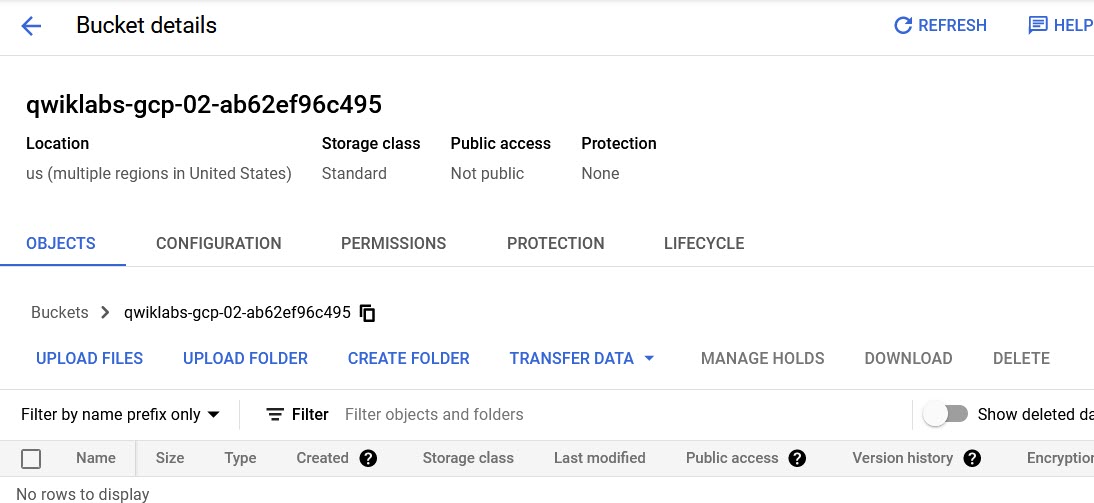
**PDE Prep—Cloud Dataproc Cluster Operations and Maintenance**

Activate Google Cloud Shell

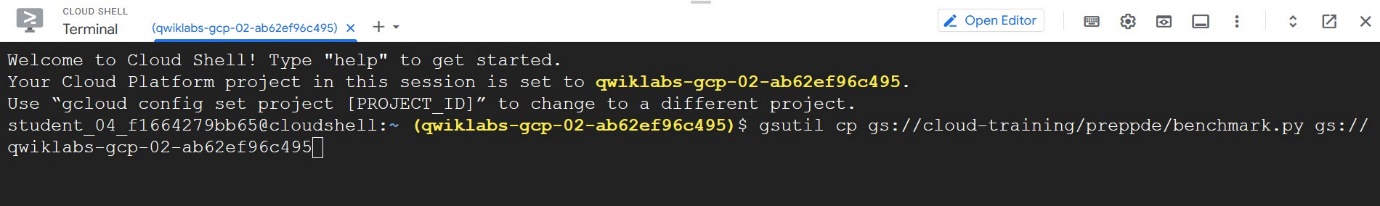
Check project permissions

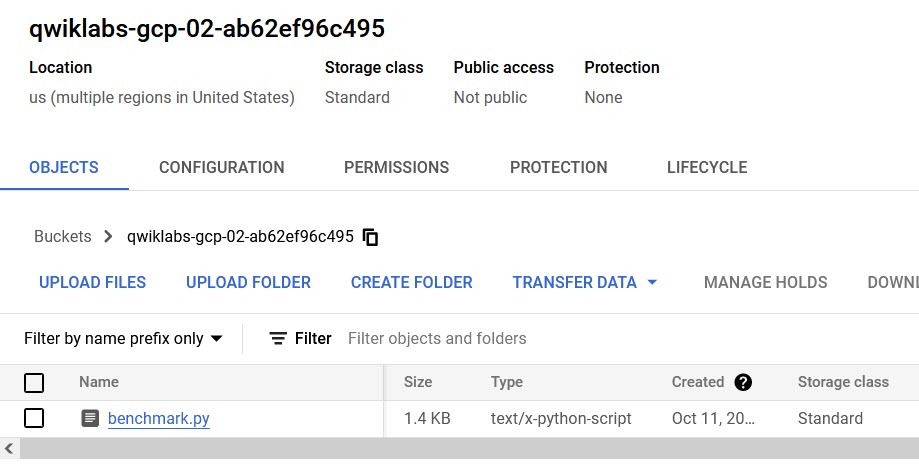
Task 1: Stage the benchmark PySpark application

