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 exercise should take approximately ****30**** minutes to complete.

## Provision an Azure Databricks workspace

****Tip****: If you already have have a Premium or Trial Azure Databricks workspace, you can skip this procedure and use your existing workspace.

This exercise includes a script to provision a new Azure Databricks workspace. The script attempts to create a Premium tier Azure Databricks workspace resource in a region in which your Azure subscription has sufficient quota for the compute cores required in this exercise; and assumes your user account has sufficient permissions in the subscription to create an Azure Databricks workspace resource. If the script fails due to insufficient quota or permissions, you can try creating an Azure Databricks workspace interactively in the Azure portal.

1. In a web browser, sign into the [Azure portal](https://portal.azure.com/) at https://portal.azure.com.
2. In the PowerShell pane, enter the following commands to clone this repo:

rm -r mslearn-databricks -f

git clone https://github.com/MicrosoftLearning/mslearn-databricks

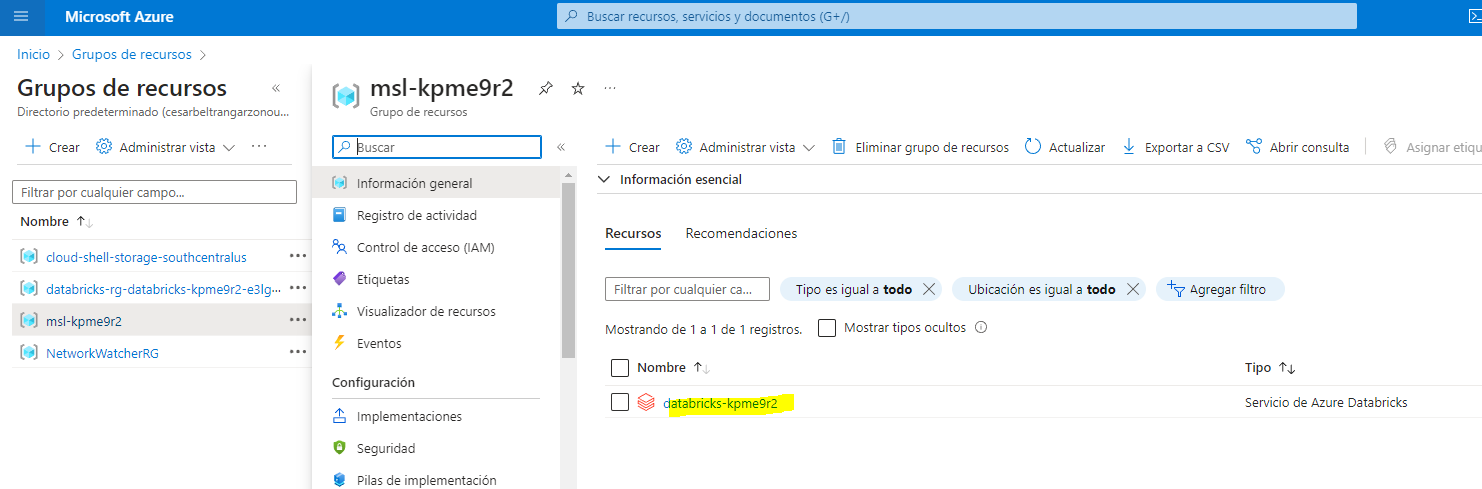
1. After the repo has been cloned, enter the following command to run the ****setup.ps1**** script, which provisions an Azure Databricks workspace in an available region:

