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.

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

## Provision an Azure Databricks workspace

****Tip****: If you already have an 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 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

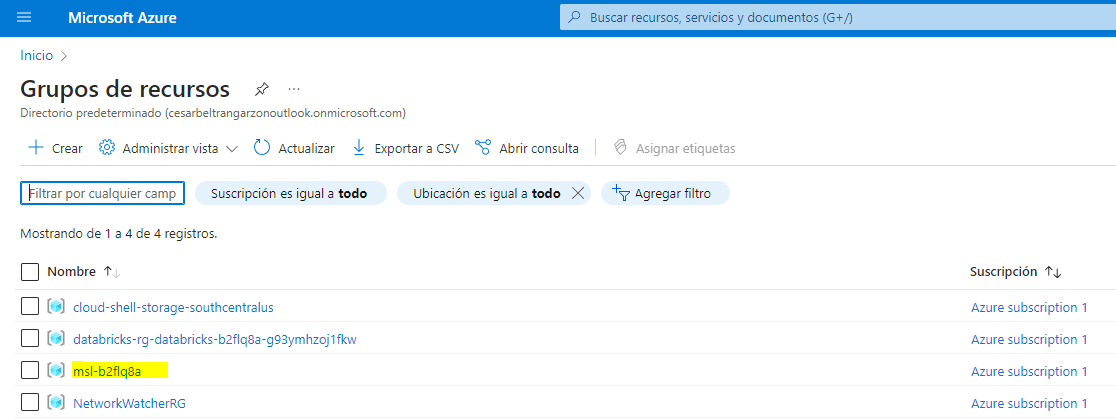
1. If prompted, choose which subscription you want to use (this will only happen if you have access to multiple Azure subscriptions).
2. Wait for the script to complete - this typically takes around 5 minutes, but in some cases may take longer. While you are waiting, review the [What is Azure Databricks?](https://learn.microsoft.com/azure/databricks/introduction/) article in the Azure Databricks documentation.

## Create a cluster

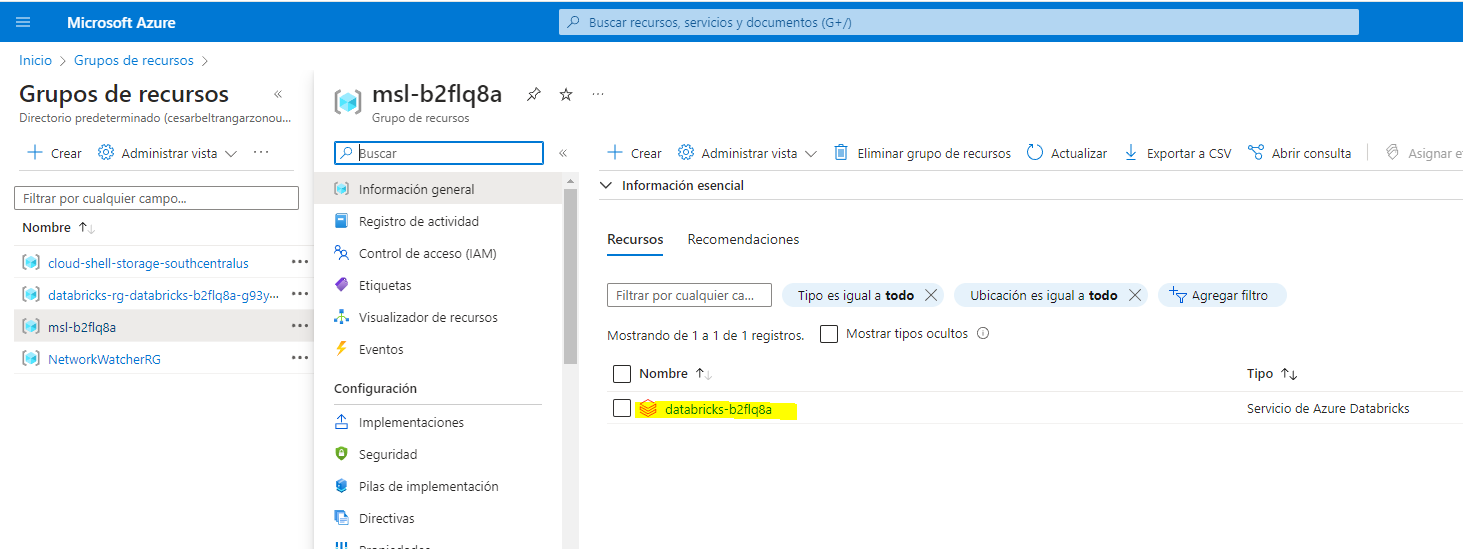
Azure Databricks is a distributed processing platform that uses Apache Spark clusters to process data in parallel on multiple nodes. Each cluster consists of a driver node to coordinate the work, and worker nodes to perform processing tasks. In this exercise, you’ll create a single-node cluster to minimize 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 ****msl-**xxxxxxx** resource group that was created by the script (or the resource group containing your existing Azure Databricks workspace)

