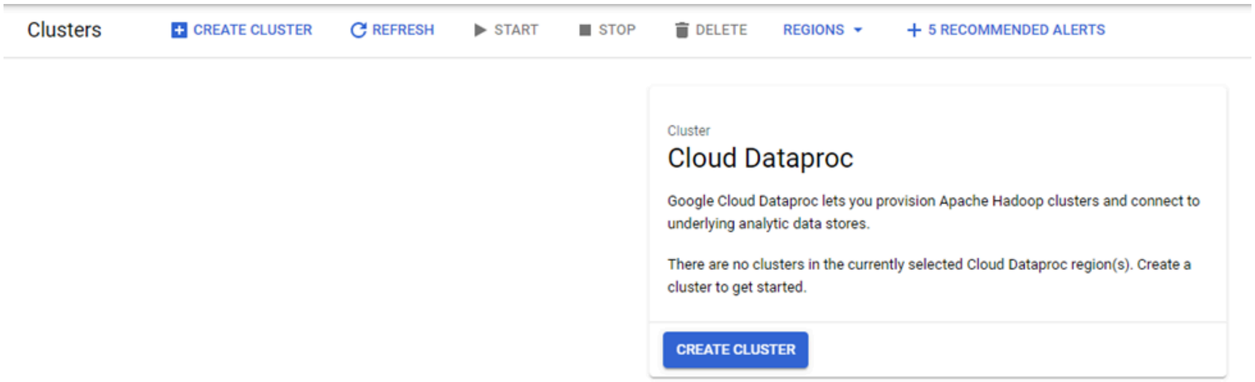
Learning Spark with Google Dataproc and Jupyter Notebook

In this instruction, we will show you how to install and run the Jupyter Notebook with Google Dataproc. Google Dataproc is the managed Hadoop and Spark framework for creating either single-node or autoscaling clusters by the Google Cloud Platform.

Create a Dataproc Cluster with Jupyter Notebook

Start at the Google Dataproc console to create a new cluster: [console]()