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

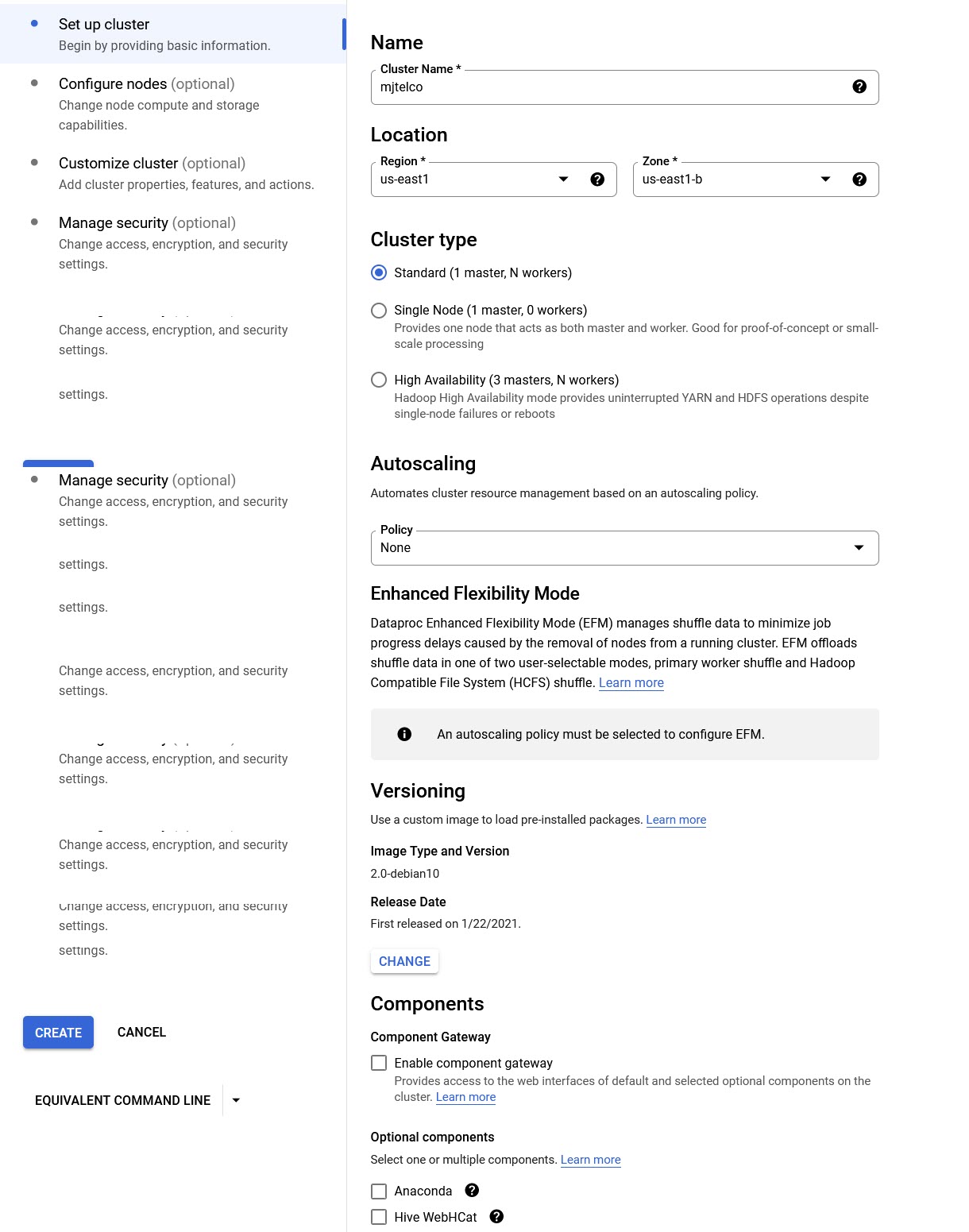


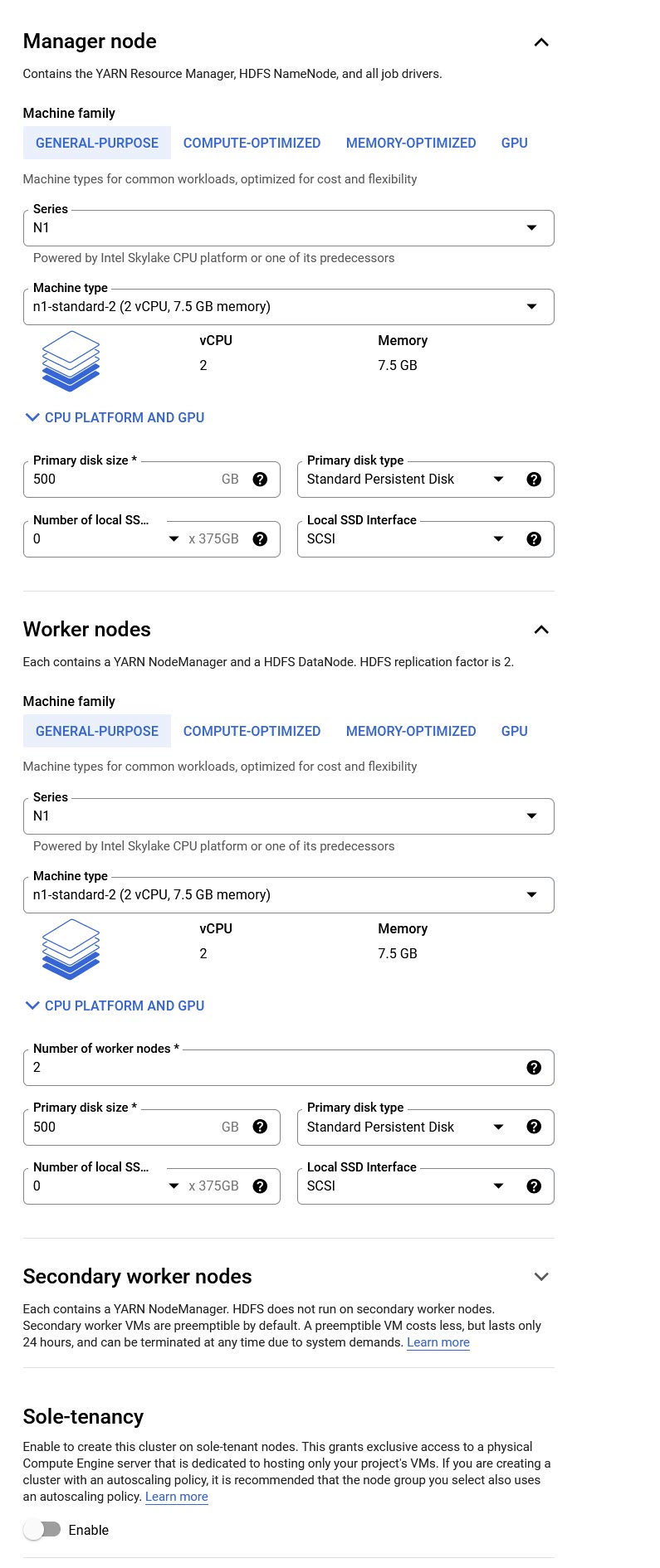


