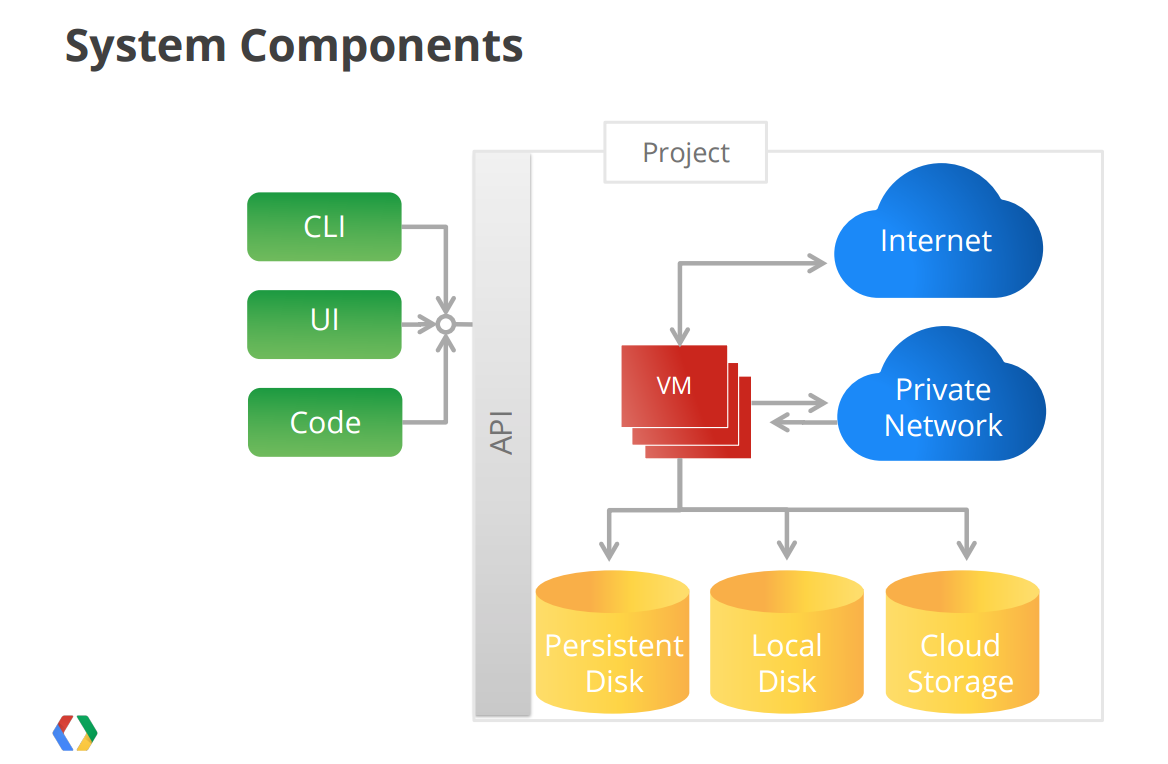
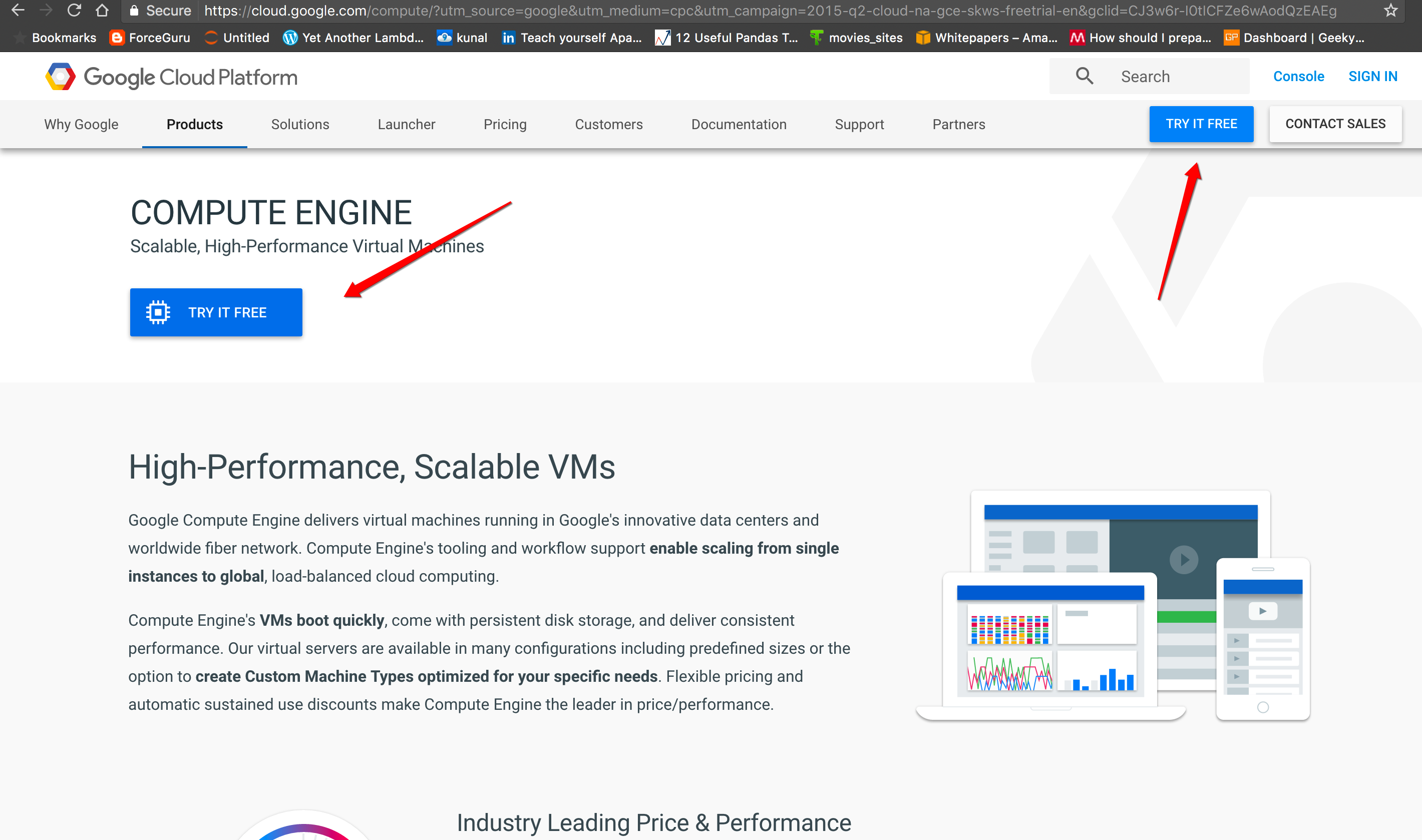
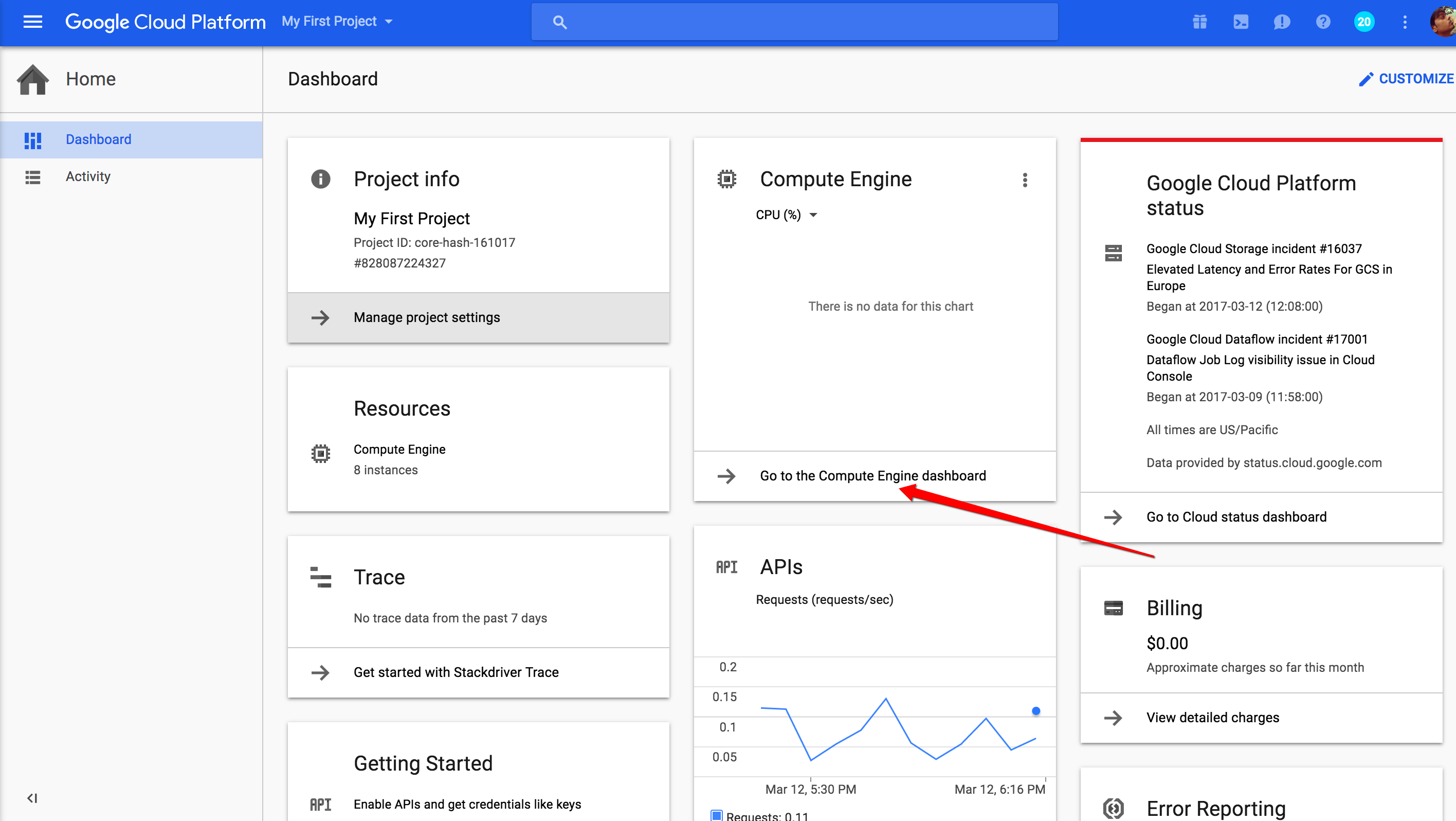
**Google’s Compute Engine