Cluster

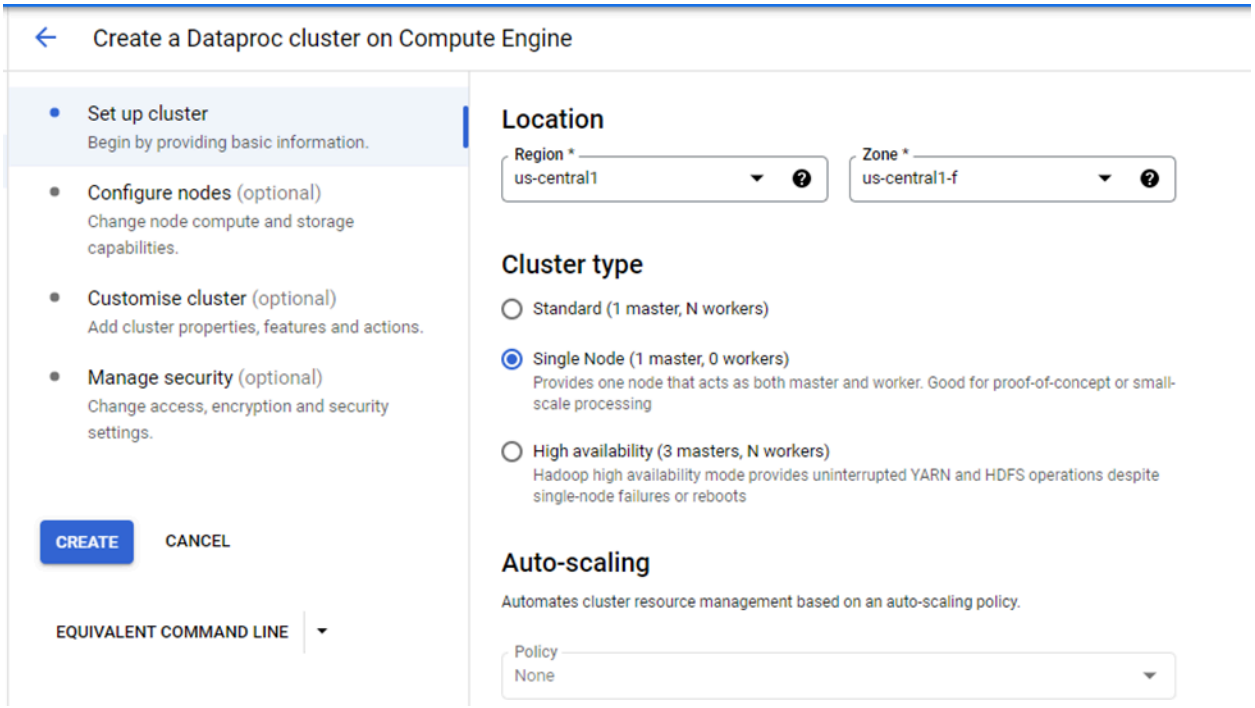
Cloud Dataproc

Google Cloud Dataproc lets you provision Apache Hadoop clusters and connect to underlying analytic data stores.

There are no clusters in the currently selected Cloud Dataproc region(s). Create a cluster to get started.

Start Dataproc cluster creation

Click "Create Cluster" and select the cluster options like Cluster Type, Name of Cluster, Location, Auto-Scaling Options, and more.



← Create a Dataproc cluster on Compute Engine
Set up cluster
Configure nodes (optional)
Customise cluster (optional)
Manage security (optional)
CREATE
CANCEL
EQUIVALENT COMMAND LINE

- Set up cluster: Begin by providing basic information.
- Configure nodes (optional): Change node compute and storage capabilities.
- Customise cluster (optional): Add cluster properties, features and actions.
- Manage security (optional): Change access, encryption and security settings.

Location

Region *: us-central1

Zone *: us-central1-f

Cluster type

○ Standard (1 master, N workers)

● Single Node (1 master, 0 workers)
Provides one node that acts as both master and worker. Good for proof-of-concept or small-scale processing

○ High availability (3 masters, N workers)
Hadoop high availability mode provides uninterrupted YARN and HDFS operations despite single-node failures or reboots

Auto-scaling

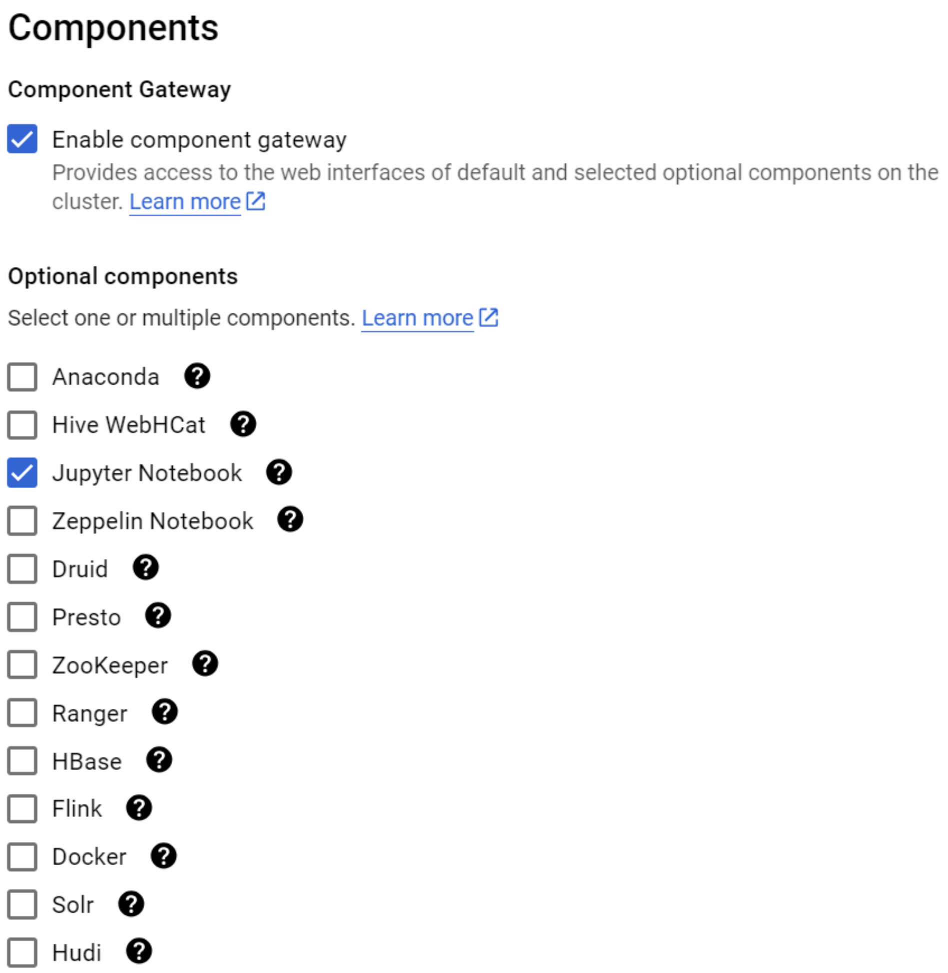
Automates cluster resource management based on an auto-scaling policy.

