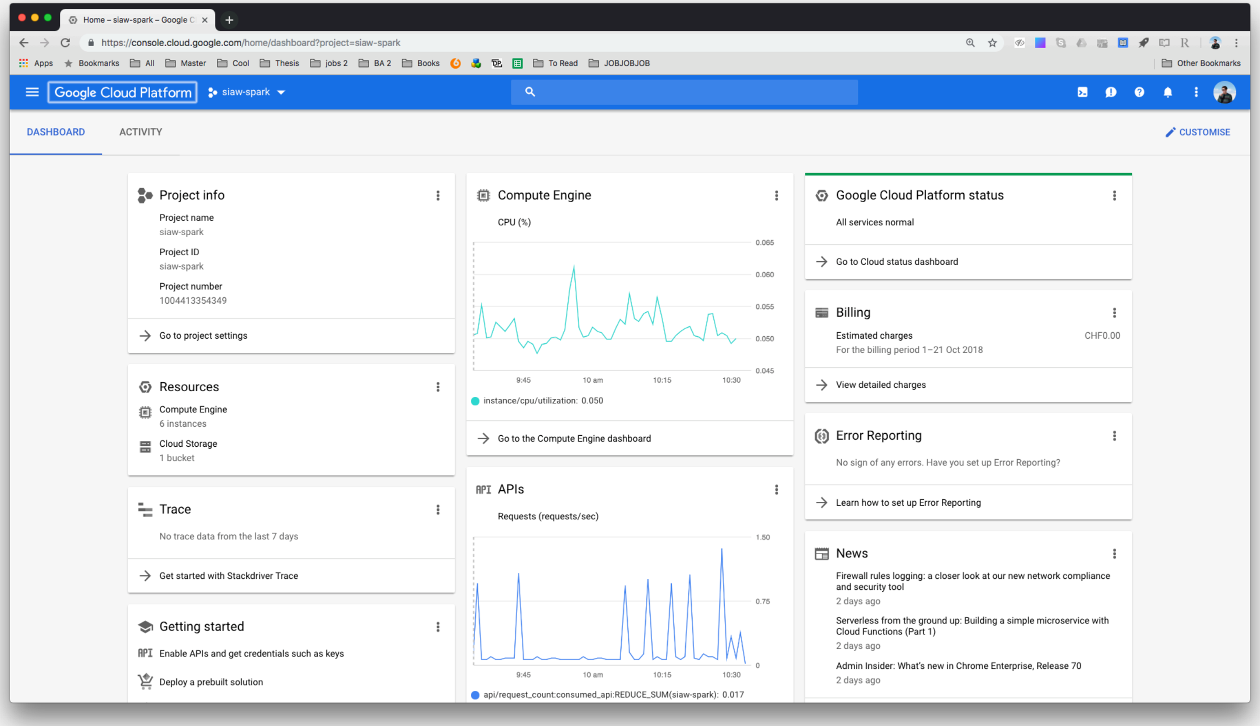
RUNNING A PYSPARK CLUSTER ON A SERVER

This tutorial will show you how to set-up a server cluster with Apache Spark, how to load data into it, and interact with your cluster using a Jupyter Notebook PySpark interface

STEP 1: OPEN GOOGLE CLOUD PLATFORM

- If you already have an account simply log-in
- If you do not have a Google Cloud Platform (GCP) account go to https://cloud.google.com/ and create a free account
- If this is your first account, you will receive 300 USD of free platform credit so you should not be incurring any costs for the sake of this tutorial
- As soon as you have your account set up create a new project. The name used for this example will be called "siaw-spark"
- You should now be able to see your dashboard



Project info

Project name
siaw-spark

Project ID
siaw-spark

Project number
1004413354349

→ Go to project settings

Resources

Compute Engine
6 instances

Cloud Storage
1 bucket

Trace

No trace data from the last 7 days

→ Get started with Stackdriver Trace

Getting started

API Enable APIs and get credentials such as keys

Deploy a prebuilt solution

Compute Engine

CPU (%)

● instance/cpu/utilization: 0.050

→ Go to the Compute Engine dashboard

API APIs

Requests (requests/sec)

● api/request_count:consumed_api:REDUCE_SUM(siaw-spark): 0.017

Google Cloud Platform status

All services normal

→ Go to Cloud status dashboard

Billing

Estimated charges CHF0.00
For the billing period 1–21 Oct 2018

→ View detailed charges

Error Reporting

No sign of any errors. Have you set up Error Reporting?

→ Learn how to set up Error Reporting

News

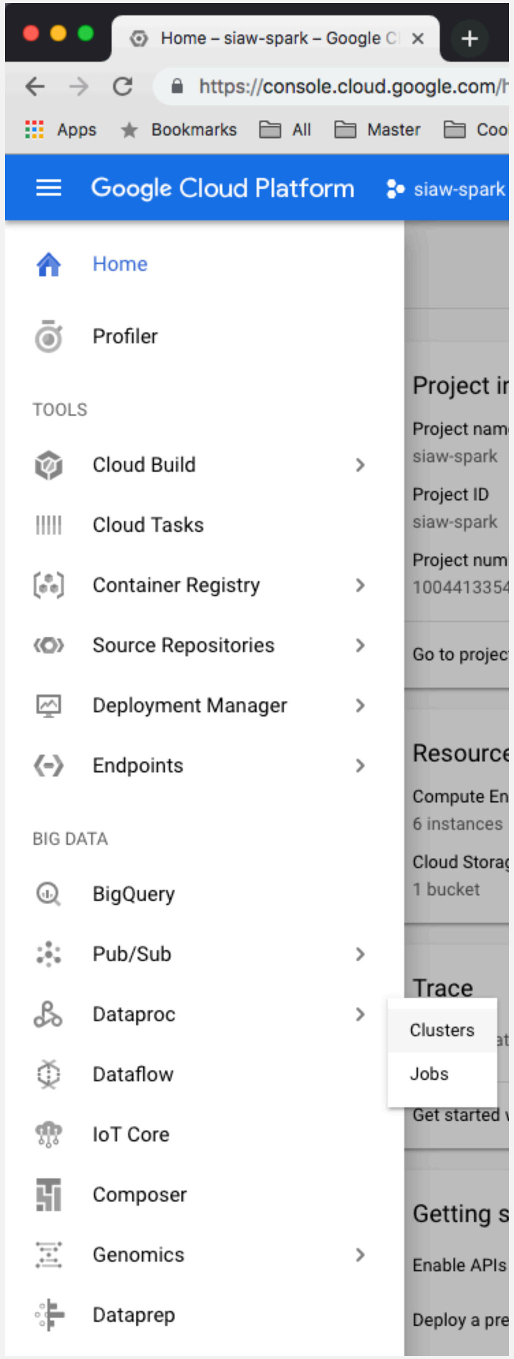
Firewall rules logging: a closer look at our new network compliance and security tool
2 days ago

Serverless from the ground up: Building a simple microservice with Cloud Functions (Part 1)
2 days ago

