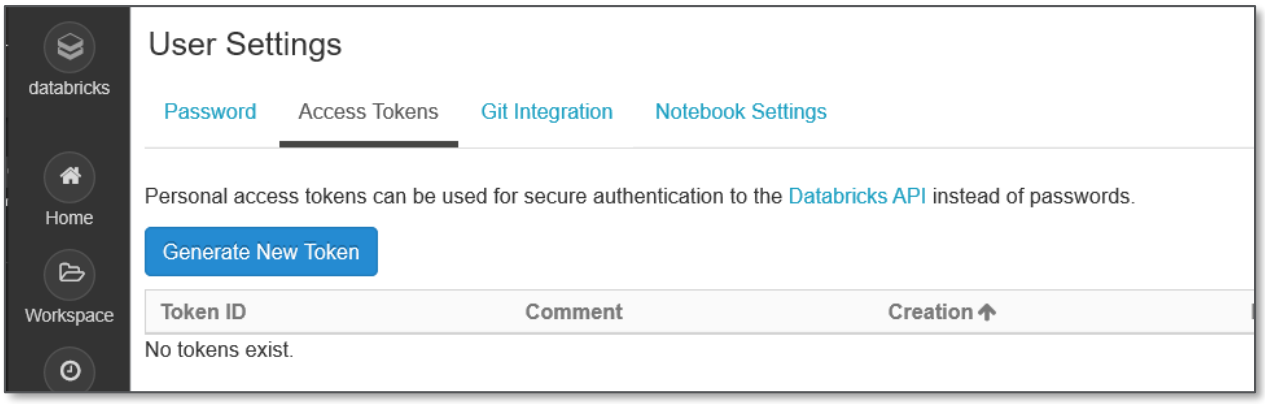
Please refer to Databricks dokocumentation and setup ODBC Driver for Windows/Linux Server Qlik Replicate is running on - https://docs.databricks.com/integrations/bi/jdbc-odbc-bi.html#connect-bi-tools

## 2. Create Databricks Token and Edit Spark Configuration

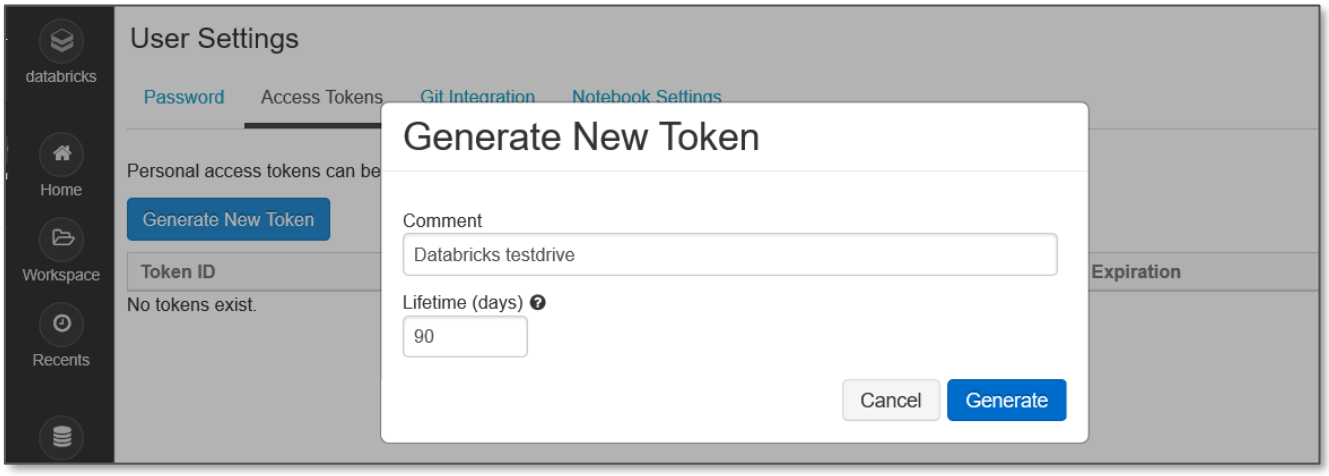
Replicate uses a Databricks “access token” to access Databricks. The first thing we will do is create one. This is done from the “User Settings” drop down of the workspace.