Policy: None

Parameters required for Cluster

Select the **Single Node Cluster** option, this means that auto-scaling is disabled as the cluster consists of only 1 master node.

Next, in the “Components” section, be sure to select “Enable Component Gateway” and “Jupyter Notebook”. This will enable the web interface of Jupyter Notebook to be available.



Components

Component Gateway

☑ Enable component gateway
Provides access to the web interfaces of default and selected optional components on the cluster. Learn more

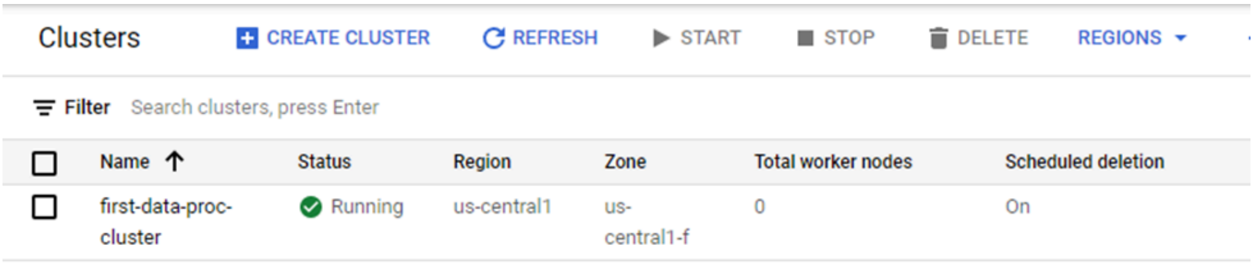
Optional components

Select one or multiple components. Learn more

- □ Anaconda
- □ Hive WebHCat
- ☑ Jupyter Notebook
- □ Zeppelin Notebook
- □ Druid
- □ Presto
- □ ZooKeeper
- □ Ranger
- □ HBase
- □ Flink
- □ Docker
- □ Solr
- □ Hudi

When you click "Create", it'll start creating the cluster.

After a few minutes the cluster with 1 master node will be ready for use.