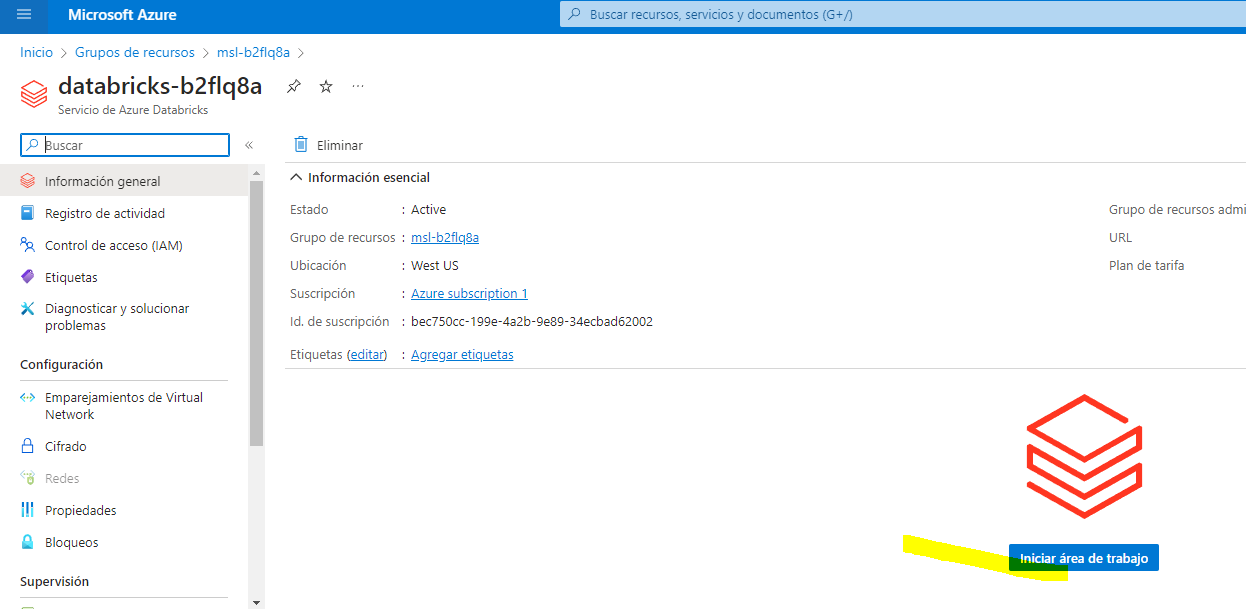


1. Select your Azure Databricks Service resource (named ****databricks-**xxxxxxx** if you used the setup script to create it).

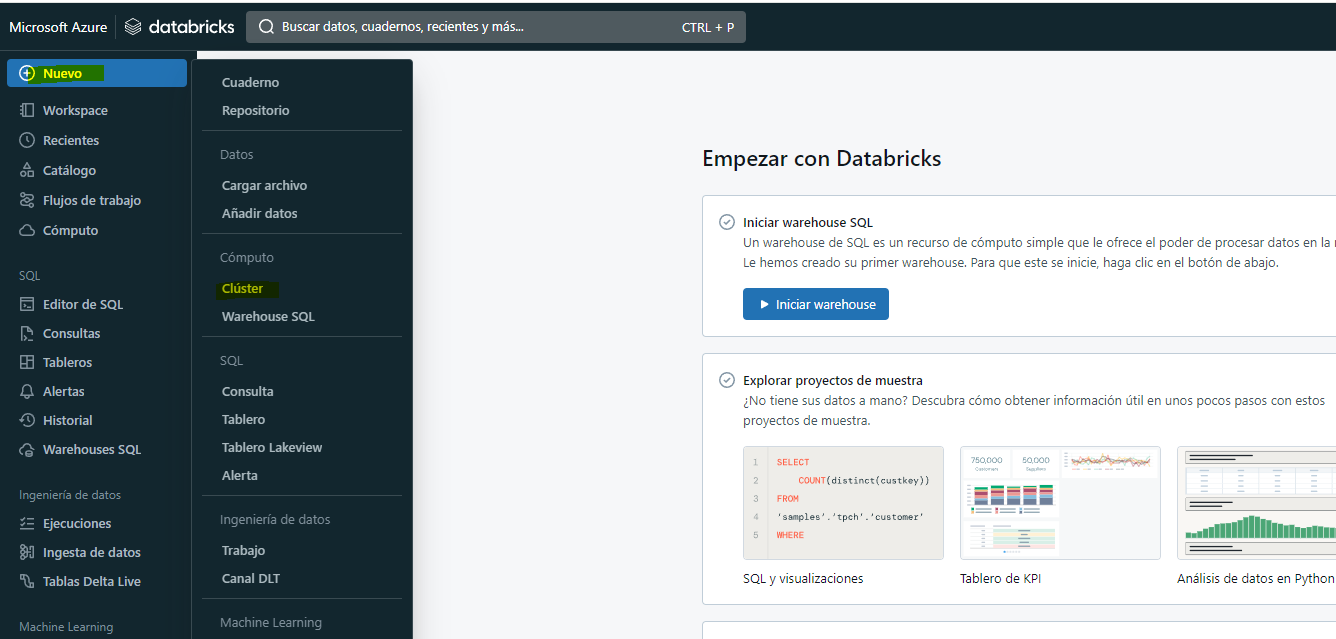


1. In the ****Overview**** page for your workspace, use the ****Launch Workspace**** button to open your Azure Databricks workspace in a new browser tab; signing in if prompted.

