Steps to install Starburst Enterprise Presto on Azure K8s

Setup a demo environment for Starburst (Presto) on Azure Kubernetes Service.

# Setup Starburst in Your Subscription

Login to the Azure Portal and open a bash CloudShell.

Run the following script, changing the variables as needed at the top:

# variables to change

SUBSCRIPTION="YOUR-SUBSCRIPTION-Demo" ← Change to your subscription that you want to use

AKS\_NAME="starburst" ← Change as you want

RES\_GROUP\_NAME="starburstdemo" ← Change as you want

LOCATION="eastus" ← please point to a location/zone you want to

VNET\_NAME="aksvnet" ← please point to your vnet\_name

SUBNET\_NAME="starburst" ← Please point to your subnet\_name

#Run the below as 1 command, it refers to the above set variables

SUBNET\_RES\_ID=$(az network vnet subnet list \

--vnet-name $VNET\_NAME \

--resource-group $RES\_GROUP\_NAME \

--query "([? name == '$SUBNET\_NAME' ])[0].id" | cut -d '"' -f 2)

#Run this command now

az aks create --resource-group $RES\_GROUP\_NAME \

--name $AKS\_NAME \

--enable-cluster-autoscaler \

--node-count 5 \

--min-count 5 \

--max-count 8 \

--node-vm-size Standard\_DS3\_v2 \

--generate-ssh-keys \

--vnet-subnet-id $SUBNET\_RES\_ID \

--network-plugin azure \

--enable-addons monitoring

#The above command will take a few minutes to complete

#Most errors happen when the VNET, Subnet are not configured properly, they need to be in the same Zone/Region

**Important**: To destroy the demo environment, just delete the resource group.

## KUBECTL Apply files:

Run the following kubectl apply commands on the Azure bash CloudShell.

* kubectl apply -f <https://starburstdata.s3.us-east-2.amazonaws.com/mission-control/0.18/k8s/postgres.yaml>
* kubectl apply -f <https://starburstdata.s3.us-east-2.amazonaws.com/mission-control/0.18/k8s/missioncontrol.yaml>
* kubectl get pods
* kubectl get svc

# Mission Control Setup

Now you can log into the Starburst Mission Control UI

Run kubectl get svc - to get the external-IP of the load balancer which is attached to the mission-control service.

Example :

|  |
| --- |
| $ kubectl get svc  NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE  mission-control-int-lb LoadBalancer 10.0.6.83 172.56.0.38 5042:31774/TCP 37m |

Open a browser window & point the the following URL

http://<load balancer ip>:5042 [username/password :: admin/admin]

