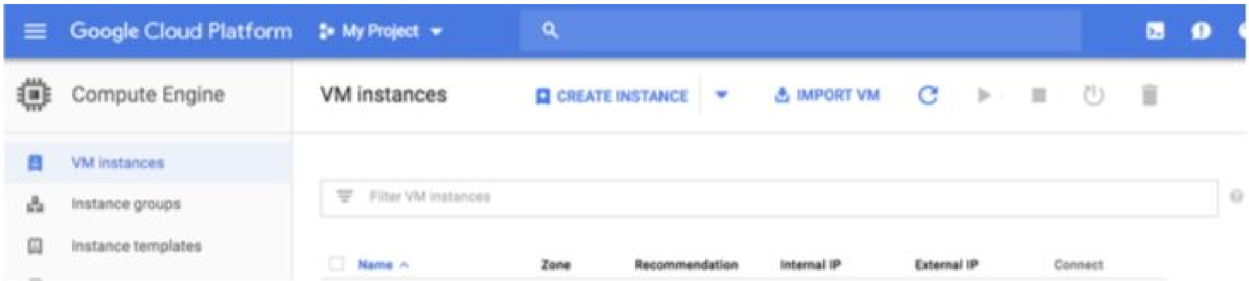
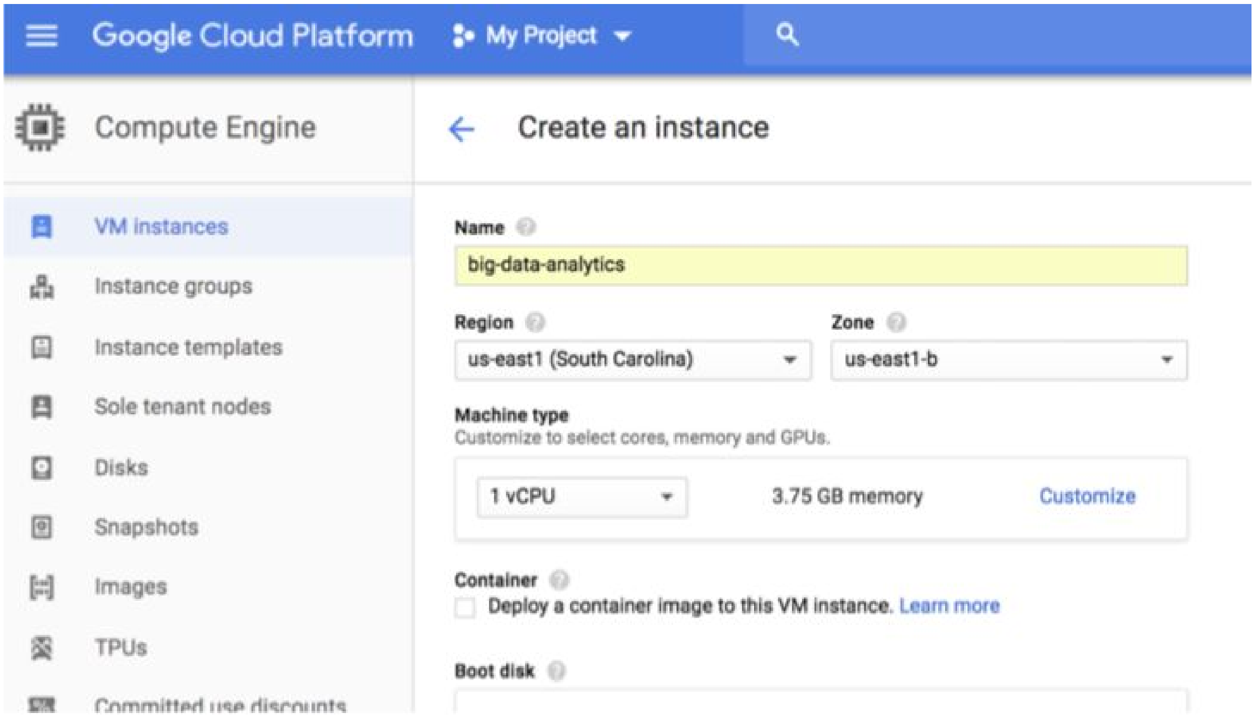
Tutorial for Beginners

1. Google Cloud Setup

Head over to https://console.cloud.google.com/
Everyone could use the free $300 Google credits valid for one year from signing up on Google Cloud.