****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.

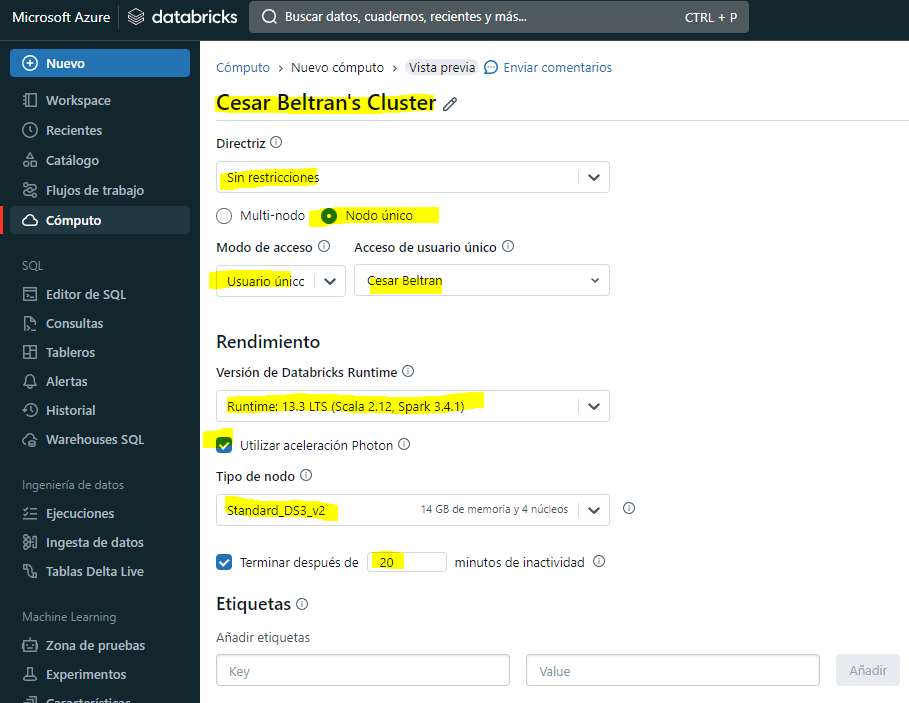


1. In the sidebar on the left, select the ****(+) New**** task, and then select ****Cluster****.



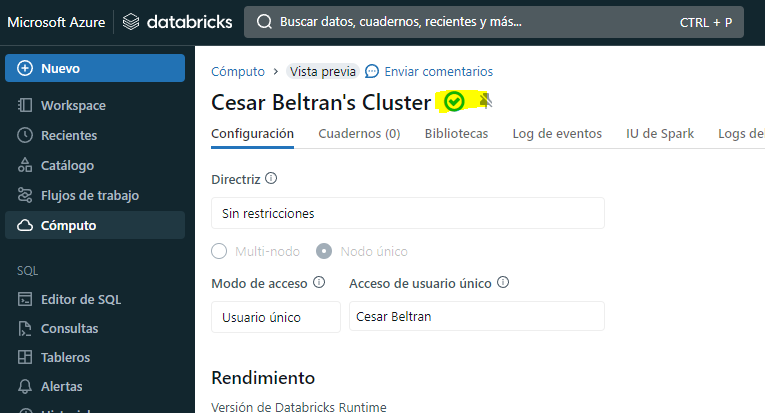
1. In the ****New Cluster**** page, create a new cluster with the following settings:

* ****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****



1. Wait for the cluster to be created. It may take a minute or two.

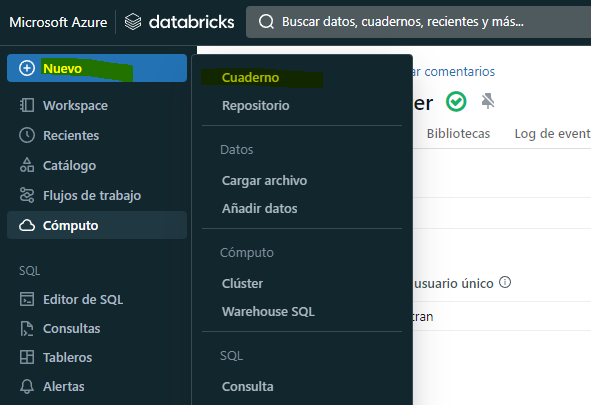
****Note****: If your cluster fails to start, your subscription may have insufficient quota in the region where your Azure Databricks workspace is provisioned. See [CPU core limit prevents cluster creation](https://docs.microsoft.com/azure/databricks/kb/clusters/azure-core-limit) for details. If this 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: ./mslearn-databricks/setup.ps1 eastus