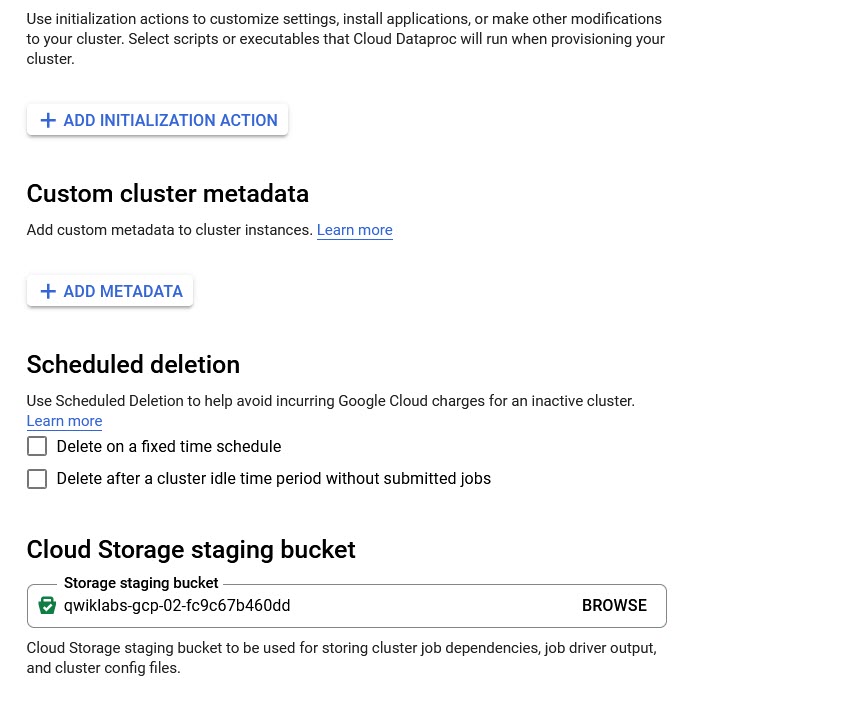
Task 2: Create a Cloud Dataproc Cluster that matches the Data Analyst's configuration