Name: big-data-analytics

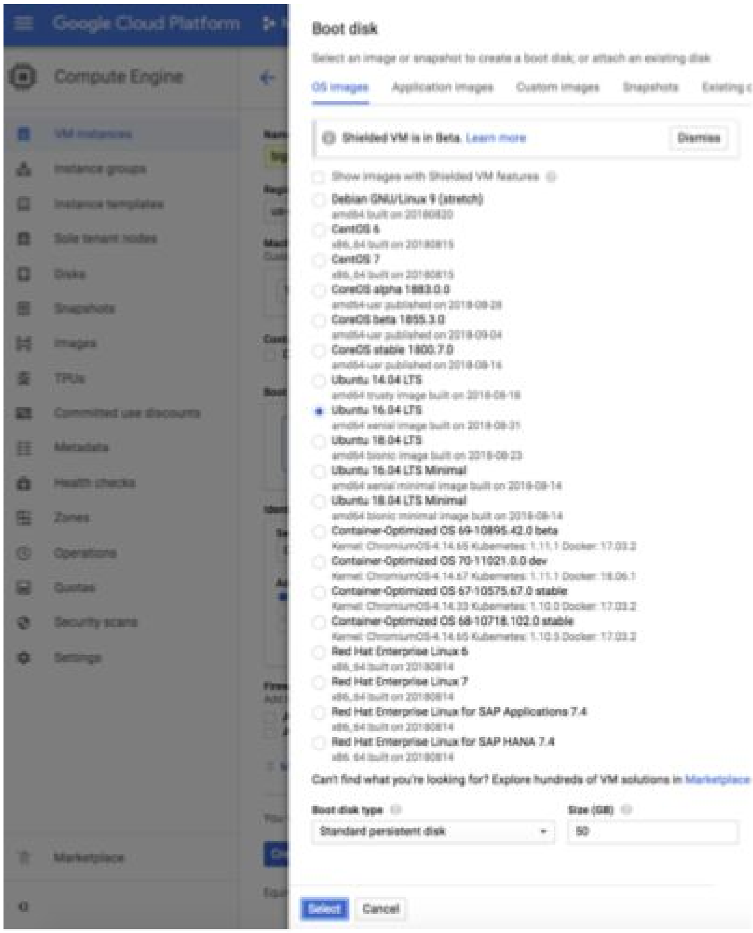
Region: us-east1 (South Carolina)