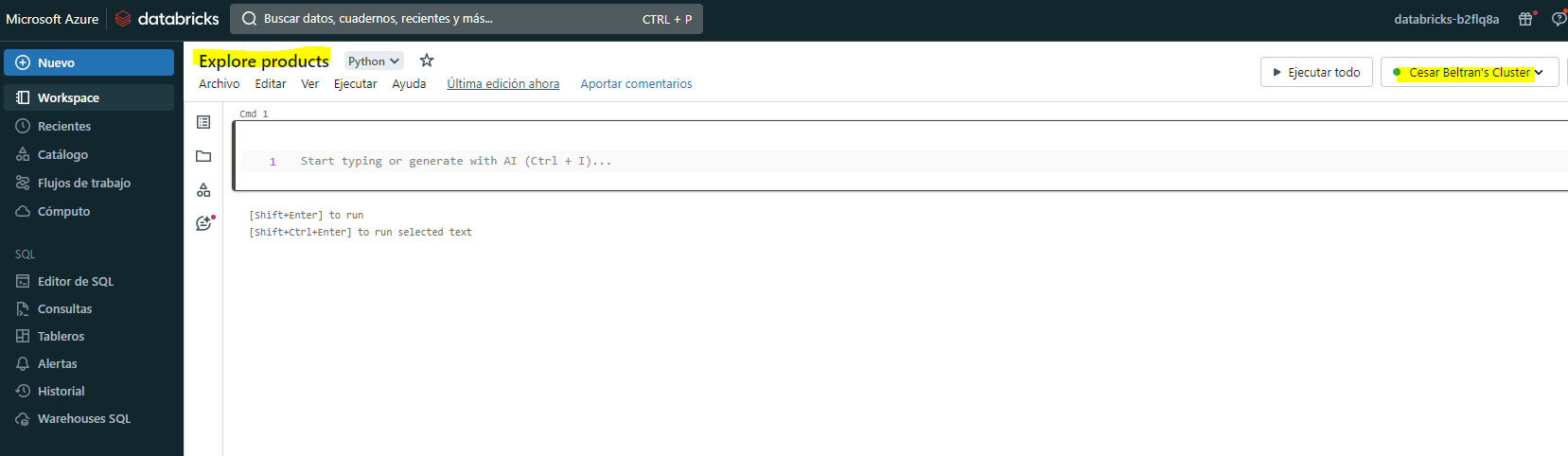
## Use Spark to analyze a data file

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.

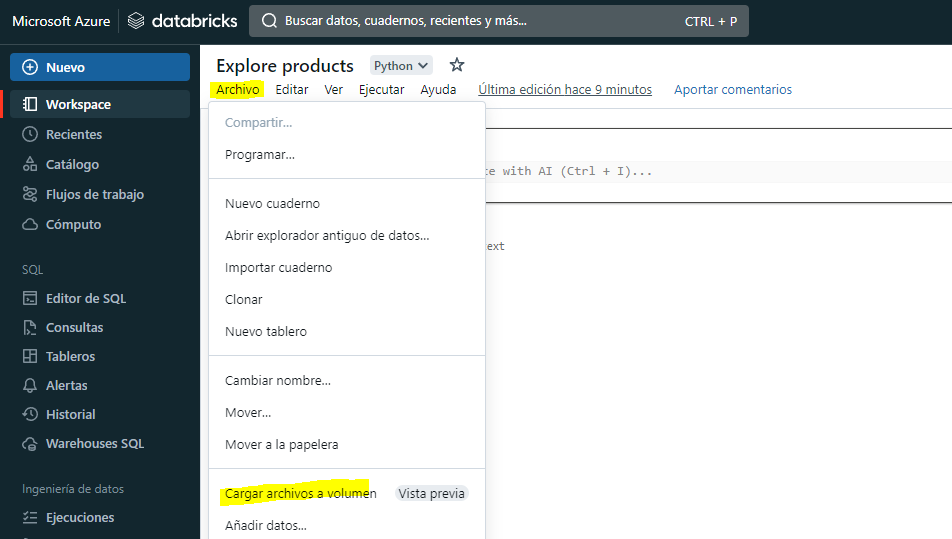
1. In the sidebar, use the ****(+) New**** link to create a ****Notebook****.



1. Change the default notebook name (****Untitled Notebook**[date]**) to ****Explore products**** and in the ****Connect**** drop-down list, select your cluster if it is not already selected. If the cluster is not running, it may take a minute or so to start.

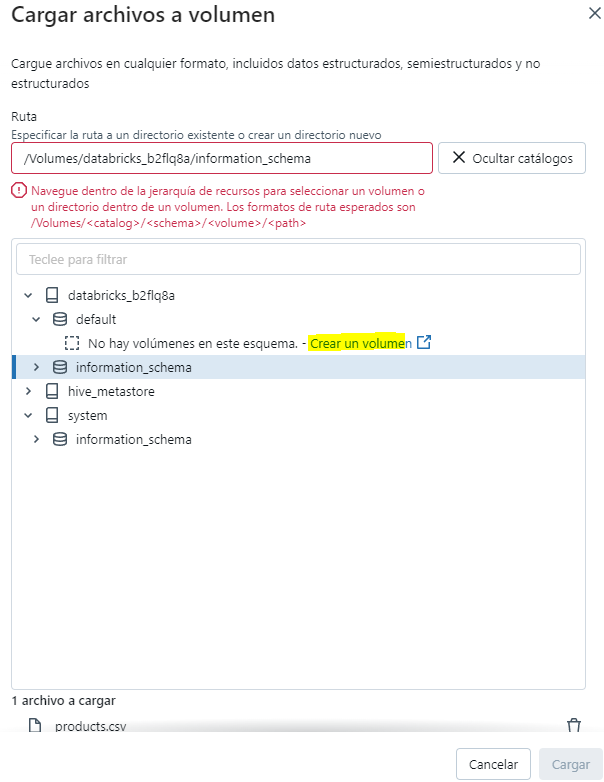
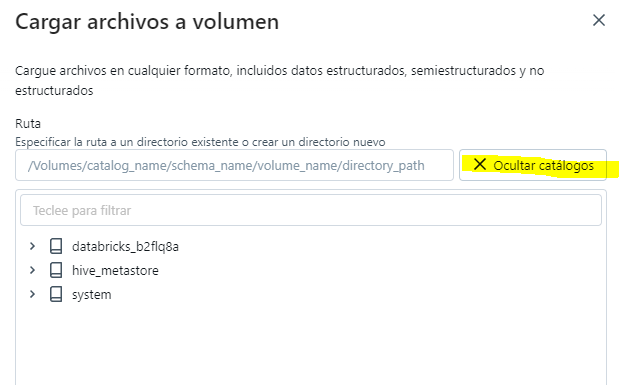


