Quick set up guide for Jupyter Notebook and Pyspark on Google Cloud

Intro: Similar to Amazon EMR, Google Dataproc is the platform designed for distributed clusters computation, which is built for big data analysis including Hadoop, Spark, HBase and others.

Compared with common line tools, Jupyter Notebook is an easier way to interact with Spark, as the notebook can easily visualize all the codes and the results.

But by default, Jupyter is not installed on both platforms, and users need to run custom script during cluster initialization if they wish to.

Advantages of Dataproc compared with EMR:

	Google Dataproc	AWS EMR
Default Provisioning	1 minute	8 minutes
Custom Provisioning	30+ minutes	Less than 5 minutes
Accessibility	Both Web shell and SSH	Only SSH
Charging cycle	By Minute	By Hour
Pricing[1]	$0.24 per hour	$0.336 per hour

[1] Default Provisioning with 4 vCPUs and 15 GB of RAM

We set up a trial to compare the performance and cost of a typical Spark workload. The trial used clusters with one master and five core instances of AWS’s m3.xlarge and GCP’s n1-standard-4.

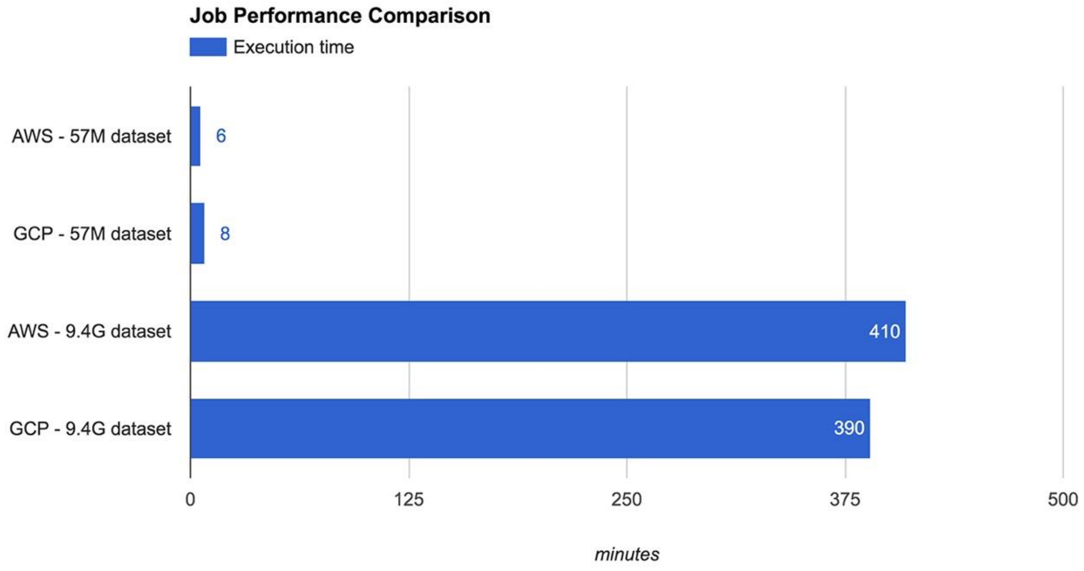


Figure 1. Computation Efficiently Credit: Michael Li and Ariel M'ndange-Pfupfu.

