Before you start

Provision an Azure Databricks workspace

View and start a SQL Warehouse

Create a database

Create a table

Create a query

Create a dashboard

Delete Azure Databricks resources

Use a SQL Warehouse in Azure Databricks

SQL is an industry-standard language for querying and manipulating data. Many data analysts perform data analytics by using SQL to query tables in a relational database. Azure Databricks includes SQL functionality that builds on Spark and Delta Lake technologies to provide a relational database layer over files in a data lake.

This lab will take approximately **30** minutes to complete.

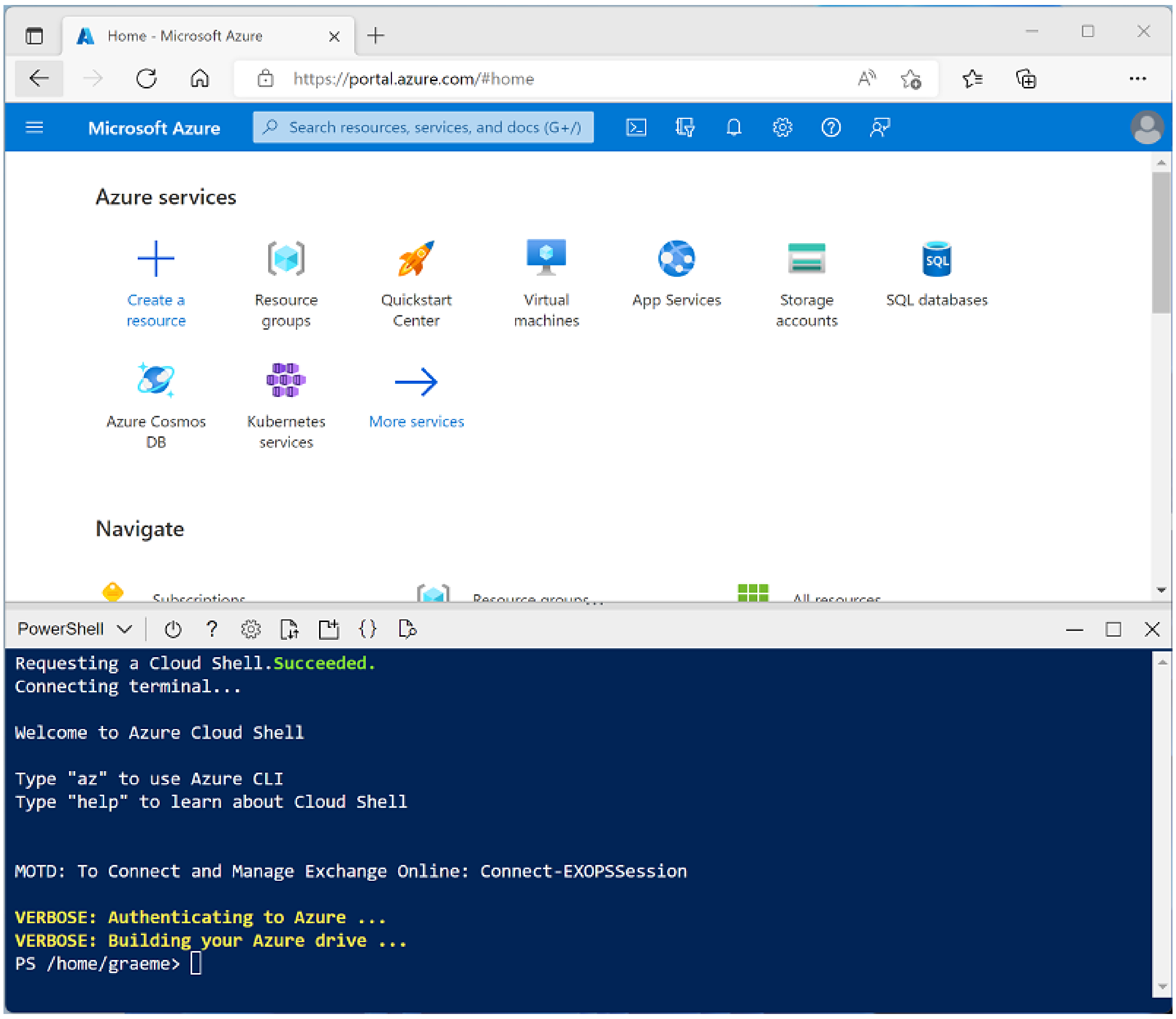
Before you start

You'll need an Azure subscription in which you have administrative-level access and sufficient quota in at least one region to provision an Azure Databricks SQL Warehouse.

