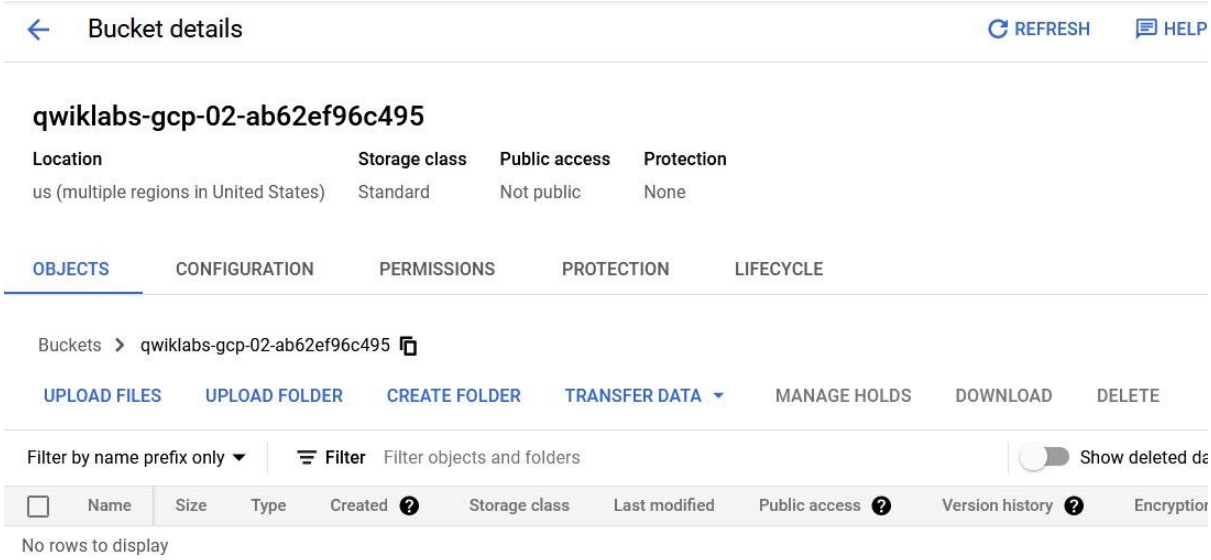
PDE Prep—Cloud Dataproc Cluster Operations and Maintenance