Admin Insider: What's new in Chrome Enterprise, Release 70
2 days ago

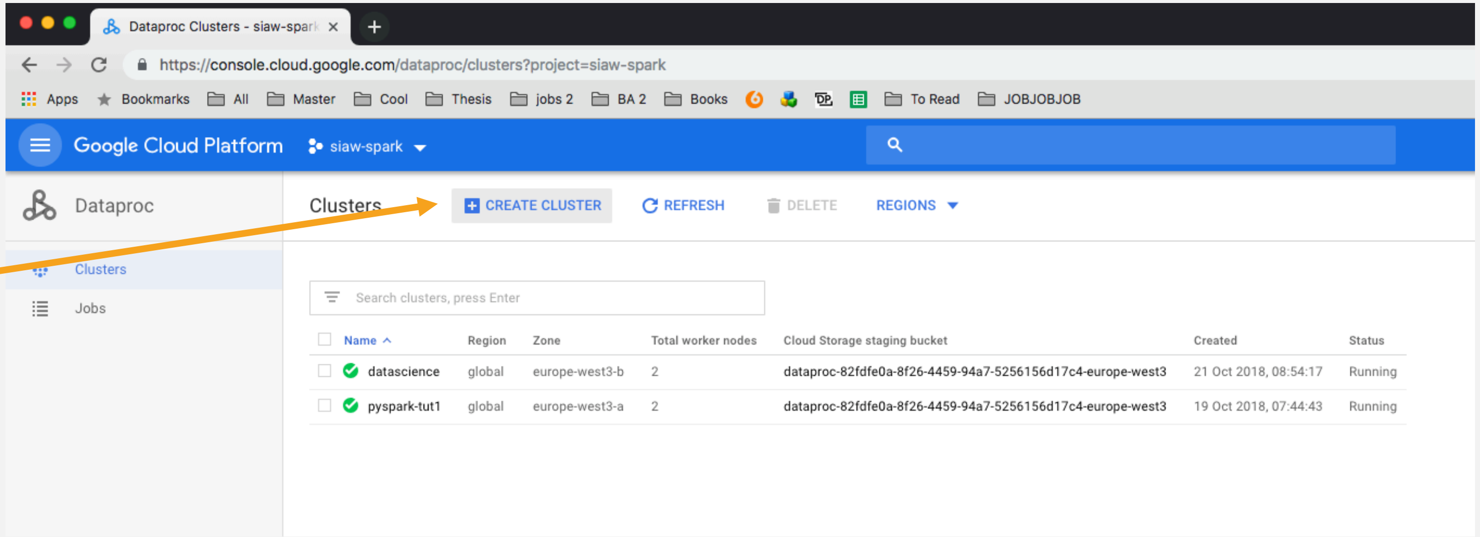
2. CLUSTER CREATION

- Go to the menu in the top left corner and look for the item “Dataproc” under the subcategory “Big Data”
- Click on “Clusters”



2. CLUSTER CREATION

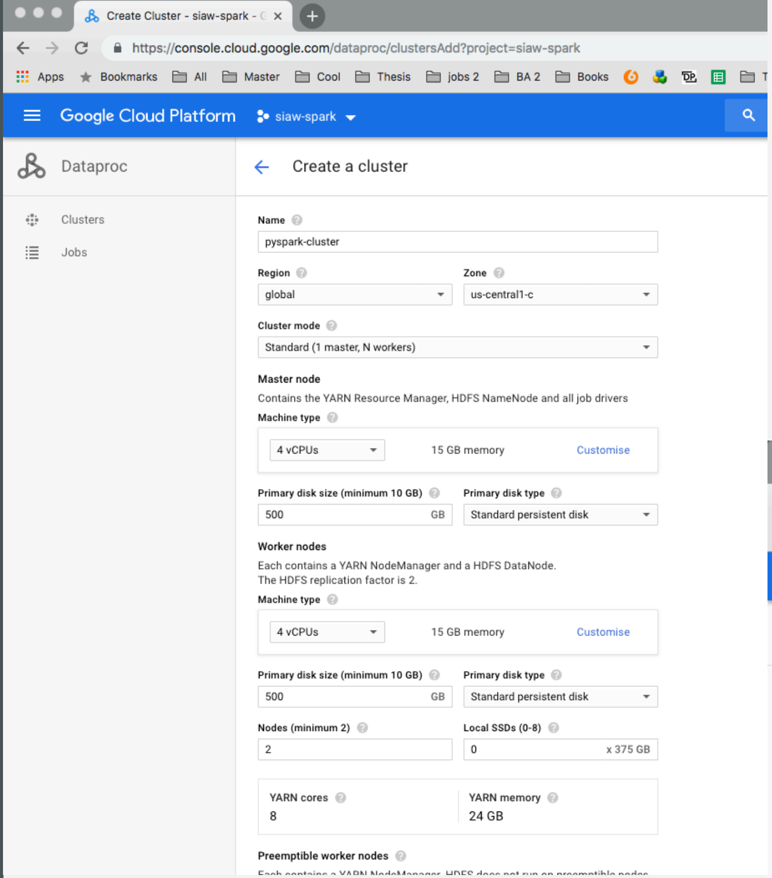
- In the new menu you will see no clusters running so far.
- If prompted, first enable the Cloud Dataproc API by following the instructions provided
- Next, click on “create cluster”



Name	Region	Zone	Total worker nodes	Cloud Storage staging bucket	Created	Status
datascience	global	europe-west3-b	2	dataproc-82fdfe0a-8f26-4459-94a7-5256156d17c4-europe-west3	21 Oct 2018, 08:54:17	Running
pyspark-tut1	global	europe-west3-a	2	dataproc-82fdfe0a-8f26-4459-94a7-5256156d17c4-europe-west3	19 Oct 2018, 07:44:43	Running

2. CLUSTER CREATION

- Assign a name to your cluster. For this installation example we will be using "pyspark-cluster"
- Change the region to a location close to you, I will choose Europe West
- You can change many other preferences here but for now we will ignore most of them
- Open "advanced options" at the bottom and click on "add initialisation action"
- Insert the link as explained on the right. The link directs to a script which allows us to use the Jupyter interface later on
- Click: "create"*



Create Cluster - siaw-spark
https://console.cloud.google.com/dataproc/clustersAdd?project=siaw-spark
Apps
Bookmarks
All
Master
Cool
Thesis
jobs 2
BA 2
Books
Google Cloud Platform
siaw-spark
Dataproc
Clusters
Jobs

Create a cluster

Name: pyspark-cluster

Region: global

Zone: us-central1-c

Cluster mode: Standard (1 master, N workers)

