As we work to finalize work on our Course Progress system, some variations may be seen. Please know that your progress is being tracked properly, even if it is not shown properly on the display. If you have an immediate need, please contact Support so we can investigate your account individually. Thank you for your continued patience!

Running a Pyspark Job on Cloud Dataproc Using Google Cloud Storage

116 Min. Remaining Intermediate

Cancel Lab Complete Lab

How was this lab?

Credentials

How do I connect?

Google Labs Account

Username

cloud_user_p_f1cd16@linuxacademygclabs.com

Password

igbtUprP

Open Google Console

Additional Information and Resources

In this lab, you will create a single node Dataproc cluster and a GCS bucket for your Pyspark job output. Separating the storage from the compute allows you to treat your cluster as ephemeral, and we will delete the cluster when we are done while preserving the results.

Launch your lab in incognito mode (or another browser private browsing mode) to avoid issues with cached logins.

For detailed instructions on how to complete these tasks, expand each learning objective below, or click the ***Guide*** tab above the video player.

Learning Objectives

Prepare Our Environment

1. First, we need to enable the Dataproc API:

```
gcloud services enable
dataproc.googleapis.com
```

2. Then create a Cloud Storage bucket:

```
gsutil mb -l us-central1
gs://$DEVSHELL_PROJECT_ID-data
```

Tools

Instant Terminal Diagram

Video Guide

```
gcloud dataproc jobs submit pyspark
wordcount.py --cluster=wordcount -- \
gs://la-gcp-labs-resources/data-
engineer/dataproc/romeoandjuliet.txt \
gs://$DEVSHELL_PROJECT_ID-data/output/
```

13. View the progress by going back to the ***Dataproc*** page's ***Cluster Details*** section, and click **Jobs** on the top-left menu to access the job.

    Note: This job may take approximately 30-45 seconds to complete, and we should see a confirmation message in the ***Job Details*** section when clicking on the job.

14. Navigate to the top-left menu, and then click **Storage**.

15. Click on the **data location** bucket.

    Note: Do ***not*** click on the **staging bucket** that has `dataproc` its name.

16. Click the `output\` folder.

Review the Pyspark Output

1. In Cloud Shell, download output files from the GCS output location:

```
gsutil cp -r gs://$DEVSHELL_PROJECT_ID-
data/output/* .
```

Note: Alternatively, we could download them to our local machine via the web console.

2. We can view the contents again, with the `ls` command.

3. Use the following to see an output file:

3. Now create the `dataproc` cluster:

4. ```
   gcloud dataproc clusters create wordcount --zone=us-central1-f --single-node --master-machine-type=n1-standard-2
   ```

5. And finally, download the `wordcount.py` file that will be used for the `pyspark` job:

   ```
   gsutil cp -r gs://la-gcp-labs-resources/data-engineer/dataproc/* .
   ```

Submit the Pyspark Job to the Dataproc Cluster

In Cloud Shell, type:

```
gcloud dataproc jobs submit pyspark wordcount.py --cluster=wordcount -- \
gs://la-gcp-labs-resources/data-engineer/dataproc/romeoandjuliet.txt \
gs://$DEVSHELL_PROJECT_ID-data/output/
```

Review the Pyspark Output

1. In Cloud Shell, download output files from the GCS output location:

   ```
   gsutil cp -r gs://$DEVSHELL_PROJECT_ID-data/output/* .
   ```

   Note: Alternatively, we could download them to our local machine via the web console.

Delete the Dataproc Cluster

1. We don't need our cluster any longer, so let's delete it. In the web console, go to the top-left menu and into **BIGDATA** > **Dataproc**.
2. Select the **wordcount** cluster, then click **DELETE**, and **OK** to confirm.

   Our job output still remains in Cloud Storage, allowing us to delete Dataproc clusters when no longer in use to save costs, while preserving input and output resources.
   ```