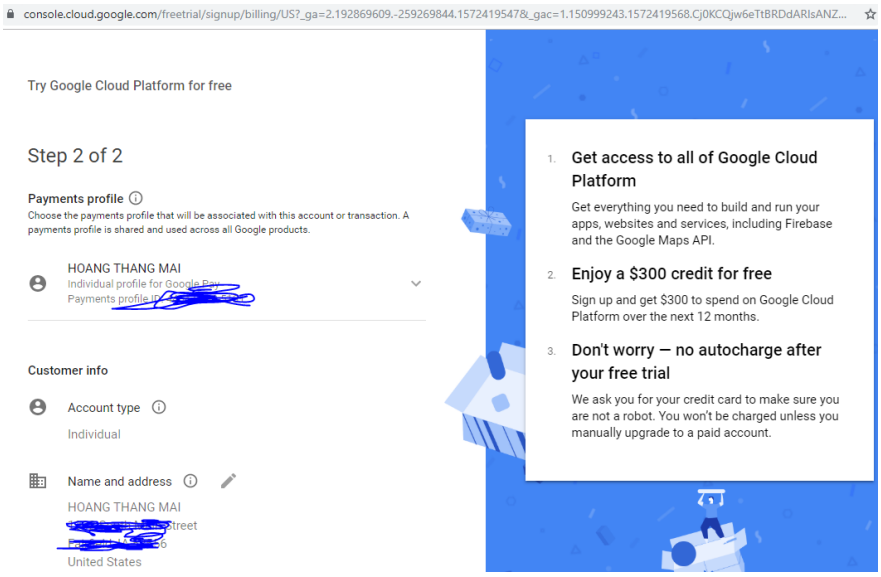
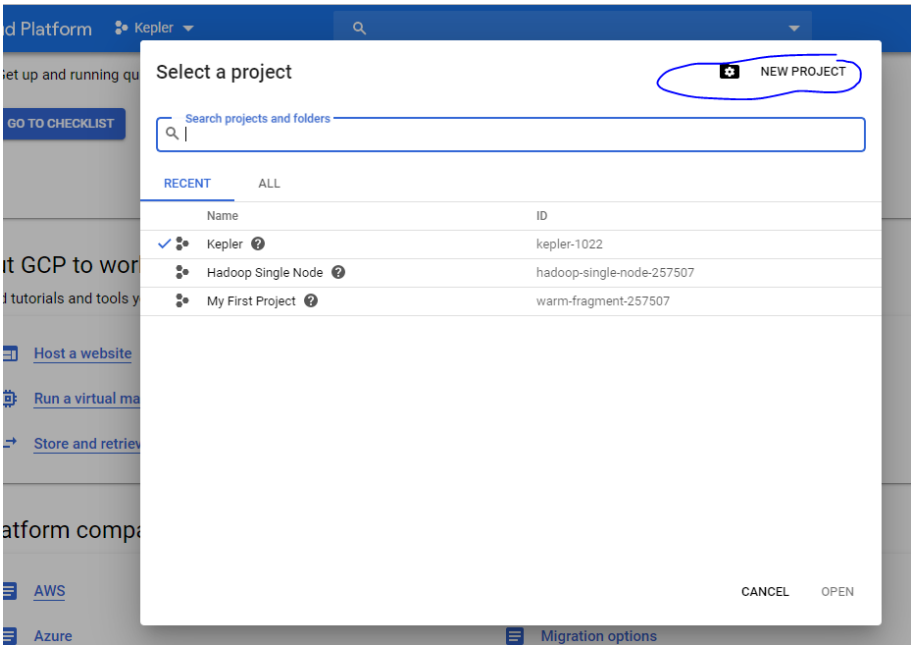
# Set Up a Single Node Cluster using Google Cloud Instance