Master node

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

Machine type: 4 vCPUs 15 GB memory Customise

Primary disk size (minimum 10 GB): 500 GB

Primary disk type: Standard persistent disk

Worker nodes

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

Machine type: 4 vCPUs 15 GB memory Customise

Primary disk size (minimum 10 GB): 500 GB

Primary disk type: Standard persistent disk

Nodes (minimum 2): 2

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

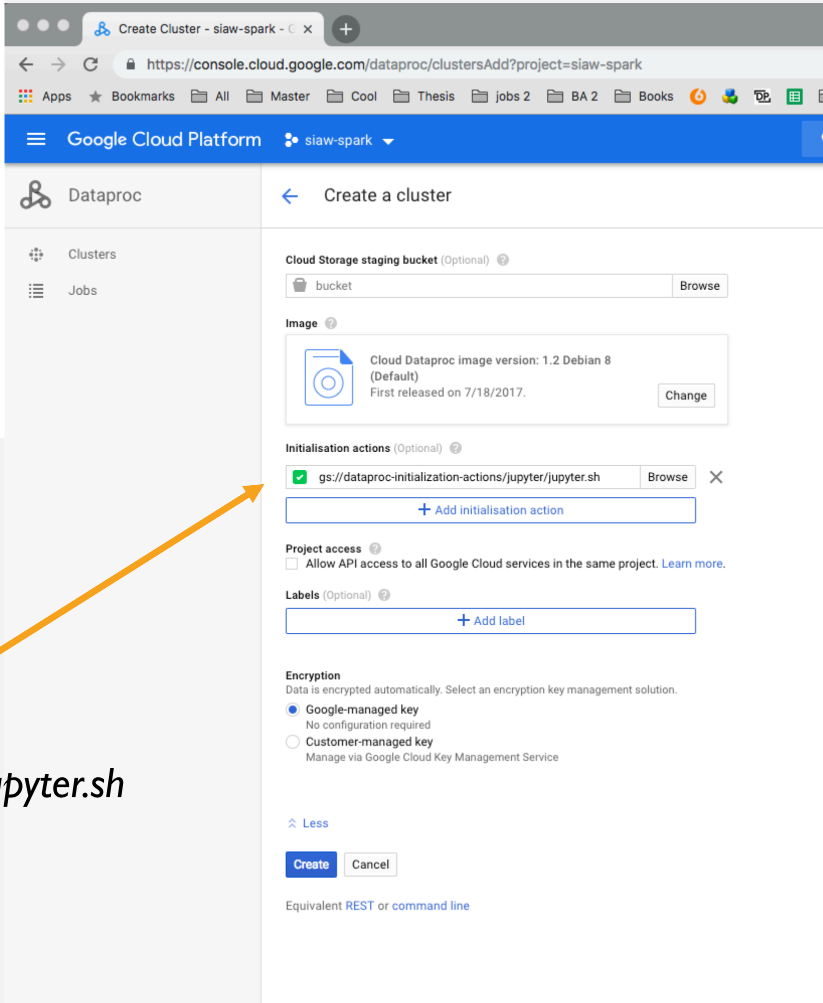
YARN cores: 8

YARN memory: 24 GB

Preemptible worker nodes

Insert this link here:

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



Create Cluster - siaw-spark
https://console.cloud.google.com/dataproc/clustersAdd?project=siaw-spark
Apps
Bookmarks
All
Master
Cool
Thesis
jobs 2
BA 2
Books
Google Cloud Platform
siaw-spark
Dataproc
Clusters
Jobs

Create a cluster

Cloud Storage staging bucket (Optional): bucket Browse

Image: Cloud Dataproc image version: 1.2 Debian 8 (Default)
First released on 7/18/2017. Change

Initialisation actions (Optional): ☑ gs://dataproc-initialization-actions/jupyter/jupyter.sh Browse

+ Add initialisation action

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

Labels (Optional)

+ Add label

Encryption
Data is encrypted automatically. Select an encryption key management solution.

● Google-managed key
No configuration required

○ Customer-managed key
Manage via Google Cloud Key Management Service

Less

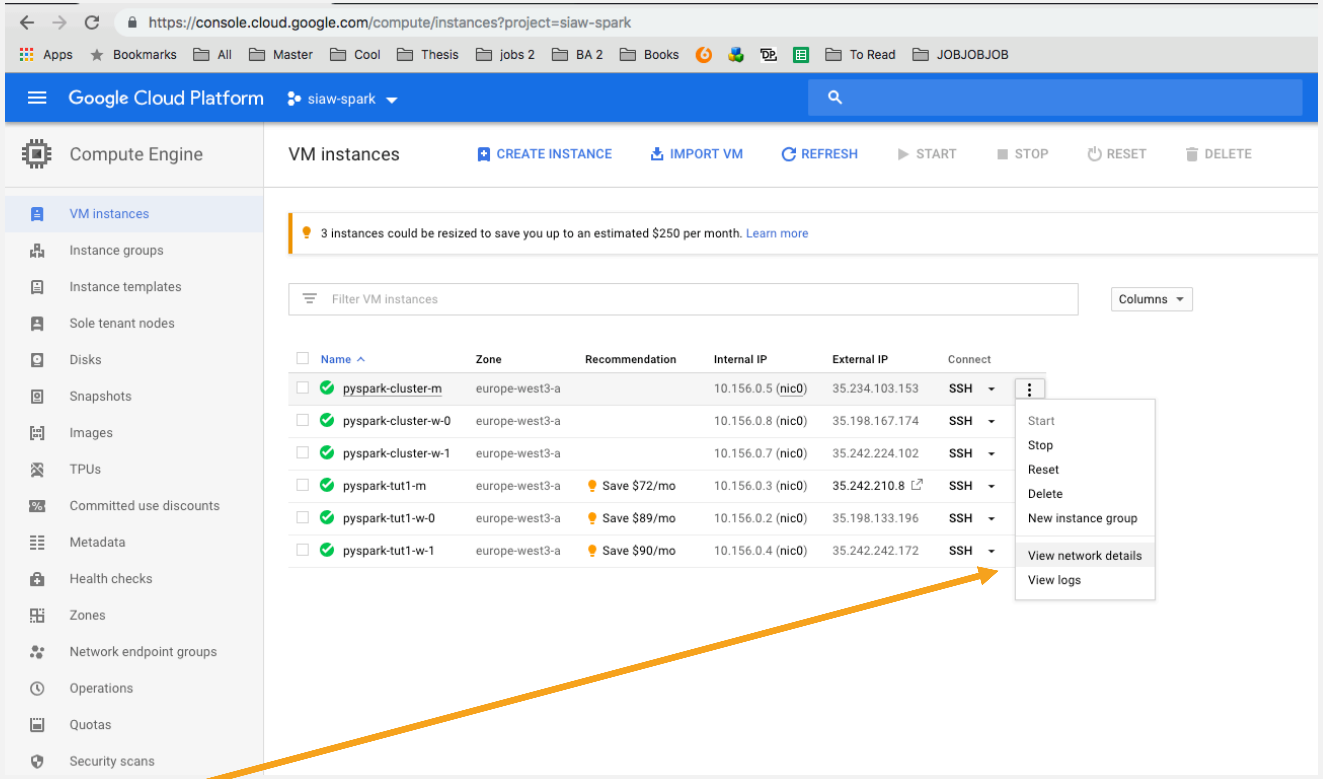
Create Cancel

Equivalent REST or command line

* Depending on which size you choose your cluster to be, you might have to enable billing first. Enabling Billing, however, does not mean that you lose your credit. You will only be charged should you surpass your available credit

3. SETTING UP THE JUPYTER INTERFACE

- Our cluster will a few minutes to be launched. Its status will change to “Running” when ready
- Click on the menu in the top left again and click on “Compute Engine”
- Click on the three vertical dots to the right of your cluster with the suffix “-m” (this is the master node)
- Click on “View Network Details”