Zone: us-east1-b

Machine type
Customize to select cores, memory and GPUs.
1 vCPU 3.75 GB memory

Container
□ Deploy a container image to this VM instance. Learn more

Boot disk



Boot disk

Select an image or snapshot to create a boot disk; or attach an existing disk

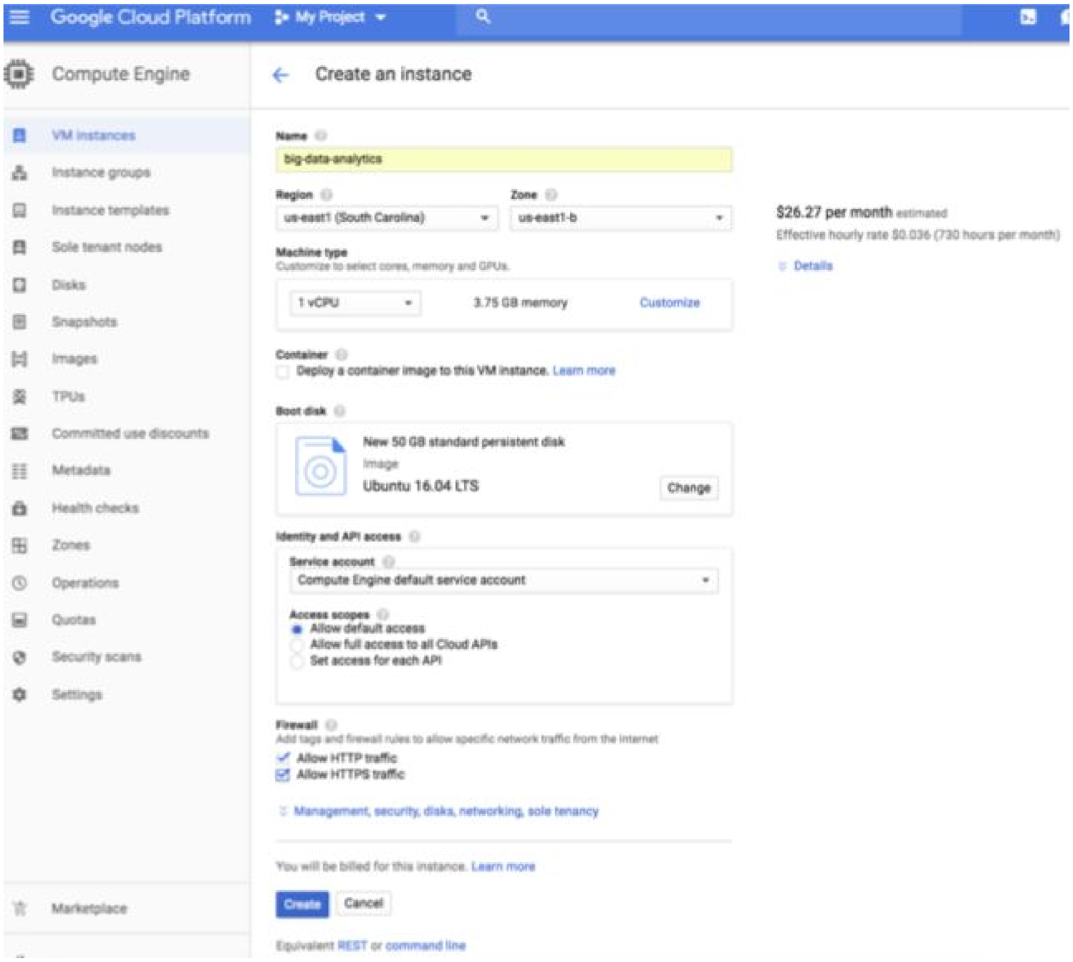
Shielded VM is in Beta. Learn more

□ Show images with Shielded VM features

○ Debian GNU/Linux 9 (stretch)
amd64 built on 20180820
○ CentOS 6
x86_64 built on 20180815
○ CentOS 7
x86_64 built on 20180815
○ CoreOS alpha 1883.0.0
amd64-usr published on 2018-08-28
○ CoreOS beta 1855.3.0
amd64-usr published on 2018-09-04
○ CoreOS stable 1800.7.0
amd64-usr published on 2018-08-16
○ Ubuntu 14.04 LTS
amd64 trusty image built on 2018-08-18
● Ubuntu 16.04 LTS
amd64 xenial image built on 2018-08-31
○ Ubuntu 18.04 LTS
amd64 bionic image built on 2018-08-23
○ Ubuntu 16.04 LTS Minimal
amd64 xenial minimal image built on 2018-08-14
○ Ubuntu 18.04 LTS Minimal
amd64 bionic minimal image built on 2018-08-14
○ Container-Optimized OS 69-10895.42.0 beta
Kernel: ChromiumOS-4.14.65 Kubernetes: 1.11.1 Docker: 17.03.2
○ Container-Optimized OS 70-11021.0.0 dev
Kernel: ChromiumOS-4.14.67 Kubernetes: 1.11.1 Docker: 18.06.1
○ Container-Optimized OS 67-10575.67.0 stable
Kernel: ChromiumOS-4.14.33 Kubernetes: 1.10.0 Docker: 17.03.2
○ Container-Optimized OS 68-10718.102.0 stable
Kernel: ChromiumOS-4.14.65 Kubernetes: 1.10.5 Docker: 17.03.2
○ Red Hat Enterprise Linux 6
x86_64 built on 20180814
○ Red Hat Enterprise Linux 7
x86_64 built on 20180814
○ Red Hat Enterprise Linux for SAP Applications 7.4
x86_64 built on 20180814
○ Red Hat Enterprise Linux for SAP HANA 7.4
x86_64 built on 20180814