## Step 1: Access into https://cloud.google.com to register account

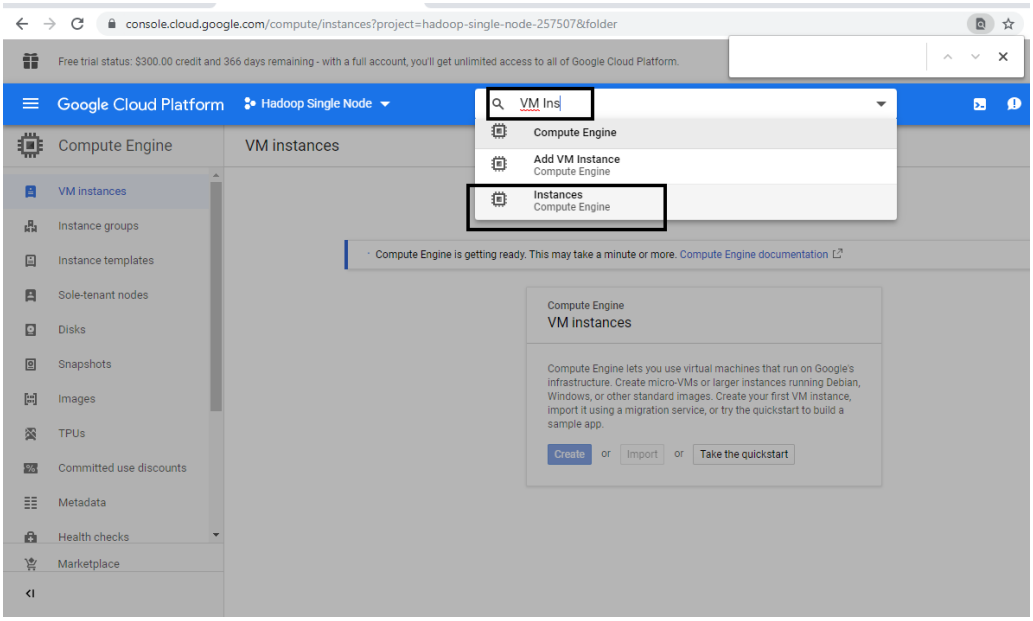
You will get $300 credit for 1 year

