

**Practical 10**  
**Cloud Computing**  
2CSDE67

**Mistry Unnat**  
20BCE515

**Date**  
May 6, 2022



Department of Computer Science and Engineering  
Institute of Technology  
Nirma University  
Ahmedabad

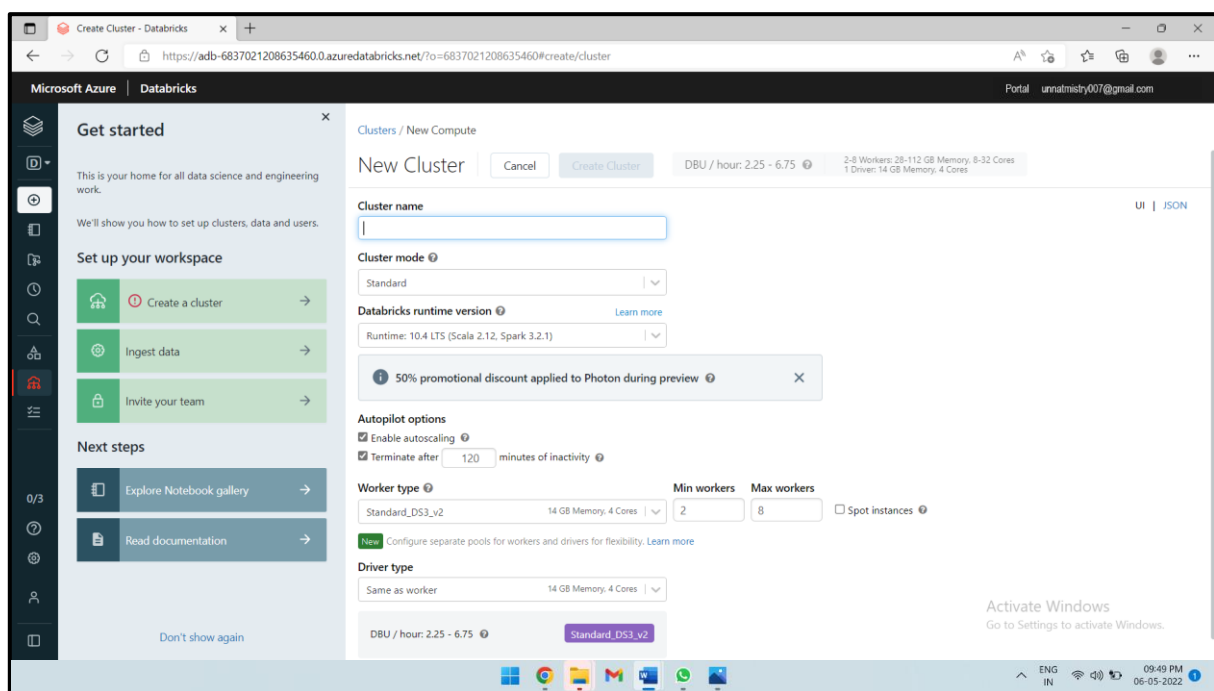
## Practical 10

**Aim:** Microsoft-AZURE[data science- AZURE webapp-Data bricks]

### Create an Azure Databricks workspace

In this section, we create an Azure Databricks workspace using the Azure portal or the Azure CLI.

1. In the Azure portal, select **Create a resource > Analytics > Azure Databricks**.
2. Under **Azure Databricks Service**, provide the values to create a Databricks workspace.