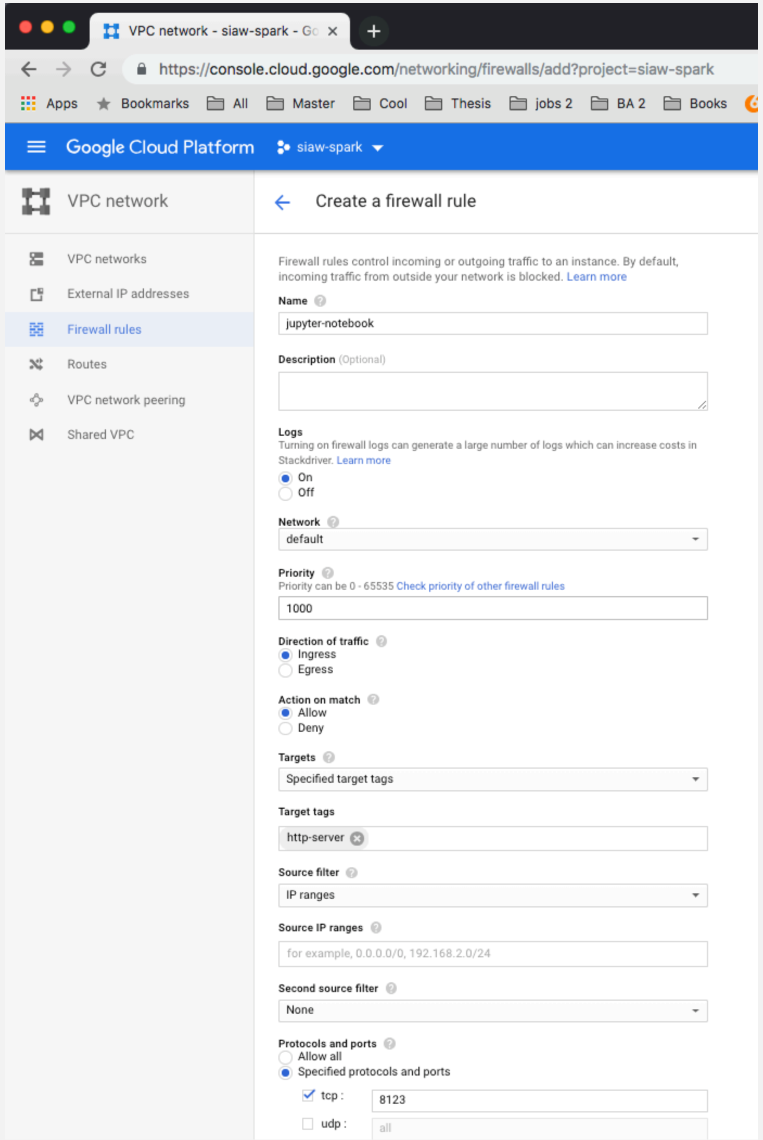


3 instances could be resized to save you up to an estimated $250 per month. Learn more

Name	Zone	Recommendation	Internal IP	External IP	Connect
pyspark-cluster-m	europe-west3-a		10.156.0.5 (nic0)	35.234.103.153	SSH
pyspark-cluster-w-0	europe-west3-a		10.156.0.8 (nic0)	35.198.167.174	SSH
pyspark-cluster-w-1	europe-west3-a		10.156.0.7 (nic0)	35.242.224.102	SSH
pyspark-tut1-m	europe-west3-a	Save $72/mo	10.156.0.3 (nic0)	35.242.210.8	SSH
pyspark-tut1-w-0	europe-west3-a	Save $89/mo	10.156.0.2 (nic0)	35.198.133.196	SSH
pyspark-tut1-w-1	europe-west3-a	Save $90/mo	10.156.0.4 (nic0)	35.242.242.172	SSH

3. SETTING UP THE JUPYTER INTERFACE

- From the left, select "Firewall Rules"
- Select "Create Firewall Rule"
- Use the following settings:
  - Name: jupyter
  - Target tags: http-server
  - Source IP ranges: your v4 IP
  - tcp: 8123 (the script sets up Jupyter on this port)
- Once created, you can use this rule for all future spark clusters
- Click "Create"



Create a firewall rule

Firewall rules control incoming or outgoing traffic to an instance. By default, incoming traffic from outside your network is blocked. Learn more

Name
jupyter-notebook

Description (Optional)

Logs
Turning on firewall logs can generate a large number of logs which can increase costs in Stackdriver. Learn more
● On
○ Off

Network
default

Priority
Priority can be 0 - 65535 Check priority of other firewall rules
1000

Direction of traffic
● Ingress
○ Egress

Action on match
● Allow
○ Deny

Targets
Specified target tags

Target tags
http-server

Source filter
IP ranges

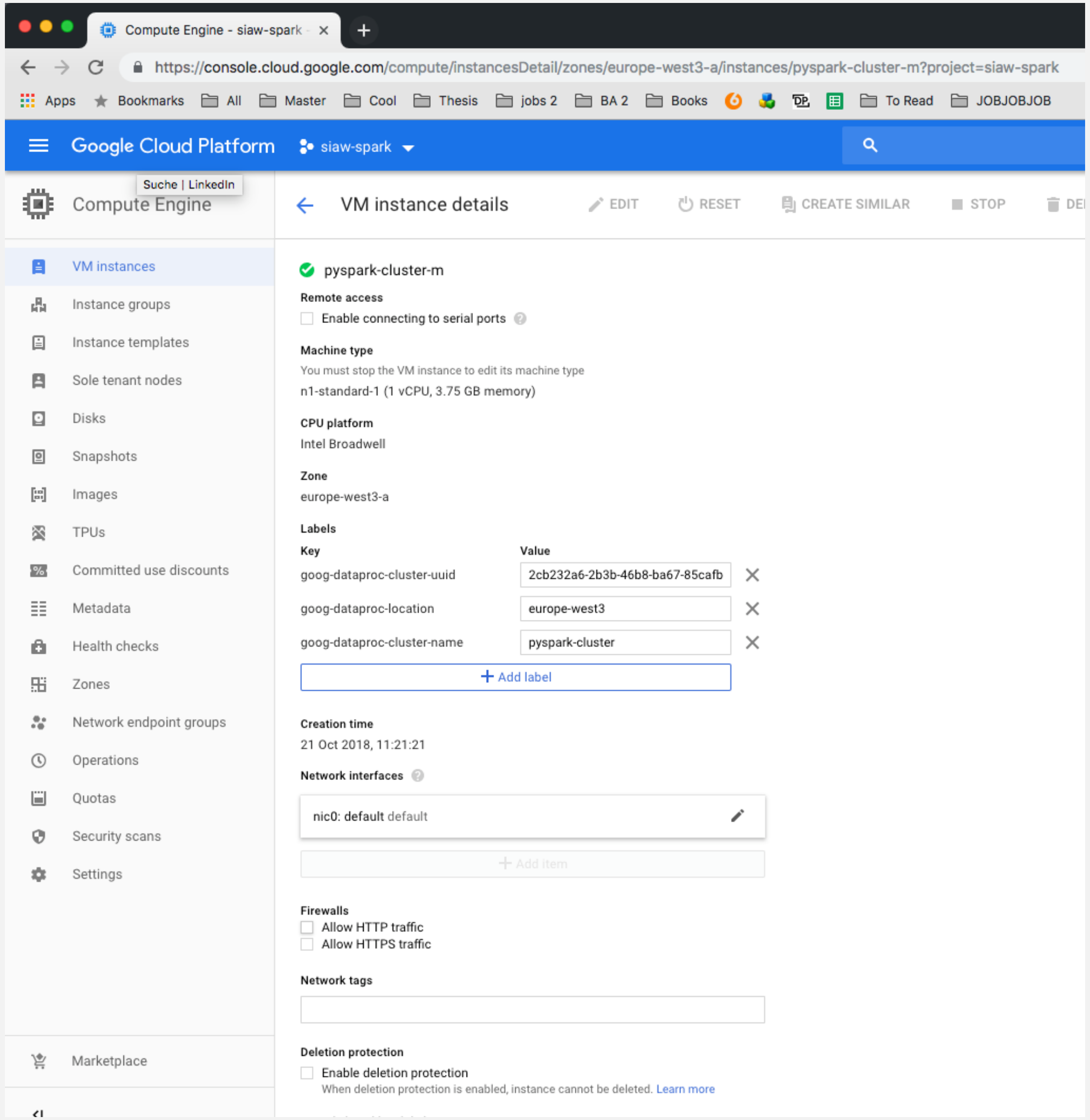
Source IP ranges
for example, 0.0.0.0/0, 192.168.2.0/24

Second source filter
None

Protocols and ports
○ Allow all
● Specified protocols and ports
☑ tcp : 8123
□ udp : all

3. SETTING UP THE JUPYTER INTERFACE

- Go back to the overview of your compute engines and click on your master node
- Click on edit
- Tick the box stating: ”Allow HTTP traffic”
- Click “save”



pyspark-cluster-m

Remote access

□ Enable connecting to serial ports

Machine type

You must stop the VM instance to edit its machine type

n1-standard-1 (1 vCPU, 3.75 GB memory)

CPU platform

Intel Broadwell

Zone

europe-west3-a

Labels

Key	Value
goog-dataproc-cluster-uuid	2cb232a6-2b3b-46b8-ba67-85cafb
goog-dataproc-location	europe-west3
goog-dataproc-cluster-name	pyspark-cluster