Activate Google Cloud Shell

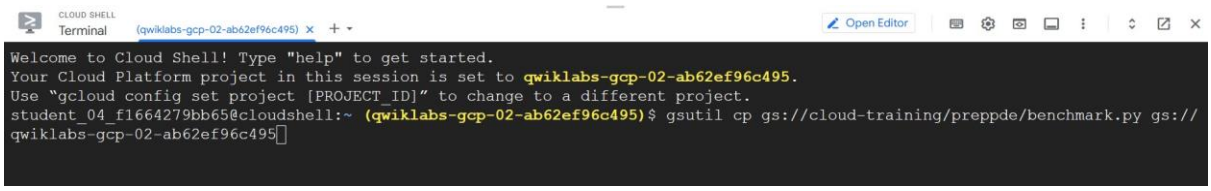
Check project permissions

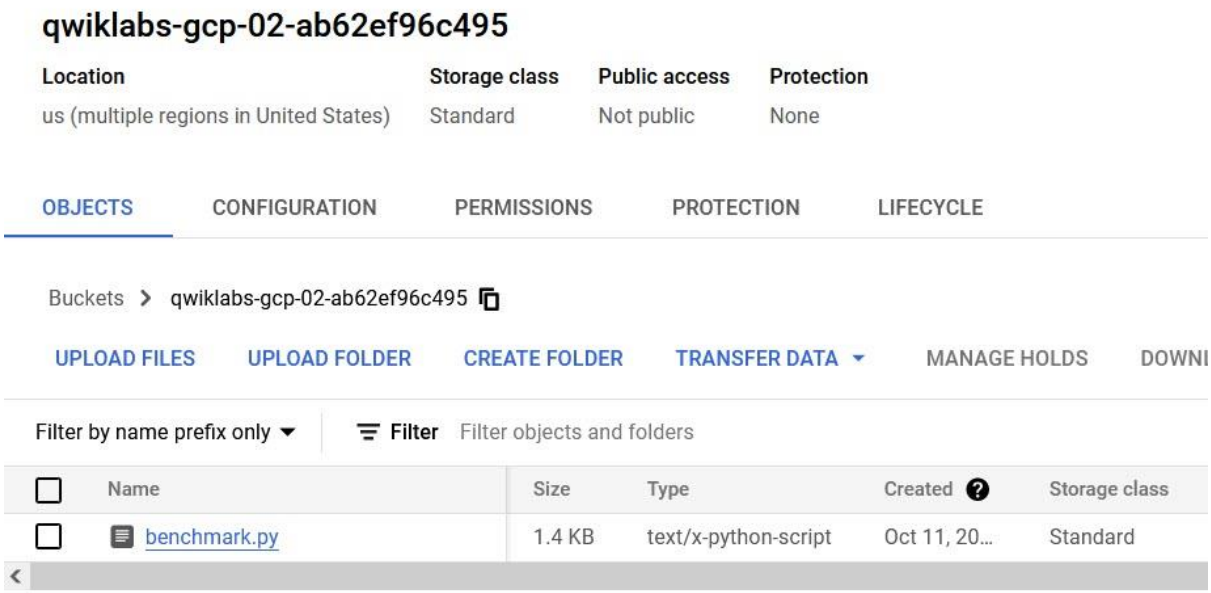
Task 1: Stage the benchmark PySpark application

Create a Cloud Storage bucket for use by your Cloud Dataproc cluster. Give the bucket the same name as your project. Copy the benchmark Python Spark application to the bucket in your project.



Copy benchmark.py file into storage bucket





Task 2: Create a Cloud Dataproc Cluster that matches the Data Analyst's configuration

Create a Cloud Dataproc cluster named mjtelco using version 2.0 (Debian 10, Hadoop 3.2, Spark 3.1) with a master node of n1-standard-2 and two worker nodes of n1-standard-2 in us-east1 region and us-east1-b zone. Use the default settings on everything else. Remember to set advanced options to give the cluster access to your Cloud Storage staging bucket.

Go to Dataproc Menu:

- 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.

Change access, encryption, and security settings.

settings.

- Manage security (optional)
  Change access, encryption, and security settings.

settings.

settings.

Change access, encryption, and security settings.

Change access, encryption, and security settings.

Change access, encryption, and security settings.

Change access, encryption, and security settings.

settings.

CREATE CANCEL

EQUIVALENT COMMAND LINE

Name

Cluster Name *
mjtelco

Location

Region *
us-east1

Zone *
us-east1-b

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

Autoscaling

Automates cluster resource management based on an autoscaling policy.

Policy
None

Enhanced Flexibility Mode

Dataproc Enhanced Flexibility Mode (EFM) manages shuffle data to minimize job progress delays caused by the removal of nodes from a running cluster. EFM offloads shuffle data in one of two user-selectable modes, primary worker shuffle and Hadoop Compatible File System (HCFS) shuffle. Learn more

An autoscaling policy must be selected to configure EFM.

Versioning

Use a custom image to load pre-installed packages. Learn more

Image Type and Version

2.0-debian10

Release Date

First released on 1/22/2021.

CHANGE

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

Manager node

Contains the YARN Resource Manager, HDFS NameNode, and all job drivers.

Machine family

GENERAL-PURPOSE | COMPUTE-OPTIMIZED | MEMORY-OPTIMIZED | GPU

Machine types for common workloads, optimized for cost and flexibility

Series
N1

Powered by Intel Skylake CPU platform or one of its predecessors

Machine type
n1-standard-2 (2 vCPU, 7.5 GB memory)

vCPU	Memory
2	7.5 GB

CPU PLATFORM AND GPU

Primary disk size *
500 GB

Primary disk type
Standard Persistent Disk

Number of local SS...
0 x 375GB

Local SSD Interface
SCSI

Worker nodes

Each contains a YARN NodeManager and a HDFS DataNode. HDFS replication factor is 2.

