Run a Spark job on Azure Databricks using the Azure portal

**Step 1:** Sign into your Azure Portal using your credentials and click on **Create a resource** in the upper left-hand corner of the Azure portal and select **Azure Databricks** from the **Analytics** category. Fill the necessary details as below to create your Azure Databricks

**Workplace Name**: Any name as an identifier

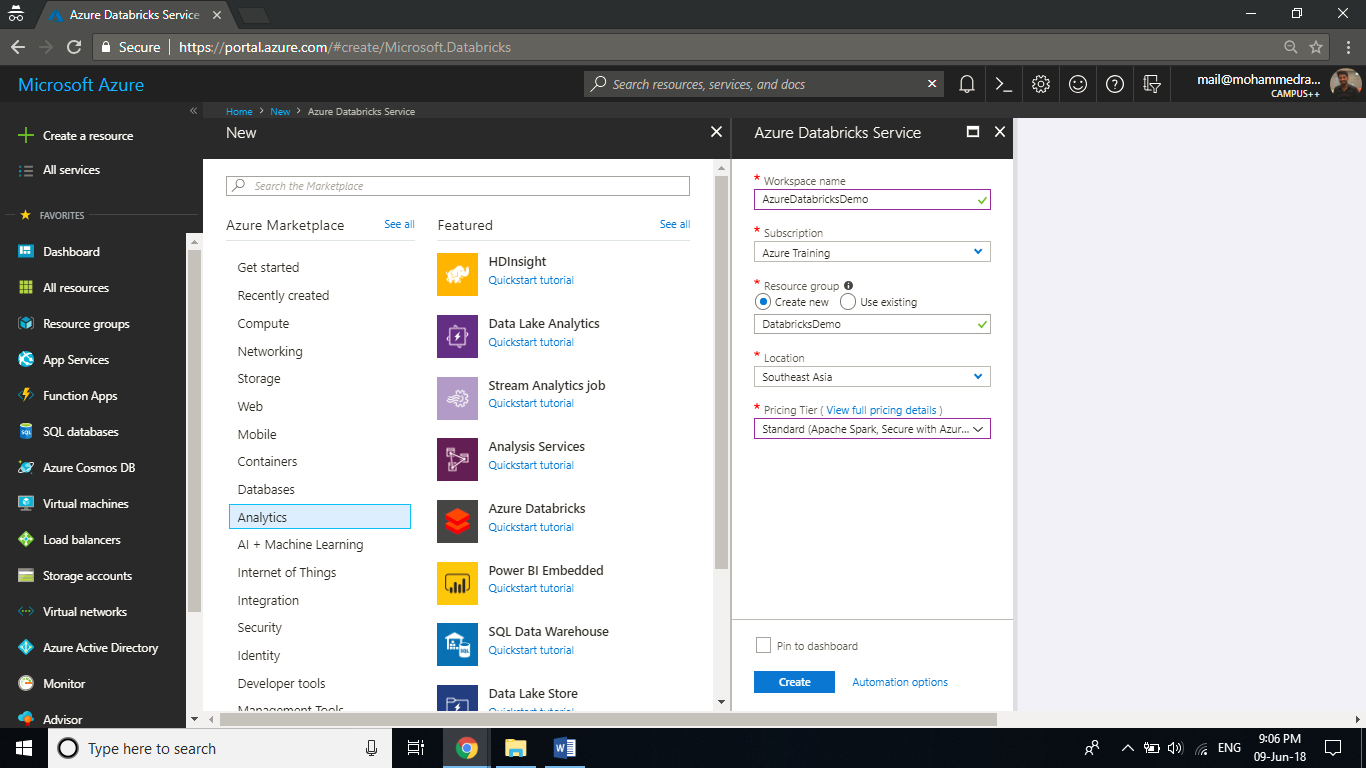
**Subscription:** Select your subscription

**Resource group**: Create a new resource group with a name

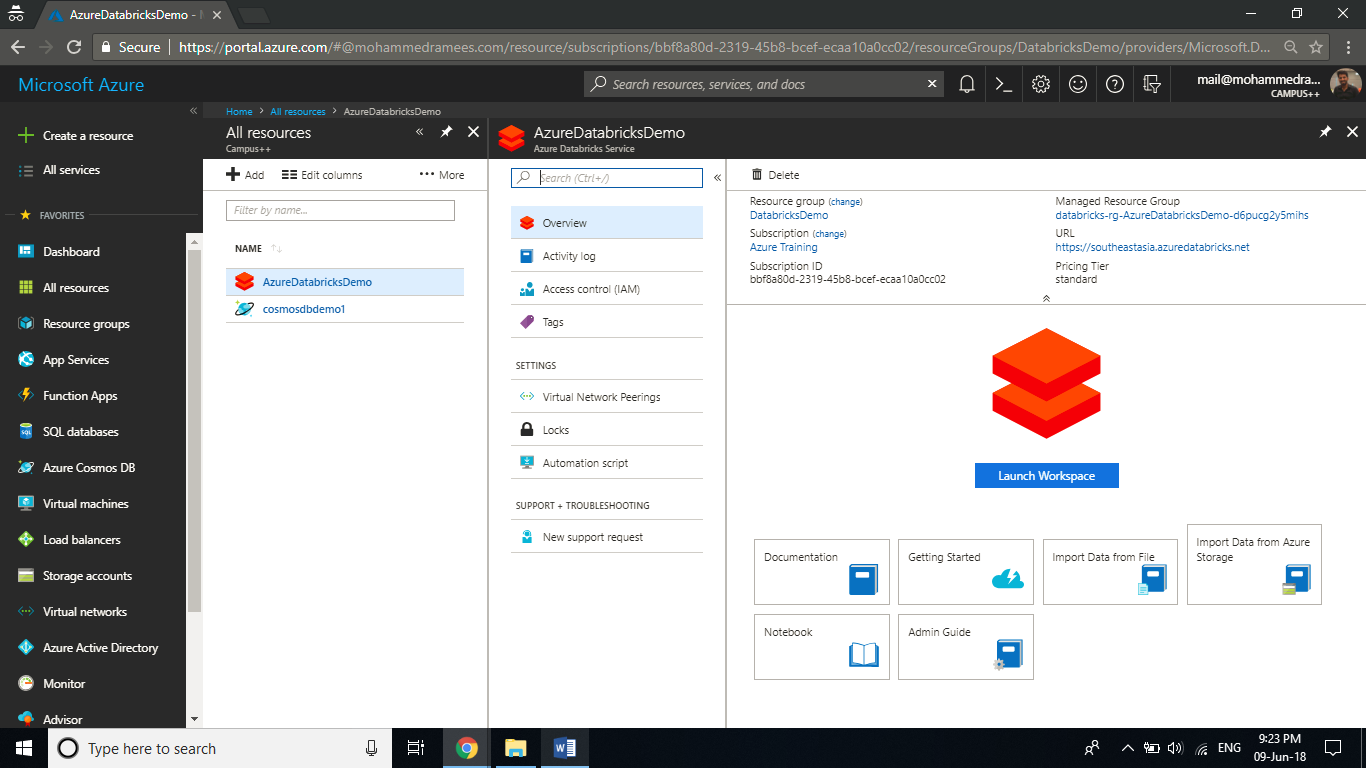
**Location:** Select the nearest location