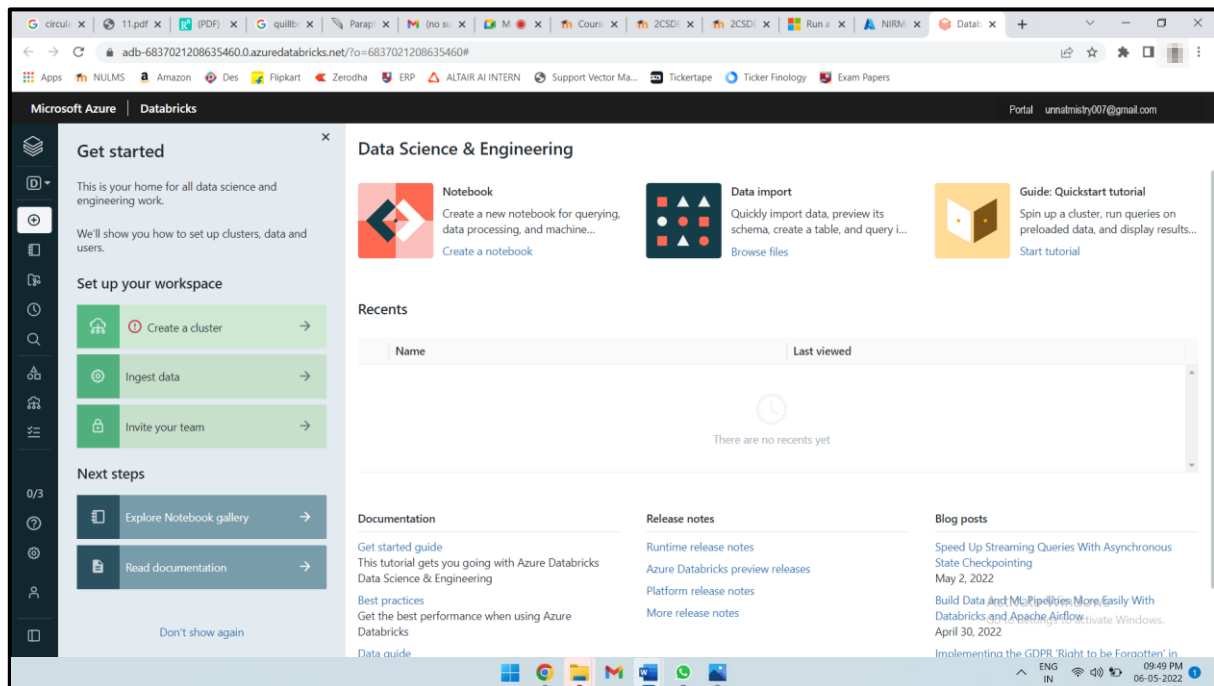
3. Select **Review + Create**, and then **Create**. The workspace creation takes a few minutes. During workspace creation, you can view the deployment status in **Notifications**. Once this process is finished, your user account is automatically added as an admin user in the workspace.