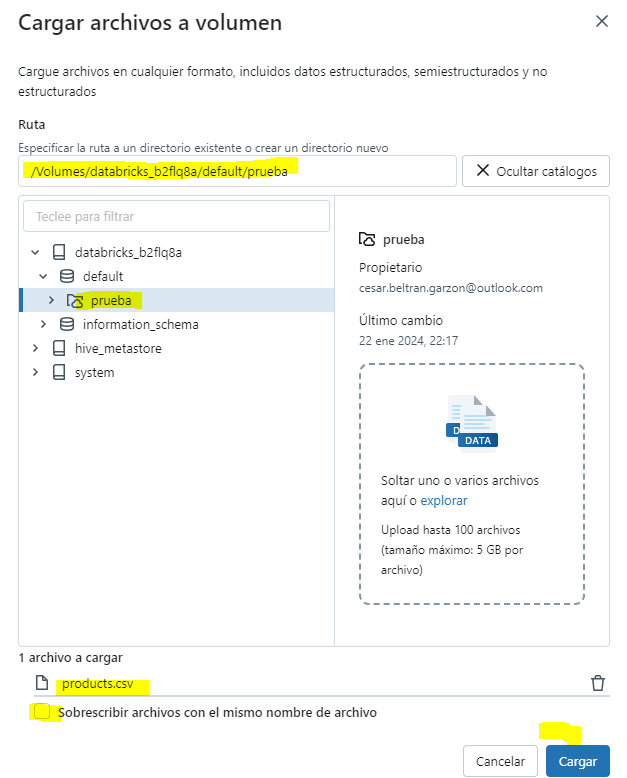
1. Download the **[products.csv](https://raw.githubusercontent.com/MicrosoftLearning/mslearn-databricks/main/data/products.csv)** file from https://raw.githubusercontent.com/MicrosoftLearning/mslearn-databricks/main/data/products.csv to your local computer, saving it as ****products.csv****. Then, in the ****Explore products**** notebook, on the ****File**** menu, select ****Upload data to DBFS****.





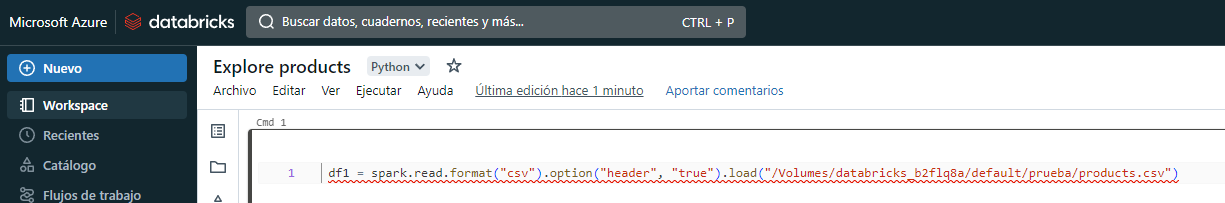
1. In the ****Upload Data**** dialog box, note the ****DBFS Target Directory**** to where the file will be uploaded. Then select the ****Files**** area, and upload the ****products.csv**** file you downloaded to your computer. When the file has been uploaded, select ****Next****



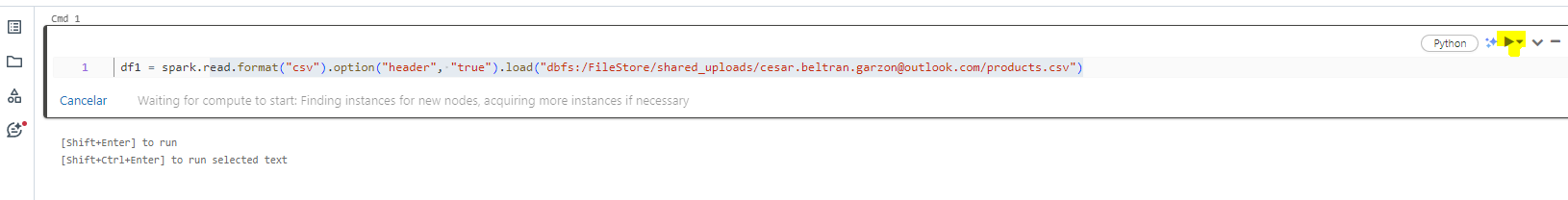


1. In the ****Access files from notebooks**** pane, select the sample PySpark code and copy it to the clipboard. You will use it to load the data from the file into a DataFrame. Then select ****Done****.
2. In the ****Explore products**** notebook, in the empty code cell, paste the code you copied; which should look similar to this:

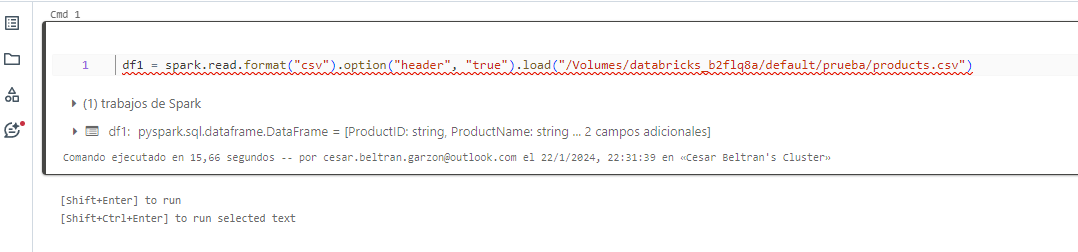
df1 = spark.read.format("csv").option("header", "true").load("dbfs:/FileStore/shared\_uploads/user@outlook.com/products.csv")



1. Use the ****▸ Run Cell**** menu option at the top-right of the cell to run it, starting and attaching the cluster if prompted.