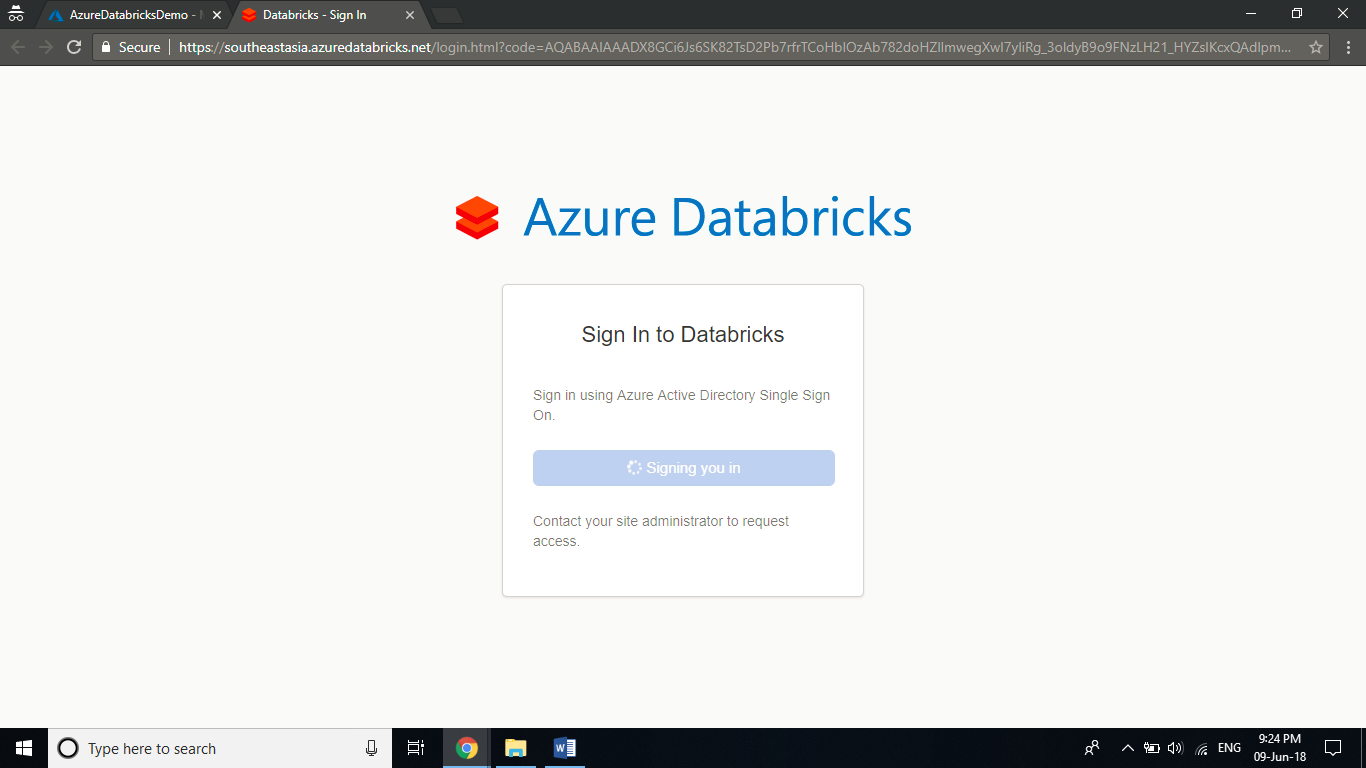
**Pricing Tier:** Standard

Click on **Create** to provision and wait for few mintues for the provision to be succeded.