Can't find what you're looking for? Explore hundreds of VM solutions in Marketplace

Boot disk type: Standard persistent disk
Size (GB): 50



Create an instance

Name: big-data-analytics

Region: us-east1 (South Carolina)
Zone: us-east1-b

$26.27 per month estimated
Effective hourly rate $0.036 (730 hours per month)

Machine type
Customize to select cores, memory and GPUs.
1 vCPU 3.75 GB memory

Container
□ Deploy a container image to this VM instance. Learn more

Boot disk
New 50 GB standard persistent disk
Image
Ubuntu 16.04 LTS

Identity and API access
Service account: Compute Engine default service account
Access scopes
● Allow default access
○ Allow full access to all Cloud APIs
○ Set access for each API

Firewall
Add tags and firewall rules to allow specific network traffic from the Internet
☑ Allow HTTP traffic
☑ Allow HTTPS traffic

Management, security, disks, networking, sole tenancy

You will be billed for this instance. Learn more

Equivalent REST or command line

2. Installations

2.1 Install JDK, JRE

```
sudo apt-get update
sudo apt-get install openjdk-8-jdk-headless default-jre ssh rsync readlink -f
/usr/bin/java | sed "s:bin/java::"
```

OUTPUT for last command: /usr/lib/jvm/java-8-openjdk-amd64/jre/

2.2 Install Hadoop

Firstly, you need to find out your username using

```
whoami
```

Then, download it using

```
wget http://www.eu.apache.org/dist/hadoop/common/stable/hadoop-2.9.2.tar.gz tar -xvzf
hadoop-2.9.2.tar.gz
mv hadoop-2.9.2 hadoop
```

(There may be more advanced version, like 2.9.3, so you would need to verify it by yourself.)
Then, edit the configurations using

```
vim ./.bashrc
```

Put these at the end of the file (To insert, type "i" for insert.)

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64 export PATH=$PATH:$HADOOP_PREFIX/bin
```

Now save the file with <esc> key followed by typing ":wq" for write-quit

```
vim ./hadoop/etc/hadoop/hadoop-env.sh
```

Then, comment the line "export JAVA_HOME=${JAVA_HOME}" and replace with these lines:

```
#export JAVA_HOME=${JAVA_HOME}
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
```

```
vim ./hadoop/etc/hadoop/core-site.xml
```

Put these lines inside <configuration></configuration>

```
<property>
<name>fs.default.name</name>
<value>hdfs://localhost:1234</value>
</property>
```

```
vim ./hadoop/etc/hadoop/hdfs-site.xml
```

Put these lines inside <configuration></configuration>

```
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
```

```
cp ./hadoop/etc/hadoop/mapred-site.xml.template ./hadoop/etc/hadoop/mapred-site.xml
vim ./hadoop/etc/hadoop/mapred-site.xml
```

Now Put these lines inside <configuration></configuration>

```
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
```

```
vim ./hadoop/etc/hadoop/yarn-site.xml
```

Put these lines inside <configuration></configuration>

```
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
```

Next line has two single quotes (take caution on typesetting).