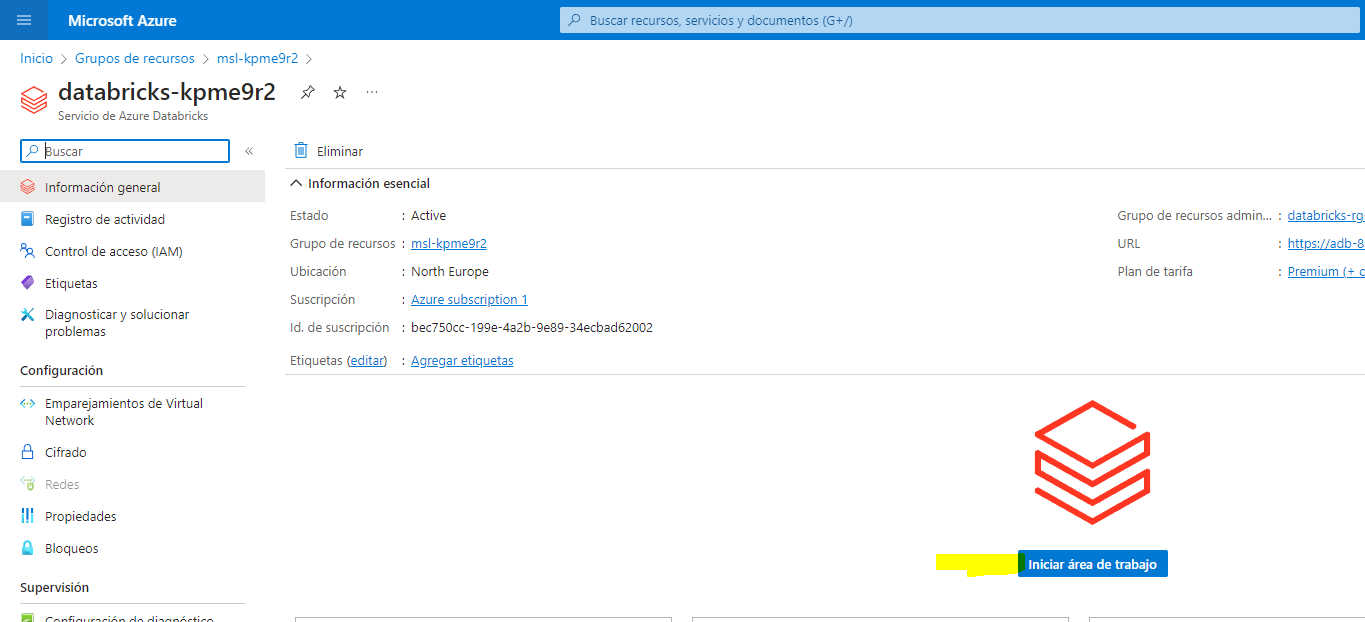
./mslearn-databricks/setup.ps1

## View and start a SQL Warehouse

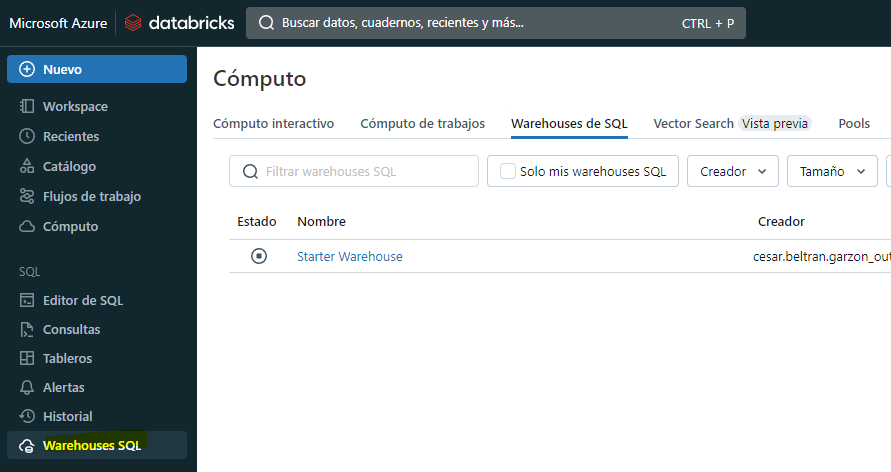
1. When the Azure Databricks workspace resource has been deployed, go to it in the Azure portal.
2. In the ****Overview**** page for your Azure Databricks workspace, use the ****Launch Workspace**** button to open your Azure Databricks workspace in a new browser tab; signing in if prompted.

****Tip****: As you use the Databricks Workspace portal, various tips and notifications may be displayed. Dismiss these and follow the instructions provided to complete the tasks in this exercise.

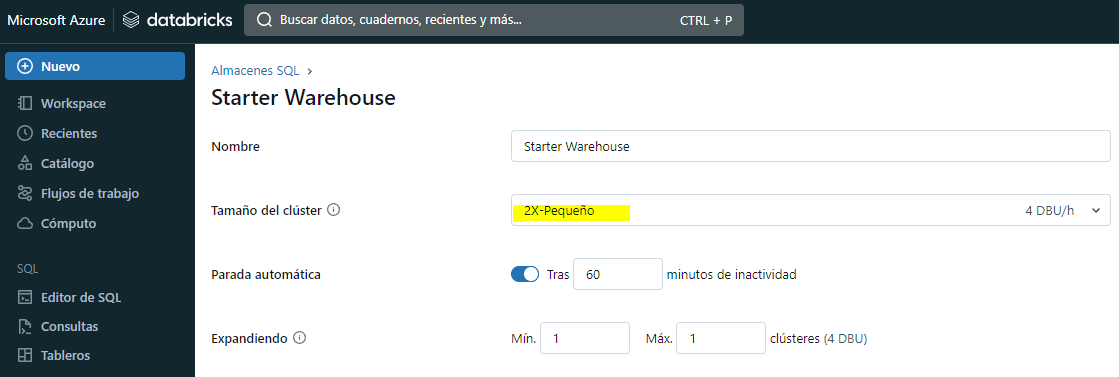




1. View the Azure Databricks workspace portal and note that the sidebar on the left side contains the names of the task categories.
2. In the sidebar, under ****SQL****, select ****SQL Warehouses****.