1. Wait for the Spark job run by the code to complete. The code has created a dataframe object named ****df1**** from the data in the file you uploaded.

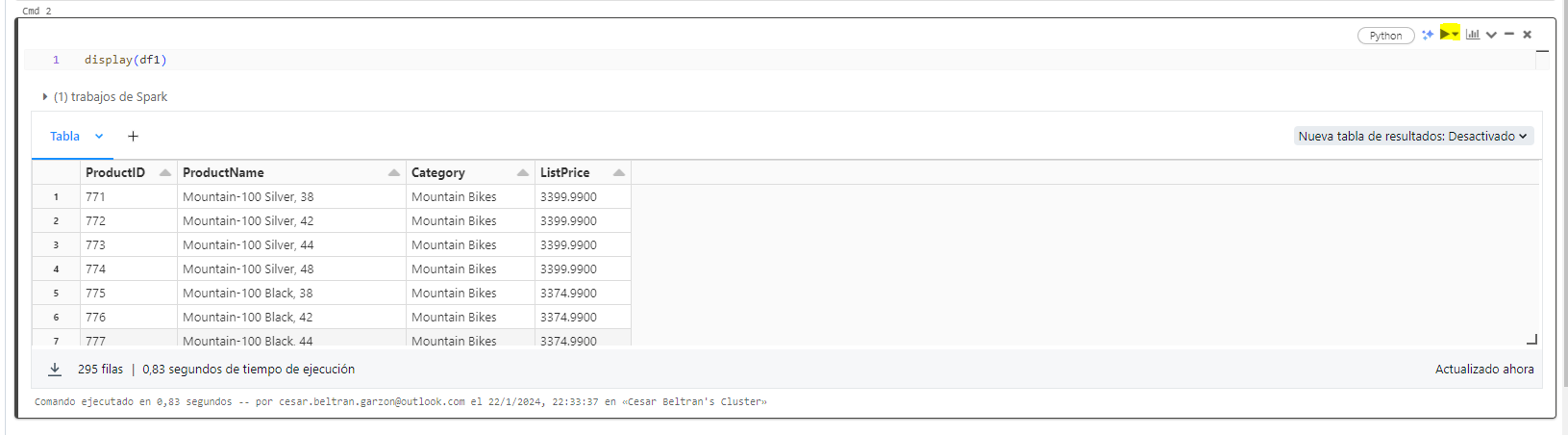


1. Under the existing code cell, use the ****+**** icon to add a new code cell. Then in the new cell, enter the following code:

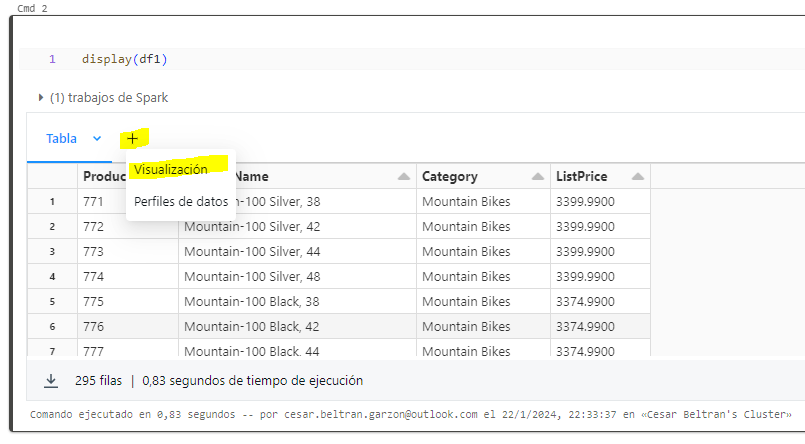
display(df1)

1. Use the ****▸ Run Cell**** menu option at the top-right of the new cell to run it. This code displays the contents of the dataframe, which should look similar to this:

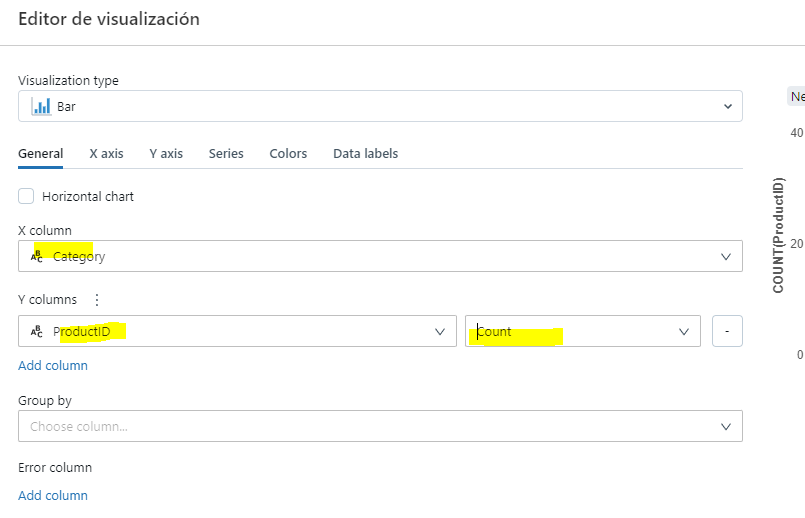
| ProductID | ProductName | Category | ListPrice |
| --- | --- | --- | --- |
| 771 | Mountain-100 Silver, 38 | Mountain Bikes | 3399.9900 |
| 772 | Mountain-100 Silver, 42 | Mountain Bikes | 3399.9900 |
| … | … | … | … |