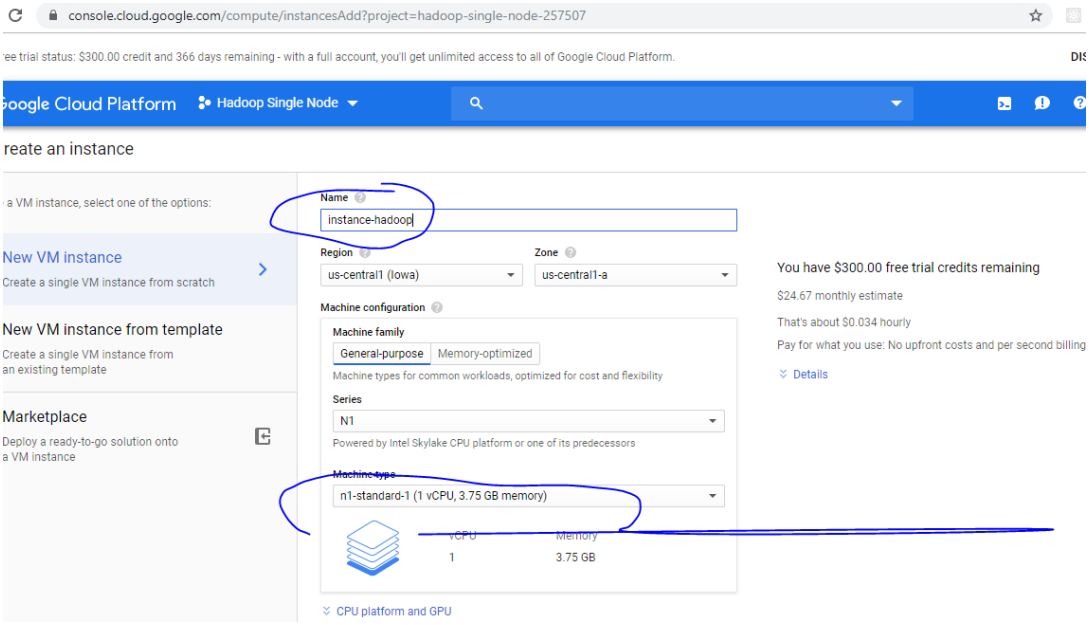


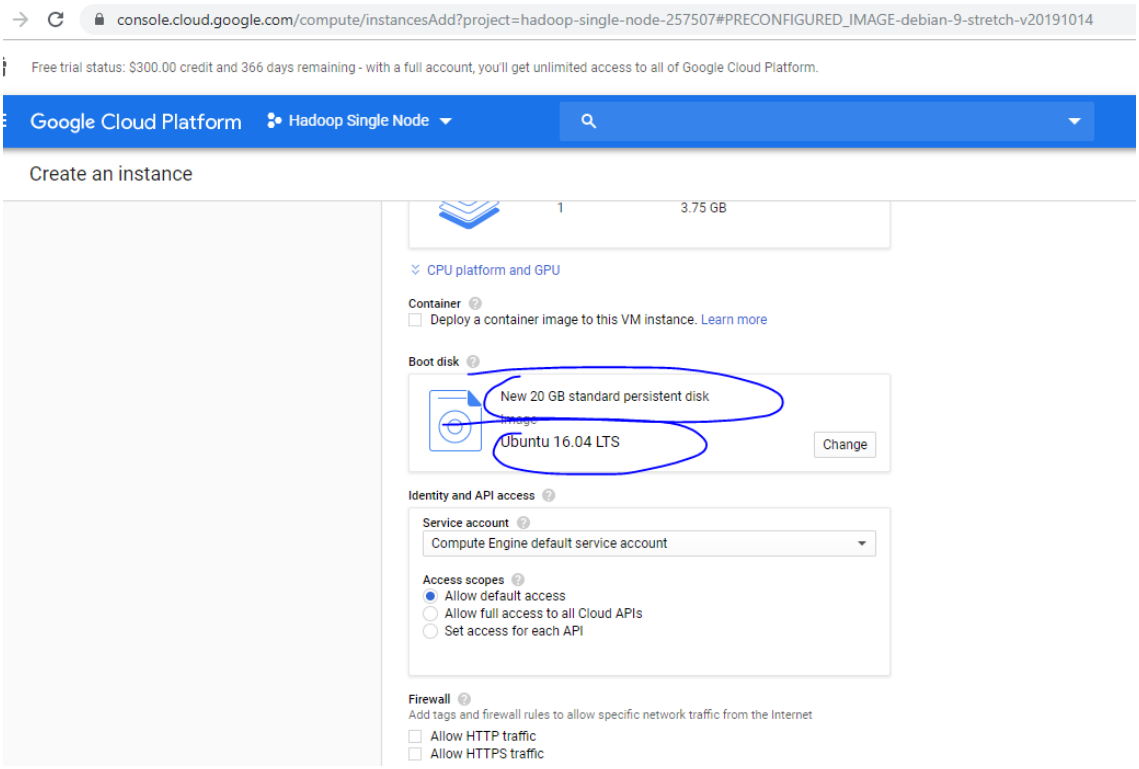

## Step 1: Create a new project