Creation time

21 Oct 2018, 11:21:21

Network interfaces

nic0: default default

Firewalls

□ Allow HTTP traffic

□ Allow HTTPS traffic

Network tags

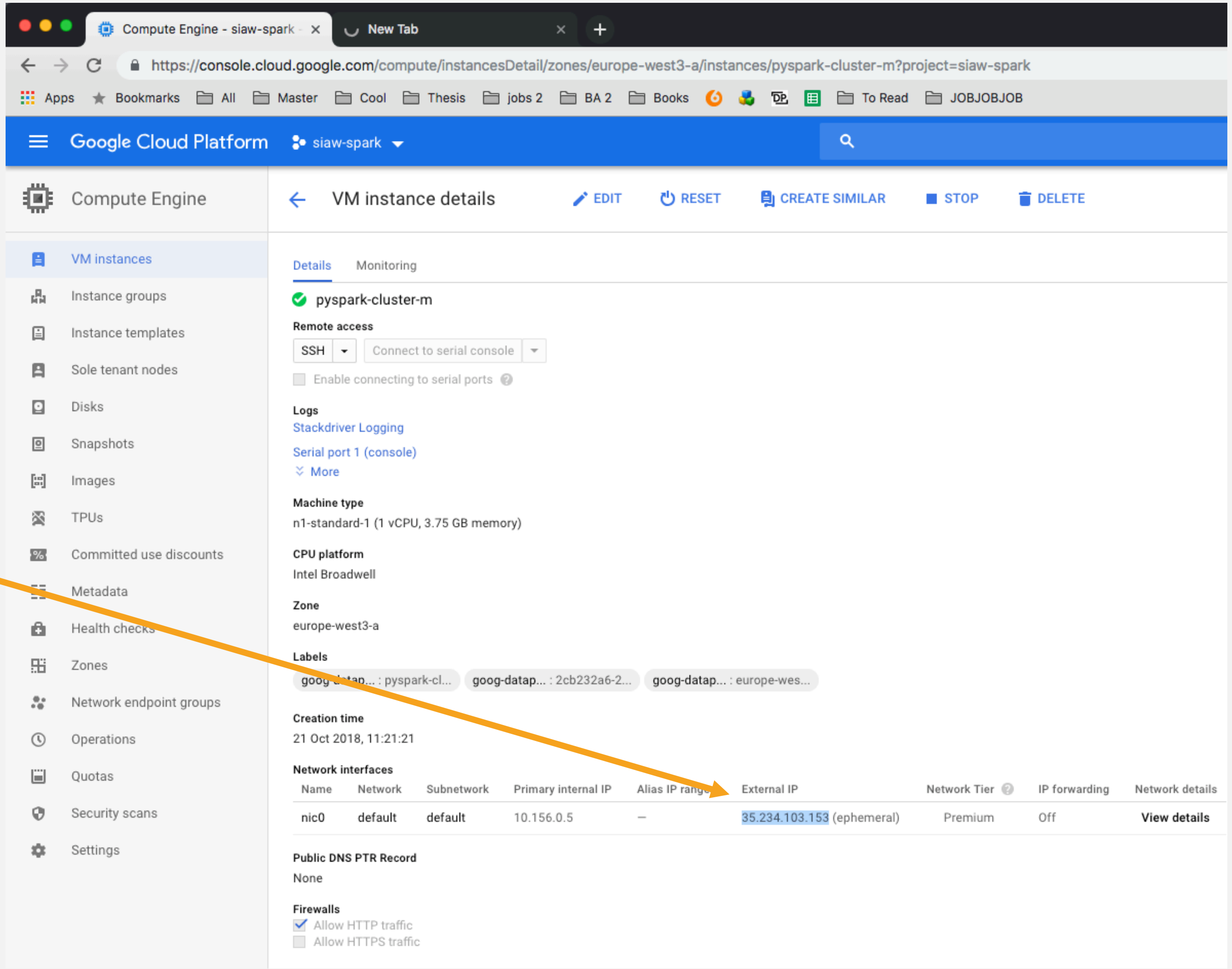
Deletion protection

□ Enable deletion protection

When deletion protection is enabled, instance cannot be deleted.

3. SETTING UP THE JUPYTER INTERFACE

- Go back to the overview of your master node and copy your master nodes IP address into your browser
- add “:8123”
- This is port where the jupyter notebook is available
- Hit enter



pyspark-cluster-m

Remote access

SSH ▾ Connect to serial console ▾

□ Enable connecting to serial ports

Logs

Stackdriver Logging

Serial port 1 (console)

More

Machine type

n1-standard-1 (1 vCPU, 3.75 GB memory)

CPU platform

Intel Broadwell

Zone

europe-west3-a

Labels

goog-datap... : pyspark-cl... goog-datap... : 2cb232a6-2... goog-datap... : europe-wes...

Creation time

21 Oct 2018, 11:21:21

Network interfaces

Name	Network	Subnetwork	Primary internal IP	Alias IP ranges	External IP	Network Tier	IP forwarding	Network details
nic0	default	default	10.156.0.5	–	35.234.103.153 (ephemeral)	Premium	Off	View details

Public DNS PTR Record

None

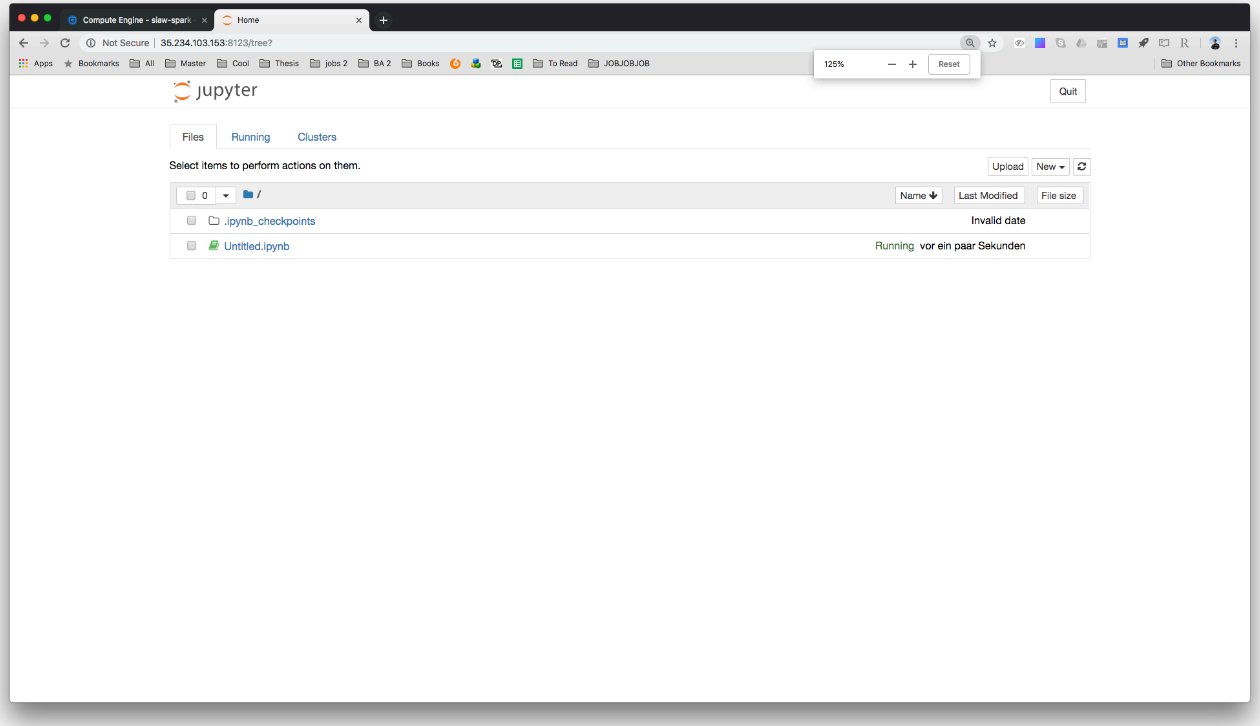
Firewalls

☑ Allow HTTP traffic

□ Allow HTTPS traffic

4. DONE

- You can now use a regular Jupyter Notebook interface for working with PySpark
- Note: even though not covered specifically your your cluster comes with a storage bucket upon instantiation. You can upload data into this bucket and load it in your notebook