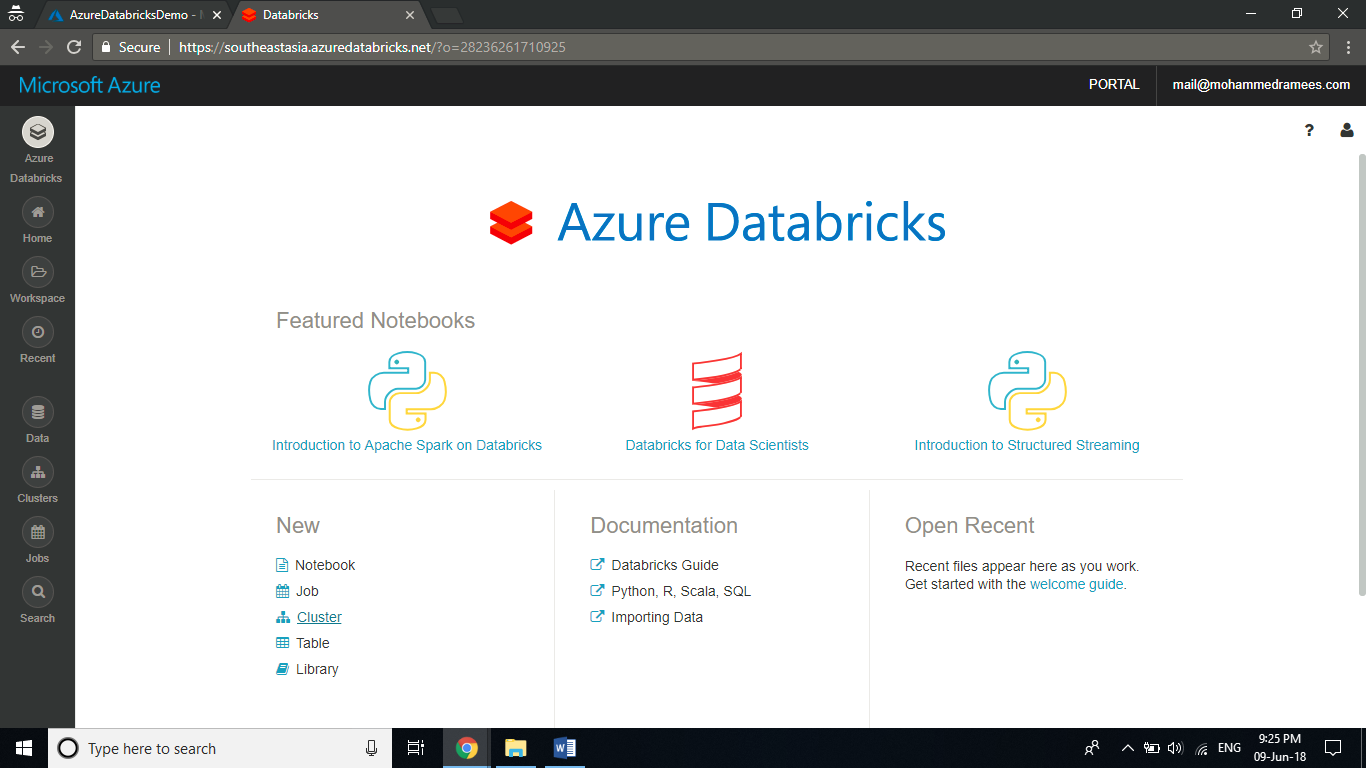


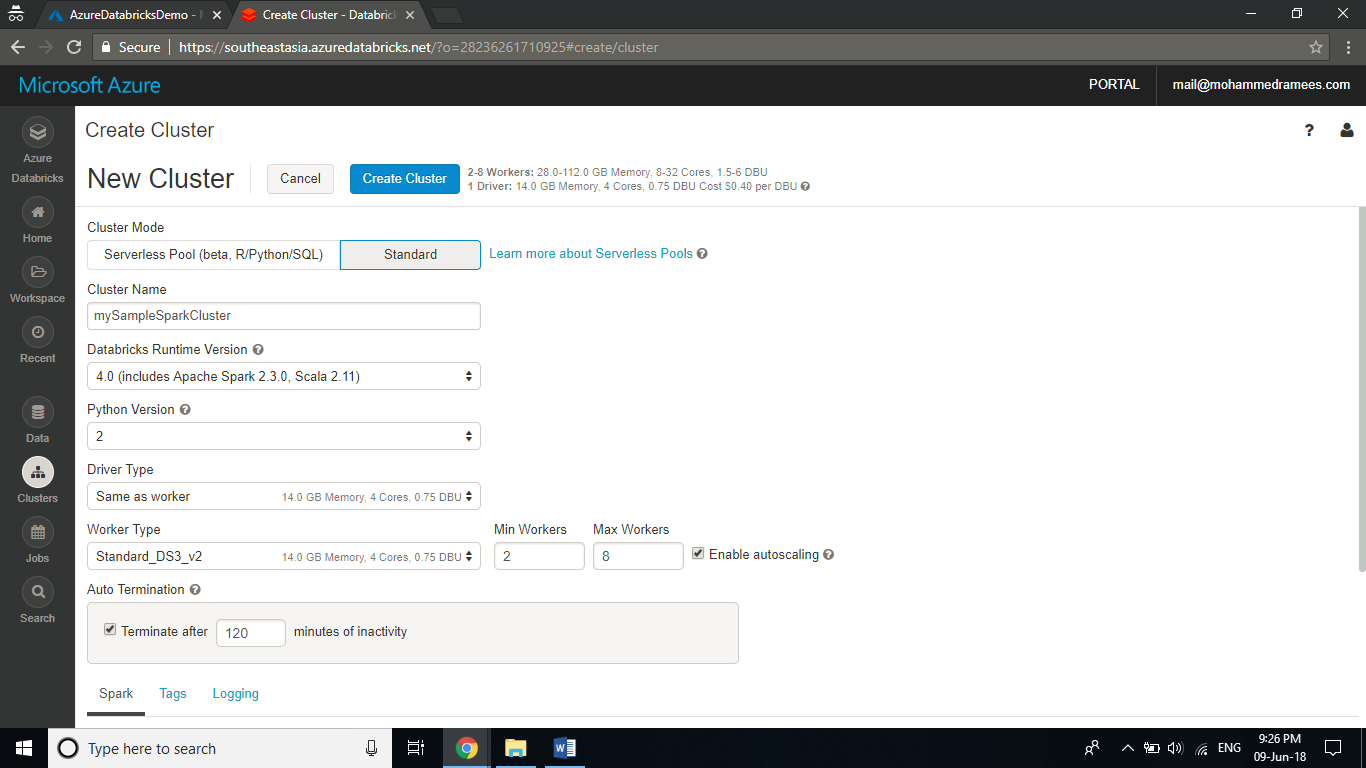
**Step 2:** In the Azure portal, go to the Databricks workspace that you created, and then click **Launch Workspace**. You are redirected to the Azure Databricks portal by singing with Azure Active Directory.