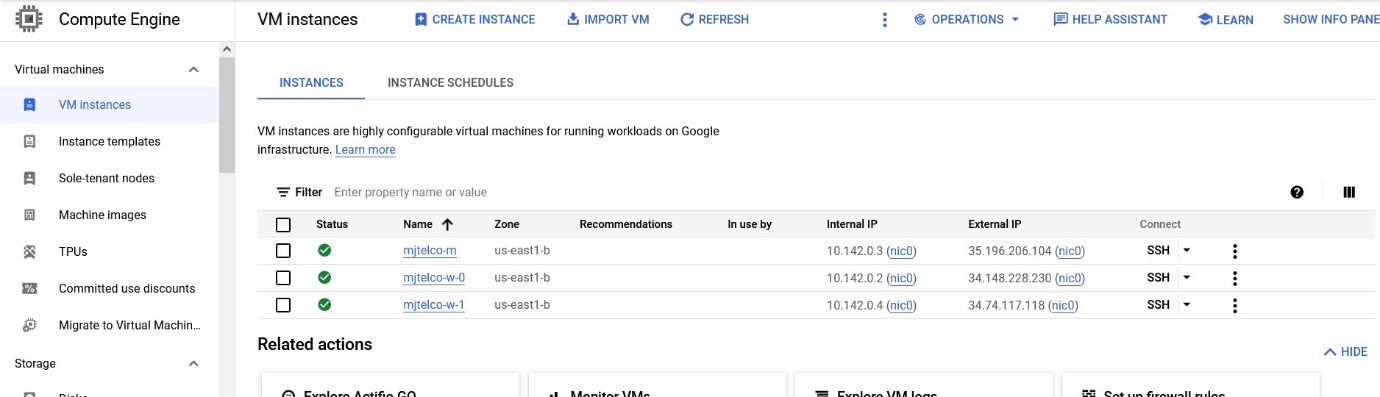
Create a Cloud Dataproc cluster named mjtelco using version 2.0 (Debian 10, Hadoop 3.2, Spark 3.1) with a master node of n1-standard-2 and two worker nodes of n1-standard-2 in us-east1 region and us-east1-b zone. Use the default settings on everything else. Remember to set advanced options to give the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

Go to Dataproc Menu:

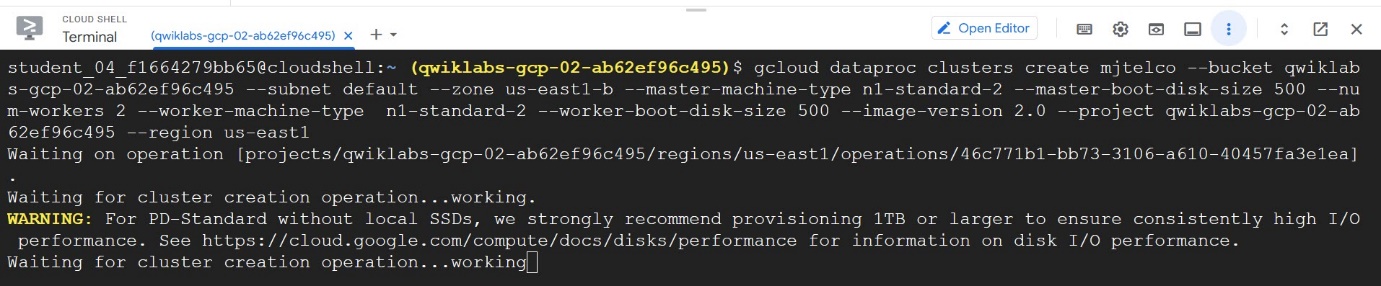


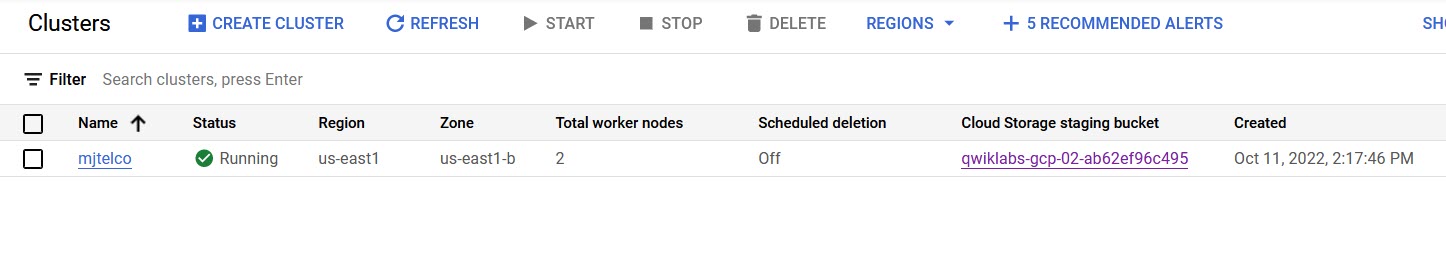






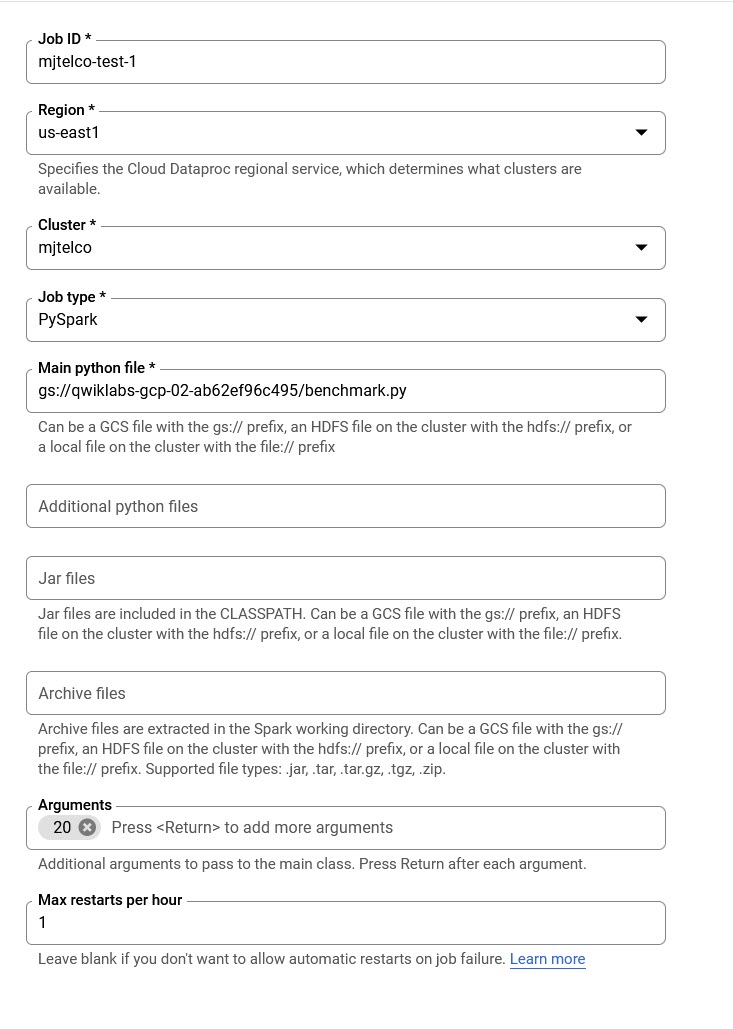
gcloud dataproc clusters create mjtelco --bucket qwiklabs-gcp-02-ab62ef96c495 --subnet default --zone us-east1-b --master-machine-type n1-standard-2 --master-boot-disk-size 500 --num-workers 2 --worker-machine-type n1-standard-2 --worker-boot-disk-size 500 --image-version 2.0 --project qwiklabs-gcp-02-ab62ef96c495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