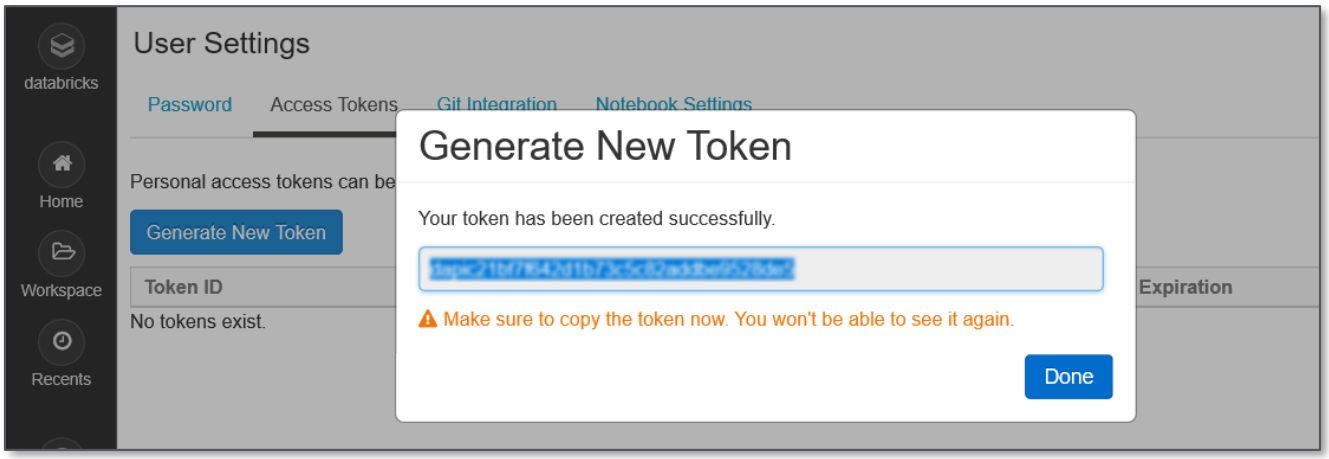

From “User Settings” select the “Access Tokens” tab and then Press “Generate New Token”.



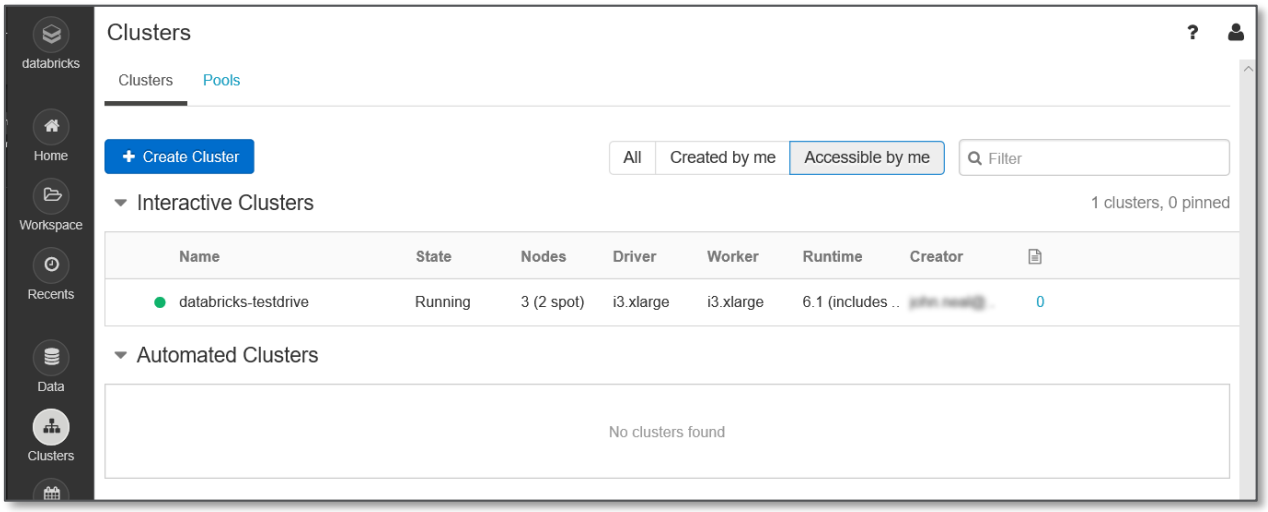
Enter a comment and select “Generate”.



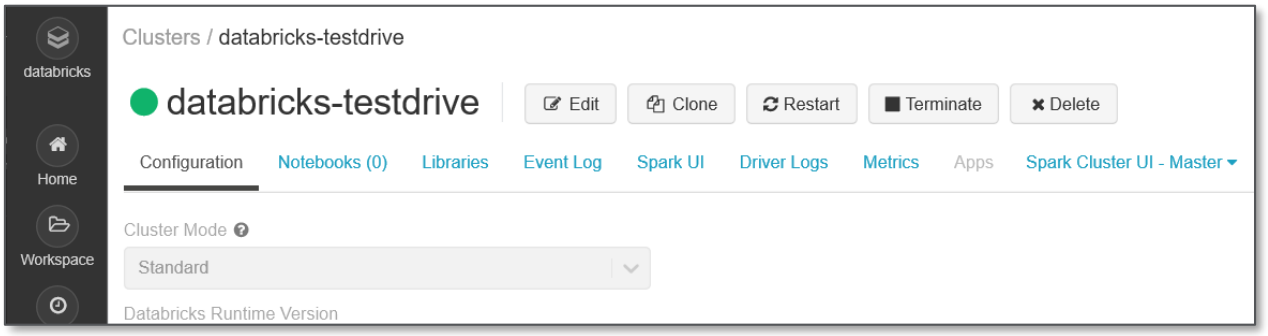

Please make sure to copy the Token value and save the generated token off now. You will not be able to retrieve it later.



The Spark Config section of your Databricks cluster configuration must contain the line "spark.hadoop.hive.server2.enable.doAs false" when using ADLS Gen2 storage. First select the cluster we will be using for this test drive by clicking on "Cluster" Icon on left side of screen.