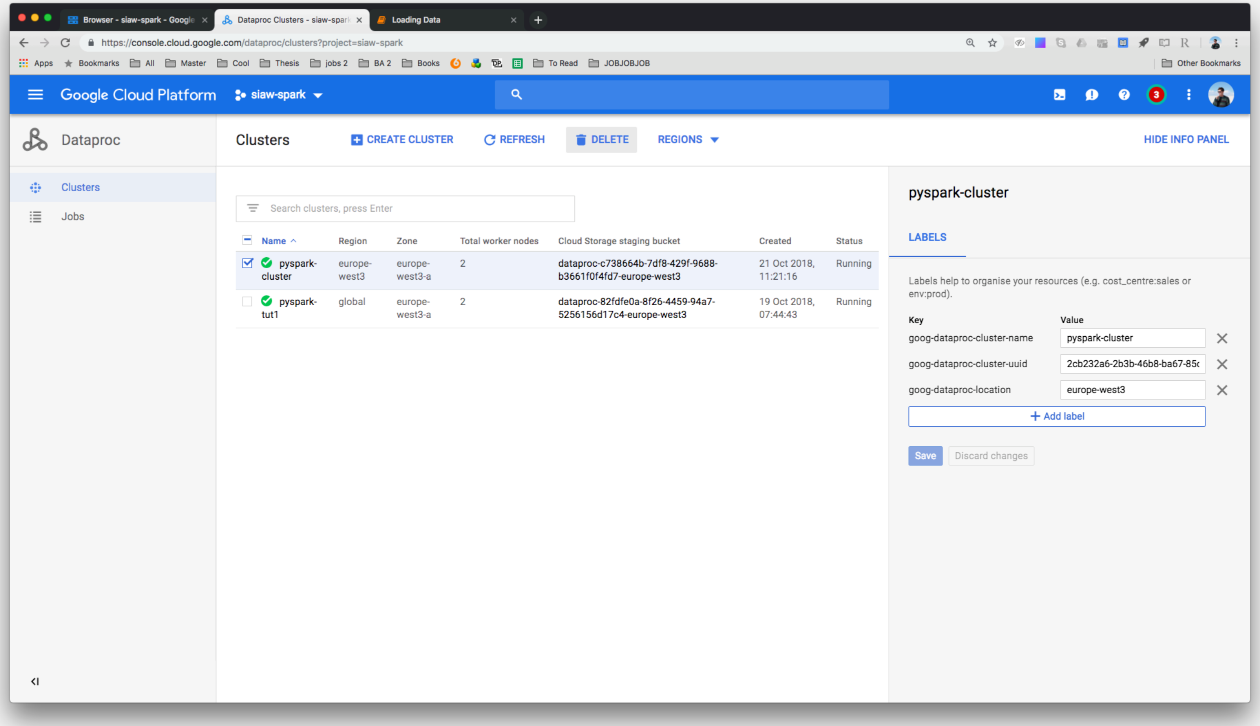


Select items to perform actions on them.

Name	Last Modified
.ipynb_checkpoints	Invalid date
Untitled.ipynb	Running vor ein paar Sekunden

4. DELETION

- Don’t remember to use your cluster when finished to avoid incurring high costs despite not using it.
- Open the menu and click on the Dataproc item
- Select your cluster on the left and press ”Delete” in the top menu
- Confirm



Clusters

	Name	Region	Zone	Total worker nodes	Cloud Storage staging bucket	Created	Status
☑	pyspark-cluster	europe-west3	europe-west3-a	2	dataproc-c738664b-7df8-429f-9688-b3661f0f4fd7-europe-west3	21 Oct 2018, 11:21:16	Running
□	pyspark-tut1	global	europe-west3-a	2	dataproc-82fdfe0a-8f26-4459-94a7-5256156d17c4-europe-west3	19 Oct 2018, 07:44:43	Running

pyspark-cluster

LABELS

Labels help to organise your resources (e.g. cost_centre:sales or env:prod).

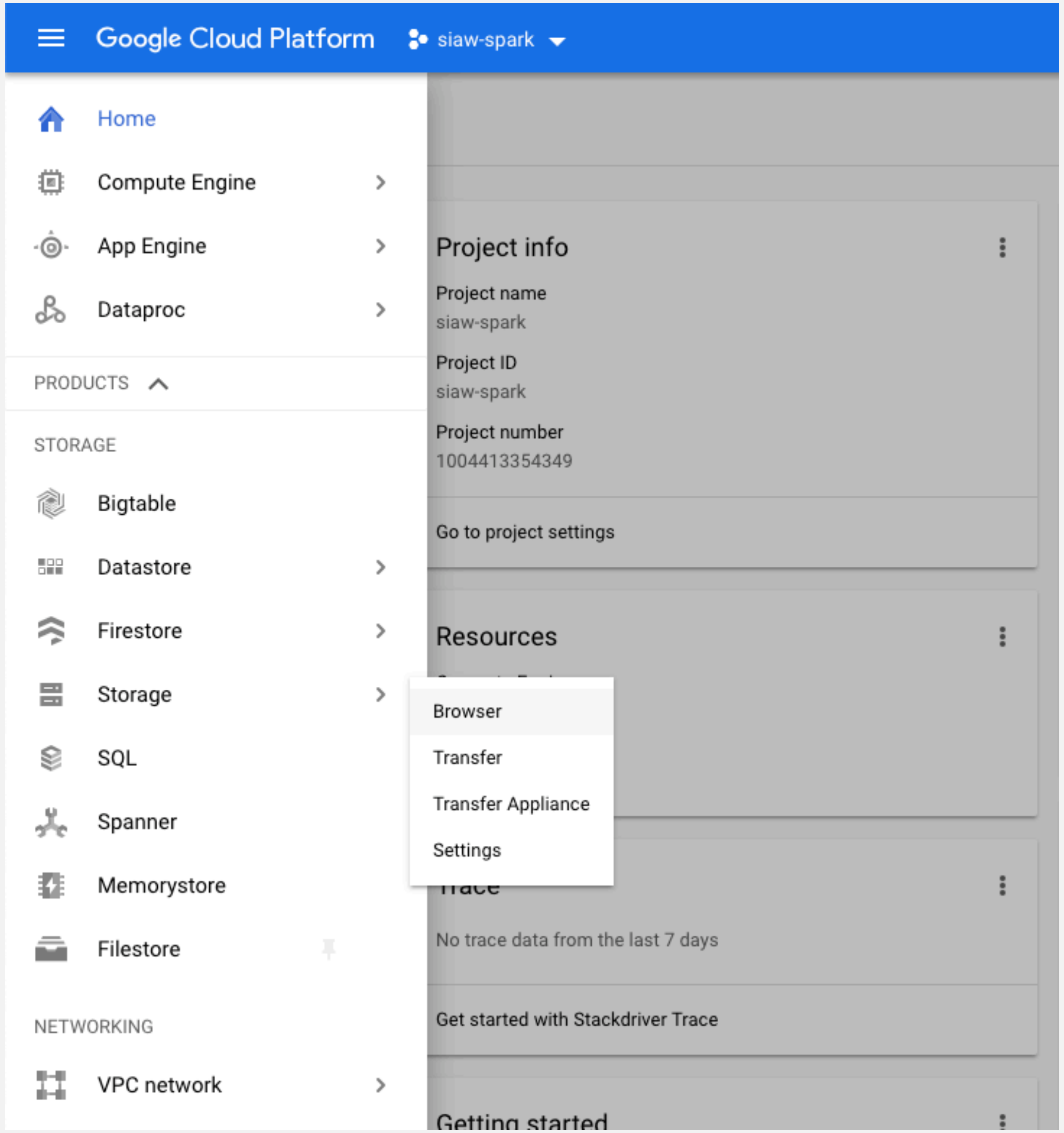
Key	Value
goog-dataproc-cluster-name	pyspark-cluster
goog-dataproc-cluster-uuid	2cb232a6-2b3b-46b8-ba67-85(
goog-dataproc-location	europe-west3