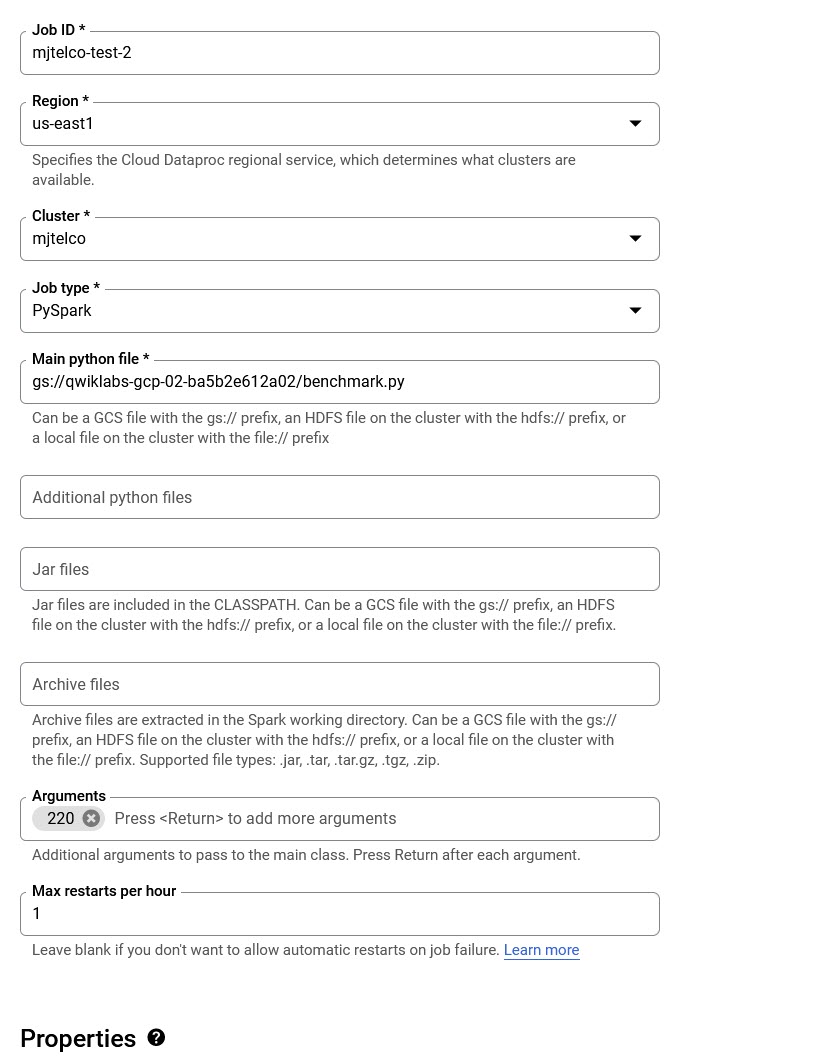
Task 3: Demonstrate the successful benchmark job without the required input value

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.



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.