### Create a Spark cluster in Databricks

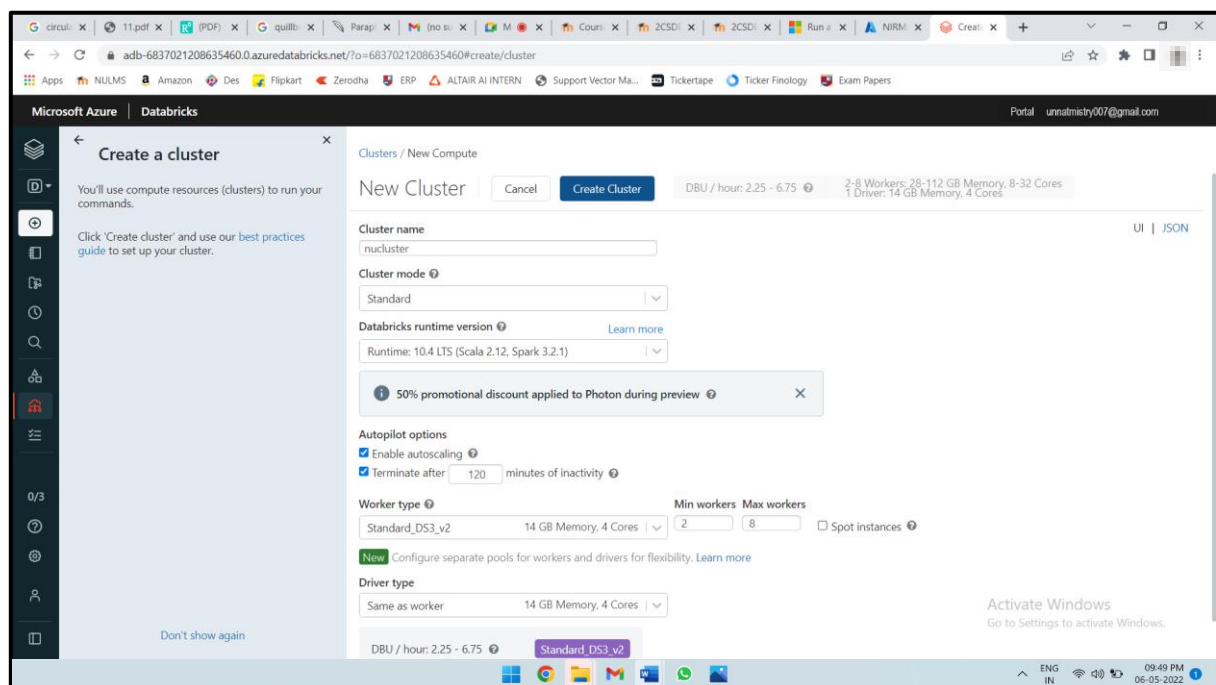
#### Note

To use a free account to create the Azure Databricks cluster, before creating the cluster, go to your profile and change your subscription to **pay-as-you-go**. For more information, see [Azure free account](#).

1. In the Azure portal, go to the Databricks workspace that you created, and then click **Launch Workspace**.
2. You are redirected to the Azure Databricks portal. From the portal, click **New Cluster**.