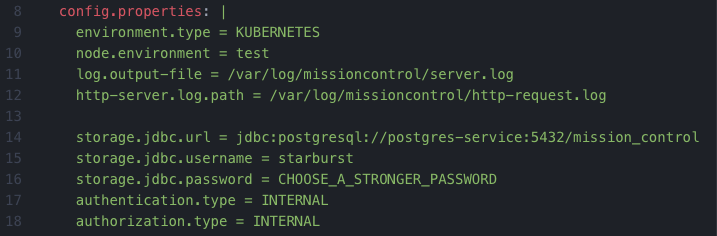
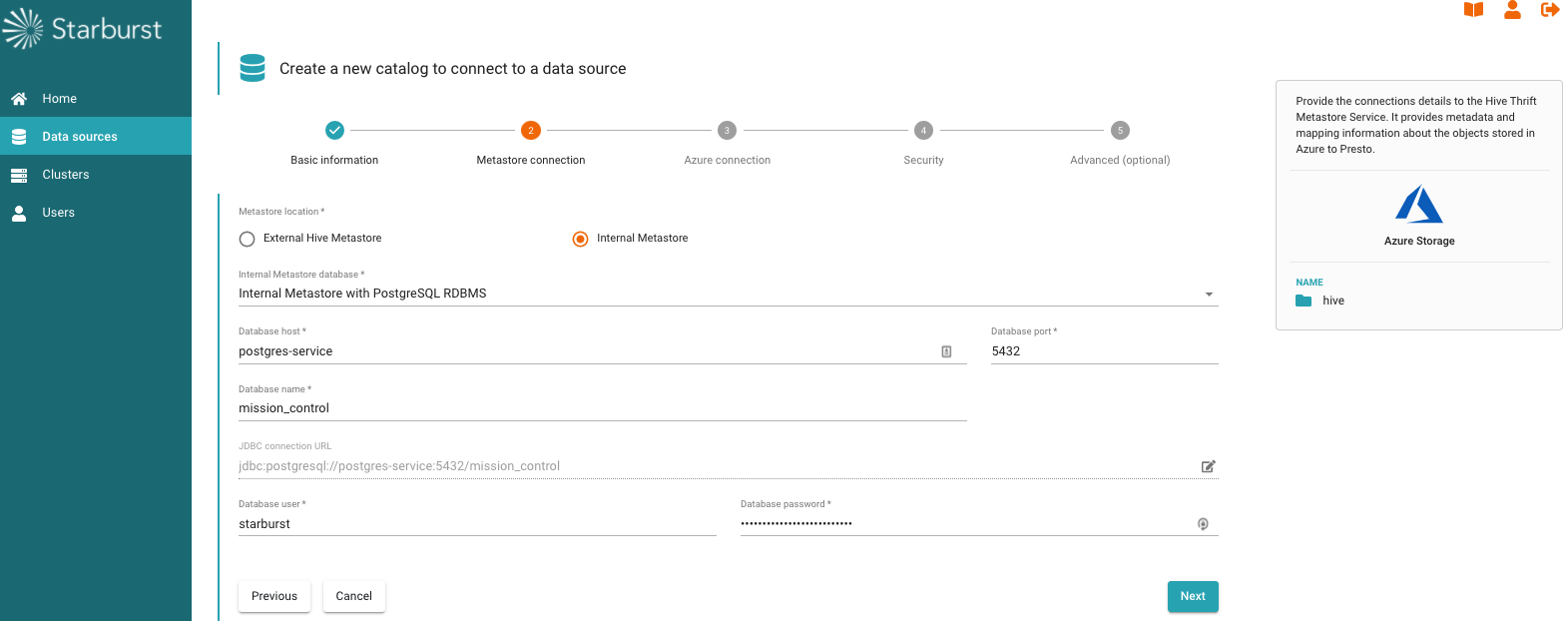
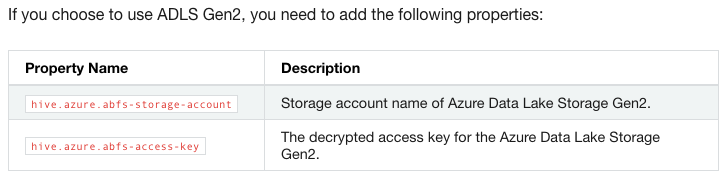