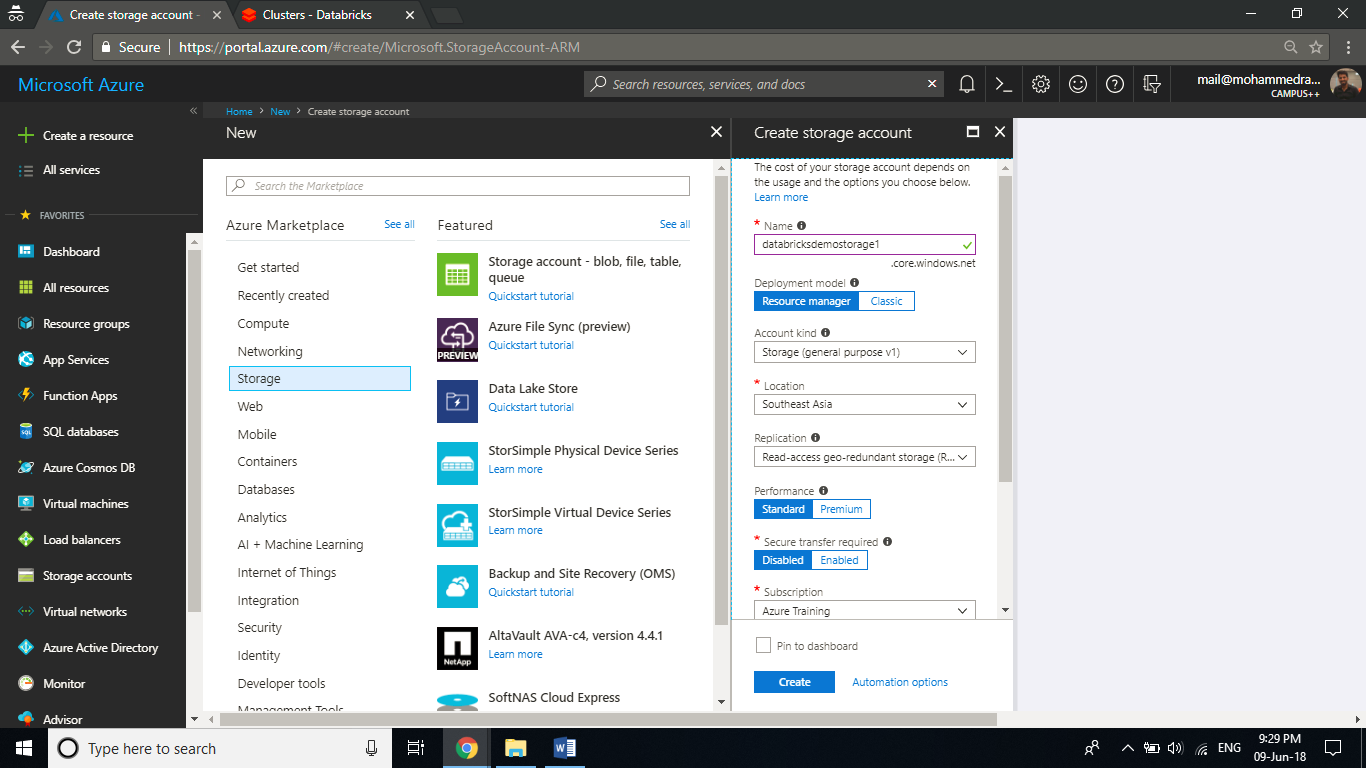
**Step 3:** From the portal, click **Cluster** and in the new cluster page, provide the values **to create a cluster**.