```
ssh-keygen -t rsa -P ‘’ -f ./.ssh/id_rsa
cat ./.ssh/id_rsa.pub >> ./.ssh/authorized_keys
```

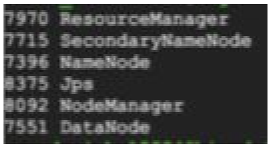
Execute "ssh localhost" and enter "yes" when prompted. And then enter "exit" Followed by this, execute:

```
source ./.bashrc
```

Start Hadoop (first time execution)
Remember to execute these commands in order (even if they may seem duplicated commands):

```
cd ./hadoop
./sbin/start-dfs.sh
./bin/hdfs namenode -format
./sbin/stop-dfs.sh
./sbin/start-dfs.sh
./bin/hdfs dfs -mkdir /user
./bin/hdfs dfs -mkdir /user/your_username
./sbin/start-yarn.sh
jps
cd ../
```

After jps, it should show like this:



```
7970 ResourceManager
7715 SecondaryNameNode
7396 NameNode
8375 Jps
8092 NodeManager
7551 DataNode
```

Sample Hadoop Examples

```
./hadoop/bin/hadoop jar
./hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.9.1.jar pi 10 100
```

Note, above is one line code.

2.3 Setup HBase

Downloads and Edit Configuration

```
wget http://www-eu.apache.org/dist/hbase/stable/hbase-1.4.7-bin.tar.gz tar -zxvf
hbase-1.4.7-bin.tar.gz
mv hbase-1.4.7 hbase
vim ./.bashrc
```

Add to bashrc file:

```
export HBASE_HOME=/home/your_username/hbase
export PATH=$PATH:$HBASE_HOME/bin
```

Save the file and execute this:

```
source ./.bashrc
```

```
vim ./hbase/conf/hbase-env.sh
```

Add to file:

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
```

```
vim ./hbase/conf/hbase-site.xml
```

```
<property>
<name>hbase.rootdir</name>
<value>hdfs://localhost:1234/hbase</value>
</property>
<property>
<name>hbase.zookeeper.property.dataDir</name>
<value>/home/your_username/zookeeper</value>
</property>
<property>
<name>hbase.cluster.distributed</name>
<value>true</value>
</property>
```

Start HBase

```
start-hbase.sh ./hadoop/bin/hdfs dfs -ls /hbase
```

If HBase setup is successful, a folder /hbase would be created and will be listed with the last command.

2.4 Install Hive

Downloads and Edit Configuration

```
wget https://archive.apache.org/dist/hive/stable/apache-hive-1.2.2-bin.tar.gz
tar -xvzf apache-hive-1.2.2-bin.tar.gz
mv apache-hive-1.2.2-bin hive
```

```
vim ./.bashrc
```

Add to bashrc file:

```
export HIVE_HOME=/home/your_username/hive export PATH=$PATH:$HIVE_HOME/bin
```

Save the file and execute this:

```
source ./.bashrc
```

```
cp ./hive/conf/hive-env.sh.template ./hive/conf/hive-env.sh
vim ./hive/conf/hive-env.sh
```

Add to file:

```
HADOOP_HOME=/home/your_username/hadoop
```

Start Hive

```
hive
exit;
```

2.5 Install spark

Downloads and Edit Configuration

```
wget https://archive.apache.org/dist/spark/spark-2.3.1/spark-2.3.1-bin-hadoop2.7.tgz
tar -xvzf spark-2.3.1-bin-hadoop2.7.tgz
mv spark-2.3.1-bin-hadoop2.7 spark
vim ./.bashrc
```

```
export SPARK_HOME=/home/your_username/spark
export PATH=$SPARK_HOME/bin:$PATH
```

```
source ./.bashrc
```

Start Spark

```
spark-shell
:quit
```

2.6 Install Jupyter notebook

Downloads and Execution

```
wget https://repo.continuum.io/archive/Anaconda3-5.1.0-Linux-x86_64.sh
bash Anaconda3-5.1.0-Linux-x86_64.sh
```