UPLOADING DATA TO GCP

This section will explain how to upload data into a Google Cloud Platform storage bucket and read it from a PySpark script

1. UPLOADING DATA

- Open your GCP console, navigate to the hamburger menu and click on "Storage"



Project info

Project name
siaw-spark

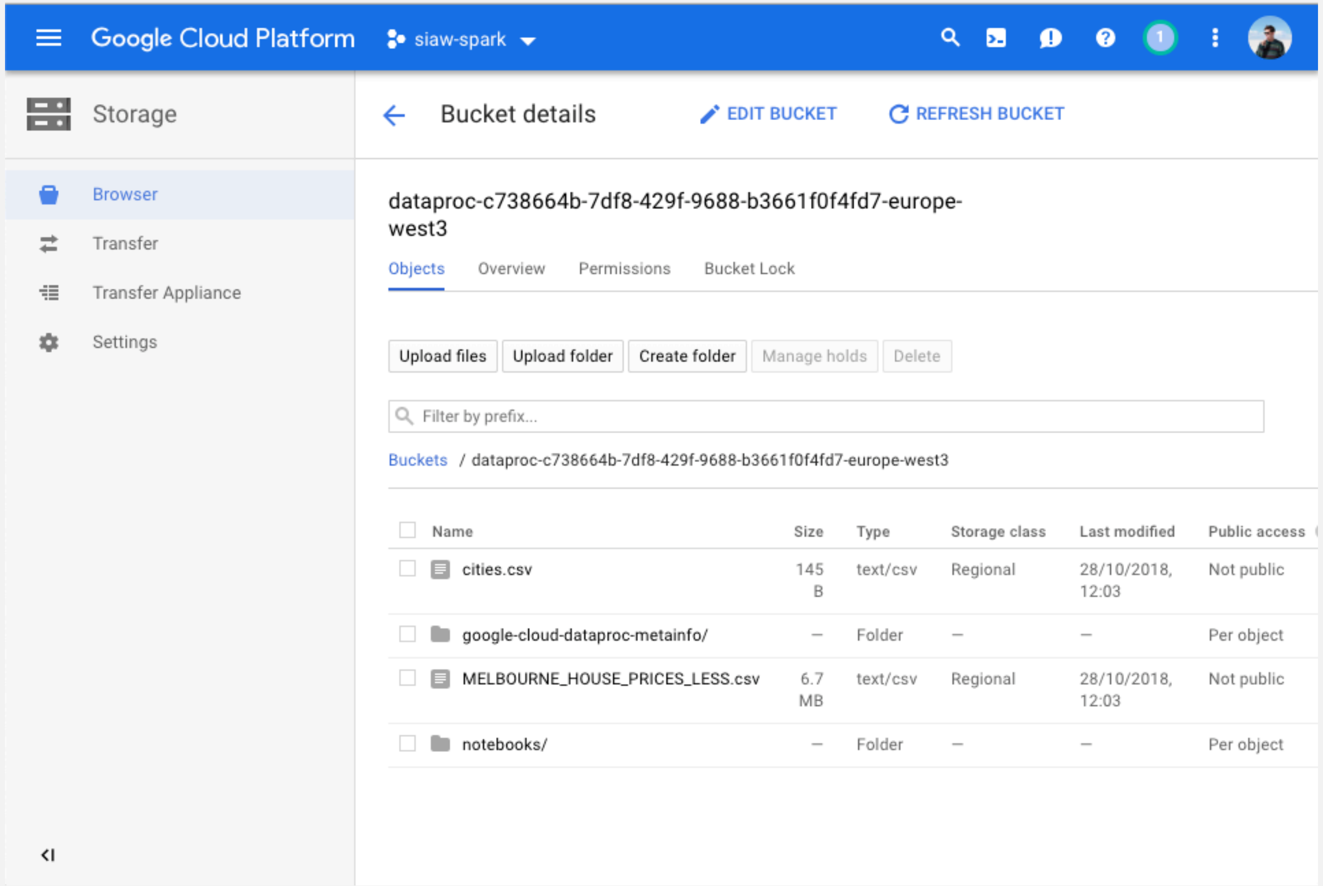
Project ID
siaw-spark

Project number
1004413354349

No trace data from the last 7 days

1. UPLOADING DATA

- You should be seeing a storage bucket that was automatically created when you set up your Spark Cluster. Click on it.
- As of now, only metafiles and a folder for Jupyter notebooks should be living in your notebook.
- Click on "Upload Files" in the top menu and select your file(s) to upload. In our case we will use a csv containing information on Avocado sales in the US



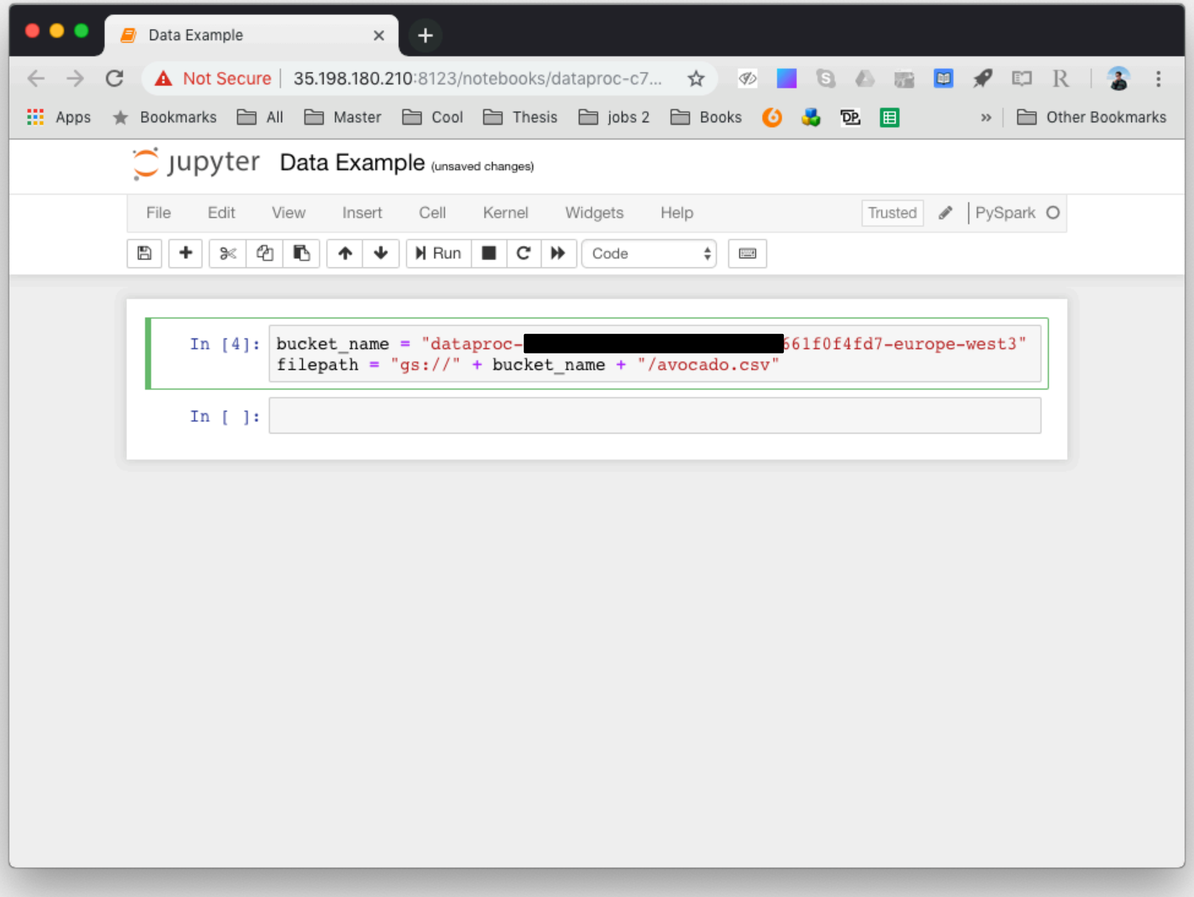
dataproc-c738664b-7df8-429f-9688-b3661f0f4fd7-europe-west3