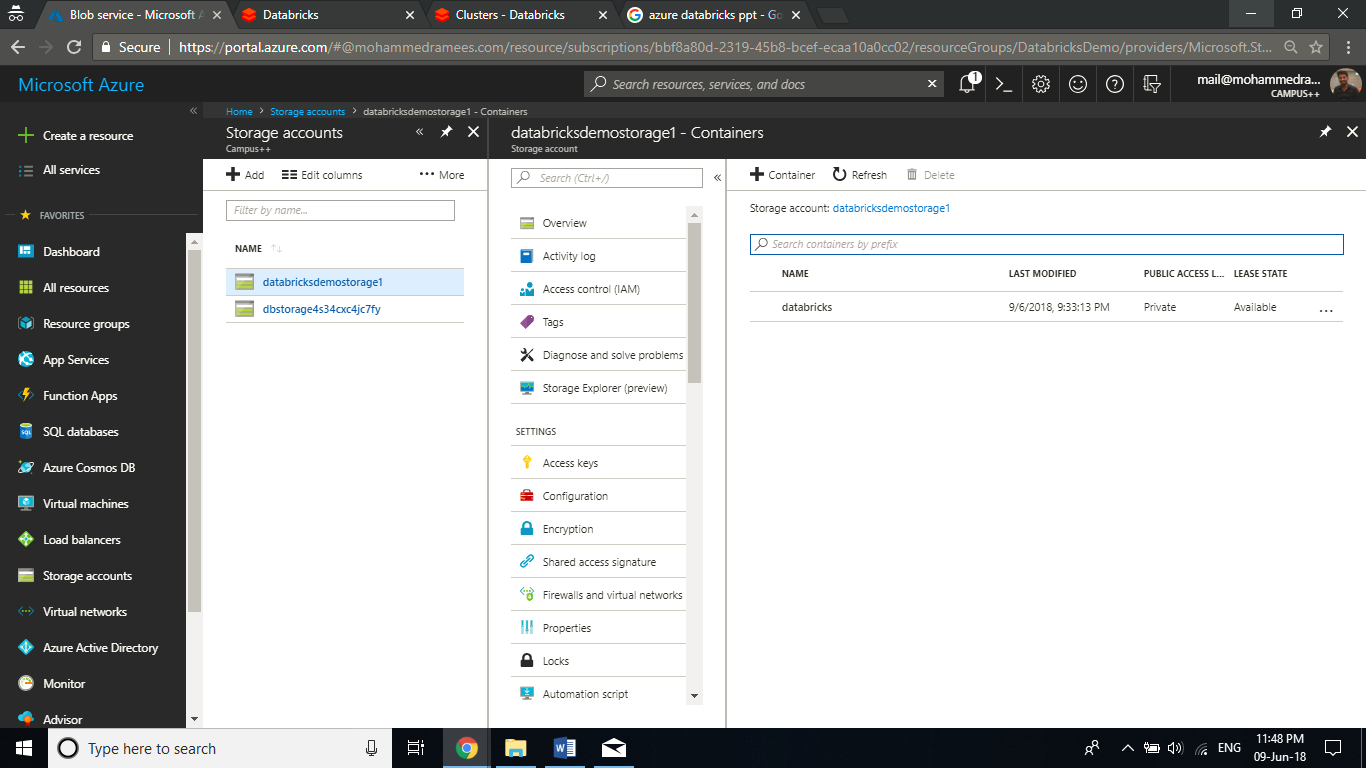


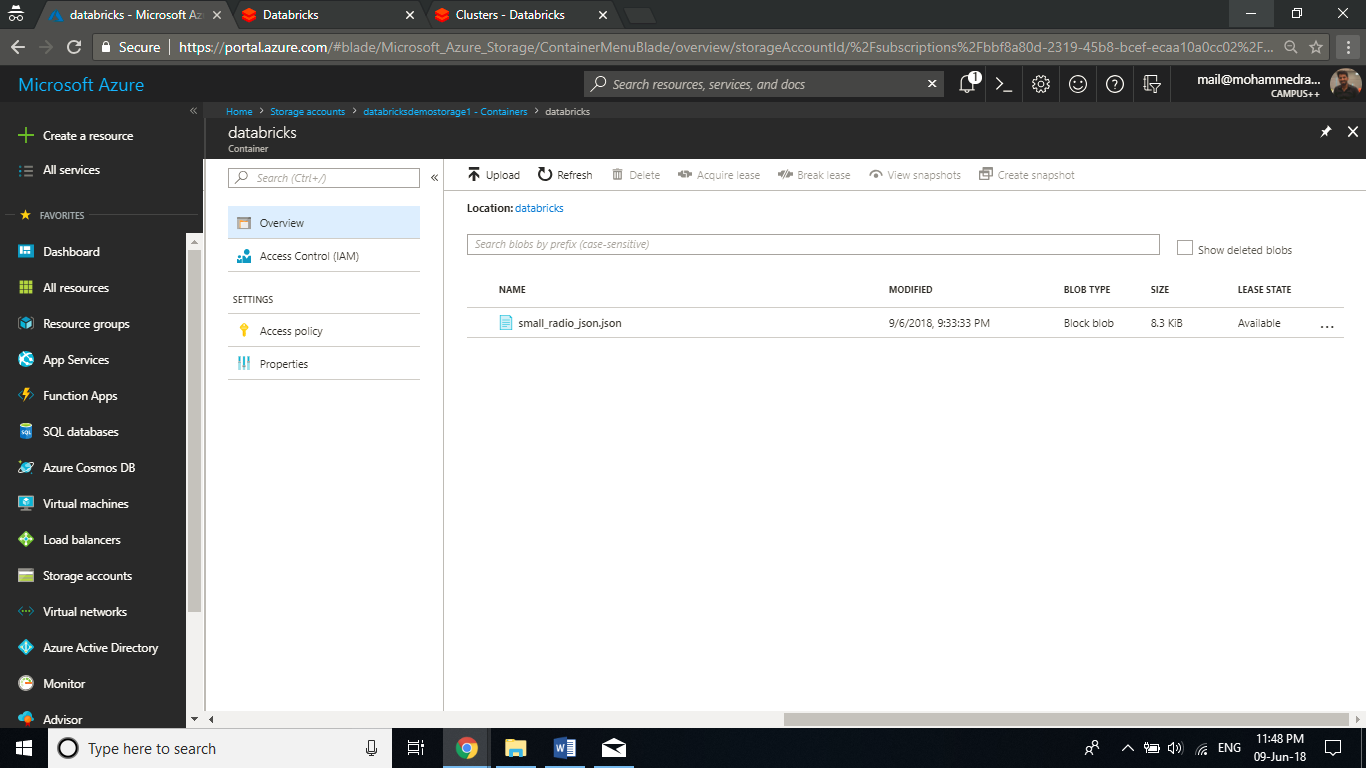


**Step 4:** Create a new storage account, and create a container inside the blob and upload the json file from the from the following link