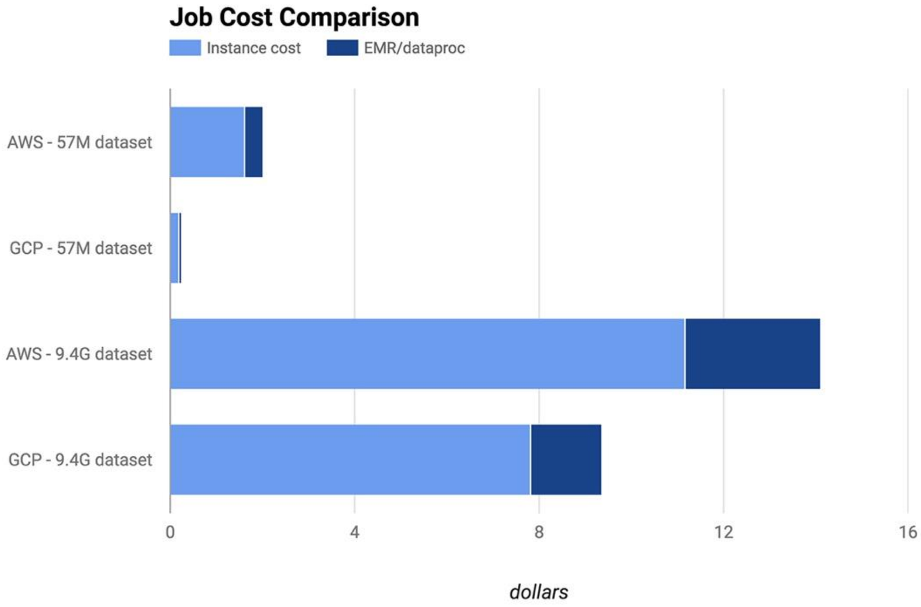
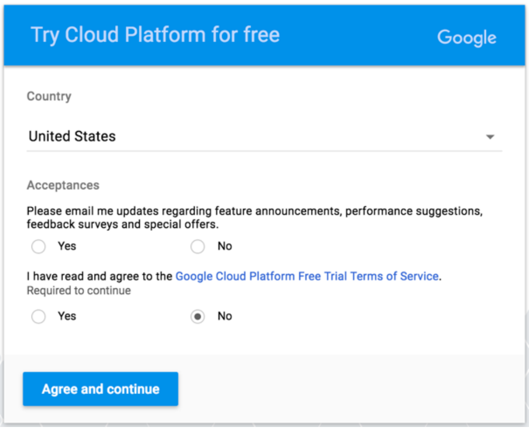


Figure 2. Cost Comparison Credit: Michael Li and Ariel M'ndange-Pfupfu.

Set up

1. You can sign up for free trial on Google Cloud, which would offer $300 credit and free tier service.



Country

United States

Acceptances

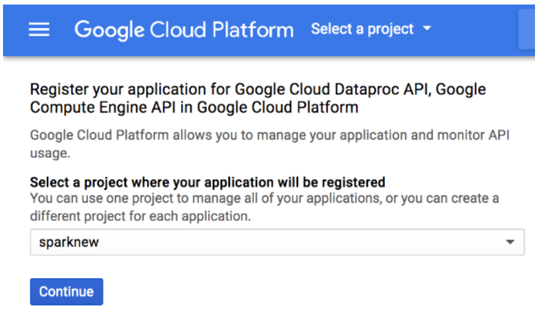
Please email me updates regarding feature announcements, performance suggestions, feedback surveys and special offers.

○ Yes ○ No

I have read and agree to the Google Cloud Platform Free Trial Terms of Service.
Required to continue

○ Yes ● No

2. Select or create a Cloud Platform project. All the service in Google cloud is based on ‘Project’. And you should also enable billing for the project and the Cloud Dataproc APIs.



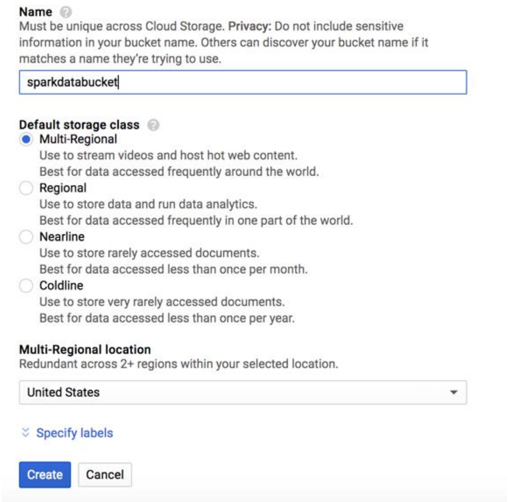
Register your application for Google Cloud Dataproc API, Google Compute Engine API in Google Cloud Platform

Google Cloud Platform allows you to manage your application and monitor API usage.