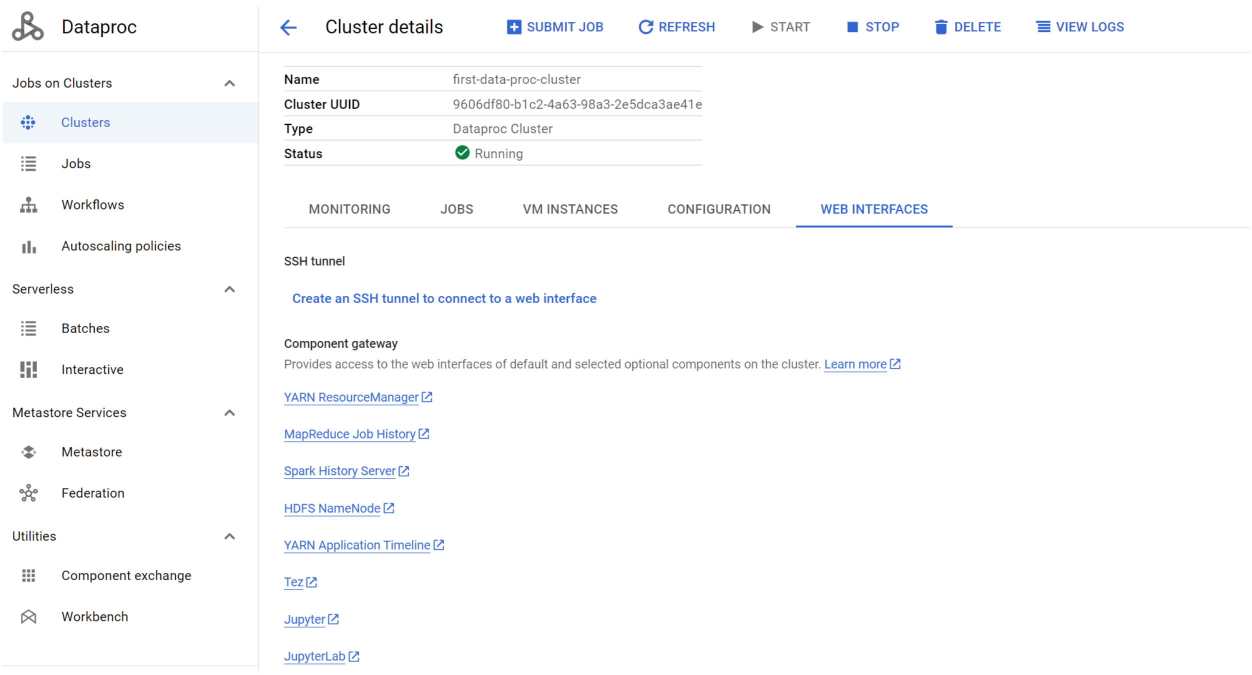
□	Name ↑	Status	Region	Zone	Total worker nodes	Scheduled deletion
□	first-data-proc-cluster	✓ Running	us-central1	us-central1-f	0	On

Cluster Up and Running

Open the JupyterLab on Master Node

Since we have created a single-node Dataproc cluster, we can simply launch a Jupyter Notebook on the Master Node to access and control the Spark cluster. We will be using JupyterLab, which is more advanced web interface of Jupyter Notebook.

To access the pre-installed JupyterLab environment, click the name of the Dataproc cluster you just created and switch to the tab of “Web interfaces”:



Name	first-data-proc-cluster
Cluster UUID	9606df80-b1c2-4a63-98a3-2e5dca3ae41e
Type	Dataproc Cluster
Status	✓ Running

