Dataproc: Qwik Start - Command Line

Activate Cloud Shell

Cloud Shell is a virtual machine that is loaded with development tools. It offers a persistent 5GB home directory and runs on the Google Cloud. Cloud Shell provides command-line access to your Google Cloud resources.

1. Click **Activate Cloud Shell** >_ at the top of the Google Cloud console.

2. Click through the following windows:

   - Continue through the Cloud Shell information window.
   - Authorize Cloud Shell to use your credentials to make Google Cloud API calls.

When you are connected, you are already authenticated, and the project is set to your **Project_ID**, `qwiklabs-gcp-01-5a5e7a45bd7c`. The output contains a line that declares the **Project_ID** for this session:

```
Your Cloud Platform project in this session is set to qwiklabs-gcp-01-5a5e7a45bd7c
```

`gcloud` is the command-line tool for Google Cloud. It comes pre-installed on Cloud Shell and supports tab-completion.

3. (Optional) You can list the active account name with this command:

```
gcloud auth list
```

Copied!

content_copy

4. Click **Authorize**.

Output:

```
ACTIVE: *
ACCOUNT: student-01-91d7fd156d5d@qwiklabs.net

To set the active account, run:
    $ gcloud config set account `ACCOUNT`
```

5. (Optional) You can list the project ID with this command:

```
gcloud config list project
```

Copied!

content_copy

Output:

```
[core]
project = qwiklabs-gcp-01-5a5e7a45bd7c
```

Note: For full documentation of `gcloud`, in Google Cloud, refer to the gcloud CLI overview guide.

Task 1. Create a cluster

1. In Cloud Shell, run the following command to set the Region:

```
gcloud config set dataproc/region us-central1
```

Copied!

content_copy

2. Dataproc creates staging and temp buckets that are shared among clusters in the same region. Since we're not specifying an account for Dataproc to use, it will use the Compute Engine default service account, which doesn't have storage bucket permissions by default. Let's add those.

- First, run the following commands to grab the PROJECT_ID and PROJECT_NUMBER:

```
PROJECT_ID=$(gcloud config get-value project) && \
gcloud config set project $PROJECT_ID

PROJECT_NUMBER=$(gcloud projects describe $PROJECT_ID --
format='value(projectNumber)')
```

Copied!

content_copy

- Now run the following command to give the Storage Admin role to the Compute Engine default service account:

```
gcloud projects add-iam-policy-binding $PROJECT_ID \
  --member=serviceAccount:$PROJECT_NUMBER-
compute@developer.gserviceaccount.com \
  --role=roles/storage.admin
```

Copied!

content_copy

3. Enable Private Google Access on your subnetwork by running the following command:

```
gcloud compute networks subnets update default --region=us-central1  --
enable-private-ip-google-access
```

Copied!

content_copy

4. Run the following command to create a cluster called `example-cluster` with e2-standard-4 VMs and default Cloud Dataproc settings:

```
gcloud dataproc clusters create example-cluster --worker-boot-disk-size
500 --worker-machine-type=e2-standard-4 --master-machine-type=e2-
standard-4
```

Copied!

content_copy

5. If asked to confirm a zone for your cluster. Enter **Y**.

Your cluster will build for a couple of minutes.

```
Waiting for cluster creation operation...done.
Created [... example-cluster]
```

When you see a "Created" message, you're ready to move on.

Task 2. Submit a job

- Run this command to submit a sample Spark job that calculates a rough value for pi:

```
gcloud dataproc jobs submit spark --cluster example-cluster \
  --class org.apache.spark.examples.SparkPi \
  --jars file:///usr/lib/spark/examples/jars/spark-examples.jar -- 1000
```

Copied!

content_copy

The command specifies:

- That you want to run a spark job on the `example-cluster` cluster
- The `class` containing the main method for the job's pi-calculating application
- The location of the jar file containing your job's code
- The parameters you want to pass to the job—in this case, the number of tasks, which is `1000`

Note: Parameters passed to the job must follow a double dash (--). See the gcloud documentation for more information.

The job's running and final output is displayed in the terminal window:

```

Waiting for job output...
...
```

```
Pi is roughly 3.14118528
...
state: FINISHED
```

Task 3. Update a cluster

1. To change the number of workers in the cluster to four, run the following command:

```
gcloud dataproc clusters update example-cluster --num-workers 4
```

Copied!

content_copy

Your cluster's updated details are displayed in the command's output:

```
Waiting on operation [projects/qwiklabs-gcp-
7f7aa0829e65200f/regions/global/operations/b86892cc-e71d-4e7b-aa5e-
6030c945ea67].
Waiting for cluster update operation...done.
```

2. You can use the same command to decrease the number of worker nodes:

```
gcloud dataproc clusters update example-cluster --num-workers 2
```

Copied!

content_copy

Now you can create a Dataproc cluster and adjust the number of workers from the `gcloud` command line on the Google Cloud.