Select a project where your application will be registered
You can use one project to manage all of your applications, or you can create a different project for each application.

sparknew

3.Create the Cloud Storage bucket. Multi-regional is for better access.



Name
Must be unique across Cloud Storage. **Privacy:** Do not include sensitive information in your bucket name. Others can discover your bucket name if it matches a name they're trying to use.

sparkdatabucket

Default storage class

● Multi-Regional
Use to stream videos and host hot web content.
Best for data accessed frequently around the world.

○ Regional
Use to store data and run data analytics.
Best for data accessed frequently in one part of the world.

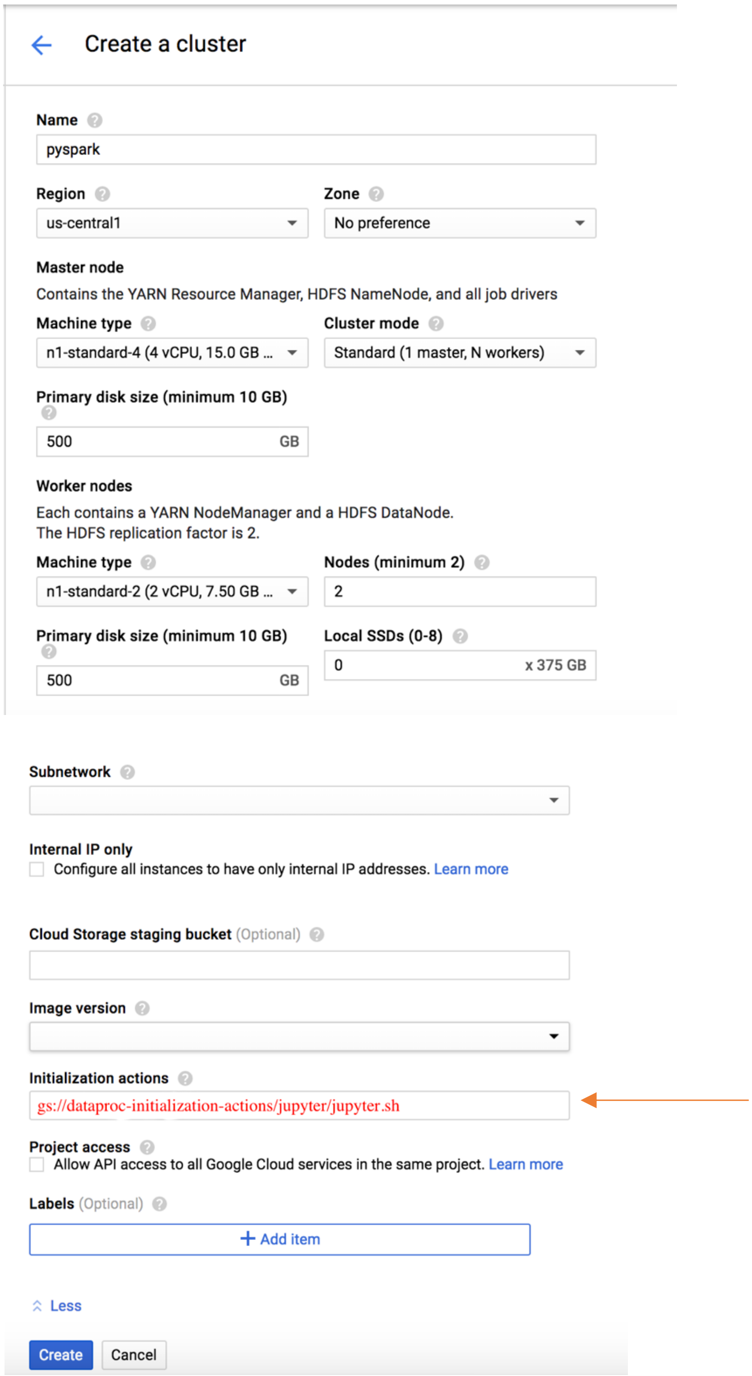
○ Nearline
Use to store rarely accessed documents.
Best for data accessed less than once per month.