3. In the **New cluster** page, provide the values to create a cluster.

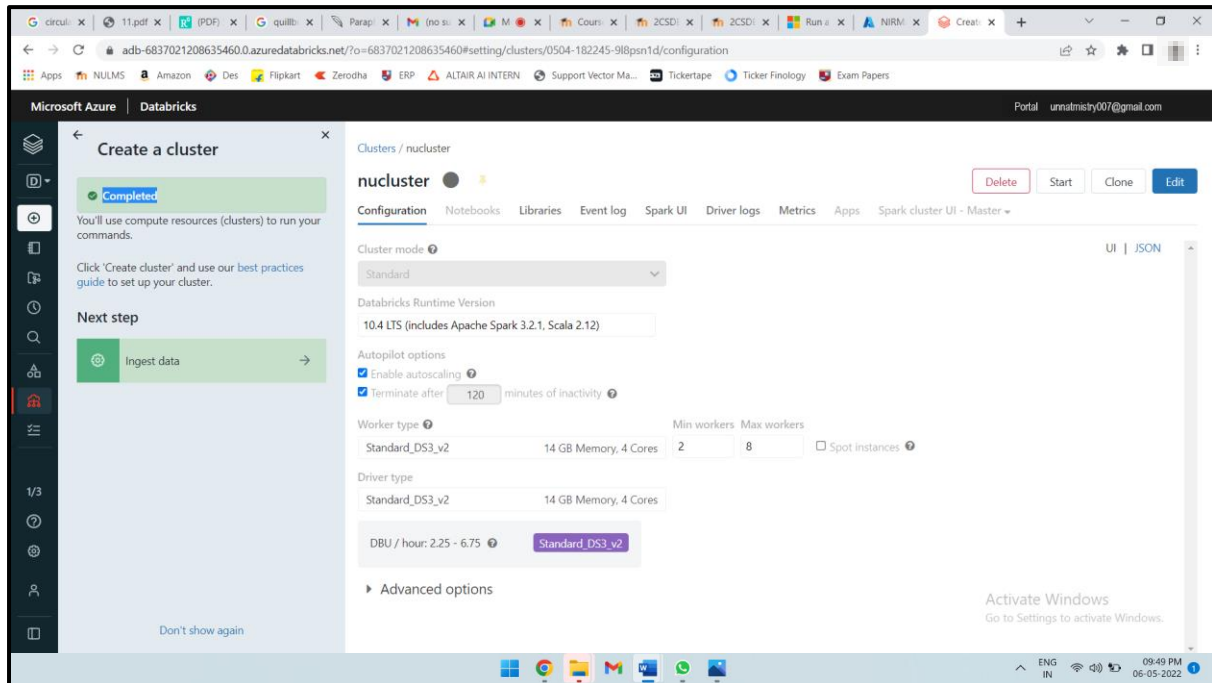


Accept all other default values other than the following:

- Enter a name for the cluster.
- For this article, create a cluster with **10.4 LTS** runtime.

- Make sure you select the **Terminate after \_\_ minutes of inactivity** checkbox. Provide a duration (in minutes) to terminate the cluster, if the cluster is not being used.

Select **Create cluster**. Once the cluster is running, you can attach notebooks to the cluster and run Spark jobs.

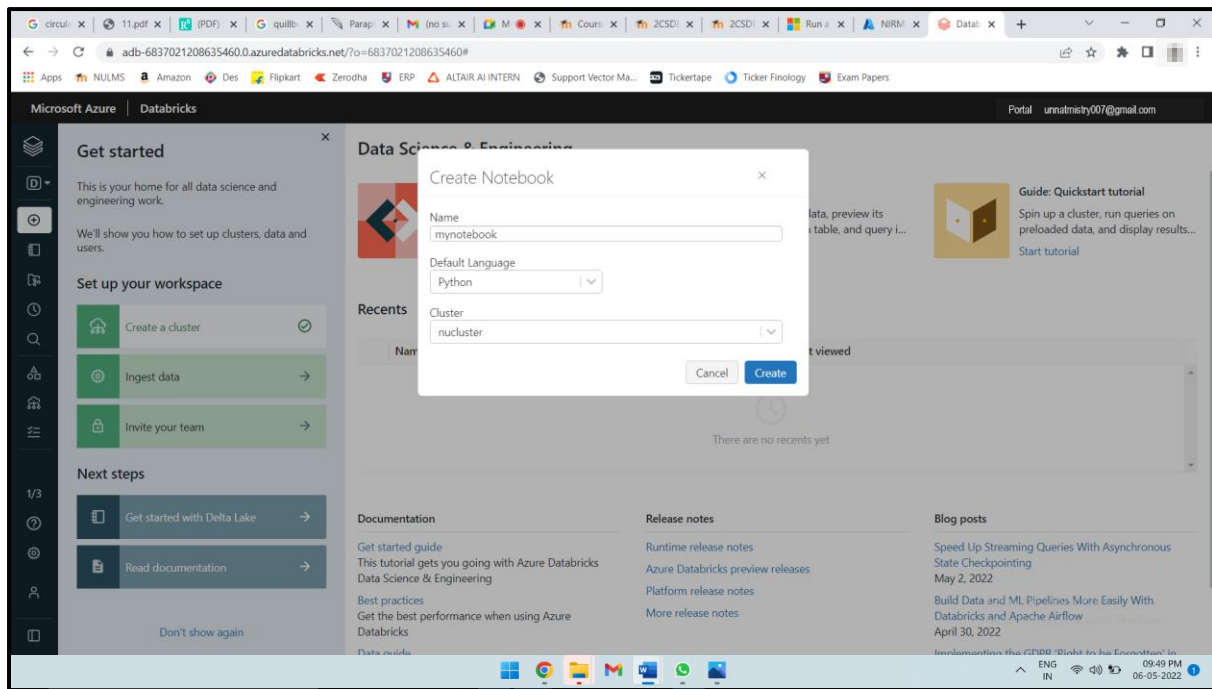


For more information on creating clusters, see [Create a Spark cluster in Azure Databricks](#).