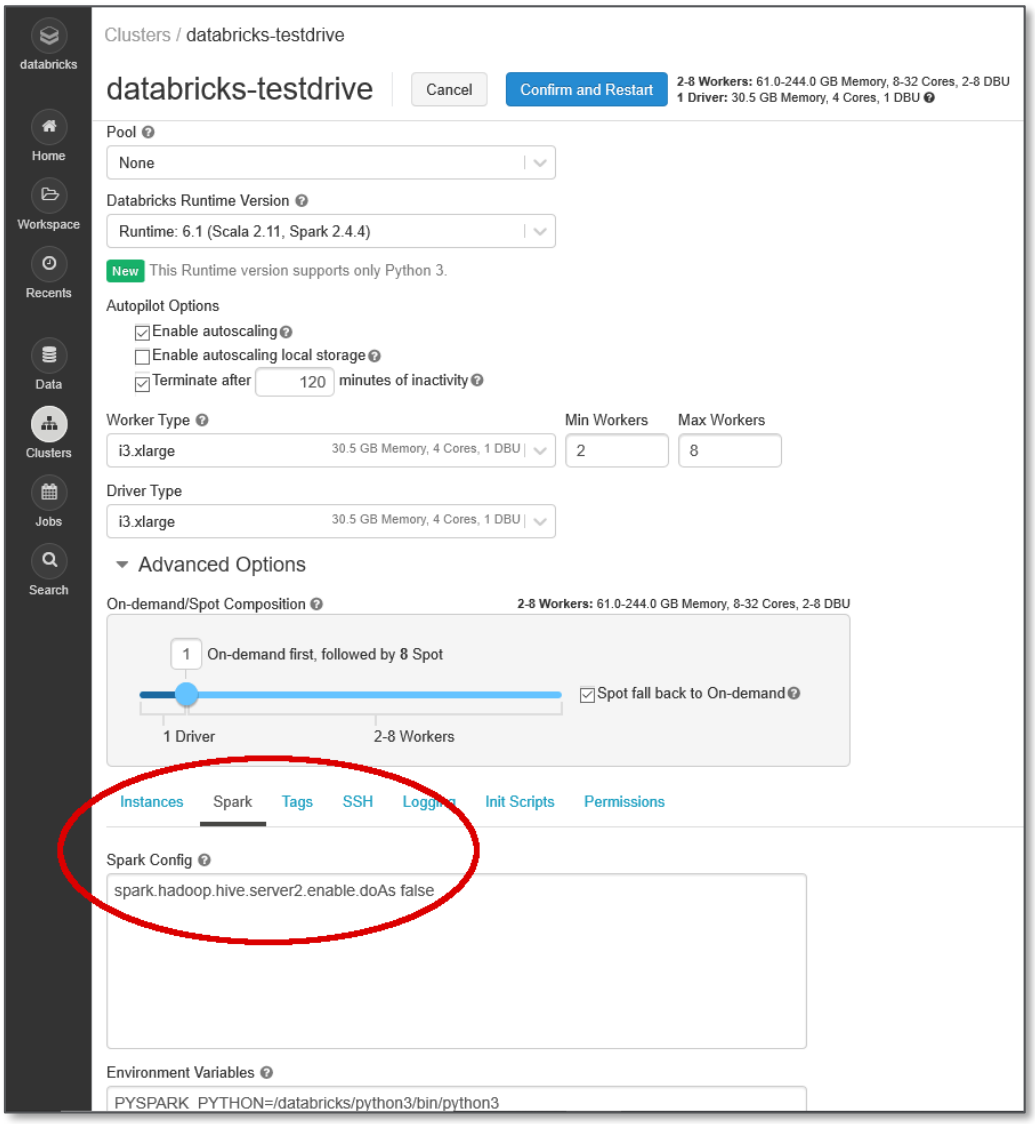



We need to make a change, so select "Edit" button.



Now scroll down,

- select > “Advanced Options”
- select the “Spark” tab
- enter the string “spark.hadoop.hive.server2.enable.doAs false” in the Spark Config section.




and then click “Confirm and Restart” button at the top of the page.

### 3. Execute Code to Mount Data Drive

The next step is to mount the Azure Data Lake Gen-2 Storage we created previously in Databricks so it can be accessed.

Open a Notebook and execute the following python command:

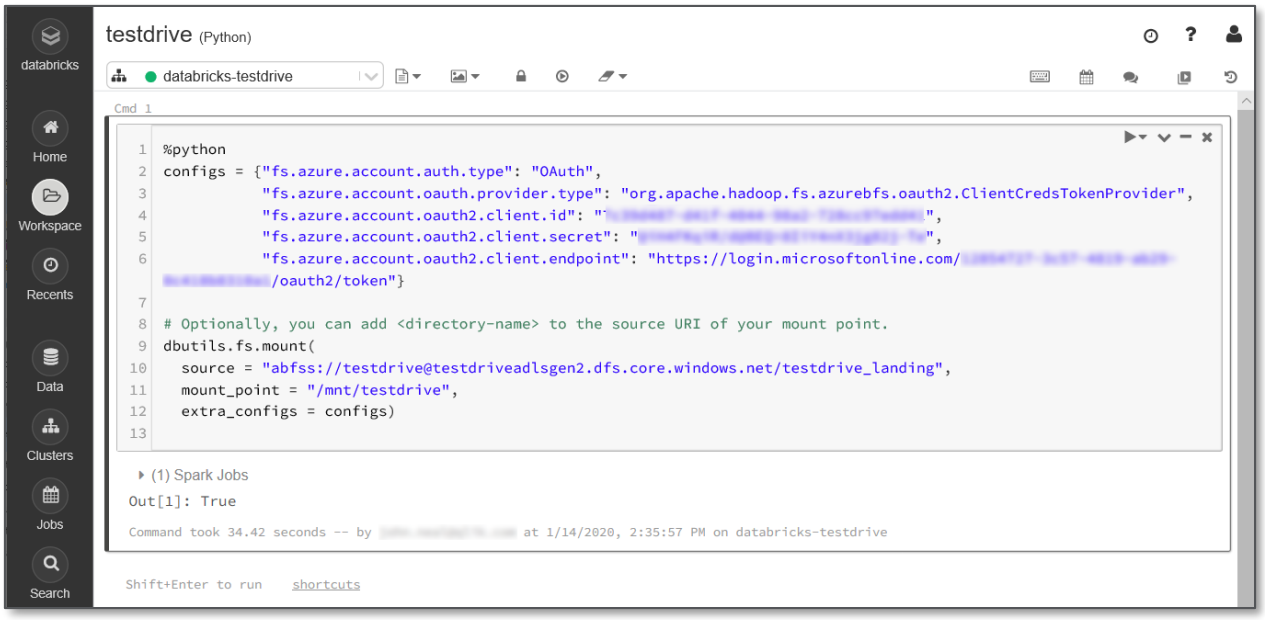
```
%python
configs = {"fs.azure.account.auth.type": "OAuth",
          "fs.azure.account.oauth.provider.type":
"org.apache.hadoop.fs.azurebfs.oauth2.ClientCredsTokenProvider",
          "fs.azure.account.oauth2.client.id": "<application-id>",
          "fs.azure.account.oauth2.client.secret": "<client-secret>"),
          "fs.azure.account.oauth2.client.endpoint": "https://login.microsoftonline.com/<directory-
id>/oauth2/token"}

# Optionally, you can add <directory-name> to the source URI of your mount point.
dbutils.fs.mount(
  source = "abfss://<file-system-name>@<storage-account-
name>.dfs.core.windows.net/<directory-name>/",
  mount_point = "/mnt/<mount-name>",
  extra_configs = configs)
```