Provision an Azure Databricks workspace

In this exercise, you'll need a premium-tier Azure Databricks workspace.

1. In a web browser, sign into the Azure portal at `https://portal.azure.com`.
2. Use the **[>_]** button to the right of the search bar at the top of the page to create a new Cloud Shell in the Azure portal, selecting a ***PowerShell*** environment and creating storage if prompted. The cloud shell provides a command line interface in a pane at the bottom of the Azure portal, as shown here:



```
Requesting a Cloud Shell.Succeeded.
Connecting terminal...

Welcome to Azure Cloud Shell

Type "az" to use Azure CLI
Type "help" to learn about Cloud Shell


MOTD: To Connect and Manage Exchange Online: Connect-EXOPSSession

VERBOSE: Authenticating to Azure ...
VERBOSE: Building your Azure drive ...
PS /home/graeme> 
```

> **Note**: If you have previously created a cloud shell that uses a *Bash* environment, use the the drop-down menu at the top left of the cloud shell pane to change it to ***PowerShell***.

3. Note that you can resize the cloud shell by dragging the separator bar at the top of the pane, or by using the **—**, **◻**, and **X** icons at the top right of the pane to minimize, maximize, and close the pane. For more information about using the Azure Cloud Shell, see the Azure Cloud Shell documentation.

4. In the PowerShell pane, enter the following commands to clone this repo:

   Code Copy

   ```
   rm -r dp-000 -f
   git clone https://github.com/MicrosoftLearning/mslearn-databricks dp-000
   ```

5. After the repo has been cloned, enter the following commands to change to the folder for this lab and run the **setup.ps1** script it contains:

   Code Copy

   ```
   cd dp-000/Allfiles/Labs/04
   ./setup.ps1
   ```

6. If prompted, choose which subscription you want to use (this will only happen if you have access to multiple Azure subscriptions).
7. Wait for the script to complete - this typically takes around 5 minutes, but in some cases may take longer. While you are waiting, review the What is Databricks SQL? article in the Azure Databricks documentation.

View and start a SQL Warehouse

1. When the Azure Databricks workspace resource has been deployed, go to it in the Azure portal.
2. In the **Overview** page for your Azure Databricks workspace, use the **Launch Workspace** button to open your Azure Databricks workspace in a new browser tab; signing in if prompted.
3. If a **What's your current data project?** message is displayed, select **Finish** to close it. Then view the Azure Databricks workspace portal and note that the sidebar on the left side contains icons for the various tasks you can perform. The sidebar expands to show the names of the task categories.
4. In the sidebar, expand **[D] Data Science & Engineering** and select **[S] SQL**. to change the portal interface to reflect the SQL persona (this persona is only available in *premium-tier* workspaces).
5. In the **Get Started** pane, select **Review SQL Warehouses** (or alternatively, in the sidebar, select **SQL Warehouses**).
6. Observe that the workspace already includes a SQL Warehouse named **Starter Warehouse**.
7. In the **Actions** (**⁝**) menu for the SQL Warehouse, select **Edit**. Then set the **Cluster size** property to **2X-Small** and save your changes.
8. Use the **Start** button to start the SQL Warehouse (which may take a minute or two).