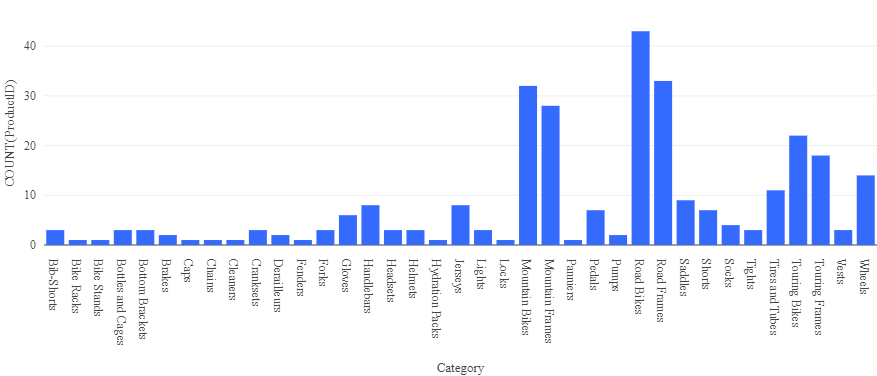
1. Above the table of results, select ****+**** and then select ****Visualization**** to view the visualization editor, and then apply the following options:

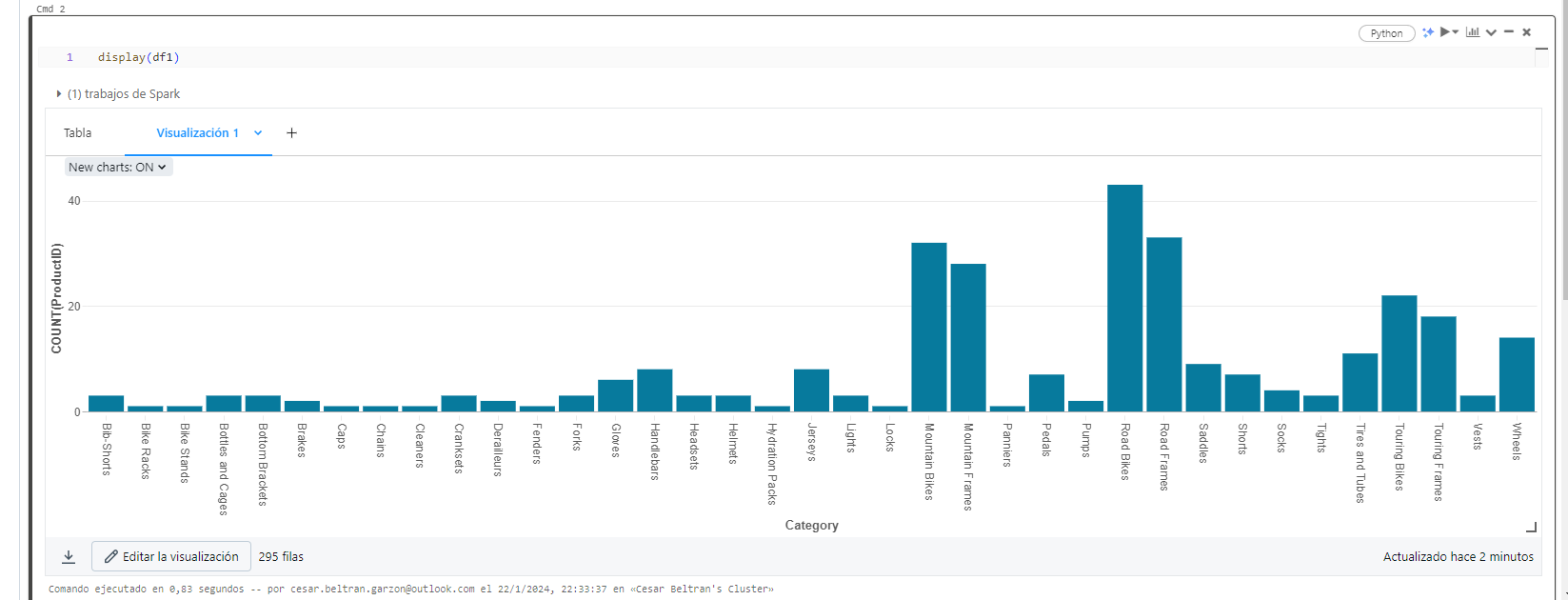


* ****Visualization type****: Bar
* ****X Column****: Category
* ****Y Column****: Add a new column and select ****ProductID****. Apply the ****Count**** aggregation.