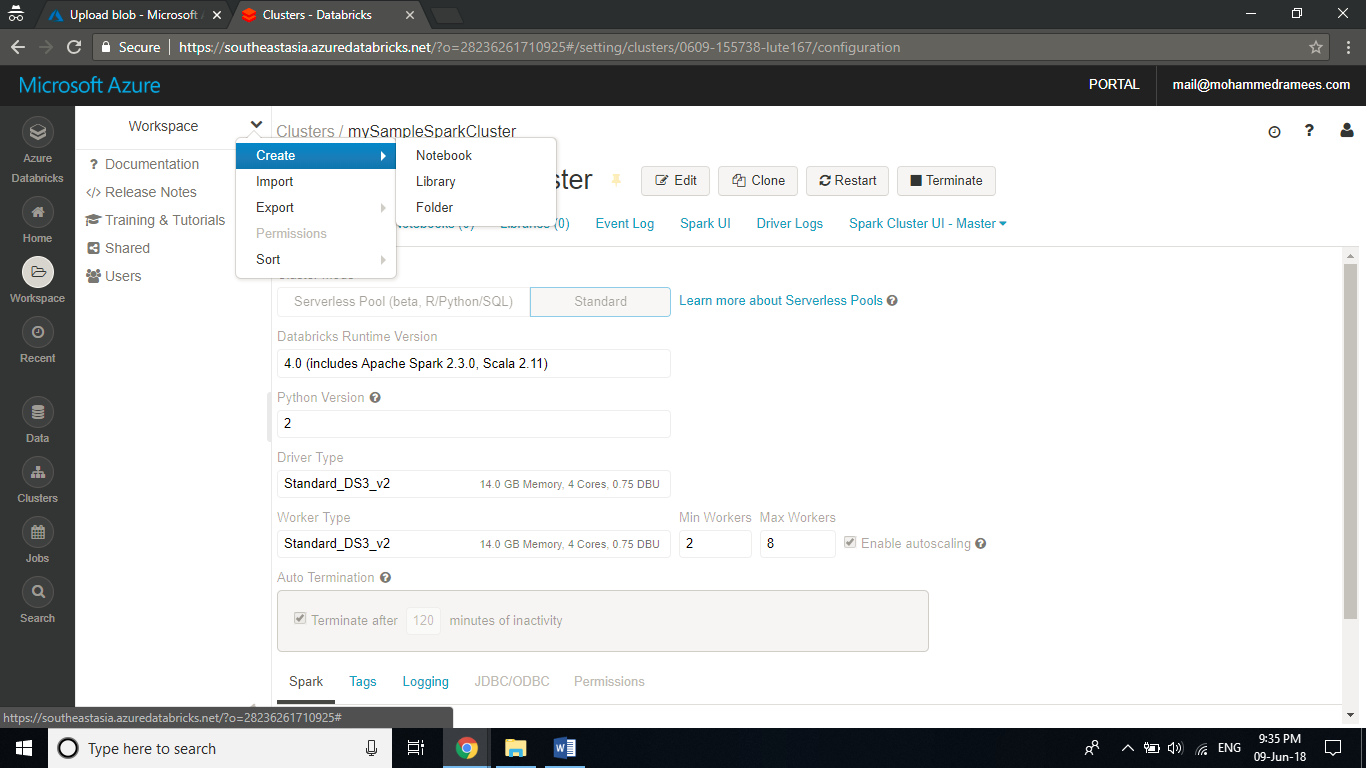
<https://github.com/Azure/usql/blob/master/Examples/Samples/Data/json/radiowebsite/small_radio_json.json>



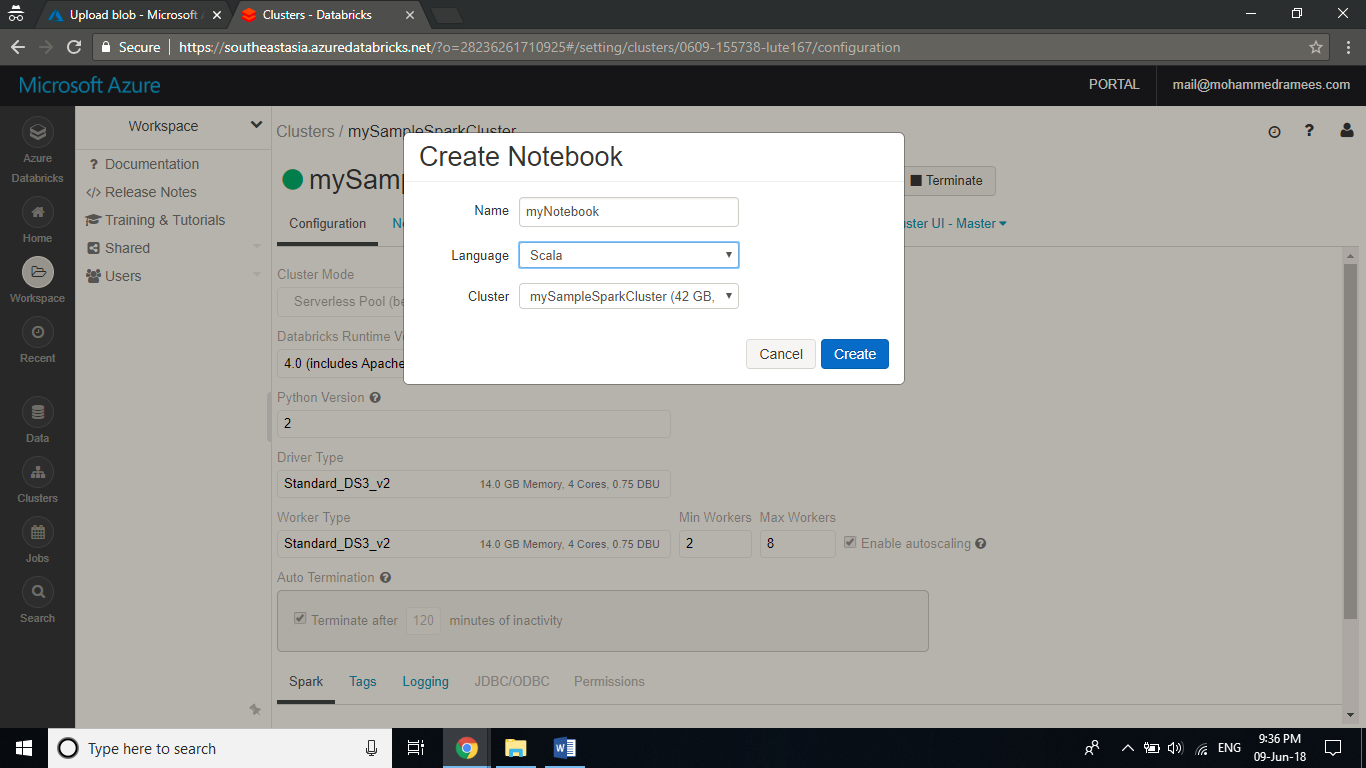




**Step 5:** Now we have to create a notebook in Databricks, configure the notebook to read data from an Azure Blob storage account, and then run a Spark SQL job on the data. In the left pane, click **Workspace**. From the **Workspace** drop-down, click **Create**, and then click Notebook.



**Step 6:** In the Create **Notebook** dialog box, enter a name, select **Scala** as the language, and select the Spark cluster that you created earlier.