Architecture:**



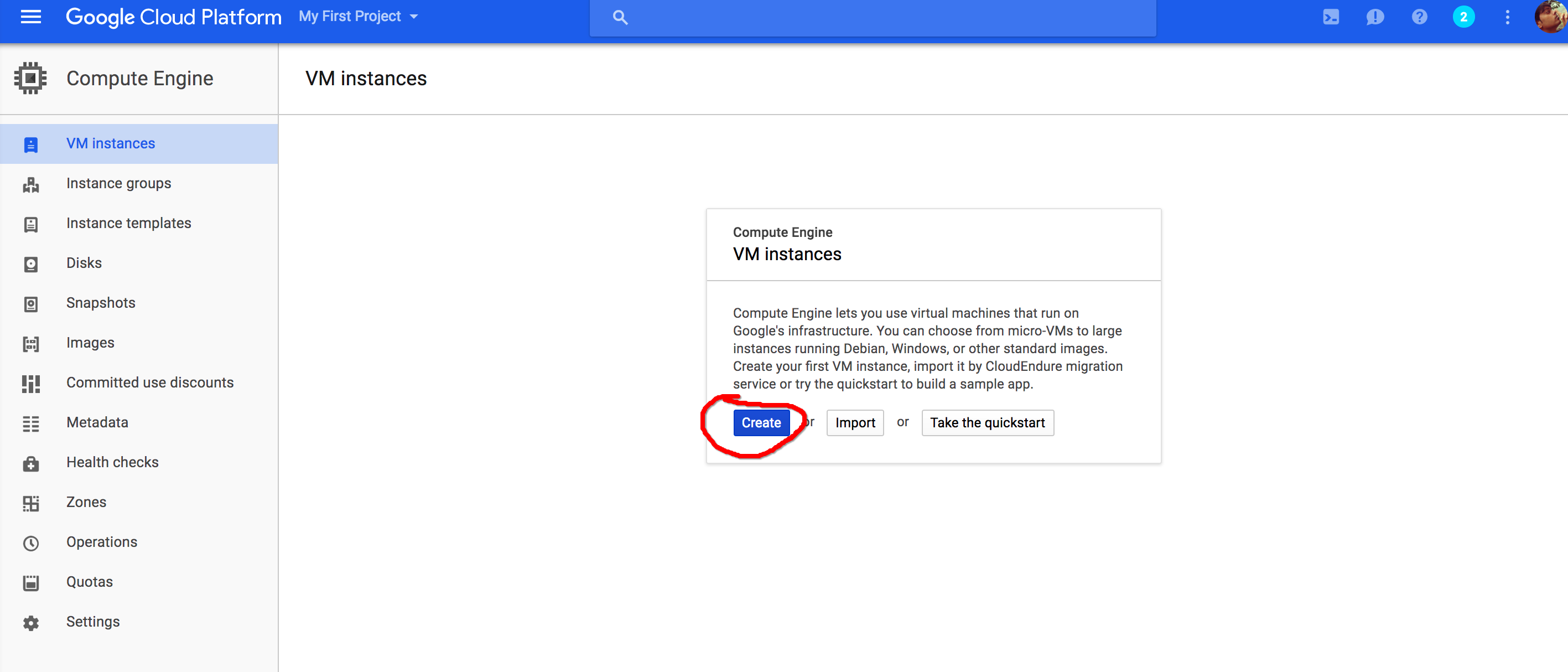
**Step 1:** Navigate to the URL https://cloud.**google**.com/**compute**/ 🡪 Create an ID, or use your existing gmail ID to sign up. Initially every user gets $300 free credit that can be used upto 1 year from the date of sign up.