> ! **Note**: If your SQL Warehouse fails to start, your subscription may have insufficient quota in the region where your Azure Databricks workspace is provisioned. See Required Azure vCPU quota for details. If this happens, you can try deleting your workspace and creating a new one in a different region. You can specify a region as a parameter for the setup script like this: `./setup.ps1 eastus`

Create a database

1. When your SQL Warehouse is *running*, close the **Review SQL warehouses** pane on the left. Then in the sidebar, select **SQL Editor**.
2. In the **Schema browser** pane, observe that the hive metastore contains a database named **default**.
3. In the **New query** pane, enter the following SQL code:

   Sql Copy

   ```
   CREATE SCHEMA adventureworks;
   ```

4. Use the **Run All** button to run the SQL code.

5. When the code has been successfully executed, in the **Schema browser** pane, use the ↻ button to refresh the list. Then select the **default** database to reveal the list of databases and select **adventureworks**. The database has been created, but contains no tables.

You can use the **default** database for your tables, but when building an analytical data store its best to create custom databases for specific data.

Create a table

1. In the sidebar, select **(+) Create** and then select **Table**.

2. In the **Upload file** area, select **browse**. Then in the **Open** dialog box, enter
   ```
   https://raw.githubusercontent.com/MicrosoftLearning/mslearn-databricks/main/Allfiles/Labs/04/data/products.csv
   ```
   and select **Open**.

   > **Tip**: If your browser or operating system doesn't support entering a URL in the **File** box, download the CSV file to your computer and then upload it from the local folder where you saved it.

3. In the **Create table in Databricks SQL** page, select the **adventureworks** database and set the table name to **products**. Then select **Create**.
4. When the table has been created, review its details.

The ability to create a table by importing data from a file makes it easy to populate a database. You can also use Spark SQL to create tables using code. The tables themselves are metadata definitions in the hive metastore, and the data they contain is stored in Delta format in Databricks File System (DBFS) storage.

Create a query

1. In the sidebar, select **(+) Create** and then select **Query**.
2. In the **Schema browser** pane, ensure the **adventureworks** database is selected and the **products** table is listed.

3. In the **New query** pane, enter the following SQL code:

   Sql

   ```sql
   SELECT ProductID, ProductName, Category
   FROM adventureworks.products;
   ```
4. Use the **Run All** button to run the SQL code.
5. When the query has completed, review the table of results.
6. Use the **Save** button at the top right of the query editor to save the query as **Products and Categories**.

Saving a query makes it easy to retrieve the same data again at a later time.

Create a dashboard

1. In the sidebar, select **(+) Create** and then select **Dashboard**.
2. In the **New dashboard** dialog box, enter the name **Adventure Works Products** and select **Save**.
3. In the **Adventure Works Products** dashboard, in the **Add** drop-down list, select **Visualization**.
4. In the **Add visualization widget** dialog box, select the **Products and Categories** query. Then select **Create new visualization**, set the title to **Products Per Category**. and select **Create visualization**.
5. In the visualization editor, set the following properties:

   - **Visualization type**: bar
   - **Horizontal chart**: selected
   - **Y column**: Category

    - **X columns**: Product ID : Count
    - **Group by**: Category
    - **Legend placement**: Automatic (Flexible)
    - **Legend items order**: Normal
    - **Stacking**: Stack
    - **Normalize values to percentage**: Unselected
6. Save the visualization and view it in the dashboard.
7. Select **Done editing** to view the dashboard as users will see it.

Dashboards are a great way to share data tables and visualizations with business users. You can schedule the dashboards to be refreshed periodically, and emailed to subscribers.

Delete Azure Databricks resources

Now you've finished exploring SQL Warehouses in Azure Databricks, you must delete the resources you've created to avoid unnecessary Azure costs and free up capacity in your subscription.

1. Close the Azure Databricks workspace browser tab and return to the Azure portal.
2. On the Azure portal, on the **Home** page, select **Resource groups**.
3. Select the resource group containing your Azure Databricks workspace (not the managed resource group).
4. At the top of the **Overview** page for your resource group, select **Delete resource group**.

5. Enter the resource group name to confirm you want to delete it, and select **Delete**.

   After a few minutes, your resource group and the managed workspace resource group associated with it will be deleted.