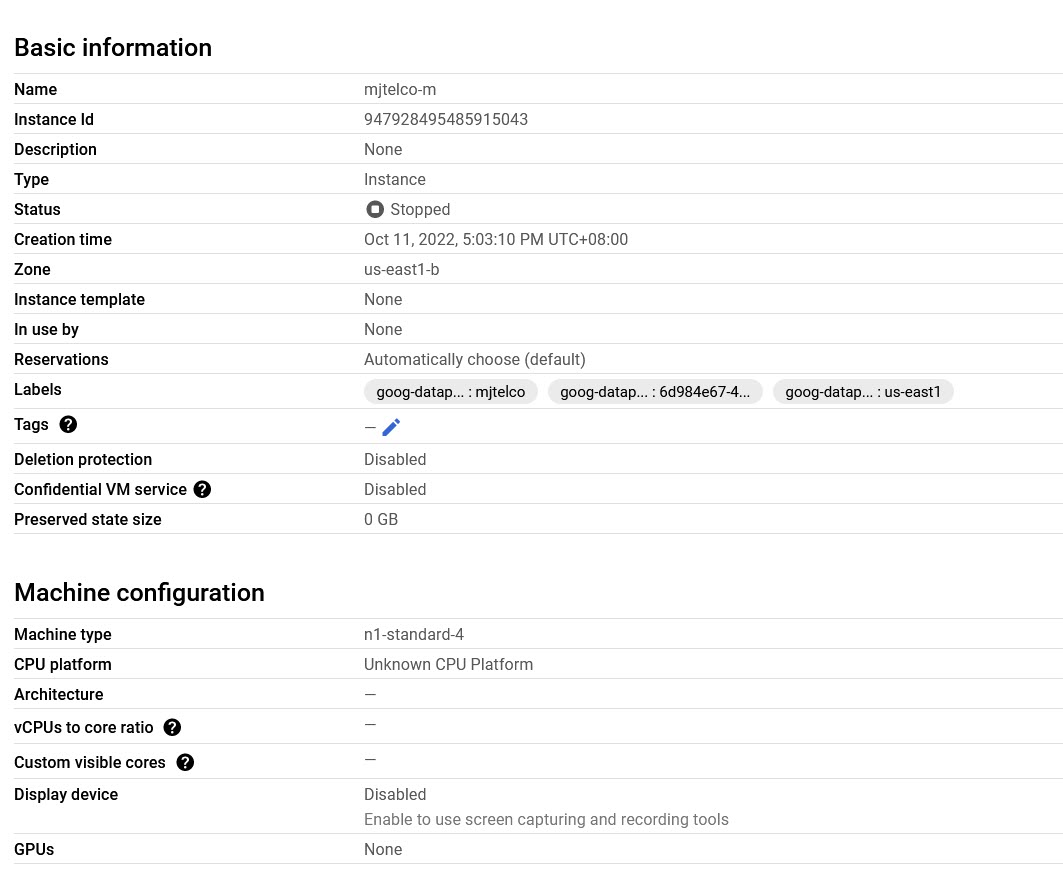


Objective 2

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

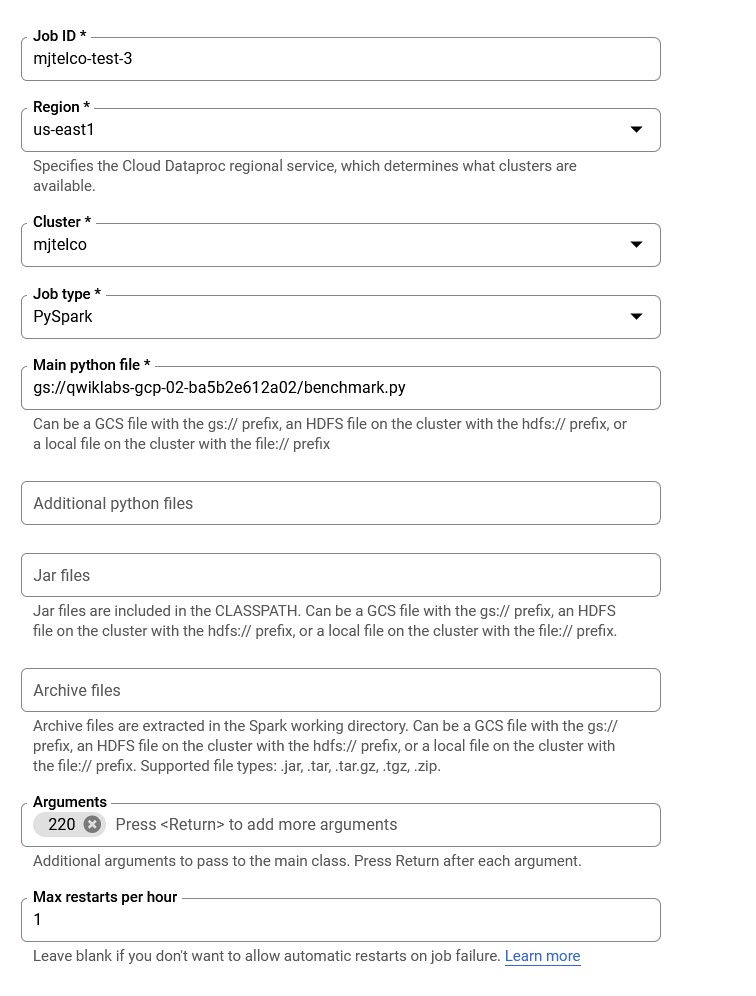
Task 5: Upgrade the master node

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



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

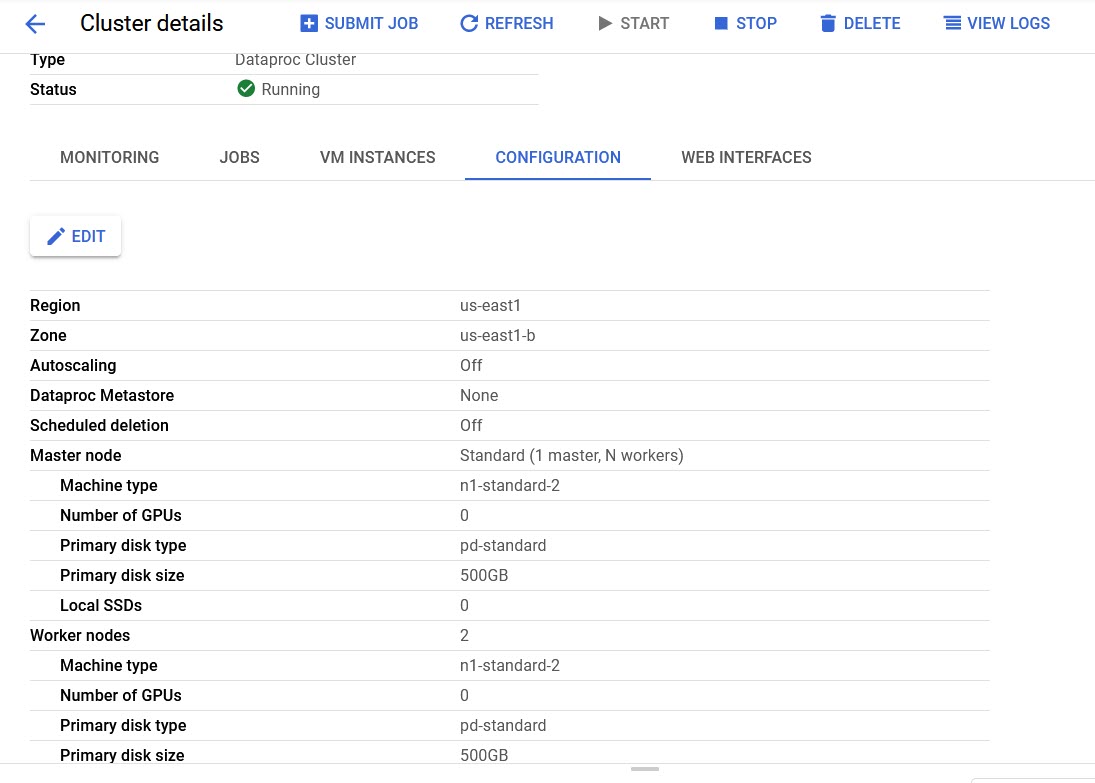
After the upgraded master node is running, submit the python job again to the cluster. Give the job the name mjtelco-test-3. Give the job the input argument of 220. For Max restarts per hour, enter 1.