Accept terms: yes
Install folder: <ENTER>
PATH in .bashrc: yes
VSCode: no

```
source ./.bashrc
cd ./anaconda3/bin/
jupyter notebook --generate-config
cd ..
vim ./.jupyter/jupyter_notebook_config.py
```

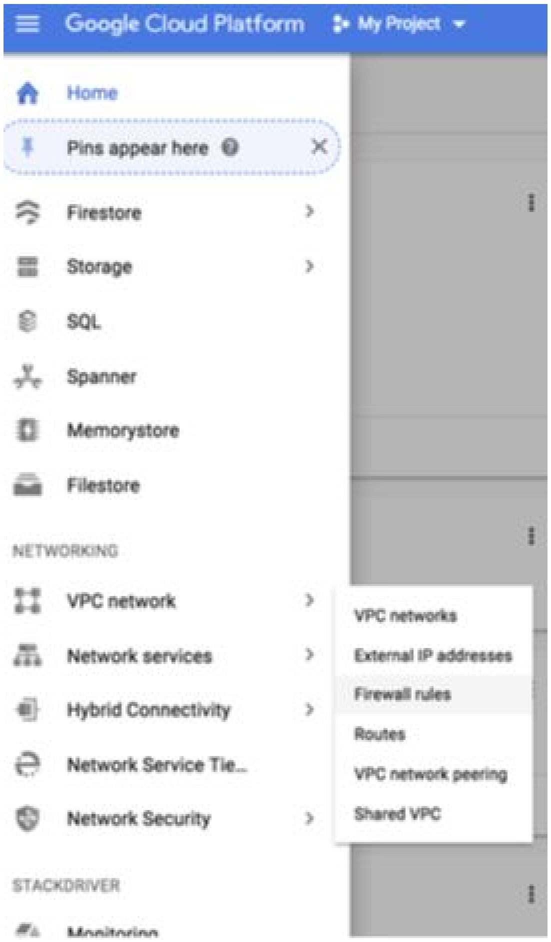
Add to the beginning of file:

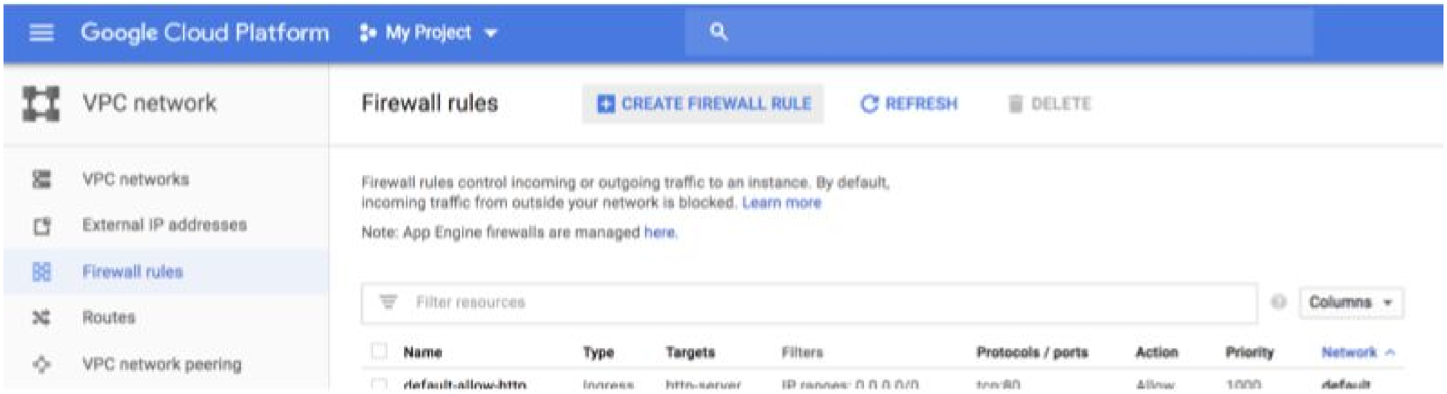
```
c = get_config()
c.NotebookApp.ip = '*'
c.NotebookApp.open_browser = False
c.NotebookApp.port = 5000
```

```
source ./.bashrc
tmux new -s jupyter
jupyter notebook
```

