
Dataproc: Qwik Start - Console

Overview

This lab demonstrates how to use **Google Cloud Dataproc**, a managed **Apache Spark** and **Hadoop** service, to create a cluster, run a Spark job, and modify the cluster size. The key objectives were:

1. **Create a Dataproc cluster** in the Google Cloud Console.
2. **Run a simple Apache Spark job** to estimate the value of Pi.
3. **Modify the number of workers** in the cluster.

Task 1: Create a Cluster

Steps:

1. Navigated to **Dataproc > Clusters** in the Google Cloud Console.
2. Clicked **Create Cluster** with the following configuration:
 - a. **Name:** example-cluster
 - b. **Region & Zone:** Default selected
 - c. **Machine Series:** E2
 - d. **Machine Type (Master & Workers):** e2-standard-2
 - e. **Primary Disk Size:** 30 GB (for both Master and Workers)
 - f. **Worker Nodes:** 2
 - g. **Internal IP Only:** Disabled (to allow external access)
3. Clicked **Create** and waited for the cluster status to change from **Provisioning** to **Running**.

Create a Dataproc cluster on Compute Engine

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

CREATE CANCEL

EQUIVALENT COMMAND LINE

EQUIVALENT REST

Name

Cluster Name *
example-cluster

Location

Region *
us-west1

Zone *
us-west1-a

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

Versioning

Create a Dataproc cluster on Compute Engine

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

CREATE CANCEL

EQUIVALENT COMMAND LINE

EQUIVALENT REST

☒ General purpose ☐ Compute optimized ☐ Memory optimized ☐ GPUs

Machine types for common workloads, optimized for cost and flexibility

Series
E2

CPU platform selection based on availability

Machine type
e2-standard-2 (2 vCPU, 1 core, 8 GB memory)

	vCPU	Memory
	2	8 GB

CPU PLATFORM AND GPU

Primary disk size *
30 GB

Primary disk type *
Standard Persistent Disk

Number of local SSDs *
x 375GB

Local SSD Interface

Dataproc / Clusters

Overview

Notebooks/IDE

BigQuery Studio

Workbench

Clusters

Jobs

Workflows

Autoscaling policies

Serverless

Batches

Interactive Sessions

Clusters

CREATE CLUSTER

REFRESH

START

STOP

DELETE

REGIONS

Filter

Search cluster by properties, press Enter

Sorry, the server was not able to fulfill your request.

	Name ↑	Status	Region	Zone	Total worker nodes	Flexible VMs
<input type="checkbox"/>	example-cluster	Running	us-west1	us-west1-a	2	No

No clusters

PERMISSION

PI

Task 2: Submit a Job

Steps:

1. Navigated to **Dataproc > Jobs**.
2. Clicked **Submit Job** with the following settings:
 - a. **Cluster:** example-cluster
 - b. **Job Type:** Spark
 - c. **Main Class:** org.apache.spark.examples.SparkPi
 - d. **Jar File:** <file:///usr/lib/spark/examples/jars/spark-examples.jar>
 - e. **Arguments:** 1000 (number of tasks for Pi estimation)
3. Clicked **Submit**.

Dataprocc / Jobs / Add job

Overview
Notebooks/IDE
BigQuery Studio
Workbench
Clusters
Clusters
Jobs
Workflows
Autoscaling policies
Serverless
Batches
Interactive Sessions
Release Notes

Submit a job

Job ID *
job-b8801731

Region *
us-west1
Specifies the Cloud Dataprocc regional service, which determines what clusters are available.

Cluster *
example-cluster

Job type *
Spark

Main class or jar *
org.apache.spark.examples.SparkPi
The fully qualified name of a class in a provided or standard jar file, for example, com.example.wordcount, or a provided jar file to use the main class of that jar file

Jar files
file:///usr/lib/spark/examples/jars/spark-examples.jar Enter file path, for exam
Jar files are included in the CLASSPATH. Can be a GCS file with the gs:// prefix, an HDFS file on the cluster with the hdfs:// prefix, or a local file on the cluster with the file:// prefix.

Dataprocc / Jobs / Job: job-b8801731 / Monitoring

Overview
Notebooks/IDE
BigQuery Studio
Workbench
Clusters
Clusters
Jobs
Workflows
Autoscaling policies
Serverless
Batches
Interactive Sessions
Release Notes

Job details

CLONE DELETE STOP REFRESH

Job ID	job-b8801731
Job UUID	4952688f-e627-40e9-8ea0-03699246abf7
Type	Dataprocc Job
Status	Running

MONITORING CONFIGURATION

Output LINE WRAP: OFF

Spark jobs take ~60 seconds to initialize resources. DISMISS

```

Press Alt+F1 for Accessibility Options.
25/06/16 04:21:01 INFO DefaultHARFaiIoverProxyProvider: Connecting to ResourceManager at example-cluster-m.us-west1-a.c.qwiklabs-gcp-01-e94elec7c41c.intern
25/06/16 04:21:01 INFO AHSPProxy: Connecting to Application History server at example-cluster-m.us-west1-a.c.qwiklabs-gcp-01-e94elec7c41c.internal./10.138.0.3:
25/06/16 04:21:02 INFO Configuration: resource-types.xml not found
25/06/16 04:21:02 INFO ResourceUtils: Unable to find 'resource-types.xml'.

```

EQUIVALENT COMMAND LINE

Job job-b8801731 successfully submitted

Task 3. View the job output

To see your completed job's output:

1. Click the job ID in the **Jobs** list.
2. Select **LINE WRAP** to ON or scroll all the way to the right to see the calculated value of Pi. Your output, with **LINE WRAP** ON, should look something like this:

Editing cluster



Worker nodes *

4

Secondary worker nodes *

0

Labels

Key 1

goog-dataproc-cluster-name

Value 1

example-cluster

Key 2

goog-dataproc-cluster-uuid

Value 2

cfbb503e-36b0-44bb-bf96-c6a8

Key 3

goog-dataproc-location

Value 3

us-west1

Key 4

goog-dataproc-drz-resource-uid

Value 4

cluster-cfbb503e-36b0-44bb-bf

+ ADD LABEL

☐ Use graceful decommissioning ?

SAVE

CANCEL

EQUIVALENT REST

1. To rerun the job with the updated cluster, you would click **Jobs** in the left pane, then click **SUBMIT JOB**.
2. Set the same fields you set in the **Submit a job** section:

Field	Value
Region	_____

Cluster	example-cluster
Job type	Spark
Main class or jar	org.apache.spark.examples.SparkPi
Jar files	file:///usr/lib/spark/examples/jars/spark-examples.jar
Arguments	1000 (This sets the number of tasks.)

3. Click **Submit**.

Conclusion

- Successfully **created a Dataproc cluster** and ran a **Spark job** to estimate Pi.
- **Modified the cluster size** from 2 to 4 workers, demonstrating **scalability**.
- The lab reinforced key concepts of **managed Spark/Hadoop clusters** in Google Cloud.