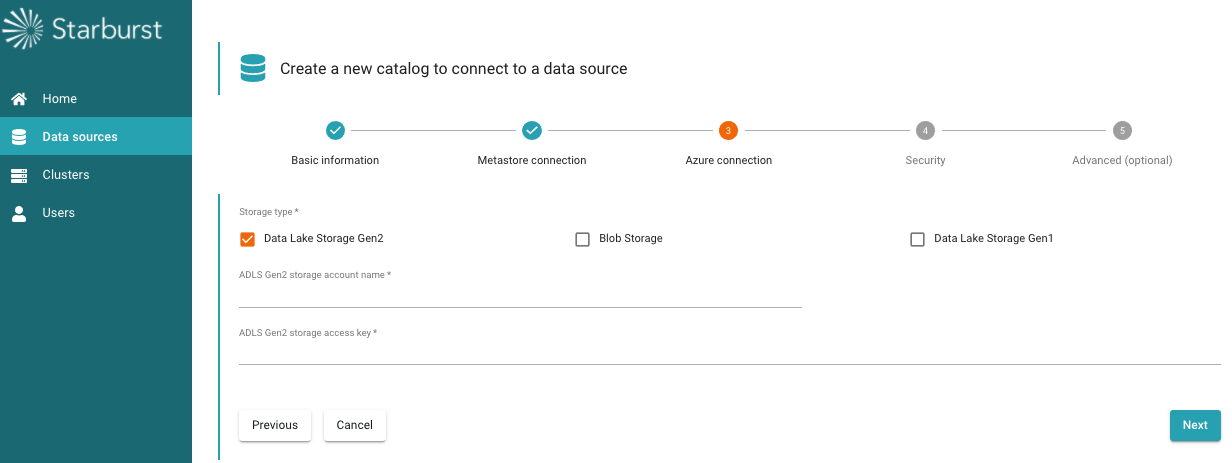
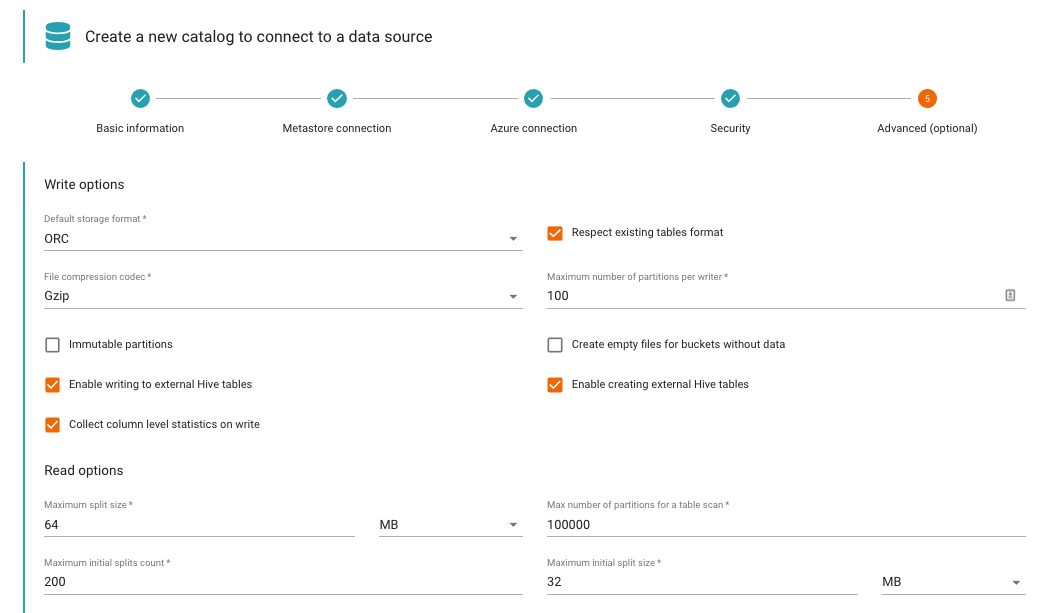
# Setup Data Sources