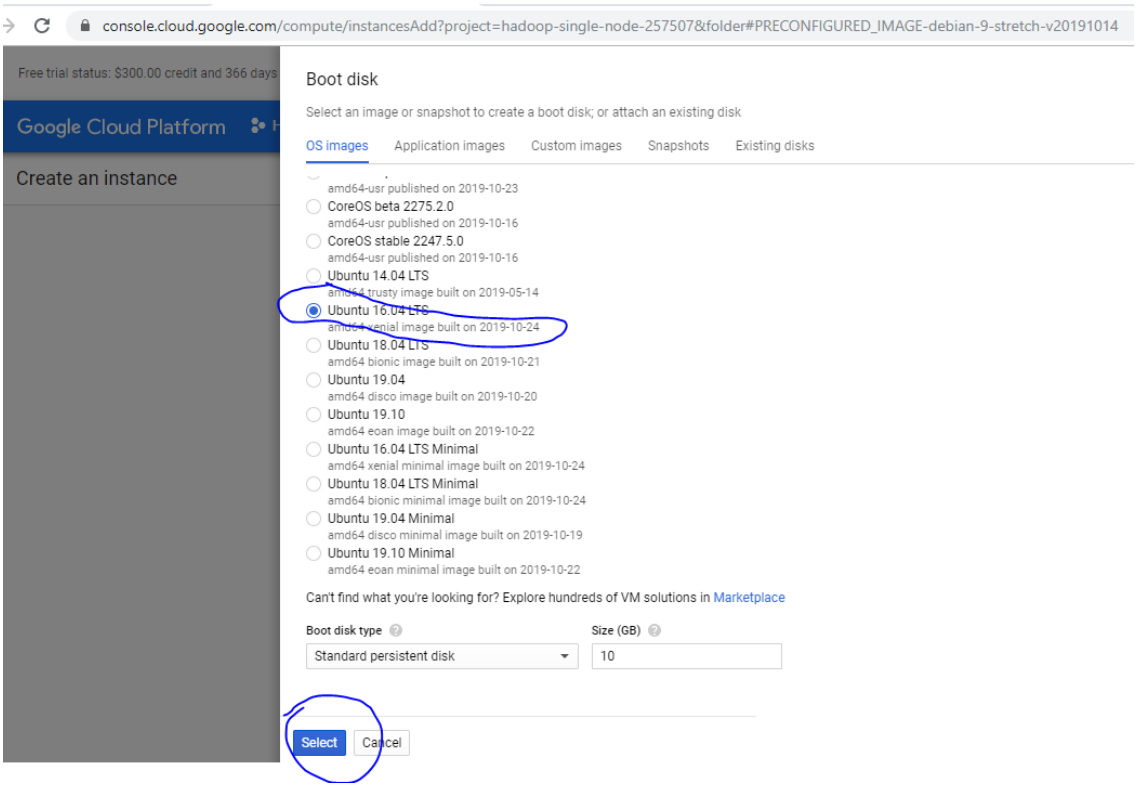
# Step 3: Init an instance

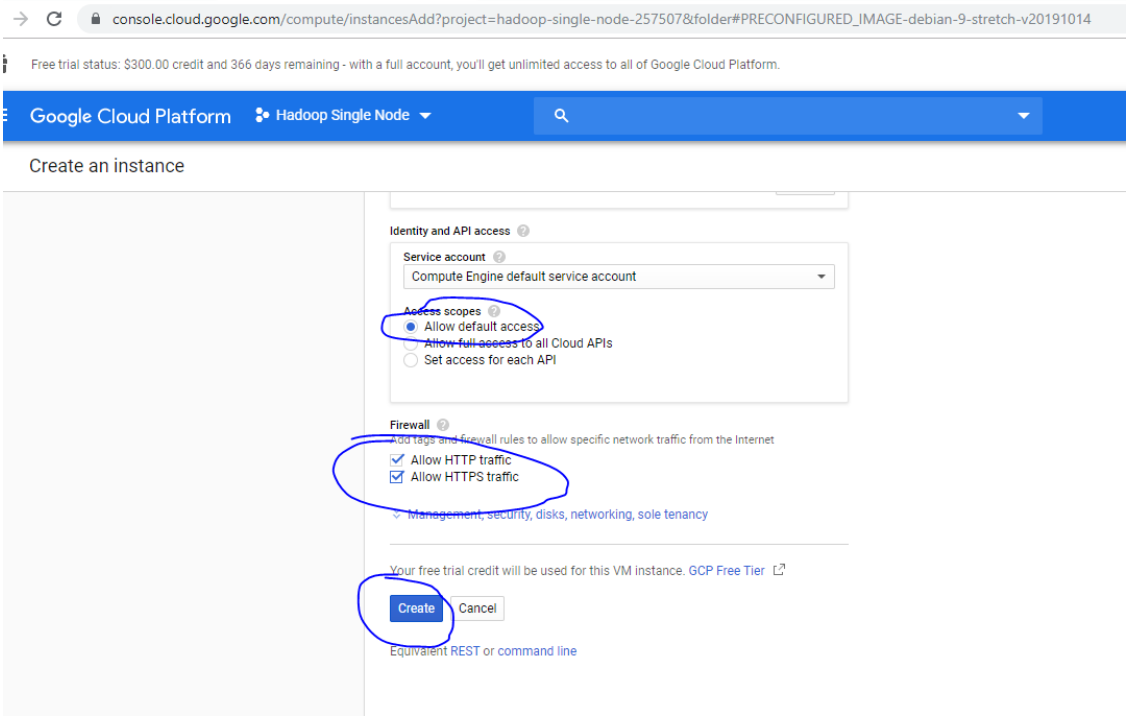


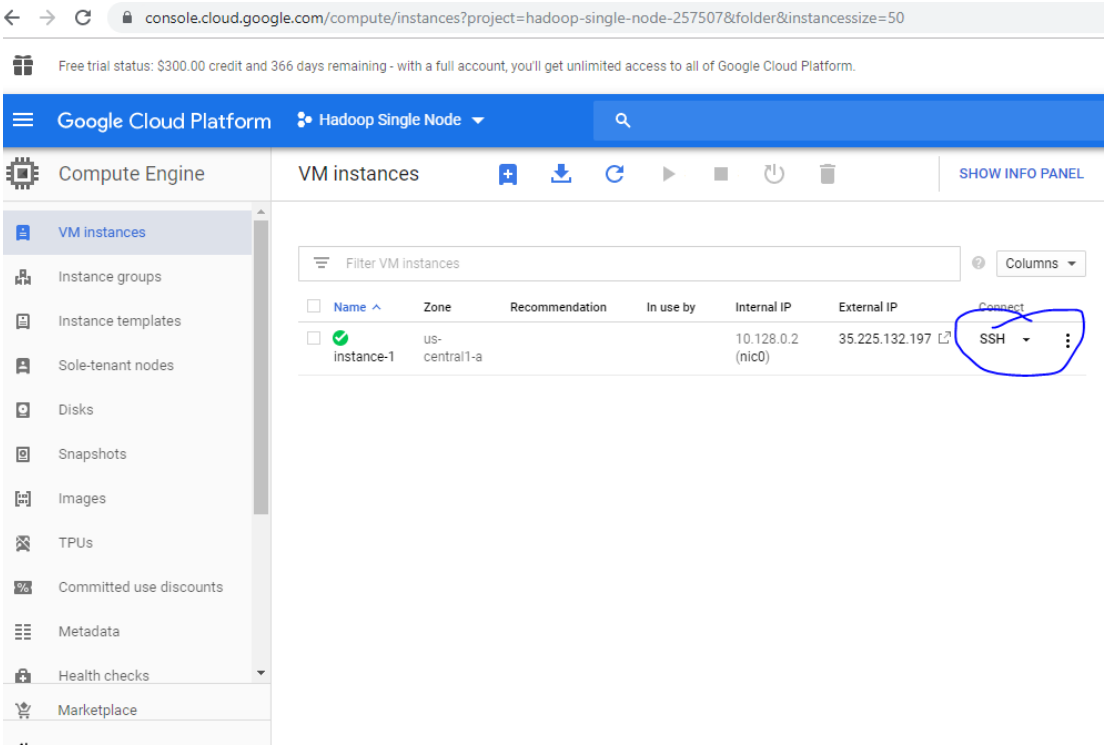


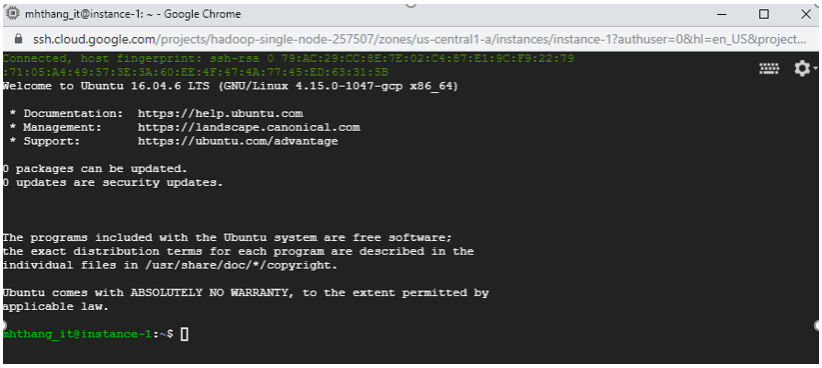

Recommend 20Gb hard-disk because we will have to download cloudera 4.7Gb then extract.