where:

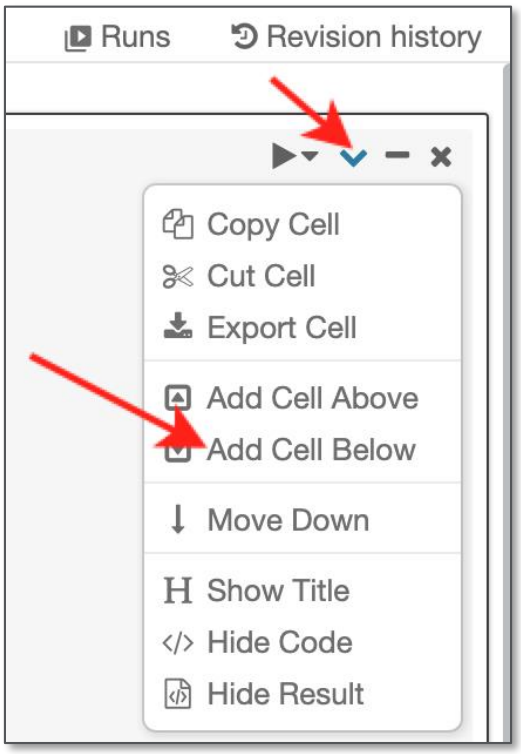
- <application-id> is the Azure Active Directory Application (client) ID we made note of earlier.
- <client-secret> is the Azure Active Directory Application (client) Key we created.
- <directory-id> is the Azure Active Directory ID (tenant ID).
- Source is the file system at target folder that will contain the data delivered by replicate.
    - <file-system-name> is the ADLS Gen2 file system we are using.
    - <storage-account-name> is the ADLS Gen2 storage account we are using.
    - <directory-name> is the name of the directory in the file system we will be writing to
- <mount-name> is where the source is mounted in Databricks. This will be used later by Replicate.

If this fails, please check your Azure storage account network settings.




## 4. Create Databricks DB and Collect ODBC Settings

Add Additional Cell in the Databricks Notebook by clicking on Down Arrow and selecting “Add Cell Below”



Execute the following Code:

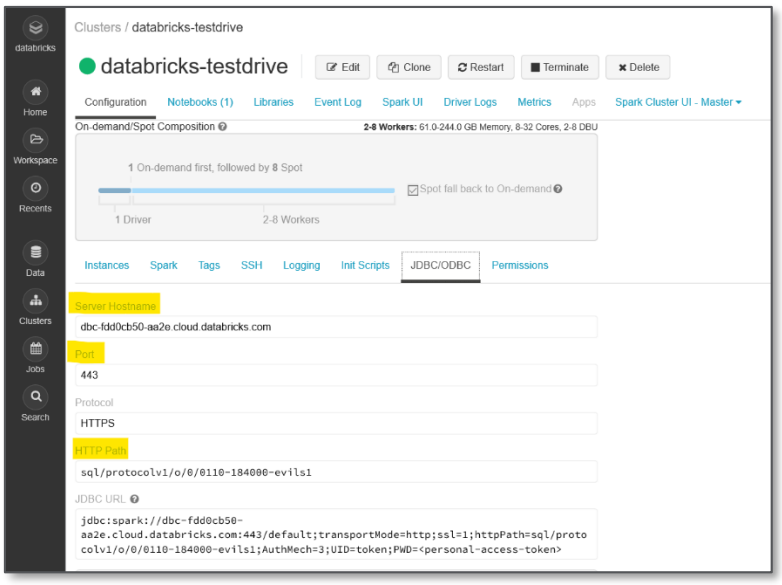
```
%sql
drop database if exists <database-name>;
create database <database-name> location '<mount-point>';
```

where:

- <database-name> is the name of the database you want to create; and
- <mount-point> is the mount point you created above.