## Setup ADSL-Gen2 connection

In Mission Control, click on the Data Sources on the left pane.

Next click on the ‘+’ (bottom right)

Select ‘Azure Storage’ as the connection type

* Enter a catalog name as ‘hive’ & provide some description. Click next.
* Select ‘Internal Metastore’
* From the drop down menu, select ‘Internal Metastore on PostgreSQL’
* Refer to the missioncontrol.yaml file to fill in the connection details for the PostgreSQL (missioncontrol.yaml file relevant section shown below)
* 
* Use the information in the missioncontrol.yaml file to fill in the connection details (sample shown below)
* 
* Click next and enter your ADLS-Gen2 credentials (sample show below). You need to get these from your Azure console in the ADLS section.  
  
* 
* Click next, accept the default configuration for the security settings and click on Advanced Settings.
* Select ‘Enable writing to external hive tables’ as show below
* 
* Click Create.

## Setup SQL Server connection

Next you can create a SQL Server Data source connection.

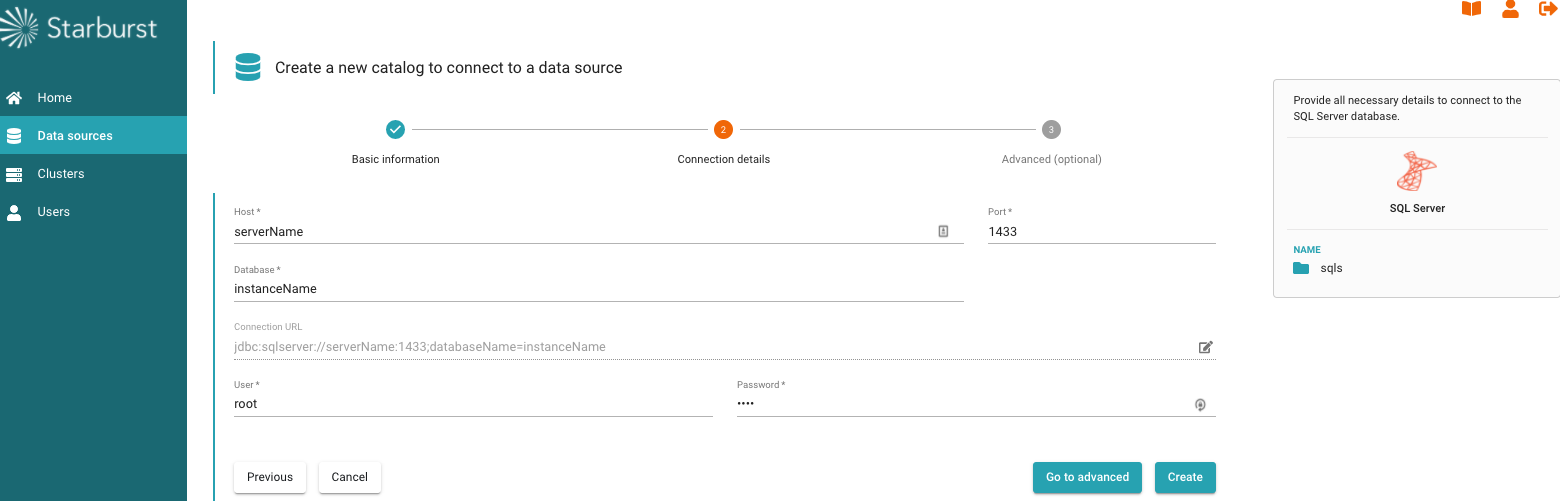
You can read more detailed information [here](https://docs.starburstdata.com/latest/connector/sqlserver.html) on the Starburst documentation site.

You will need the following information to create this connection:



**IMPORTANT**: Please make sure there is network connectivity between the K8s cluster and the SQL Server. If connectivity is not configured in Azure, the Presto Cluster will not start-up successfully (next section).

Sample below



Click on the create button after this.

At this point, you should have 2 Data Sources defined - ADLS & SQL Server.

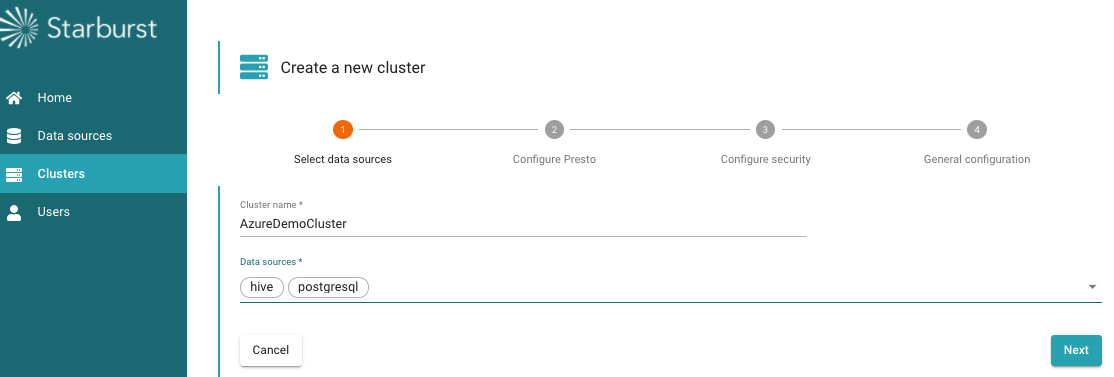
Next we will setup a Presto Cluster (Worker Compute node(s)) to read data from these 2 data sources and execute SQL against them.