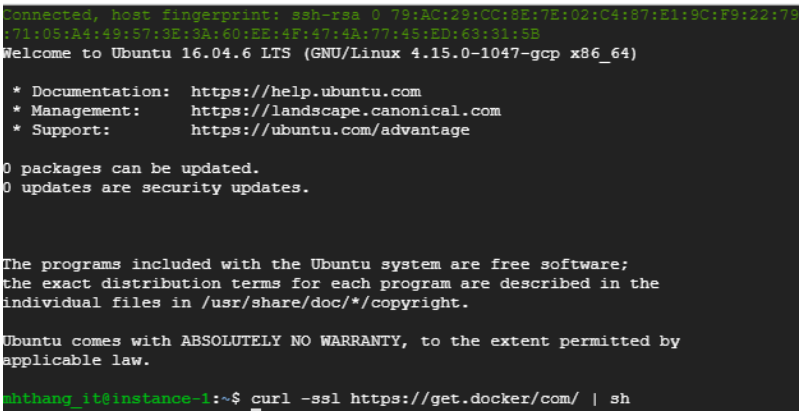

## Step 4: SSH to the instance



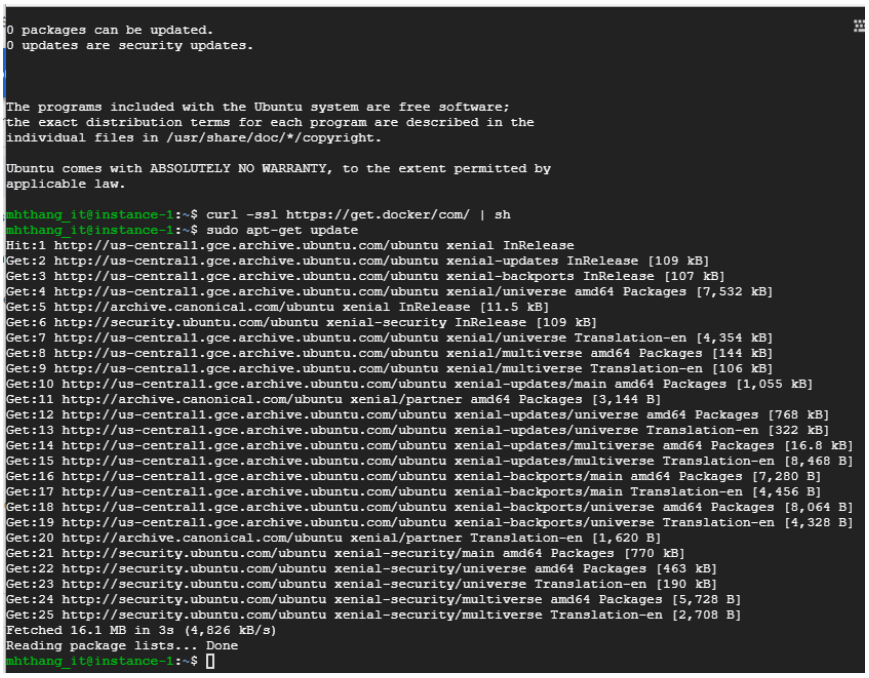



## Step 5: get Docker and update

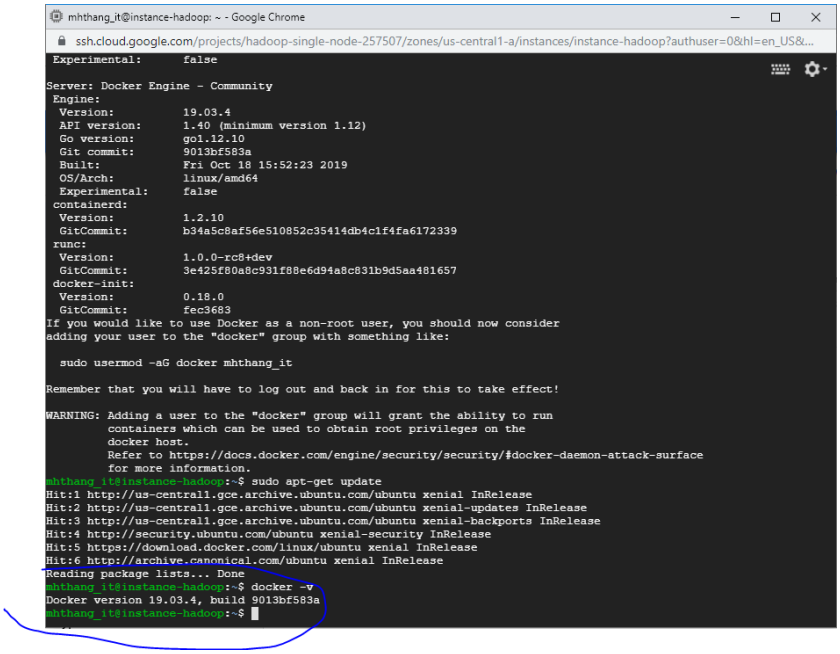
curl -ssl https://get.docker.com/ | sh




sudo apt-get update



Check version of docker




## Step 6: Get the Cloudera Quickstart Image

sudo wget https://downloads.cloudera.com/demo_vm/docker/cloudera-quickstart-vm-5.13.0-0-beta-docker.tar.gz


## Step 7: Extract the cloudera quickstart tar file

tar xzf cloudera-quickstart-vm-*-docker.tar.gz

```
2019-10-30 08:30:06 (37.1 MB/s) - 'cloudera-quickstart-vm-5.13.0-0-beta-docker.tar.gz' saved [5096138910/50
]

mhthang_it@instance-hadoop:~$ tar xzf cloudera-quickstart-vm-*-docker.tar.gz
mhthang_it@instance-hadoop:~$ ls
cloudera-quickstart-vm-5.13.0-0-beta-docker  cloudera-quickstart-vm-5.13.0-0-beta-docker.tar.gz
mhthang_it@instance-hadoop:~$
```

