Explore Azure Databricks

Azure Databricks is a Microsoft Azure-based version of the popular open-source Databricks platform.

Similarly to Azure Synapse Analytics, an Azure Databricks *workspace* provides a central point for managing Databricks clusters, data, and resources on Azure.

Provision an Azure Databricks workspace

> **Tip**: If you already have an Azure Databricks workspace, you can skip this procedure and use your existing workspace.

Create a cluster

Azure Databricks is a distributed processing platform that uses Apache Spark *clusters* to process data in parallel on multiple nodes. Each cluster consists of a driver node to coordinate the work, and worker nodes to perform processing tasks. In this exercise, you'll create a *single-node* cluster to minimize the compute resources used in the lab environment (in which resources may be constrained). In a production environment, you'd typically create a cluster with multiple worker nodes.

> **Tip**: If you already have a cluster with a 13.3 LTS or higher runtime version in your Azure Databricks workspace, you can use it to complete this exercise and skip this procedure.

1. In the Azure portal, browse to the resource group containing your existing Azure Databricks workspace
2. Select your Azure Databricks Service.
3. In the **Overview** page for your 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.

4. In the sidebar on the left, select the **(+) New** task, and then select **Cluster**.
5. In the **New Cluster** page, create a new cluster with the following settings:
   - **Cluster name**: *User Name's* cluster (the default cluster name)
   - **Policy**: Unrestricted
   - **Cluster mode**: Single Node
   - **Access mode**: Single user (*with your user account selected*)
   - **Databricks runtime version**: 13.3 LTS (Spark 3.4.1, Scala 2.12) or later
   - **Use Photon Acceleration**: Selected
   - **Node type**: Standard_DS3_v2

    - **Terminate after *20* minutes of inactivity**
6. Wait for the cluster to be created. It may take a minute or two.

Note: If your cluster fails to start, your subscription may have insufficient quota in the region where your Azure Databricks workspace is provisioned. See CPU core limit prevents cluster creation for details.

Use Spark to analyze data

As in many Spark environments, Databricks supports the use of notebooks to combine notes and interactive code cells that you can use to explore data.

1. Download the **products.csv** file from `raw.githubusercontent.com/parveenkrraina/WNS-B2/main/databricks/products.csv` to your local computer, saving it as **products.csv**.
2. In the sidebar, in the **(+) New** link menu, select **File upload**.
3. Upload the **products.csv** file you downloaded to your computer.
4. In the **Create or modify table from file upload** page, ensure that your cluster is selected at the top right of the page. Then choose the **hive_metastore** catalog and its default schema to create a new table named **products**.
5. In the **catalog Explorer** page when the **products** page has been created, in the **Create** button menu, select **Notebook** to create a notebook.
6. In the notebook, ensure that the notebook is connected to your cluster and then review the code that has been automatically been added to the first cell; which should look similar to this:

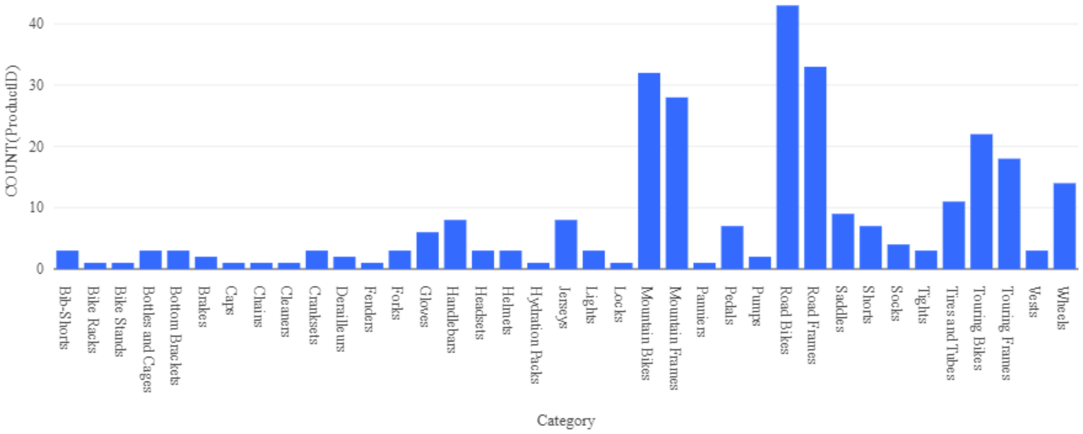
   codeCopy

   ```
   %sql
   SELECT * FROM `hive_metastore`.`default`.`products`;
   ```

7. Use the ▸ **Run Cell** menu option at the left of the cell to run it, starting and attaching the cluster if prompted.
8. Wait for the Spark job run by the code to complete. The code retrieves data from the table that was created based on the file you uploaded.
9. Above the table of results, select **+** and then select **Visualization** to view the visualization editor, and then apply the following options:
    - **Visualization type**: Bar
    - **X Column**: Category

- **Y Column**: *Add a new column and select* **ProductID**. *Apply the* **Count** *aggregation*.

Save the visualization and observe that it is displayed in the notebook, like this:



Analyze data with a dataframe

While most data analysis are comfortable using SQL code as used in the previous example, some data analysts and data scientists can use native Spark objects such as a *dataframe* in programming languages such as *PySpark* (A Spark-optimized version of Python) to work efficiently with data.

1. In the notebook, under the chart output from the previously run code cell, use the **+** icon to add a new cell.
2. Enter and run the following code in the new cell:

codeCopy

```
df = spark.sql("SELECT * FROM products")
df = df.filter("Category == 'Road Bikes'")
display(df)
```

3. Run the new cell, which returns products in the *Road Bikes* category.

Clean up

In Azure Databricks portal, on the **Compute** page, select your cluster and select ■ **Terminate** 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.