Before launching jupyter notebook, you should open the port on google cloud using step 3.
Then, copy the url shown and replace "localhost" with the IP on google cloud landing page and open this in browser.
To come out of a "tmux" session, use: Ctrl and B together, and then leaving the two keys, press D,
i.e. Ctrl+B, D.
To go back into the session, use: "tmux a -t jupyter"

3. Open Port in Google Cloud

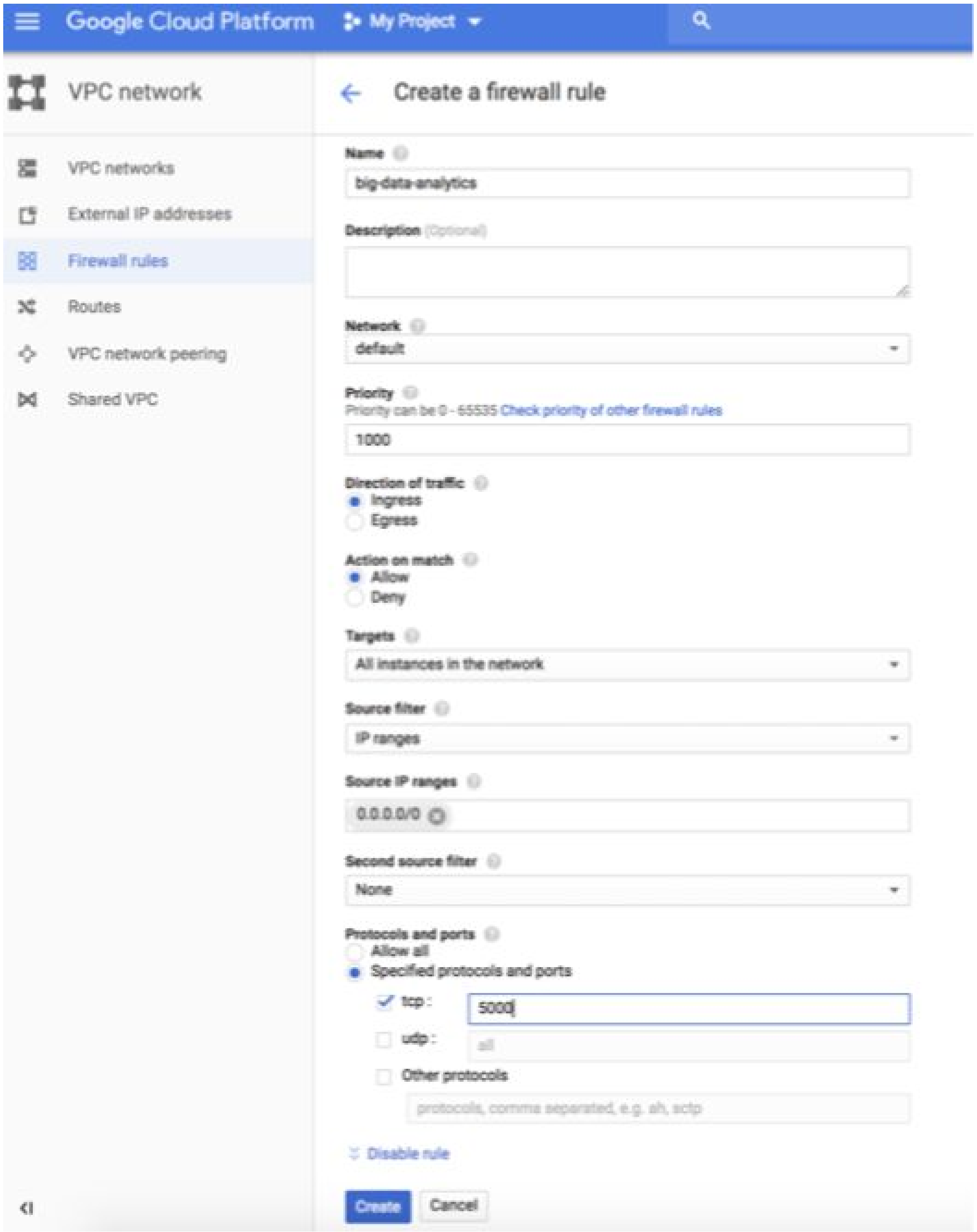




Firewall rules control incoming or outgoing traffic to an instance. By default, incoming traffic from outside your network is blocked. Learn more

Note: App Engine firewalls are managed here.

Name	Type	Targets	Filters	Protocols / ports	Action	Priority	Network
default-allow-http	ingress	http-server	IP ranges: 0.0.0.0/0	tcp:80	Allow	1000	default



Create a firewall rule

Name: big-data-analytics