# Setup Presto Cluster

In the mission control UI, click on ‘Clusters’ on the left pane.

Click on the ‘+’ sign to create a new cluster.

Give the cluster a name and select the 2 data sources from the drop down menu (as shown below)



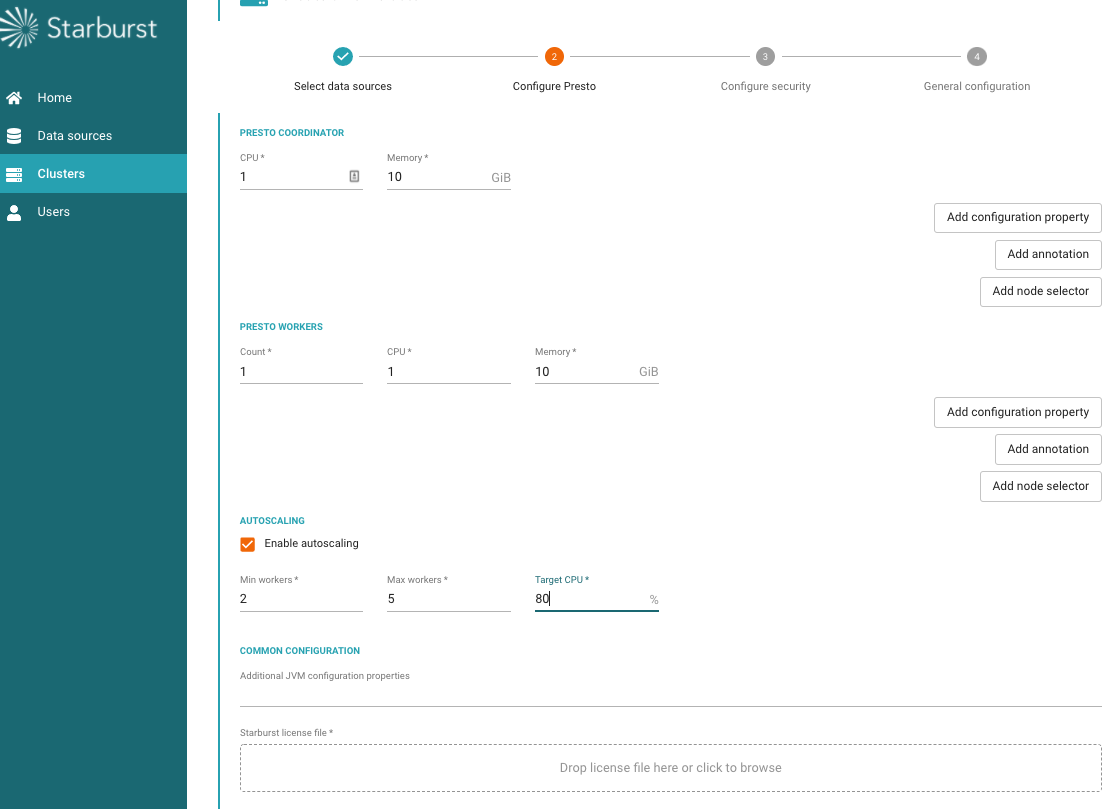
Click next and enter the information (sample show below).

You can give more CPU/Memory/Storage capacity based on your demo requirements.

Select “Enable Autoscaling”

Drop the starburst presto license file provided to you by Starburst.