○ Coldline
Use to store very rarely accessed documents.
Best for data accessed less than once per year.

Multi-Regional location
Redundant across 2+ regions within your selected location.

United States

4. You can use either Cloud SDK(Google Cloud Commend line tool) or Web console to create the cluster. For easier visualization, we would use the web console.



Create a cluster

Name: pyspark

Region: us-central1

Zone: No preference

Master node

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

Machine type: n1-standard-4 (4 vCPU, 15.0 GB ...

Cluster mode: Standard (1 master, N workers)

Primary disk size (minimum 10 GB): 500 GB

Worker nodes

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

Machine type: n1-standard-2 (2 vCPU, 7.50 GB ...

Nodes (minimum 2): 2

Primary disk size (minimum 10 GB): 500 GB

Local SSDs (0-8): 0 x 375 GB

Subnetwork:

Internal IP only

□ Configure all instances to have only internal IP addresses. Learn more

Cloud Storage staging bucket (Optional):

Image version:

Initialization actions: gs://dataproc-initialization-actions/jupyter/jupyter.sh

Project access

□ Allow API access to all Google Cloud services in the same project. Learn more

Labels (Optional)

For the 'initialization actions' part, we would use a bash shell initialization script provided by Dataproc team on Cloud Storage. This script and other initialization scripts are co-located at the GitHub [GoogleCloudPlatform/dataproc-initialization-actions repository]().

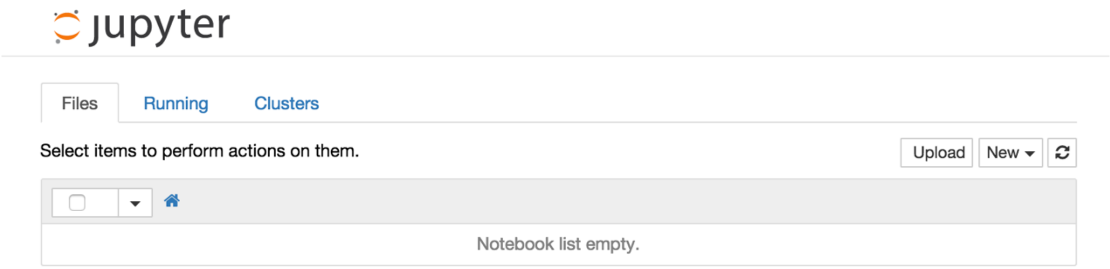
5. Establish SSH tunnel. You can use Cloud SDK to establish the SSH tunnel. It would configure keys and authentication automatically.

```
$ gcloud compute ssh "cluster-name-m" \
    --project project-id \
    --zone=cluster-zone \
    -- -D 10000 –N
```

6. SSH tunnel supports traffic proxying using the SOCKS protocol, and we need to set up browser to access the tunnel. You should use Terminals or cmd.

```
/Applications/Google\ Chrome.app/Contents/MacOS/Google\ Chrome \
      "http://<cluster-name>-m:8123" \
      --proxy-server="socks5://localhost:10000" \
      --host-resolver-rules="MAP * 0.0.0.0 , EXCLUDE localhost" \
      --user-data-dir=/tmp
```

The opening page of the Jupyter notebook displays in your browser. You can use Jupyter for Pyspark now.



Select items to perform actions on them.

Notebook list empty.

Reference

Michael Li &Ariel M'ndange-Pfupfu. Spark comparison: AWS vs. GCP. O'Reilly, August 30,2016

Install and run a Jupyter notebook in a Cloud Dataproc cluster, Google Cloud Documentation, available at: https://cloud.google.com/dataproc/docs/tutorials/jupyter-notebook

From UMN MSBA6330 Trends Marketplace Project by:

Abhijit Patil: patil074@umn.edu,

Abhilasha Pandey: pande130@umn.edu,

Ankit Agarwal: agarw145@umn.edu,

Ao Liu: liux4399@umn.edu,

Suzanne Kaminski: kamin089@umn.edu