## Run a Spark SQL job

Perform the following tasks to create a notebook in Databricks, configure the notebook to read data from an Azure Open Datasets, and then run a Spark SQL job on the data.

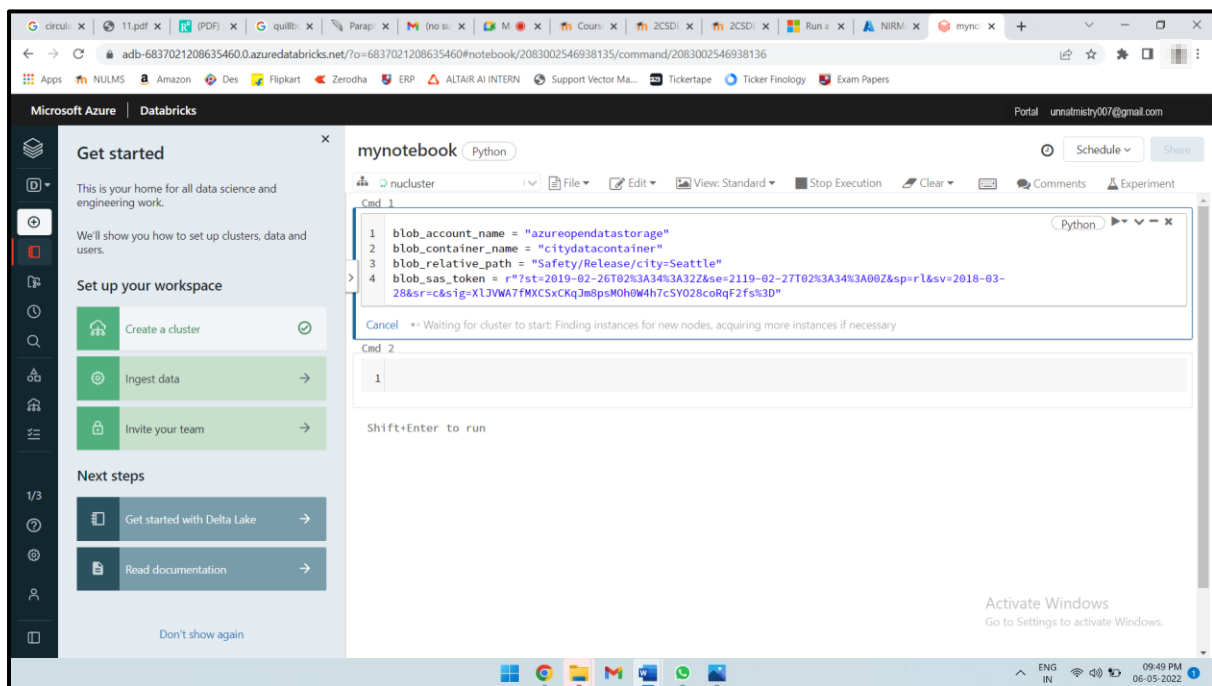
1. In the left pane, select **Azure Databricks**. From the **Common Tasks**, select **New Notebook**.



- In the **Create Notebook** dialog box, enter a name, select **Python** as the language, and select the Spark cluster that you created earlier.

Select **Create**.

- In this step, create a Spark DataFrame with Seattle Safety Data from [Azure Open Datasets](#), and use SQL to query the data.