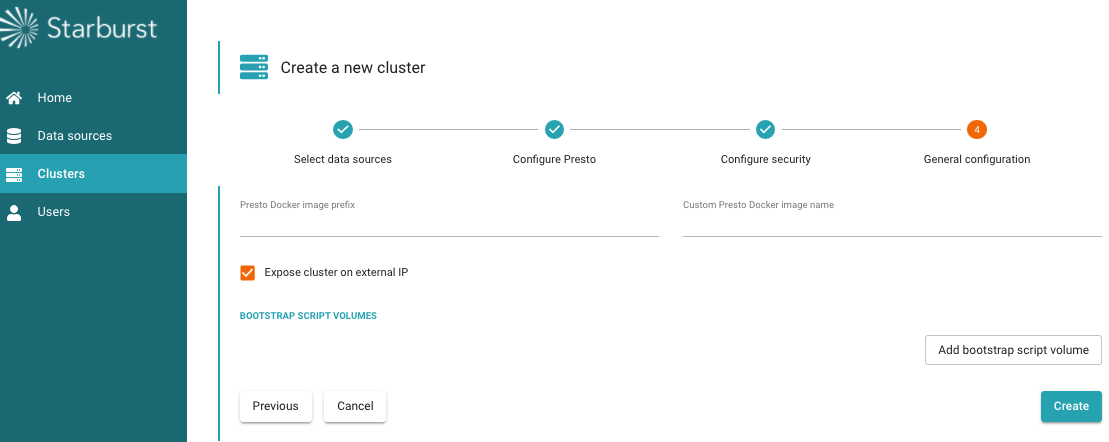
Click next.



On the security configuration tab, do not select anything. Leave the defaults.

Click next.

On the ‘General Configuration’ tab, select the ‘Expose cluster on external-IP’ option. (as shown below)



Next click on ‘create’ and run the cluster.

This will take a few minutes to start the cluster.

**IMPORTANT**: The cluster will not start (instead error out) if there are issues with any of the Data Source definitions/configurations or any network connectivity issues between the K8s cluster and the data sources. You have to resolve those before you can have a clean cluster up and running.

# Connecting to Starburst Enterprise Presto

## Connecting to Starburst Cluster Overview

Run the following command on the bash cloud shell to get the external-IP for the Presto Coordinator service running on port 8080.

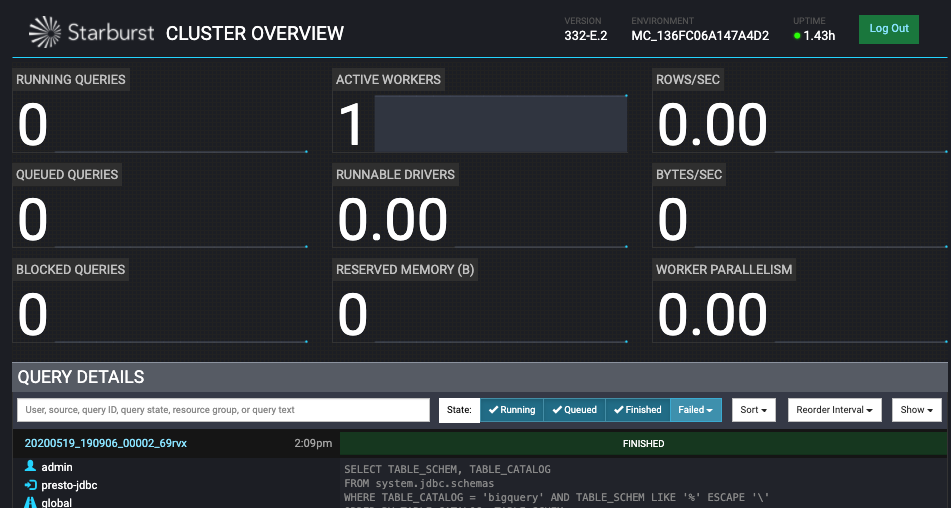
$ kubectl get svc

Copy the external-IP and paste it in a browser window along with the port 8080.

[http://[External-IP]:8080](about:blank)

Login with username : admin (no password is required for HTTP).

You should now have access to the Starburst Presto cluster overview (sample shown below).