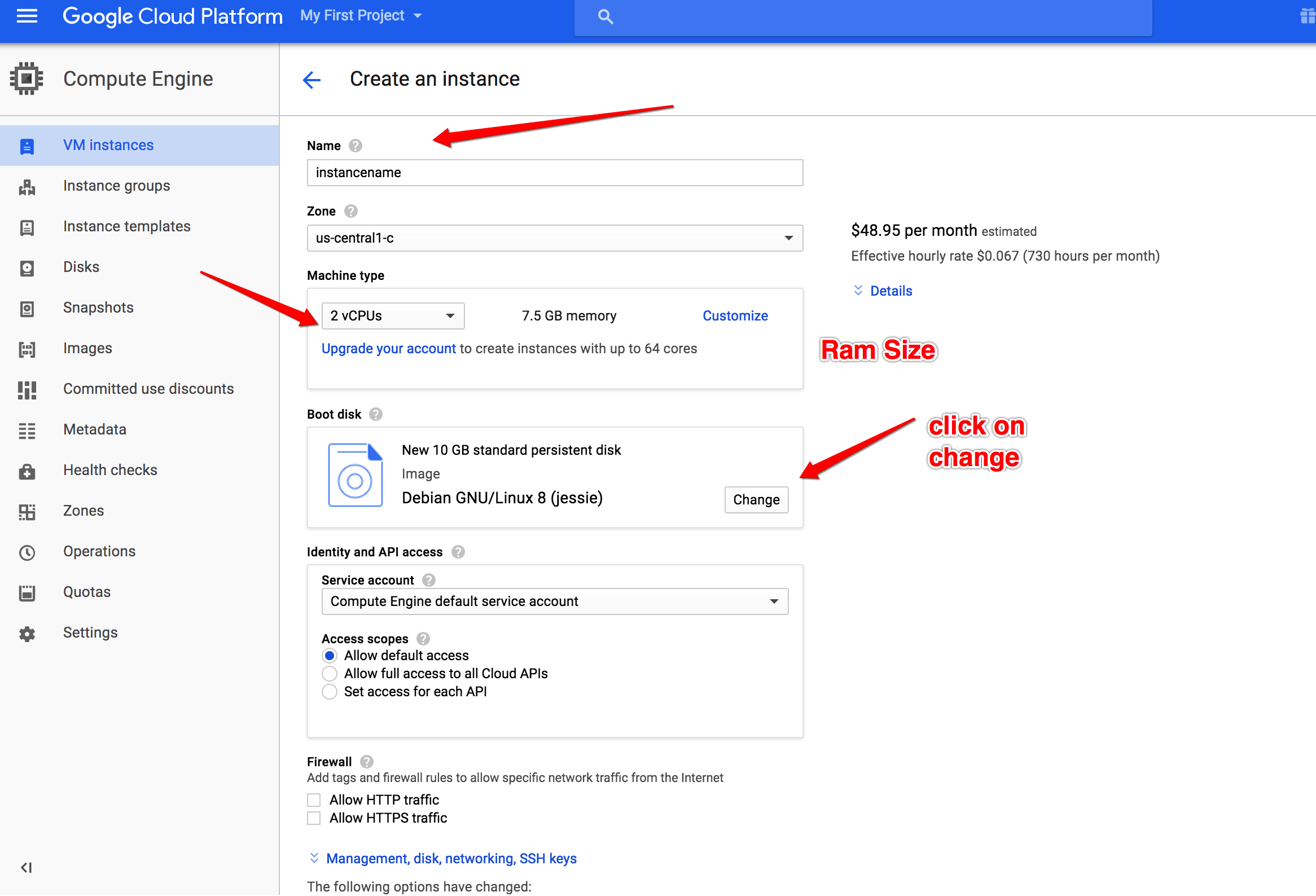
**Step 2: Go to google dashboard and select compute