The results should look something like this:

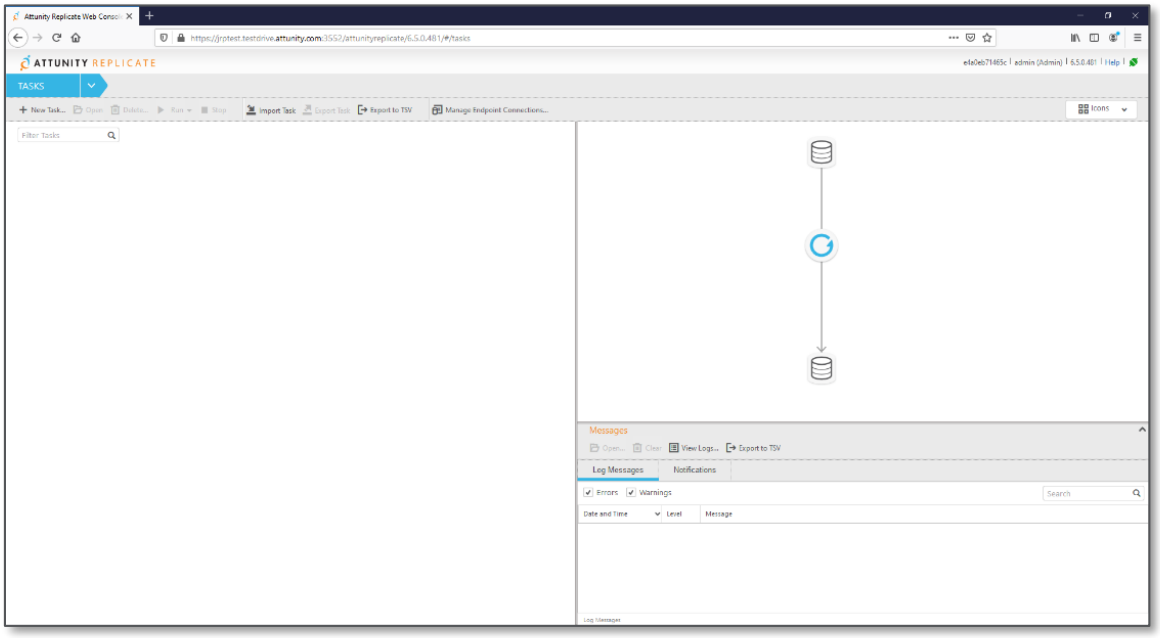

Qlik Replicate uses ODBC to write table metadata to Databricks. We need to collect some information from Databricks that we will need in the next section of the guide. Once again, select the cluster you are using and go to "> Advanced Options". From there, select the "JDBC/ODBC" tab.



You should make note of the “Server Hostname”, “Port” (normally 443), and “HTTP Path”. Now you should have all information for configuring the Qlik Replicate Azure Databricks target endpoint!

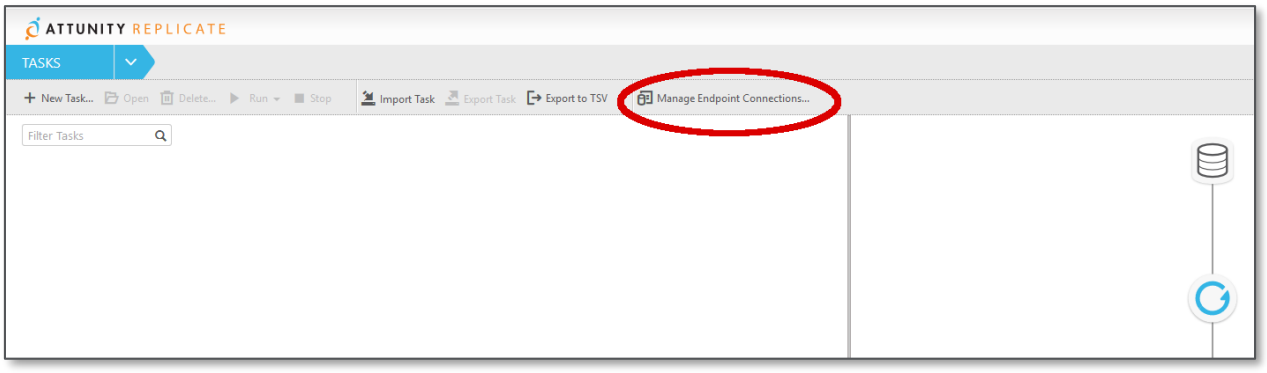
# B. Configure Azure Databricks connection on Qlik Replicate

First things first, we need to do is open Qlik Replicate. Click the Qlik Replicate icon to open Replicate in a new tab in your browser. Once you are logged in you will see the main screen for Qlik Replicate.