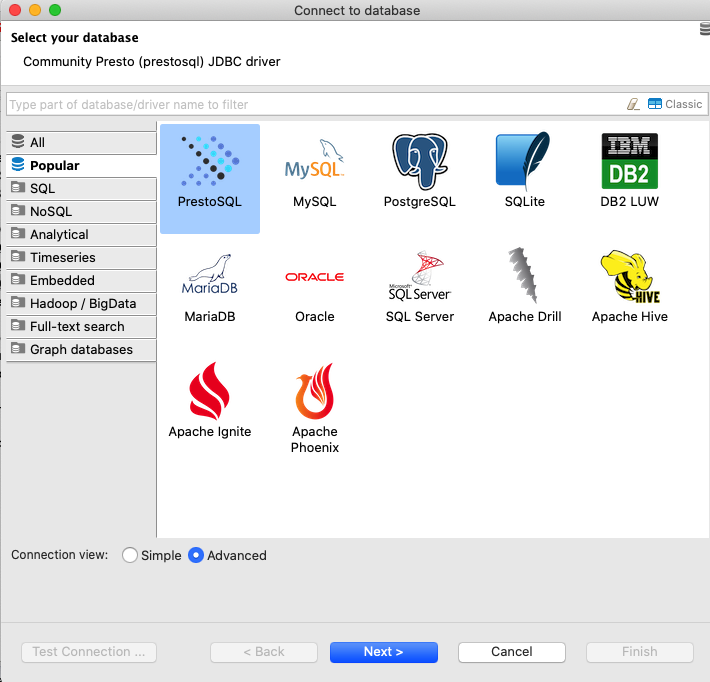
## 

## Connecting to Starburst over JDBC via DBeaver

Download DBeaver [here](https://dbeaver.io/download/). It's an open source SQL Client tool.

Once installed, open DBeaver and create a new connection.

Select type as ‘PrestoSQL’. Click next.

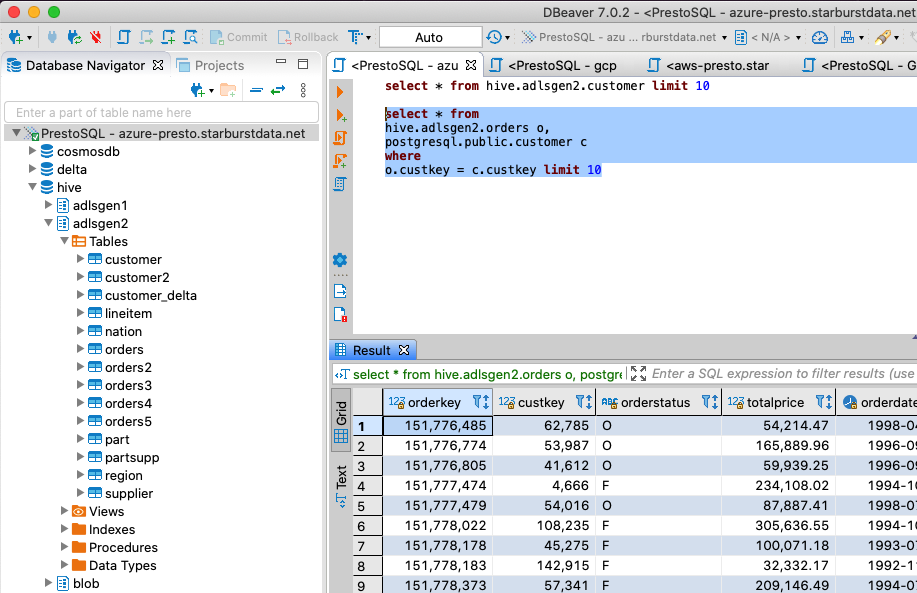


Enter the ‘External-IP’ for the Presto Coordinator (from previous step).

Enter username : ‘admin’

Click on ‘Test Connection’. This should succeed. At that point click ‘finish.

At this point you should be able to navigate the 2 data sources (as catalogs) and write SQL queries against them (as shown below)



### Sample SQL Queries you can run

select \* from tpch.sf1.customer limit 10

drop schema hive.adlsgen2

create schema hive.adlsgen2 WITH (location = 'abfs://tpch-10gb@davewdemodata.dfs.core.windows.net/presto');

create table hive.adlsgen2.customer as select \* from tpch.sf1.customer

sql server query federation example

create table hive.adlsgen2.customer as select \* from tpch.sf1.customer

select \* from airline\_oltp.dbo.airline limit 100

create table hive.adlsgen2.airline as select \* from airline\_oltp.dbo.airline limit 10