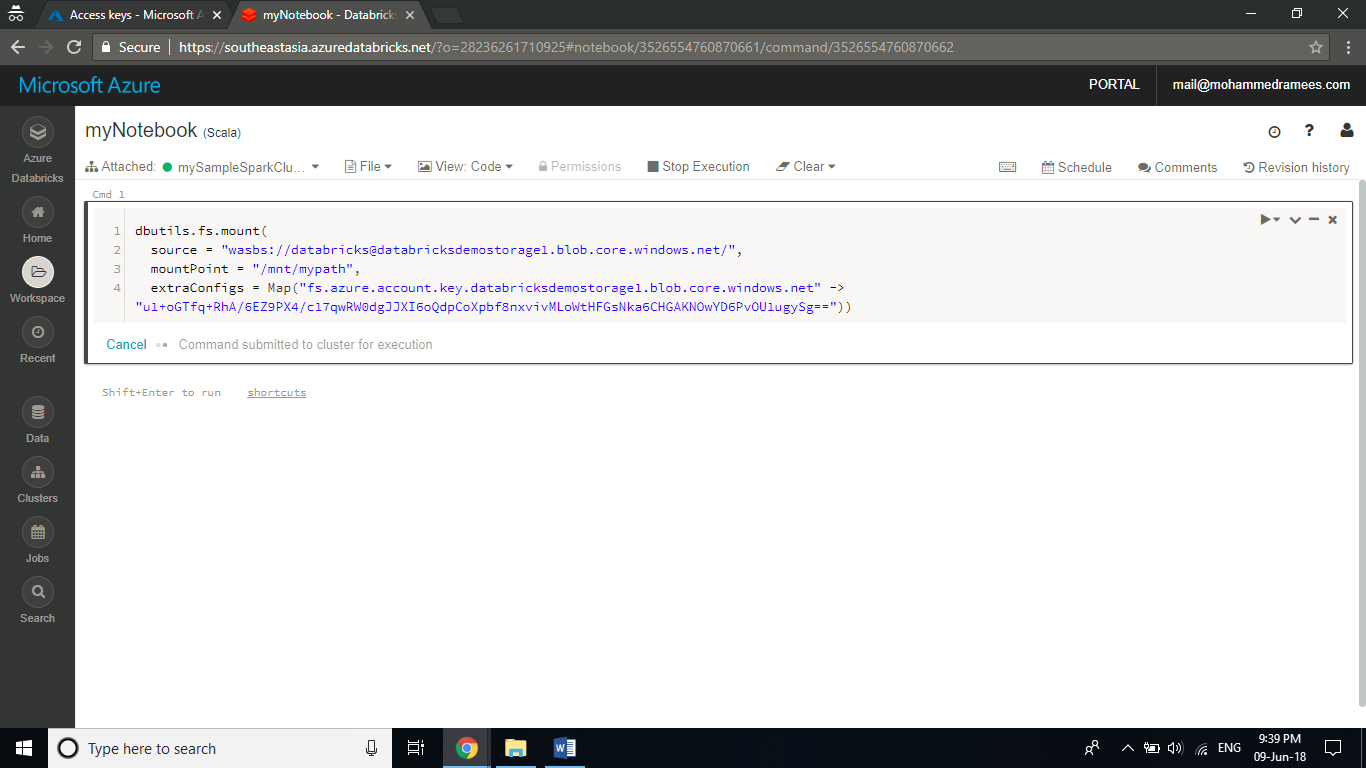
**Step 7:** In the following snippet, replace {YOUR CONTAINER NAME}, {YOUR STORAGE ACCOUNT NAME}, and {YOUR STORAGE ACCOUNT ACCESS KEY} with the appropriate values for your Azure Storage account. Paste the snippet in an empty cell in the notebook and then press SHIFT + ENTER to run the code cell**.**

dbutils.fs.mount(

source = "wasbs://{YOUR CONTAINER NAME}@{YOUR STORAGE ACCOUNT NAME}.blob.core.windows.net/",

mountPoint = "/mnt/mypath",

extraConfigs = Map("fs.azure.account.key.{YOUR STORAGE ACCOUNT NAME}.blob.core.windows.net" -> "{YOUR STORAGE ACCOUNT ACCESS KEY}")



**Step 8:** Run a SQL statement to create a temporary table using data from the sample JSON data file, small\_radio\_json.json. In the following snippet, replace the placeholder values with your container name and storage account name. Paste the snippet in a code cell in the notebook, and then press SHIFT + ENTER. In the snippet, path denotes the location of the sample JSON file that you uploaded to your Azure Storage account.

%sql

DROP TABLE IF EXISTS radio\_sample\_data;