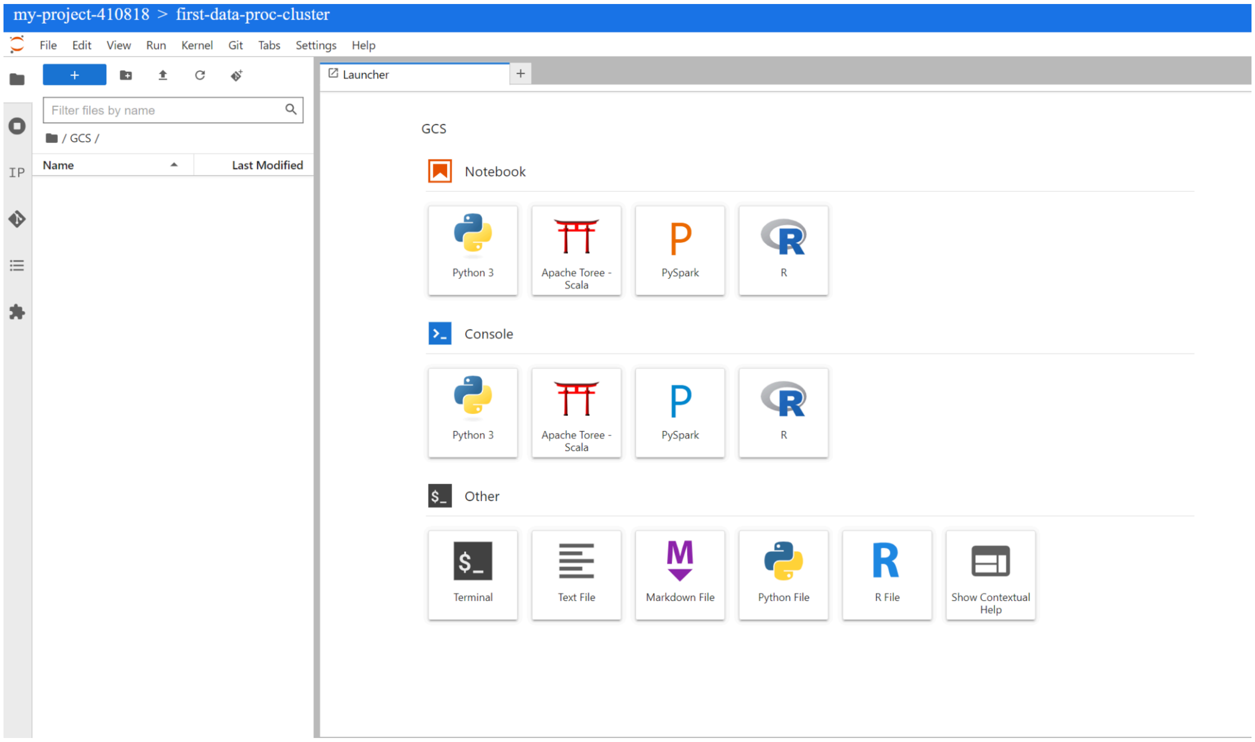
SSH tunnel

Create an SSH tunnel to connect to a web interface

Component gateway

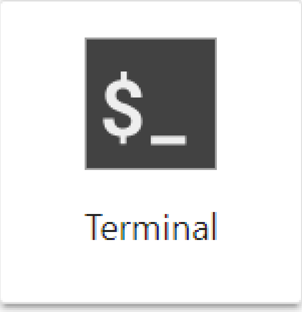
Provides access to the web interfaces of default and selected optional components on the cluster. Learn more

Click the link of “JupyterLab” to access the web interface.



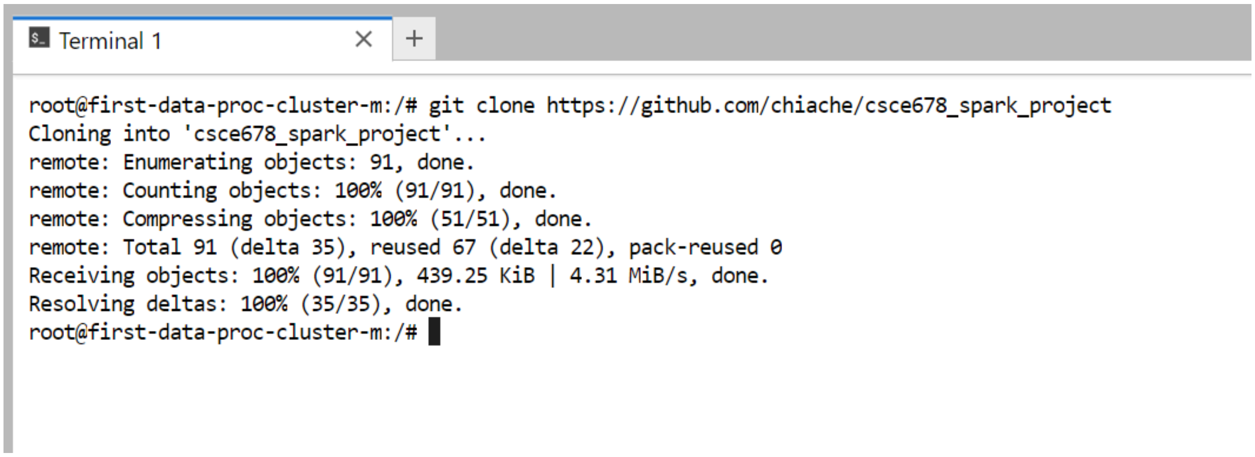
Check Out the Spark Exercises and Access in Jupyter Notebook

Next, click the Terminal button in the main window to start a Terminal in the browser:



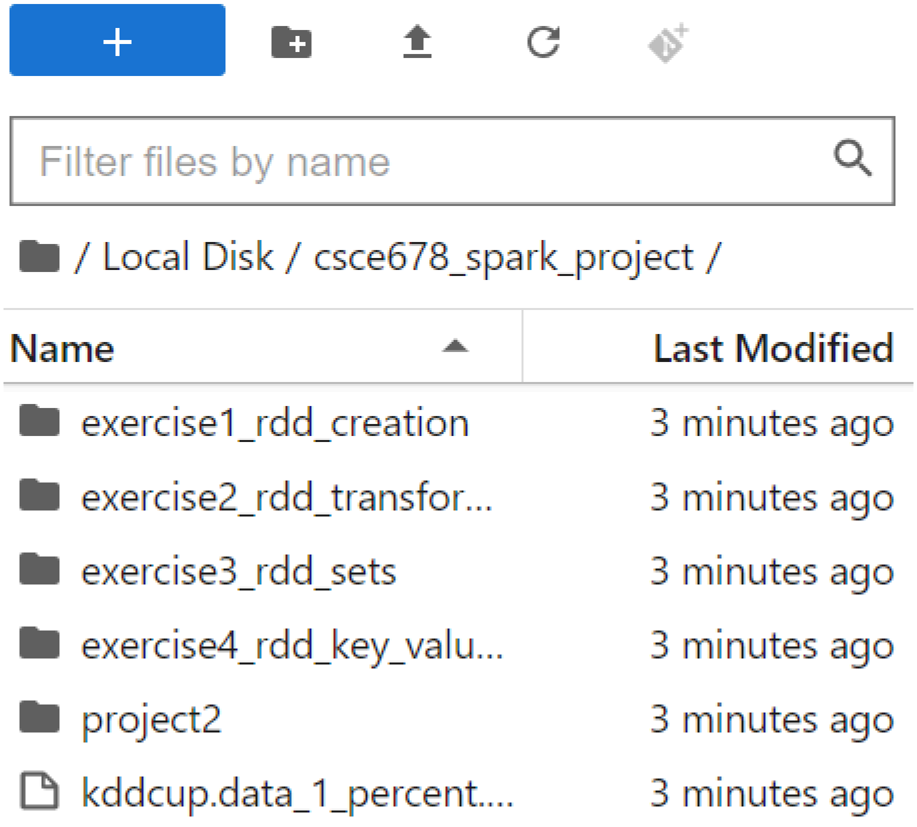
In the Terminal window, enter the following command:

```
git clone https://github.com/chiache/csce678_spark_project
```



```
root@first-data-proc-cluster-m:/# git clone https://github.com/chiache/csce678_spark_project
Cloning into 'csce678_spark_project'...
remote: Enumerating objects: 91, done.
remote: Counting objects: 100% (91/91), done.
remote: Compressing objects: 100% (51/51), done.
remote: Total 91 (delta 35), reused 67 (delta 22), pack-reused 0
Receiving objects: 100% (91/91), 439.25 KiB | 4.31 MiB/s, done.
Resolving deltas: 100% (35/35), done.
root@first-data-proc-cluster-m:/#
```