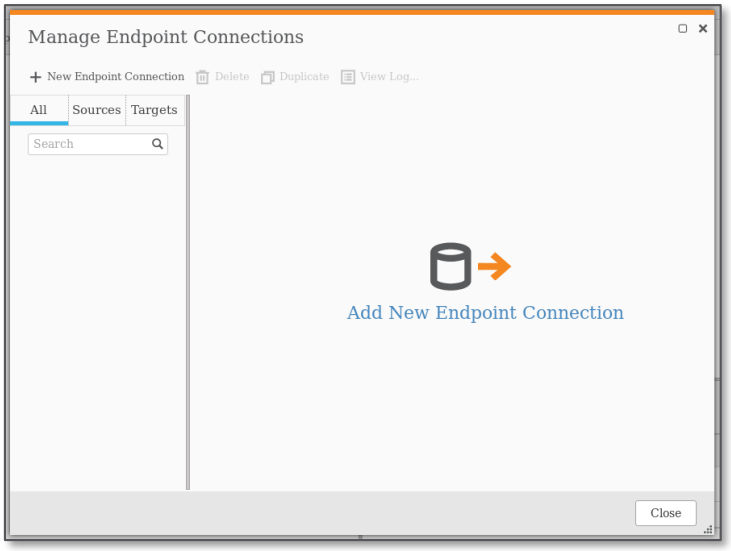



## 1. Create Microsoft Azure Databricks Endpoint Connection

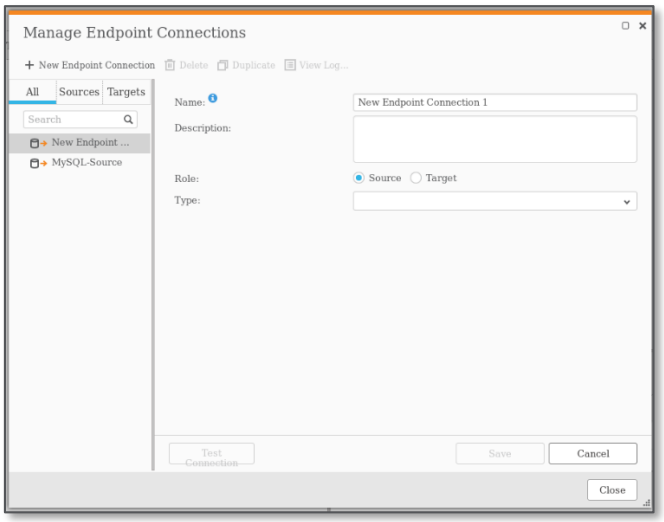
The first thing we need to do is create a target endpoint. We do this by clicking the "Manage Endpoint Connections" button at the top of the screen.



From there, click on “Add New Endpoint Connection” link or the + “New Endpoint Connection” button at the top of the screen.

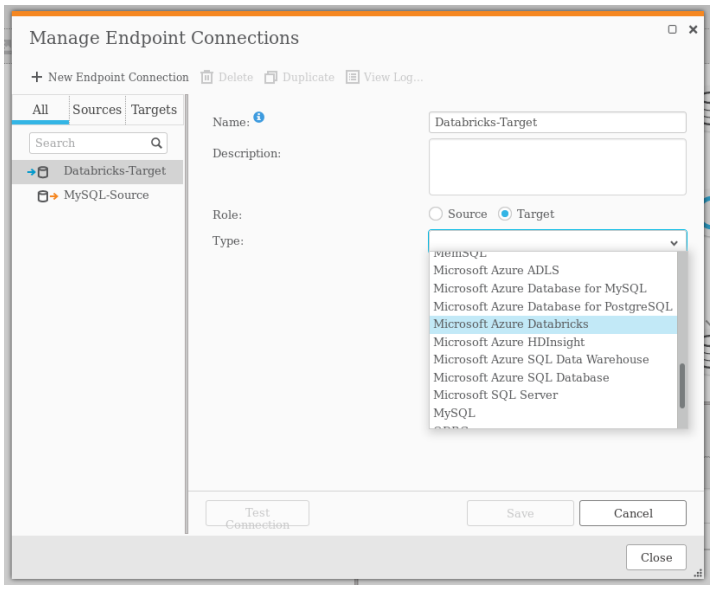



Once you do that you will see this window:



We will now create a Databricks Target endpoint:

- Replace the text “New Endpoint Connection 1” with something more descriptive like Databricks-Target, make sure the Target radio button is selected.
- Select “Microsoft Azure Databricks” from the dropdown selection box.

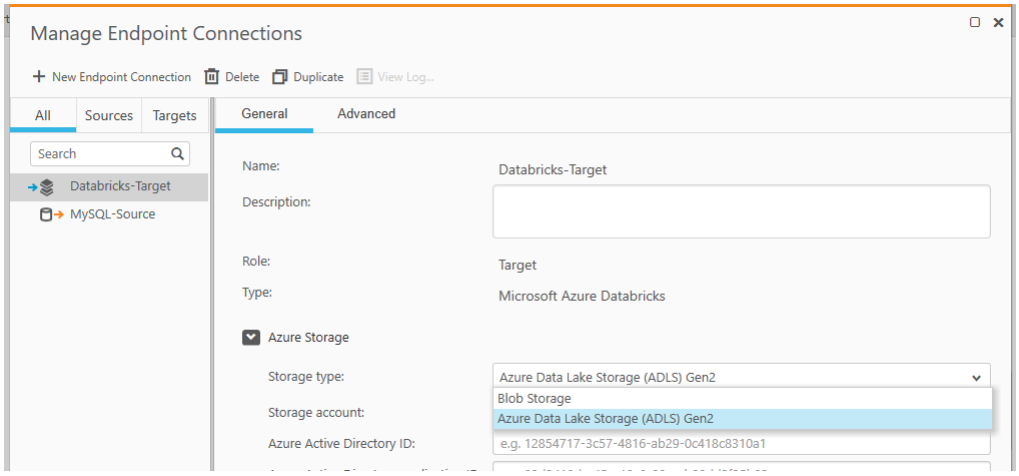



Replicate creates external tables in the Databricks metadata store using ODBC, and when running Full Load and Store Changes tasks, it writes the files to Azure storage. Like other endpoints, Replicate creates change data partitions in the Partition Control Table and in the metadata store.