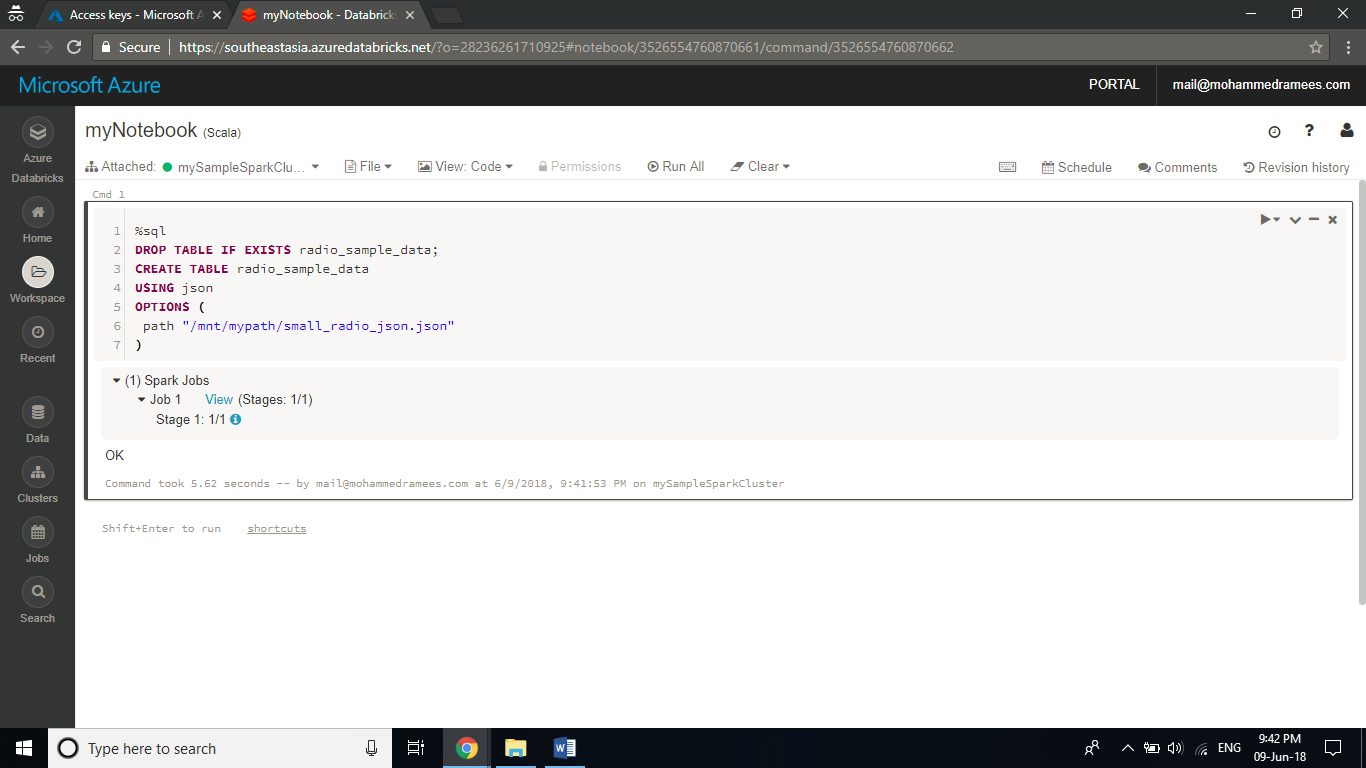
CREATE TABLE radio\_sample\_data

USING json

OPTIONS (

path "/mnt/mypath/small\_radio\_json.json"

)

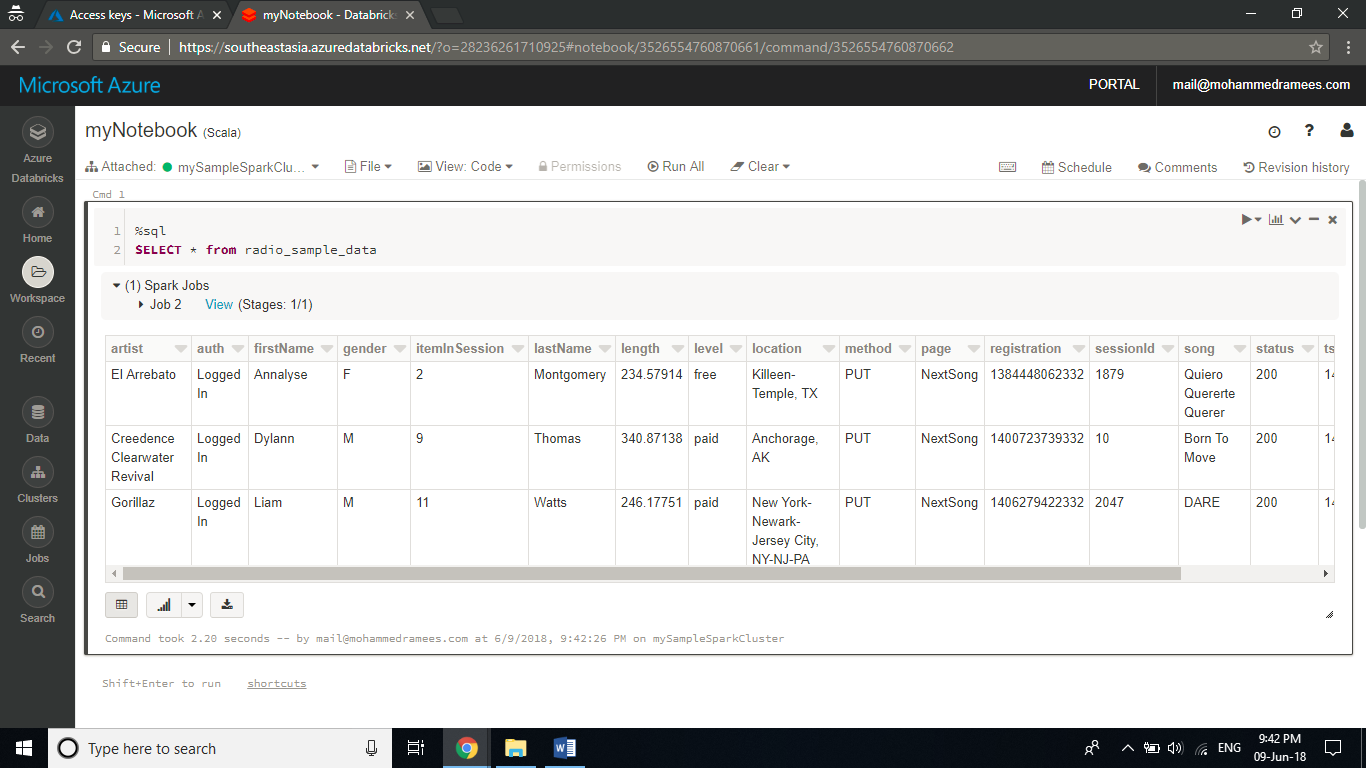


**Step 9:** Once the command successfully completes, you have all the data from the JSON file as a table in Databricks cluster. Let's look at a snapshot of the sample JSON data to better understand the query that you run. Paste the following snippet in the code cell and press SHIFT + ENTER.

%sql

SELECT \* from radio\_sample\_data

You see a tabular output like shown in the following screenshot (only some columns are shown):



**Step 10:** You now create a visual representation of this data to show for each gender, how many users have free accounts and how many are paid subscribers. From the bottom of the tabular output, click the **Bar chart icon**, and then click **Plot Options**. In Customize Plot, drag-and-drop values as shown in the screenshot.

Set Keys to gender.

Set Series groupings to level.

Set Values to level.

Set Aggregation to COUNT.

Click **Apply**.