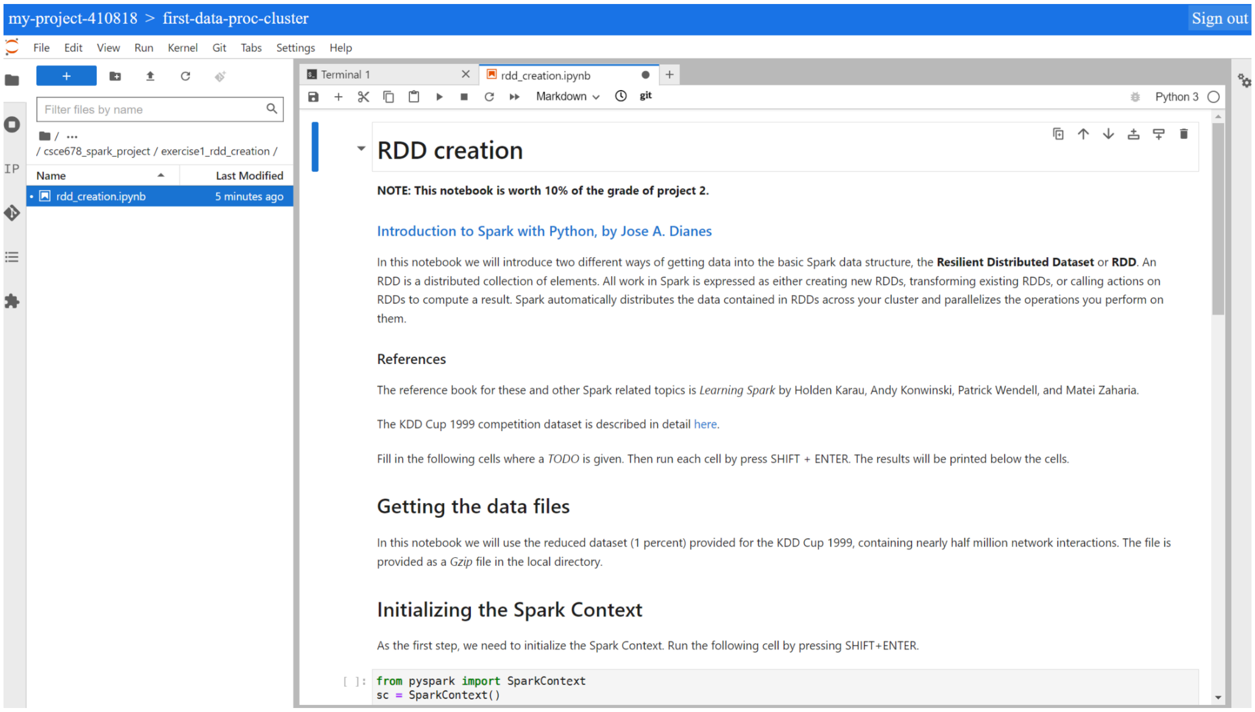
Use the file browser in the left panel and naviate to / Local Disk / csce678_spark_project:



/ Local Disk / csce678_spark_project /

Name	Last Modified
exercise1_rdd_creation	3 minutes ago
exercise2_rdd_transfor...	3 minutes ago
exercise3_rdd_sets	3 minutes ago
exercise4_rdd_key_valu...	3 minutes ago
project2	3 minutes ago
kddcup.data_1_percent....	3 minutes ago

Enter the first folder (exercise1_rdd_creation) and open the Jupyter Notebook file named **exercise1_rdd_creation.ipynb**.


my-project-410818 > first-data-proc-cluster
Sign out
File Edit View Run Kernel Git Tabs Settings Help
Filter files by name
/ csce678_spark_project / exercise1_rdd_creation /
Name
Last Modified
rdd_creation.ipynb
5 minutes ago
Terminal 1
rdd_creation.ipynb
Markdown
git
Python 3

RDD creation

NOTE: This notebook is worth 10% of the grade of project 2.

Introduction to Spark with Python, by Jose A. Dianes

In this notebook we will introduce two different ways of getting data into the basic Spark data structure, the **Resilient Distributed Dataset** or **RDD**. An RDD is a distributed collection of elements. All work in Spark is expressed as either creating new RDDs, transforming existing RDDs, or calling actions on RDDs to compute a result. Spark automatically distributes the data contained in RDDs across your cluster and parallelizes the operations you perform on them.

References

The reference book for these and other Spark related topics is *Learning Spark* by Holden Karau, Andy Konwinski, Patrick Wendell, and Matei Zaharia.

The KDD Cup 1999 competition dataset is described in detail here.

Fill in the following cells where a *TODO* is given. Then run each cell by press SHIFT + ENTER. The results will be printed below the cells.

Getting the data files

In this notebook we will use the reduced dataset (1 percent) provided for the KDD Cup 1999, containing nearly half million network interactions. The file is provided as a *Gzip* file in the local directory.

Initializing the Spark Context

As the first step, we need to initialize the Spark Context. Run the following cell by pressing SHIFT+ENTER.

```
[ ]: from pyspark import SparkContext
     sc = SparkContext()
```

Follow the instructions in the Jupyter Notebook to learn how to program with PySpark. You can use “SHIFT+ENTER” to execute the cell and advance to the next cell. Be sure to finish the code in the cells that contain “# TODO”.

In this exercise, you only need to finish the jupyter notebooks in the first three folders in the repository (**exercise1_rdd_creation**, **exercise2_rdd_transformation**, and **exercise3_rdd_sets**). The rest of the exercises are optional.