## 2. Azure Storage Configuration

To optimize delivery into the Databricks environment, Qlik Replicate delivers change data in a continual series of micro batches that are staged for bulk ingest. You can configure the Databricks on Azure endpoint to stage the data files on Databricks (i.e. internally) or on Amazon S3.

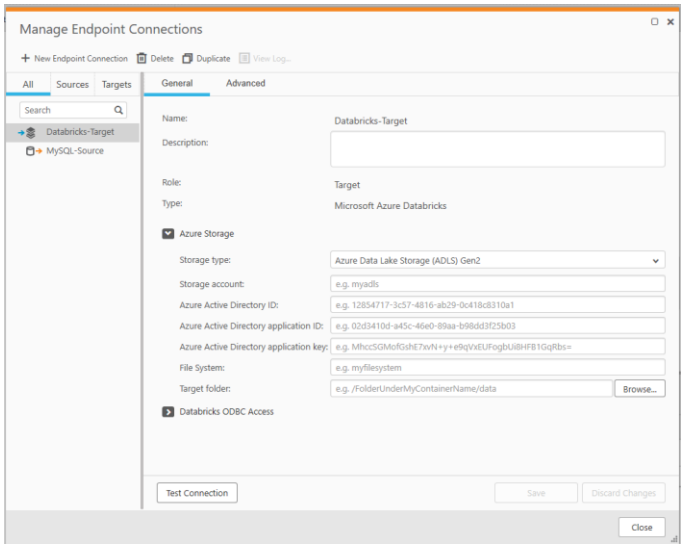
Replicate supports delivering the data for the external tables into Azure Blob Storage as well as into Azure Data Lake Storage (ADLS) Gen2. In either case, the storage location must be accessible from the Replicate server, and obviously must have write access as well. Further, for Databricks to be able to access the data, the storage that Replicate writes to needs to be mounted on the Databricks File System (DBFS).



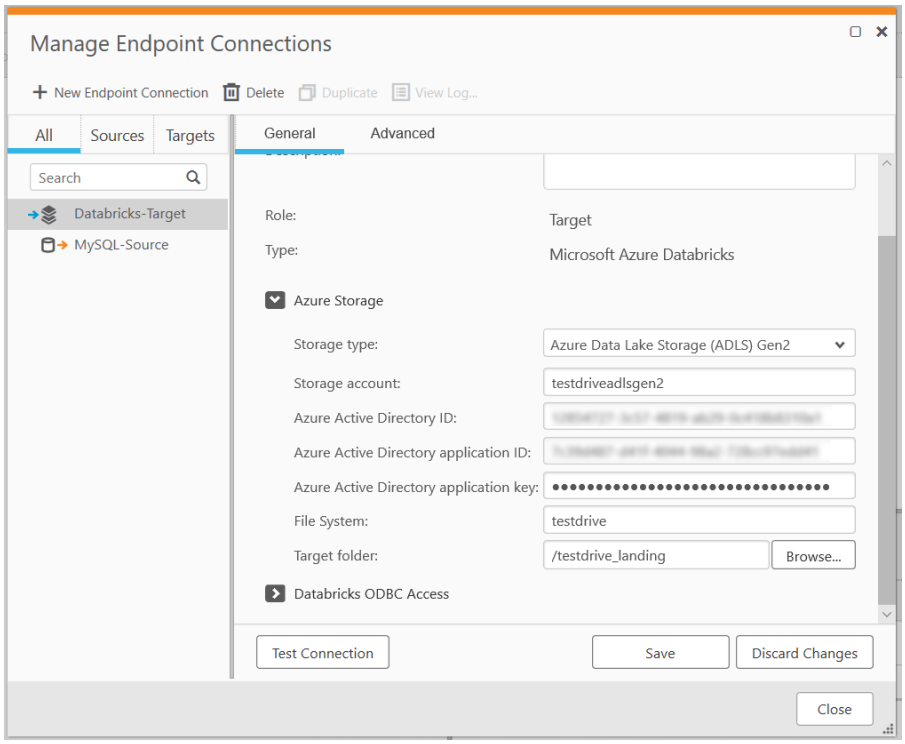

Please note Replicate only supports writing data to Blob Storage when Replicate is running on a Windows Server.

Fill in the blanks related to ADLS Gen2 storage with information specific to your Azure subscription. We worked through how to configure and obtain this information in the previous 'Configure Azure Data Lake Gen2 Storage' section.

- Storage account: specify the name of your ADLS Gen2 storage account
- Azure Active Directory ID: specify your Azure Active Directory (tenant) ID
- Azure Active Directory application ID: specify the Azure Active Directory application (client) ID
- Azure Active Directory application key: specify the Azure Active Directory application (client) key.
- File System: the ADLS Gen2 file system containing your folders and files
- Target folder: the folder where we want Replicate to create the data files on ADLS.