Description (Optional)

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: All instances in the network

Source filter: IP ranges

Source IP ranges: 0.0.0.0/0

Second source filter: None

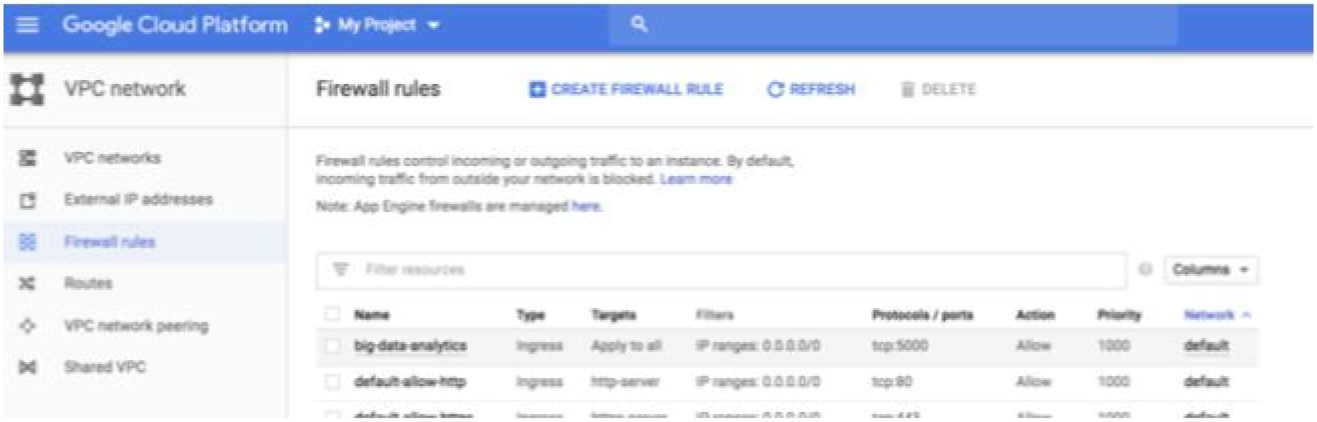
Protocols and ports
○ Allow all
● Specified protocols and ports

☑ tcp : 5000

□ udp : all

□ Other protocols

protocols, comma separated, e.g. ah, sctp



Firewall rules

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

Note: App Engine firewalls are managed here.

Name	Type	Targets	Filters	Protocols / ports	Action	Priority	Network
big-data-analytics	Ingress	Apply to all	IP ranges: 0.0.0.0/0	tcp:5000	Allow	1000	default
default-allow-http	Ingress	http-server	IP ranges: 0.0.0.0/0	tcp:80	Allow	1000	default
default-allow-https	Ingress	https-server	IP ranges: 0.0.0.0/0	tcp:443	Allow	1000	default