## Step 8: Import Cloudera quickstart docker image

cd cloudera-quickstart-vm-5.13.0-0-beta-docker

sudo docker import cloudera-quickstart-vm-5.13.0–0-beta-docker.tar

Remove downloaded file if short of disk space

```
cloudera-quickstart-vm-5.13. 100%[=========================================>]   4.75G  8.57MB/s    in 2m 11s

2019-10-30 08:30:06 (37.1 MB/s) - 'cloudera-quickstart-vm-5.13.0-0-beta-docker.tar.gz' saved [5096138910/5096138910
]

mhthang_it@instance-hadoop:~$ tar xzf cloudera-quickstart-vm-*-docker.tar.gz
mhthang_it@instance-hadoop:~$ ls
cloudera-quickstart-vm-5.13.0-0-beta-docker  cloudera-quickstart-vm-5.13.0-0-beta-docker.tar.gz
mhthang_it@instance-hadoop:~$ cd cloudera-quickstart-vm-5.13.0-0-beta-docker/
mhthang_it@instance-hadoop:~/cloudera-quickstart-vm-5.13.0-0-beta-docker$ sudo docker import cloudera-quickstart-vm
-5.13.0-0-beta-docker.tar
Error response from daemon: Error processing tar file(exit status 1): write /var/lib/hadoop-hdfs/cache/hdfs/dfs/dat
a/current/BP-1388946040-10.0.0.8-1508802350597/current/finalized/subdir0/subdir2/blk_1073742404: no space left on d
evice
mhthang_it@instance-hadoop:~/cloudera-quickstart-vm-5.13.0-0-beta-docker$ ls
cloudera-quickstart-vm-5.13.0-0-beta-docker.tar
mhthang_it@instance-hadoop:~/cloudera-quickstart-vm-5.13.0-0-beta-docker$ cd
mhthang_it@instance-hadoop:~$ sudo rm cloudera-quickstart-vm-5.13.0-0-beta-docker.tar.gz
mhthang_it@instance-hadoop:~$ cd cloudera-quickstart-vm-5.13.0-0-beta-docker/
mhthang_it@instance-hadoop:~/cloudera-quickstart-vm-5.13.0-0-beta-docker$ sudo docker import cloudera-quickstart-vm
-5.13.0-0-beta-docker.tar
```

## Step 9: Check the container image ID

sudo docker images

```
mhthang_it@instance-hadoop:~/cloudera-quickstart-vm-5.13.0-0-beta-docker$ ls
cloudera-quickstart-vm-5.13.0-0-beta-docker.tar
mhthang_it@instance-hadoop:~/cloudera-quickstart-vm-5.13.0-0-beta-docker$ cd
mhthang_it@instance-hadoop:~$ sudo rm cloudera-quickstart-vm-5.13.0-0-beta-docker.tar.gz
mhthang_it@instance-hadoop:~$ cd cloudera-quickstart-vm-5.13.0-0-beta-docker/
mhthang_it@instance-hadoop:~/cloudera-quickstart-vm-5.13.0-0-beta-docker$ sudo docker import cloudera-quickstart-vm
-5.13.0-0-beta-docker.tar
sha256:e144772eb1eace261b1c96fd18054b101e79bcd19d0eaa589522fc0a822cf778
mhthang_it@instance-hadoop:~/cloudera-quickstart-vm-5.13.0-0-beta-docker$ cd
mhthang_it@instance-hadoop:~$ sudo docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
<none>              <none>              e144772eb1ea        About a minute ago   7GB
mhthang_it@instance-hadoop:~$
```

## Step 10: Step Run docker image

sudo docker run --hostname=quickstart.cloudera --privileged=true -t -i -p 8777:8888 -p 7190:7180 -p 90:80 e144772eb1ea /usr/bin/docker-quickstart

*Now, we are ready to run Hadoop, example with wordcount map reduce. Step to create and export Hadoop project, ref to Set Up a Single Node Cluster using VM or Docker document.*

## Step 11: Create folder

[root@quickstart ~]# hadoop fs -mkdir /user/cloudera/wordcount /user/cloudera/wordcount/input

```
[root@quickstart ~]# hadoop fs -mkdir /user/cloudera/wordcount /user/cloudera/wordcount/input
```

## Step 11: Create documents

echo "Hadoop is an elephant" > file0

echo "Hadoop is as yellow as can be" > file1

echo "Oh what a yellow fellow is Hadoop" > file2

hadoop fs -put file* /user/cloudera/wordcount/input

```
[root@quickstart ~]# echo "Hadoop is an elephant" > file0
[root@quickstart ~]# echo "Hadoop is as yellow as can be" > file1
[root@quickstart ~]# echo "Oh what a yellow fellow is Hadoop" > file2
[root@quickstart ~]# hadoop fs -put file* /user/cloudera/wordcount/input
[root@quickstart ~]#
```

## Step 12: Load *.jar file into cloudera

In this step, the wordcount.jar is downloaded from an hyperlink. You can find another way to copy file from your machine into google cloud instance.

```
[root@quickstart ~]# curl -o http://cuidot.vn/data/wordcount.jar
curl: no URL specified!
curl: try 'curl --help' or 'curl --manual' for more information
[root@quickstart ~]# curl -O http://cuidot.vn/data/wordcount.jar
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
101  4282  101  4282    0     0   2629      0  0:00:01  0:00:01 --:--:-- 20198
[root@quickstart ~]# ls
file0  file1  file2  hue.json  wordcount.jar
[root@quickstart ~]#
```

## Step 13: Run the WordCount application from the JAR file we created on Eclipse, giving the paths to the input and output directories in HDFS.