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

[](https://microsoftlearning.github.io/mslearn-databricks/Instructions/Exercises/images/databricks-chart.png)



## Create and query a table

While many data analysis are comfortable using languages like Python or Scala to work with data in files, a lot of data analytics solutions are built on relational databases; in which data is stored in tables and manipulated using SQL.

1. In the ****Explore products**** 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:

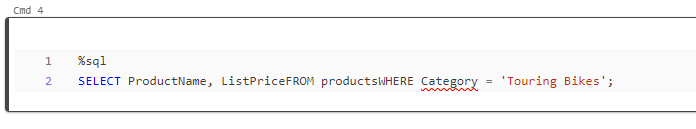
df1.write.saveAsTable("products")



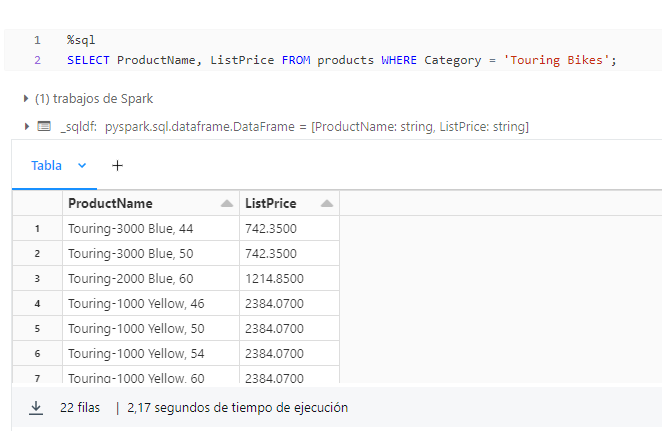
1. When the cell has completed, add a new cell under it with the following code:

%sql

SELECT ProductName, ListPrice FROM products WHERE Category = 'Touring Bikes';



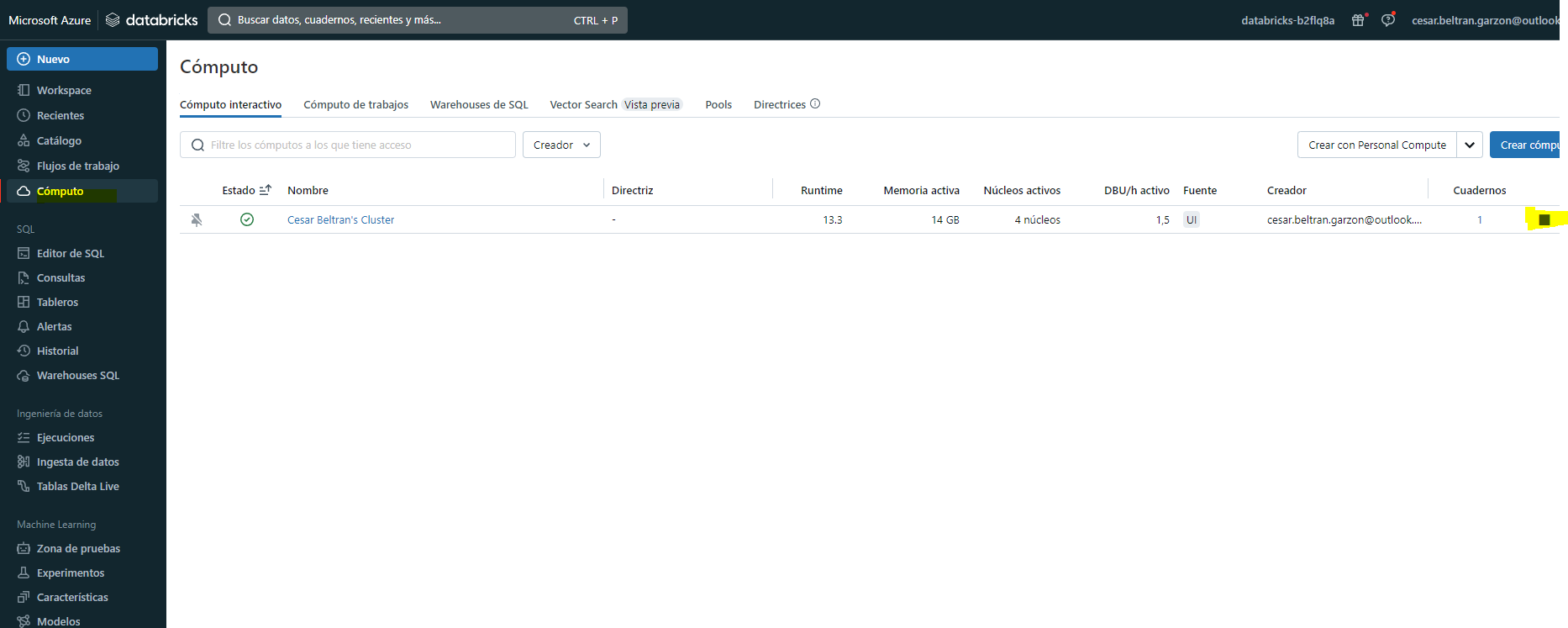
1. Run the new cell, which contains SQL code to return the name and price of products in the Touring Bikes category.



1. In the sidebar, select the ****Catalog**** link, and verify that the ****products**** table has been created in the default database schema (which is unsurprisingly named ****default****). It’s possible to use Spark code to create custom database schemas and a schema of relational tables that data analysts can use to explore data and generate analytical reports.

## 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.