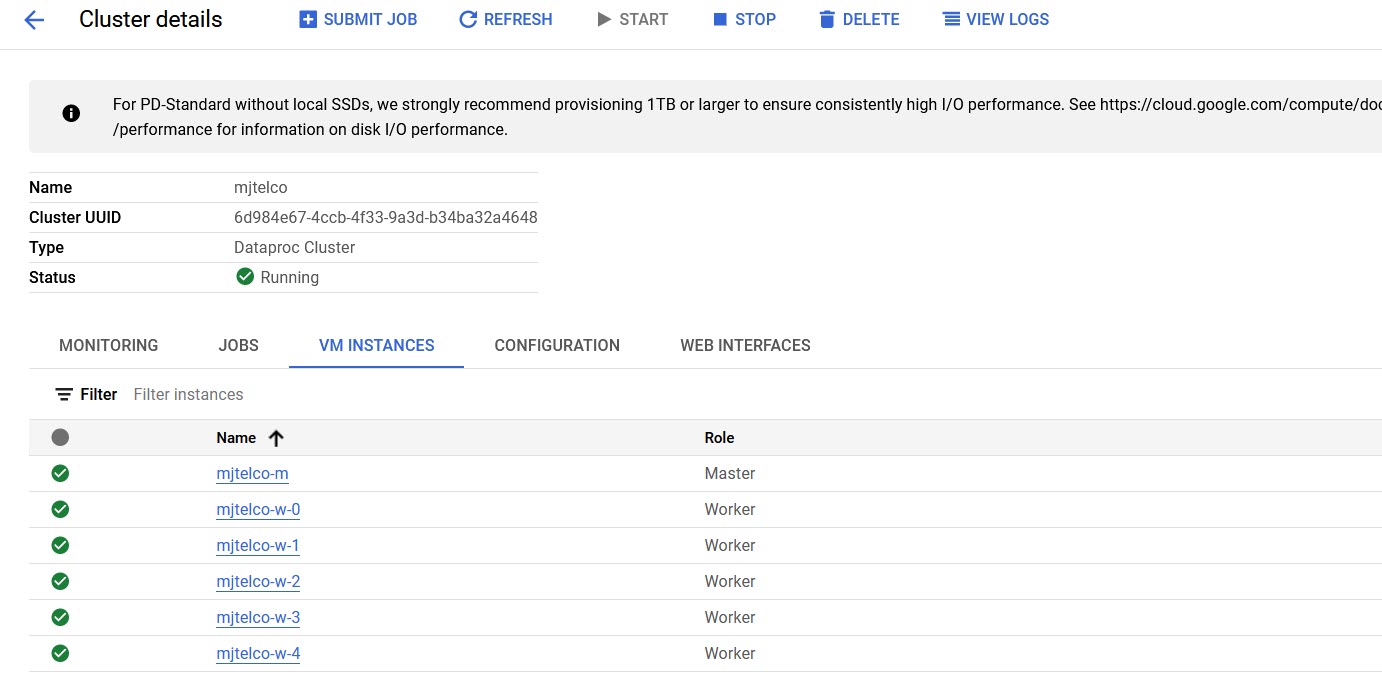


Task 7: Grow the cluster

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.





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.