hadoop jar wordcount.jar org.myorg.WordCount /user/cloudera/wordcount/input /user/cloudera/wordcount/output

```
[root@quickstart ~]# curl -o http://cuidot.vn/data/wordcount.jar
curl: no URL specified!
curl: try 'curl --help' or 'curl --manual' for more information
[root@quickstart ~]# curl -O http://cuidot.vn/data/wordcount.jar
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
101  4282  101  4282    0     0   2629      0  0:00:01  0:00:01 --:--:-- 20198
[root@quickstart ~]# ls
file0  file1  file2  hue.json  wordcount.jar
[root@quickstart ~]# hadoop jar wordcount.jar WordCount /user/cloudera/wordcount/input /user/cloudera/wordcount/out
put
19/10/30 09:03:30 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
19/10/30 09:03:31 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement t
he Tool interface and execute your application with ToolRunner to remedy this.
19/10/30 09:03:31 INFO input.FileInputFormat: Total input paths to process : 3
19/10/30 09:03:31 INFO mapreduce.JobSubmitter: number of splits:3
19/10/30 09:03:32 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1572425408290_0001
19/10/30 09:03:33 INFO impl.YarnClientImpl: Submitted application application_1572425408290_0001
19/10/30 09:03:33 INFO mapreduce.Job: The url to track the job: http://quickstart.cloudera:8088/proxy/application_1
572425408290_0001/
19/10/30 09:03:33 INFO mapreduce.Job: Running job: job_1572425408290_0001
```

## Result:

```
[root@quickstart ~]# hadoop jar wordcount.jar WordCount /user/cloudera/wordcount/input /user/cloudera/wordcount/out
put
19/10/30 09:03:30 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
19/10/30 09:03:31 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement t
he Tool interface and execute your application with ToolRunner to remedy this.
19/10/30 09:03:31 INFO input.FileInputFormat: Total input paths to process : 3
19/10/30 09:03:31 INFO mapreduce.JobSubmitter: number of splits:3
19/10/30 09:03:32 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1572425408290_0001
19/10/30 09:03:33 INFO impl.YarnClientImpl: Submitted application application_1572425408290_0001
19/10/30 09:03:33 INFO mapreduce.Job: The url to track the job: http://quickstart.cloudera:8088/proxy/application_1
572425408290_0001/
19/10/30 09:03:33 INFO mapreduce.Job: Running job: job_1572425408290_0001
19/10/30 09:03:49 INFO mapreduce.Job: Job job_1572425408290_0001 running in uber mode : false
19/10/30 09:03:49 INFO mapreduce.Job:  map 0% reduce 0%
19/10/30 09:04:11 INFO mapreduce.Job:  map 33% reduce 0%
19/10/30 09:04:12 INFO mapreduce.Job:  map 67% reduce 0%
19/10/30 09:04:13 INFO mapreduce.Job:  map 100% reduce 0%
19/10/30 09:04:31 INFO mapreduce.Job:  map 100% reduce 100%
19/10/30 09:04:32 INFO mapreduce.Job: Job job_1572425408290_0001 completed successfully
19/10/30 09:04:33 INFO mapreduce.Job: Counters: 50
        File System Counters
                FILE: Number of bytes read=200
                FILE: Number of bytes written=574817
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=482
                HDFS: Number of bytes written=80
                HDFS: Number of read operations=12
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Killed map tasks=1
                Launched map tasks=3
                Launched reduce tasks=1
                Data-local map tasks=3
                Total time spent by all maps in occupied slots (ms)=60450
                Total time spent by all reduces in occupied slots (ms)=16932
                Total time spent by all map tasks (ms)=60450
                Total time spent by all reduce tasks (ms)=16932
                Total vcore-milliseconds taken by all map tasks=60450
                Total vcore-milliseconds taken by all reduce tasks=16932
                Total megabyte-milliseconds taken by all map tasks=61900800
                Total megabyte-milliseconds taken by all reduce tasks=17338368
        Map-Reduce Framework
                Map input records=3
                Map output records=18
                Map output bytes=158
                Map output materialized bytes=212
                Input split bytes=396
                Combine input records=0
                Combine output records=0
                Reduce input groups=12
                Reduce shuffle bytes=212
                Reduce input records=18
                Reduce output records=12
                Spilled Records=36
                Shuffled Maps =3
                Failed Shuffles=0
                Merged Map outputs=3
                GC time elapsed (ms)=806
                CPU time spent (ms)=3340
                Physical memory (bytes) snapshot=828334080
                Virtual memory (bytes) snapshot=5230444544
```

Good Luck