The following command allows Spark to read from Blob storage remotely

```

1 blob_account_name = "azureopendatastorage"
2 blob_container_name = "citydatacontainer"
3 blob_relative_path = "Safety/Release/city=Seattle"
4 blob_sas_token = r"?st=2019-02-26T02%3A34%3A32Z&se=2119-02-27T02%3A34%3A00Z&sp=r&sv=2018-03-28&sr=c&sig=X1JYWA7FMKCSxCKqJm8psMOh0W4h7cSYO28coRqF2fs%3D"

1 wasbs_path = 'wasbs://%s@%s.blob.core.windows.net/%s' % (blob_container_name, blob_account_name, blob_relative_path)
2 spark.conf.set('fs.azure.sas.%s.%s.blob.core.windows.net' % (blob_container_name, blob_account_name), blob_sas_token)
3 print('Remote blob path: ' + wasbs_path)
4

```

The following command creates a DataFrame. Paste this PySpark code into the next cell and use **Shift+Enter** to run the code.

```

1 blob_account_name = "azureopendatastorage"
2 blob_container_name = "citydatacontainer"
3 blob_relative_path = "Safety/Release/city=Seattle"
4 blob_sas_token = r"?st=2019-02-26T02%3A34%3A32Z&se=2119-02-27T02%3A34%3A00Z&sp=r&sv=2018-03-28&sr=c&sig=X1JYWA7FMKCSxCKqJm8psMOh0W4h7cSYO28coRqF2fs%3D"

1 wasbs_path = 'wasbs://%s@%s.blob.core.windows.net/%s' % (blob_container_name, blob_account_name, blob_relative_path)
2 spark.conf.set('fs.azure.sas.%s.%s.blob.core.windows.net' % (blob_container_name, blob_account_name), blob_sas_token)
3 print('Remote blob path: ' + wasbs_path)
4

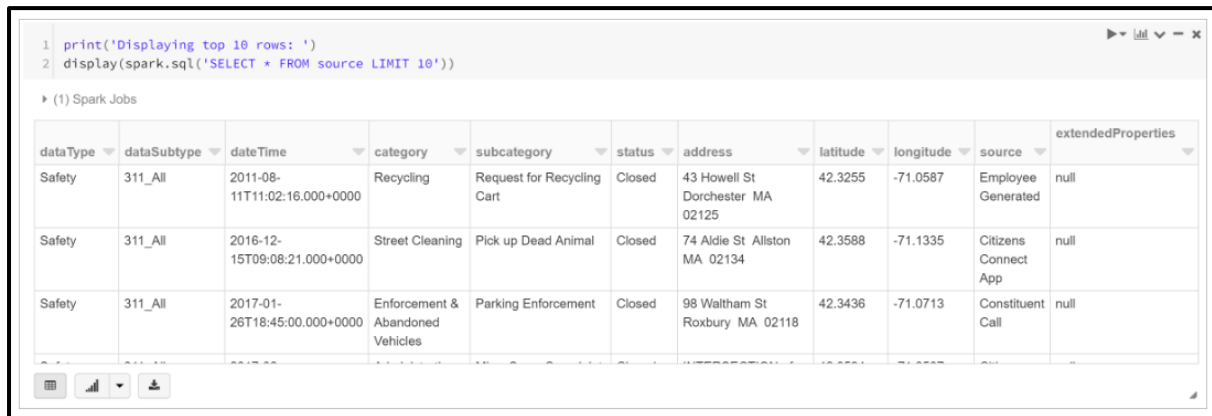
1 df = spark.read.parquet(wasbs_path)
2 print('Register the DataFrame as a SQL temporary view: source')
3 df.createOrReplaceTempView('source')
4

```

4. Run a SQL statement return the top 10 rows of data from the temporary view called **source**. Paste this PySpark code into the next cell and use **Shift+Enter** to run the code.

```
print('Displaying top 10 rows: ')
display(spark.sql('SELECT * FROM source LIMIT 10'))
```