Engine**



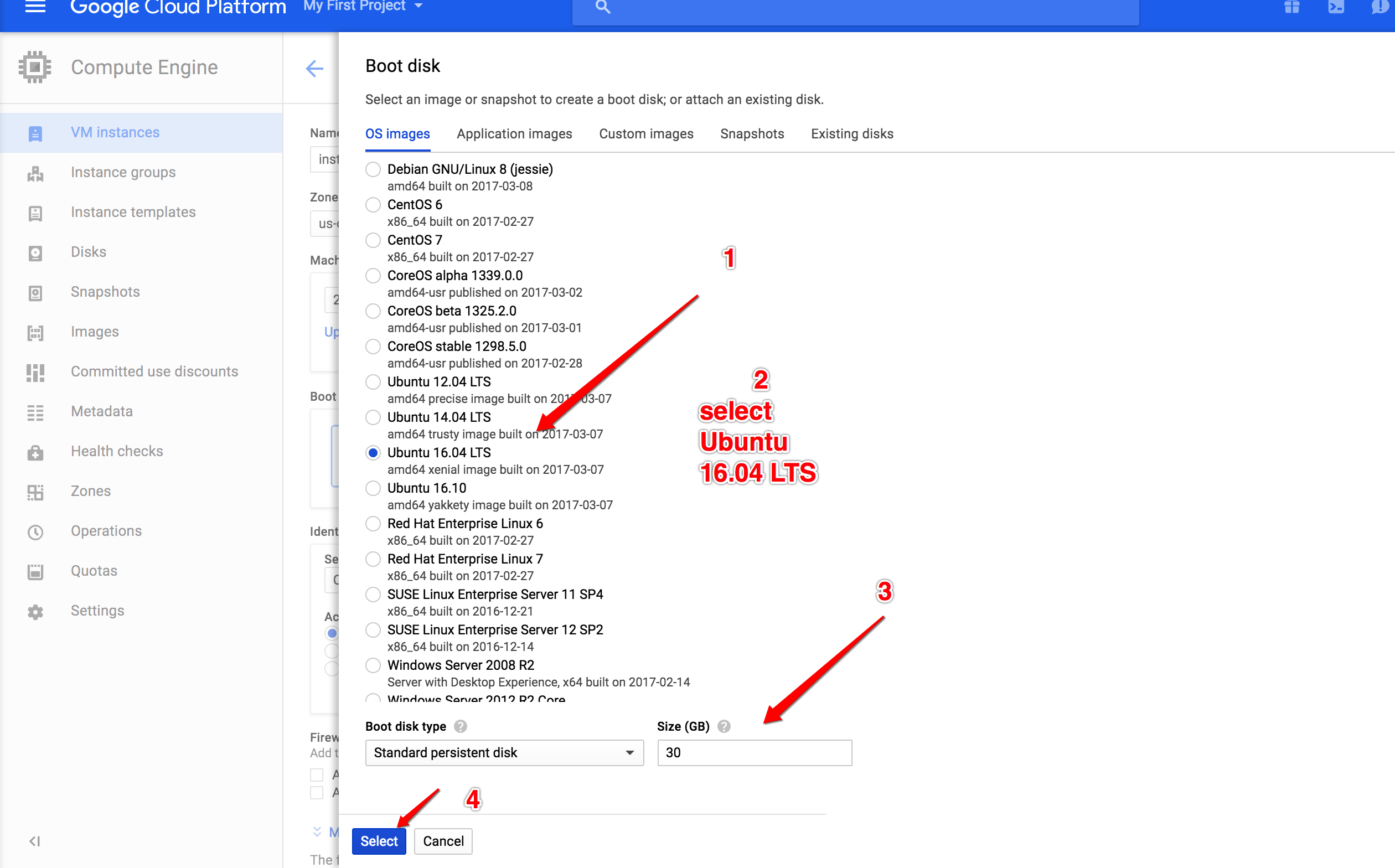
**Step 3: Create Instances:**