Buckets / dataproc-c738664b-7df8-429f-9688-b3661f0f4fd7-europe-west3

Name	Size	Type	Storage class	Last modified	Public access
cities.csv	145 B	text/csv	Regional	28/10/2018, 12:03	Not public
google-cloud-dataproc-metainfo/	–	Folder	–	–	Per object
MELBOURNE_HOUSE_PRICES_LESS.csv	6.7 MB	text/csv	Regional	28/10/2018, 12:03	Not public
notebooks/	–	Folder	–	–	Per object

2. CALLING THE DATA FROM A NOTEBOOK

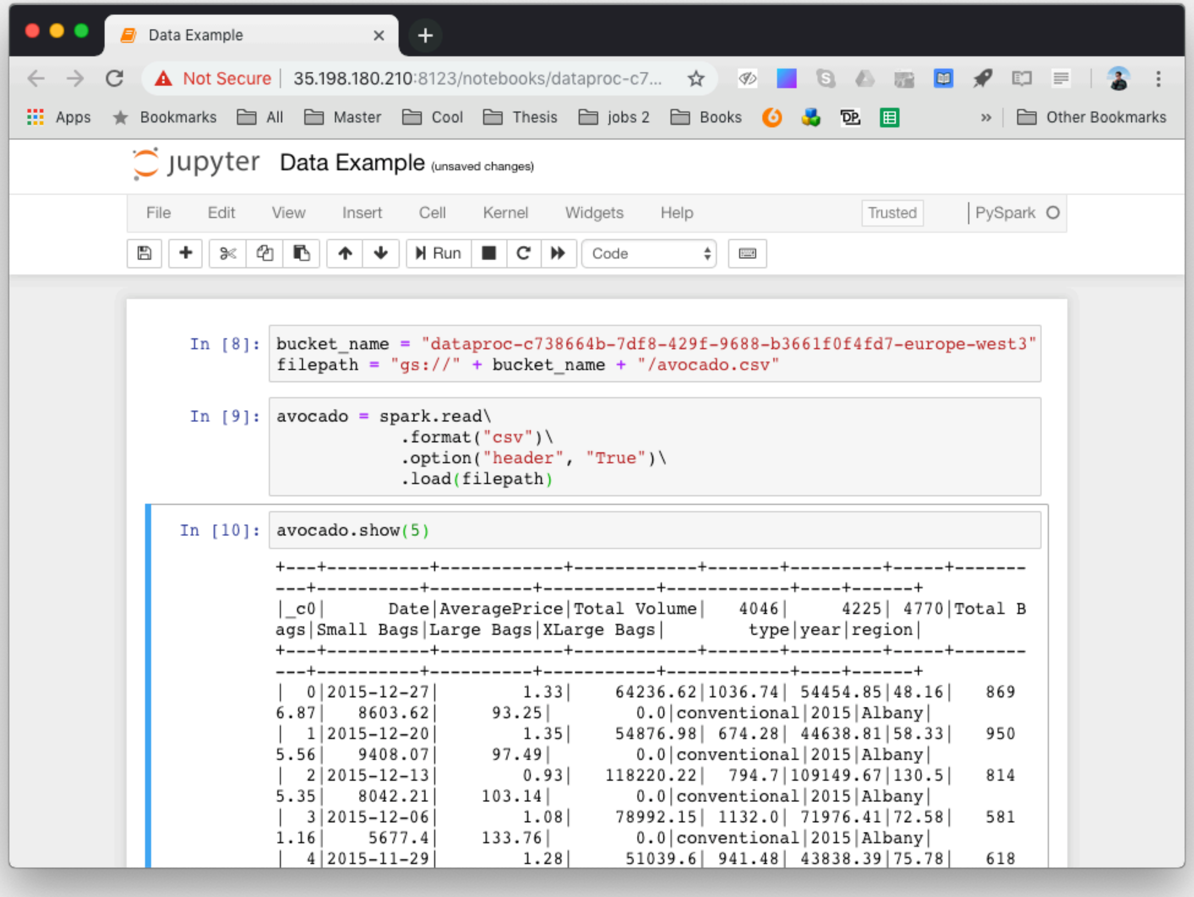
- Open a new PySpark Jupyter notebook
- You can access your data under the server file path which consists of the prefix ”gs://”, your server name and the path of the file in the bucket
- You can copy your server name from the bucket menu in the “Uploading Data” section



```
bucket_name = "dataproc-[redacted]61f0f4fd7-europe-west3"
filepath = "gs://" + bucket_name + "/avocado.csv"
```

2. CALLING THE DATA FROM A NOTEBOOK

- You can now access the files at the specified file path with the PySpark commands and use it as intended.



```
In [8]: bucket_name = "dataproc-c738664b-7df8-429f-9688-b3661f0f4fd7-europe-west3"
        filepath = "gs://" + bucket_name + "/avocado.csv"
```

```
In [9]: avocado = spark.read\
                    .format("csv")\
                    .option("header", "True")\
                    .load(filepath)
```

```
In [10]: avocado.show(5)
```

```
+---+----------+------------+------------+-------+---------+-----+-------
---+----------+----------+-----------+-----------+----+------+
|_c0|      Date|AveragePrice|Total Volume|   4046|     4225| 4770|Total B
ags|Small Bags|Large Bags|XLarge Bags|       type|year|region|
+---+----------+------------+------------+-------+---------+-----+-------
---+----------+----------+-----------+-----------+----+------+
|  0|2015-12-27|        1.33|    64236.62|1036.74| 54454.85|48.16|    869
6.87|   8603.62|     93.25|        0.0|conventional|2015|Albany|
|  1|2015-12-20|        1.35|    54876.98| 674.28| 44638.81|58.33|    950
5.56|   9408.07|     97.49|        0.0|conventional|2015|Albany|
|  2|2015-12-13|        0.93|   118220.22|  794.7|109149.67|130.5|    814
5.35|   8042.21|    103.14|        0.0|conventional|2015|Albany|
|  3|2015-12-06|        1.08|    78992.15| 1132.0| 71976.41|72.58|    581
1.16|    5677.4|    133.76|        0.0|conventional|2015|Albany|
|  4|2015-11-29|        1.28|     51039.6| 941.48| 43838.39|75.78|    618
```