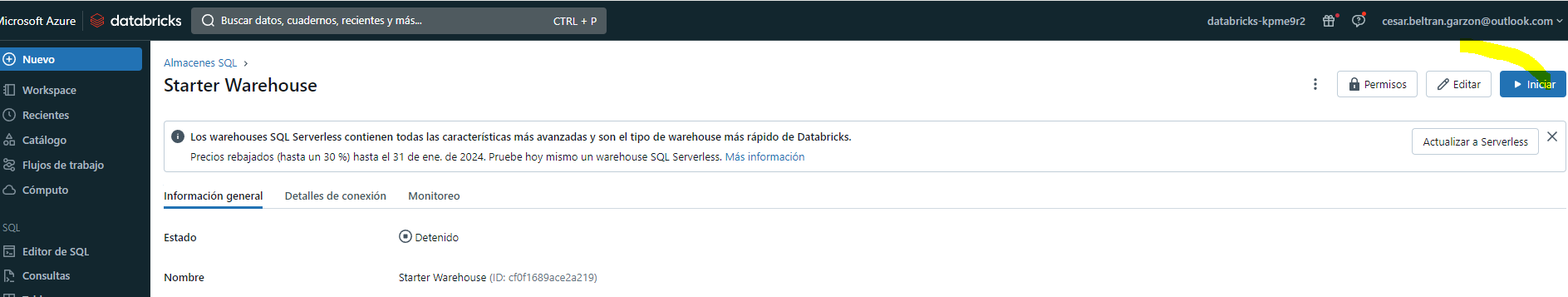


1. Observe that the workspace already includes a SQL Warehouse named ****Starter Warehouse****.
2. In the ****Actions**** (****⁝****) menu for the SQL Warehouse, select ****Edit****. Then set the ****Cluster size**** property to ****2X-Small**** and save your changes.



1. 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](https://docs.microsoft.com/azure/databricks/sql/admin/sql-endpoints" \l "required-azure-vcpu-quota) for details. If this happens, you can try requesting for a quota increase as detailed in the error message when the warehouse fails to start. Alternatively, 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 schema

1. When your SQL Warehouse is running, select ****SQL Editor**** in the sidebar.
2. In the ****Schema browser**** pane, observe that the hive\_metastore catalogue contains a database named ****default****.
3. In the ****New query**** pane, enter the following SQL code:

CREATE DATABASE retail\_db;

1. Use the ****►Run (1000)**** button to run the SQL code.
2. When the code has been successfully executed, in the ****Schema browser**** pane, use the refresh button at the bottom of the pane to refresh the list. Then expand ****hive\_metastore**** and ****retail\_db****, and observe that 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. Download the **[products.csv](https://raw.githubusercontent.com/MicrosoftLearning/mslearn-databricks/main/data/products.csv)** file to your local computer, saving it as ****products.csv****.
2. In the Azure Databricks workspace portal, in the sidebar, select ****(+) New**** and then select ****File Upload**** and upload the ****products.csv**** file you downloaded to your computer.
3. In the ****Upload data**** page, select the ****retail\_db**** schema and set the table name to ****products****. Then select ****Create table**** on the bottom left corner of the page.
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 ****(+) New**** and then select ****Query****.
2. In the ****Schema browser**** pane, expand ****hive\_metastore**** and ****retail\_db****, and verify that the ****products**** table is listed.
3. In the ****New query**** pane, enter the following SQL code:

SELECT ProductID, ProductName, CategoryFROM retail\_db.products;

1. Use the ****►Run (1000)**** button to run the SQL code.
2. When the query has completed, review the table of results.
3. 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 ****(+) New**** and then select ****Dashboard****.
2. In the ****New dashboard**** dialog box, enter the name ****Retail Dashboard**** and select ****Save****.
3. In the ****Retail Dashboard**** 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****: Leave blank
* ****Stacking****: Disabled
* ****Normalize values to percentage****: Unselected
* ****Missing and NULL values****: Do not display in chart

1. Save the visualization and view it in the dashboard.
2. 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.

## Clean up

In Azure Databricks portal, on the ****SQL Warehouses**** page, select your SQL Warehouse and select ****■ Stop**** to shut it down.

If you’ve finished exploring Azure Databricks, you can delete the resources you’ve created to avoid unnecessary Azure costs and free up capacity in your subscription.