Machine family

GENERAL-PURPOSE | COMPUTE-OPTIMIZED | MEMORY-OPTIMIZED | GPU

Machine types for common workloads, optimized for cost and flexibility

Series
N1

Powered by Intel Skylake CPU platform or one of its predecessors

Machine type
n1-standard-2 (2 vCPU, 7.5 GB memory)

vCPU	Memory
2	7.5 GB

CPU PLATFORM AND GPU

Number of worker nodes *
2

Primary disk size *
500 GB

Primary disk type
Standard Persistent Disk

Number of local SS...
0 x 375GB

Local SSD Interface
SCSI

Secondary worker nodes

Each contains a YARN NodeManager. HDFS does not run on secondary worker nodes. Secondary worker VMs are preemptible by default. A preemptible VM costs less, but lasts only 24 hours, and can be terminated at any time due to system demands. Learn more

Sole-tenancy

Enable to create this cluster on sole-tenant nodes. This grants exclusive access to a physical Compute Engine server that is dedicated to hosting only your project's VMs. If you are creating a cluster with an autoscaling policy, it is recommended that the node group you select also uses an autoscaling policy. Learn more

Enable

Use initialization actions to customize settings, install applications, or make other modifications to your cluster. Select scripts or executables that Cloud Dataproc will run when provisioning your cluster.

+ ADD INITIALIZATION ACTION

Custom cluster metadata

Add custom metadata to cluster instances. Learn more

+ ADD METADATA

Scheduled deletion

Use Scheduled Deletion to help avoid incurring Google Cloud charges for an inactive cluster. Learn more

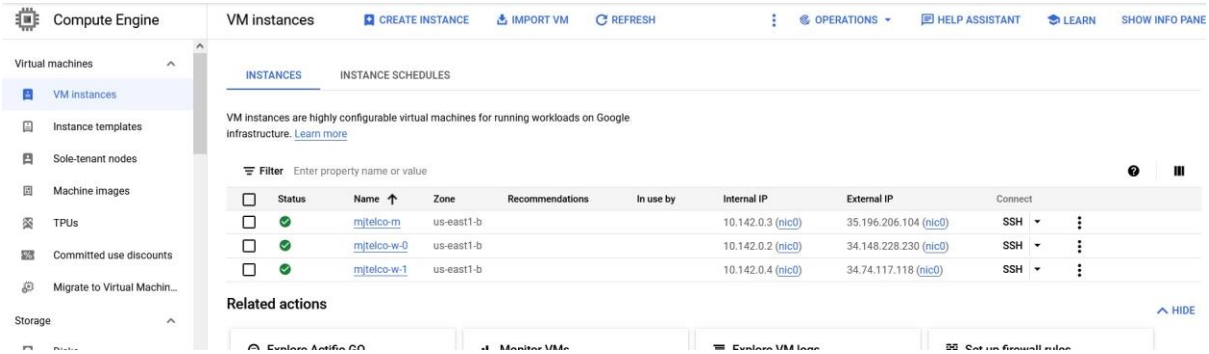
- [ ] Delete on a fixed time schedule
- [ ] Delete after a cluster idle time period without submitted jobs