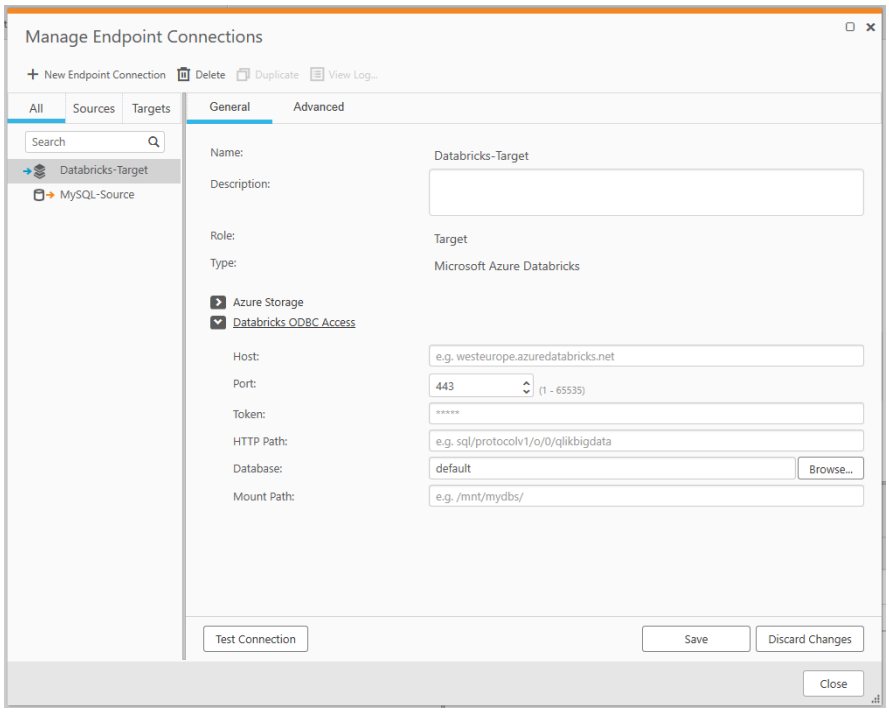


Note: if you specify a folder that does not exist, Replicate will create it for you. You may also press “Browse” to find directories in your file system that you may choose from. If you created a directory as the guide suggested in the ‘Configure Azure Data Lake Gen2 Storage’ section, you will see that directory listed when you press browse.



## 3. Databricks ODBC Access Configuration

Now click on the arrow (>) next to “Databricks ODBC Access”:

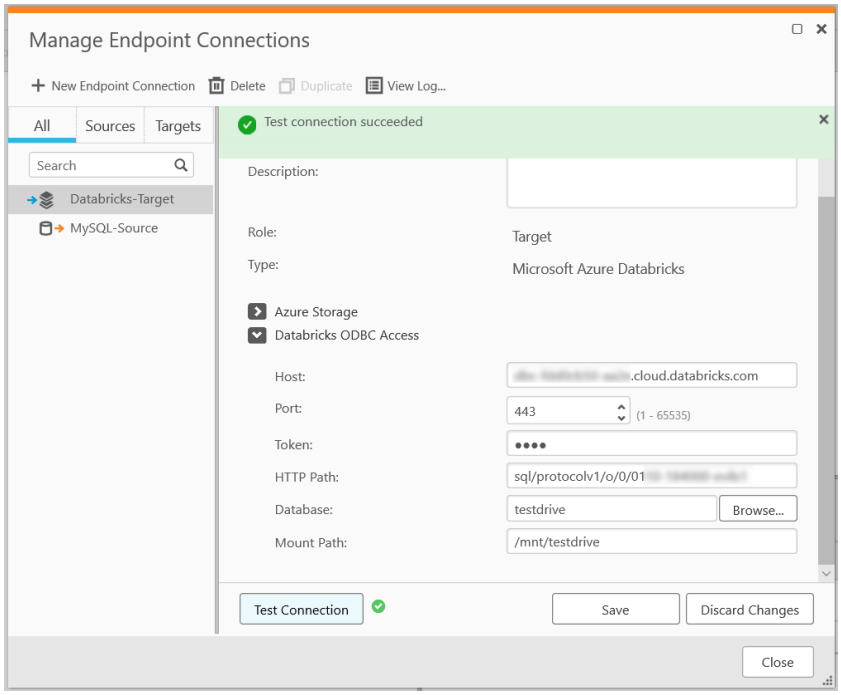



Fill in the blanks with information pertaining to your Databricks subscription. We worked through how to configure and obtain this information in the previous ‘Prepare Databricks for Data Delivery’ section.

- **Host**: specify the host name of the Databricks workspace where the ADLS storage containers are mounted
- **Port**: specify the port to use to access the workspace (443 by default)
- **Token**: specify the Databricks Token
- **HTTP Path**: specify the path to the cluster being used
- **Database**: specify the name of the Databricks target database
  *Note: you can Browse for existing databases if needed.*
- **Mount Path**: specify the mount path to the storage tables. This is the mount path we created previously
  *Note that the mount path cannot contain special characters or spaces.*

## 4. Test and Save

Once you have completed configuring the Databricks endpoint, click on “Test Connection”. Your screen should look like the following, indicating that your connection succeeded.




Assuming so, click “Save” and the configuration of your Databricks on Azure target endpoint is complete. Click “Close” to close the window.

For more details about using Databricks as a target, please review our Help Guide.


## About Qlik

Qlik's vision is a data-literate world, one where everyone can use data to improve decision-making and solve their most challenging problems. Only Qlik offers end-to-end, real-time data integration and analytics solutions that help organizations access and transform all their data into value. Qlik helps companies lead with data to see more deeply into customer behavior, reinvent business processes, discover new revenue streams, and balance risk and reward. Qlik does business in more than 100 countries and serves over 50,000 customers around the world.

**qlik.com**