Shut Down the Dataproc Cluster (if necessary)

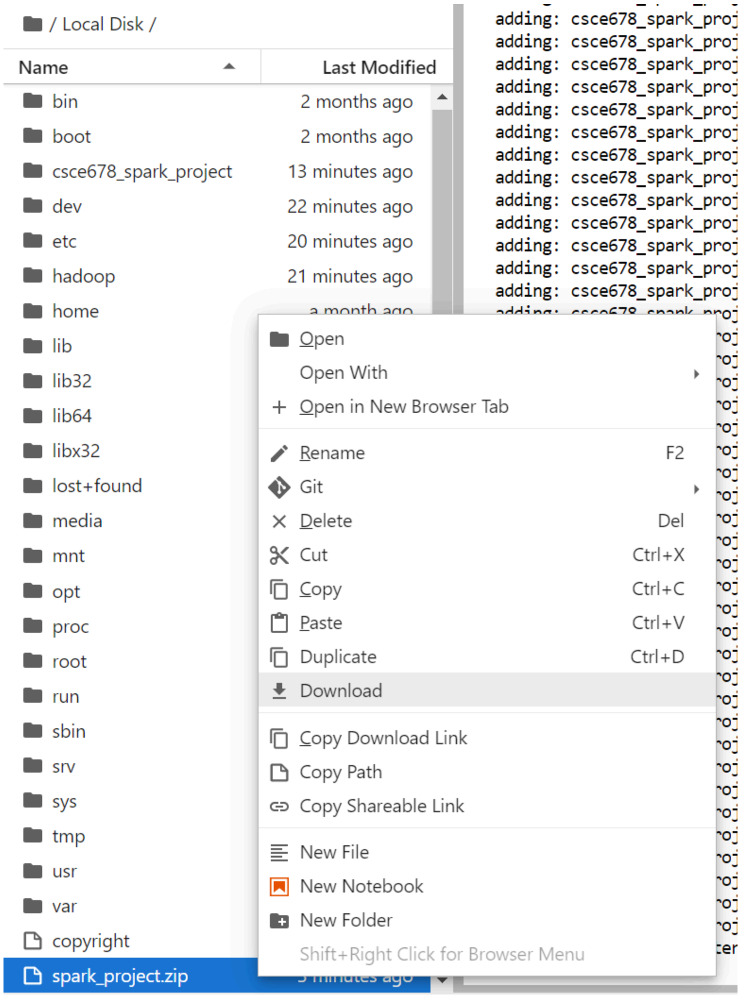
When you are not working on the Jupyter Notebooks, it is recommended to shut down (NOT deleting) the cluster to prevent unnecessary charges. Open the Dataproc Console and select the cluster you created for Spark. And then click ■ STOP to stop the cluster.

Download and Submit the Jupyter Notebooks

Once you have completed the Jupyter Notebook, switch back to the terminal and run the following command:

```
zip -r spark_project.zip csce678_spark_project/
```

In the file browser, switch to / Local Disk / and find “spark_project.zip”. Right click the file to choose “Download”.



/ Local Disk /

Name	Last Modified
bin	2 months ago
boot	2 months ago
csce678_spark_project	13 minutes ago
dev	22 minutes ago
etc	20 minutes ago
hadoop	21 minutes ago
home	a month ago
lib	
lib32	
lib64	
libx32	
lost+found	
media	
mnt	
opt	
proc	
root	
run	
sbin	
srv	
sys	
tmp	
usr	
var	
copyright	
spark_project.zip	5 minutes ago

```
adding: csce678_spark_proj
adding: csce678_spark_proj
adding: csce678_spark_proj
adding: csce678_spark_proj
adding: csce678_spark_proj
adding: csce678_spark_proj
adding: csce678_spark_proj
adding: csce678_spark_proj
adding: csce678_spark_proj
adding: csce678_spark_proj
adding: csce678_spark_proj
adding: csce678_spark_proj
adding: csce678_spark_proj
adding: csce678_spark_proj
```

Once You have downloaded spark_project.zip, head over to Canvas and upload the file to the assignment “Exercise Spark Jupyter Notebooks (Submission)”.

Delete the Dataproc Cluster and Google Cloud Storage

Finally, you can delete the dataproc cluster and the bucket you created in the Google Cloud Storage.