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

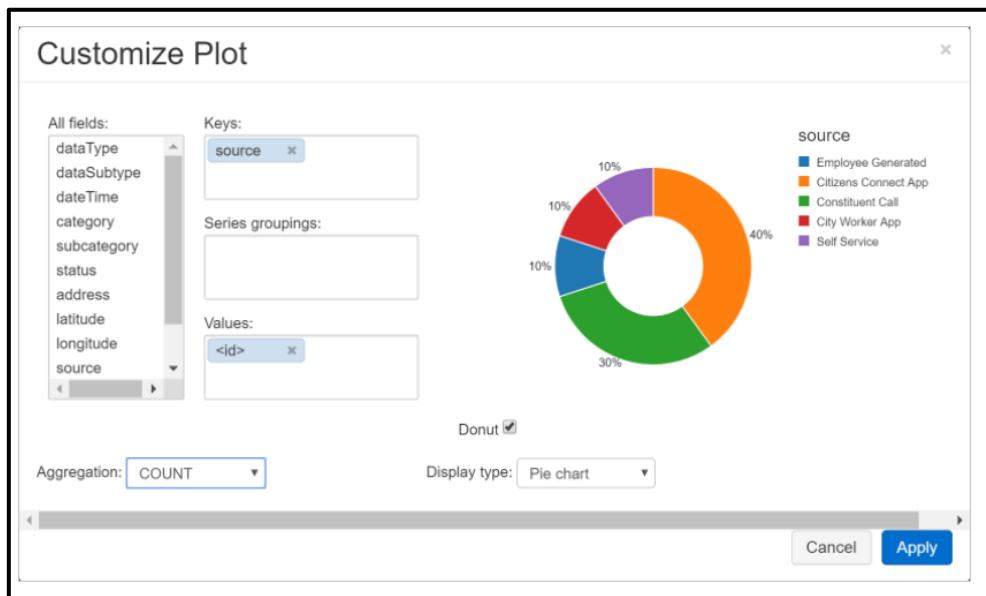


dataType	dataSubtype	dateTime	category	subcategory	status	address	latitude	longitude	source	extendedProperties
Safety	311_All	2011-08-11T11:02:16.000+0000	Recycling	Request for Recycling Cart	Closed	43 Howell St Dorchester MA 02125	42.3255	-71.0587	Employee Generated	null
Safety	311_All	2016-12-15T09:08:21.000+0000	Street Cleaning	Pick up Dead Animal	Closed	74 Aldie St Allston MA 02134	42.3588	-71.1335	Citizens Connect App	null
Safety	311_All	2017-01-26T18:45:00.000+0000	Enforcement & Abandoned Vehicles	Parking Enforcement	Closed	98 Waltham St Roxbury MA 02118	42.3436	-71.0713	Constituent Call	null

6. You now create a visual representation of this data to show how many safety events are reported using the Citizens Connect App and City Worker App instead of other sources. From the bottom of the tabular output, select the **Bar chart** icon, and then click **Plot Options**.



7. In **Customize Plot**, drag-and-drop values as shown in the screenshot.

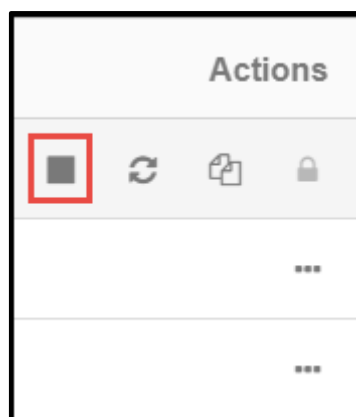


- Set **Keys** to **source**.
- Set **Values** to **<\id>**.
- Set **Aggregation** to **COUNT**.
- Set **Display type** to **Pie chart**.

Click **Apply**.

### Clean up resources

After you have finished the article, you can terminate the cluster. To do so, from the Azure Databricks workspace, from the left pane, select **Clusters**. For the cluster you want to terminate, move the cursor over the ellipsis under **Actions** column, and select the **Terminate** icon.





If you do not manually terminate the cluster it will automatically stop, provided you selected the **Terminate after \_\_ minutes of inactivity** checkbox while creating the cluster. In such a case, the cluster automatically stops, if it has been inactive for the specified time.

---

END

---