Cloud Storage staging bucket

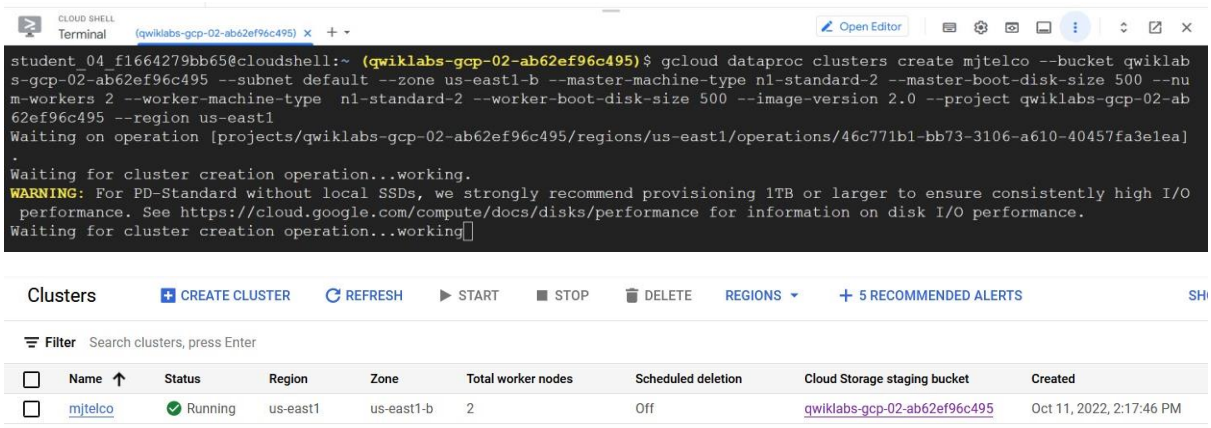
Storage staging bucket

qwiklabs-gcp-02-fc9c67b460dd BROWSE

Cloud Storage staging bucket to be used for storing cluster job dependencies, job driver output, and cluster config files.



gcloud dataproc clusters create mjtelco --bucket qwiklabs-gcp-02-ab62ef96c495 --subnet default --zone us-east1-b --master-machine-type n1-standard-2 --master-boot-disk-size 500 --num-workers 2 --worker-machine-type n1-standard-2 --worker-boot-disk-size 500 --image-version 2.0 --project qwiklabs-gcp-02-ab62ef96c495 --region us-east1



Task 3: Demonstrate the successful benchmark job without the required input value

Submit the python job to the cluster, and give the job the name mjtelco-test-1. Give the job the input argument of 20. For Max restarts per hour, enter 1.

Job ID *
mjtelco-test-1

Region *
us-east1

Specifies the Cloud Dataproc regional service, which determines what clusters are available.

Cluster *
mjtelco

Job type *
PySpark

Main python file *
gs://qwiklabs-gcp-02-ab62ef96c495/benchmark.py

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

Additional python files

Jar files

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.

Archive files

Archive files are extracted in the Spark working directory. 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. Supported file types: .jar, .tar, .tar.gz, .tgz, .zip.

Arguments
20 Press <Return> to add more arguments

Additional arguments to pass to the main class. Press Return after each argument.

Max restarts per hour
1

Leave blank if you don't want to allow automatic restarts on job failure. Learn more

Task 4: Demonstrate the slower benchmark job with the required input value

Submit the python job to the cluster, and give the job the name mjtelco-test-2. Give the job the input argument of 220. For Max restarts per hour, enter 1.

Job ID *
mjtelco-test-2

Region *
us-east1

Specifies the Cloud Dataproc regional service, which determines what clusters are available.

Cluster *
mjtelco

Job type *
PySpark

Main python file *
gs://qwiklabs-gcp-02-ba5b2e612a02/benchmark.py

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

Additional python files

Jar files

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.

Archive files

Archive files are extracted in the Spark working directory. 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. Supported file types: .jar, .tar, .tar.gz, .tgz, .zip.

Arguments
220 Press <Return> to add more arguments

Additional arguments to pass to the main class. Press Return after each argument.

Max restarts per hour
1

Leave blank if you don't want to allow automatic restarts on job failure. Learn more

Properties

Objective 2

Your second job is to improve the performance of the cluster and to reduce the time it takes to run the benchmark job.

Task 5: Upgrade the master node

Upgrade the master node to a 4-CPU instance, n1-standard-4.

Basic information

Name	mjtelco-m
Instance Id	947928495485915043
Description	None
Type	Instance
Status	Stopped
Creation time	Oct 11, 2022, 5:03:10 PM UTC+08:00
Zone	us-east1-b
Instance template	None
In use by	None
Reservations	Automatically choose (default)
Labels	goog-datap... : mjtelco goog-datap... : 6d984e67-4... goog-datap... : us-east1
Tags	—
Deletion protection	Disabled
Confidential VM service	Disabled
Preserved state size	0 GB

Machine configuration

Machine type	n1-standard-4
CPU platform	Unknown CPU Platform
Architecture	—
vCPUs to core ratio	—
Custom visible cores	—
Display device	Disabled Enable to use screen capturing and recording tools
GPUs	None

Task 6: Demonstrate that the benchmark job completes in less time

After the upgraded master node is running, submit the python job again to the cluster. Give the job the name mjtelco-test-3. Give the job the input argument of 220. For Max restarts per hour, enter 1.

Job ID *
mjtelco-test-3

Region *
us-east1

Specifies the Cloud Dataproc regional service, which determines what clusters are available.

Cluster *
mjtelco

Job type *
PySpark

Main python file *
gs://qwiklabs-gcp-02-ba5b2e612a02/benchmark.py

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

Additional python files

Jar files

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.

Archive files

Archive files are extracted in the Spark working directory. 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. Supported file types: .jar, .tar, .tar.gz, .tgz, .zip.

Arguments
220 Press <Return> to add more arguments

Additional arguments to pass to the main class. Press Return after each argument.

Max restarts per hour
1

Leave blank if you don't want to allow automatic restarts on job failure. Learn more

Task 7: Grow the cluster

You are getting closer but the job still does not complete in under the required time (under 75 seconds) when given the input value of 220.

Upgrade the cluster by adding three more n1-standard-2 worker nodes for a total of five workers.

Cluster details | SUBMIT JOB | REFRESH | START | STOP | DELETE | VIEW LOGS

Type	Dataproc Cluster
Status	Running

MONITORING | JOBS | VM INSTANCES | CONFIGURATION | WEB INTERFACES

EDIT

Region	us-east1
Zone	us-east1-b
Autoscaling	Off
Dataproc Metastore	None
Scheduled deletion	Off
Master node	Standard (1 master, N workers)
Machine type	n1-standard-2
Number of GPUs	0
Primary disk type	pd-standard
Primary disk size	500GB
Local SSDs	0
Worker nodes	2
Machine type	n1-standard-2
Number of GPUs	0
Primary disk type	pd-standard
Primary disk size	500GB

Cluster details | SUBMIT JOB | REFRESH | START | STOP | DELETE | VIEW LOGS

For PD-Standard without local SSDs, we strongly recommend provisioning 1TB or larger to ensure consistently high I/O performance. See https://cloud.google.com/compute/doc /performance for information on disk I/O performance.

Name	mjtelco
Cluster UUID	6d984e67-4ccb-4f33-9a3d-b34ba32a4648
Type	Dataproc Cluster
Status	Running

MONITORING | JOBS | VM INSTANCES | CONFIGURATION | WEB INTERFACES

Filter Filter instances

Name	Role
mjtelco-m	Master
mjtelco-w-0	Worker
mjtelco-w-1	Worker
mjtelco-w-2	Worker
mjtelco-w-3	Worker
mjtelco-w-4	Worker

Task 8: Submit the job and verify improved performance

After the additional nodes are running, submit the job again. Submit the python job to the cluster, and give the job the name mjtelco-test-4. Give the job the input argument of 220. For Max restarts per hour, enter 1.

Job ID *
mjtelco-test-4

Region *
us-east1

Specifies the Cloud Dataproc regional service, which determines what clusters are available.

Cluster *
mjtelco

Job type *
PySpark

Main python file *
gs://qwiklabs-gcp-02-ba5b2e612a02/benchmark.py

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

Additional python files

Jar files

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.

Archive files

Archive files are extracted in the Spark working directory. 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. Supported file types: .jar, .tar, .tar.gz, .tgz, .zip.

Arguments
220 Press <Return> to add more arguments

Additional arguments to pass to the main class. Press Return after each argument.

Max restarts per hour
1

Leave blank if you don't want to allow automatic restarts on job failure. Learn more

Properties