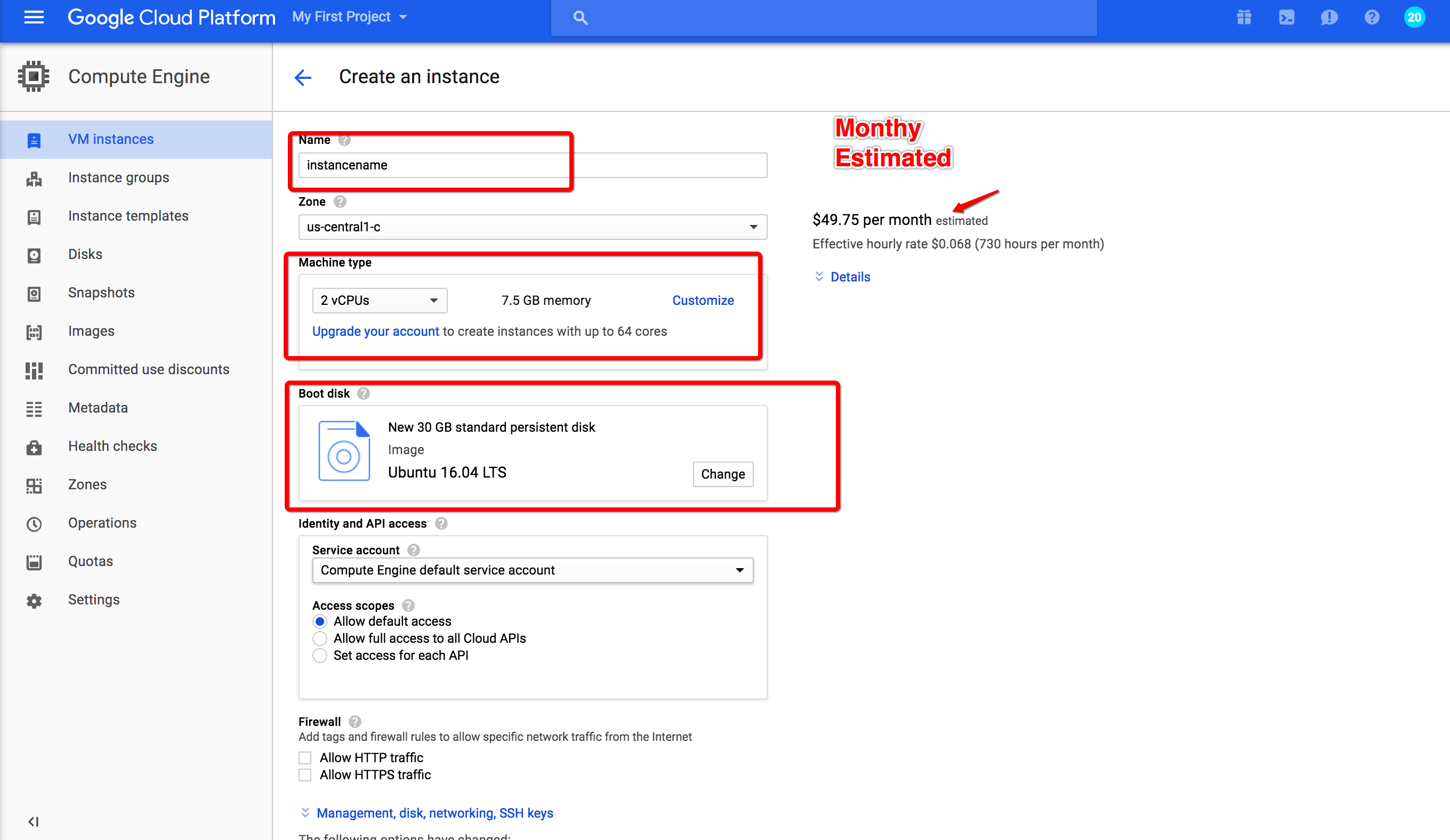
**Step 4: Compute Engine select Below Configuration,**

****

**Step 4:**

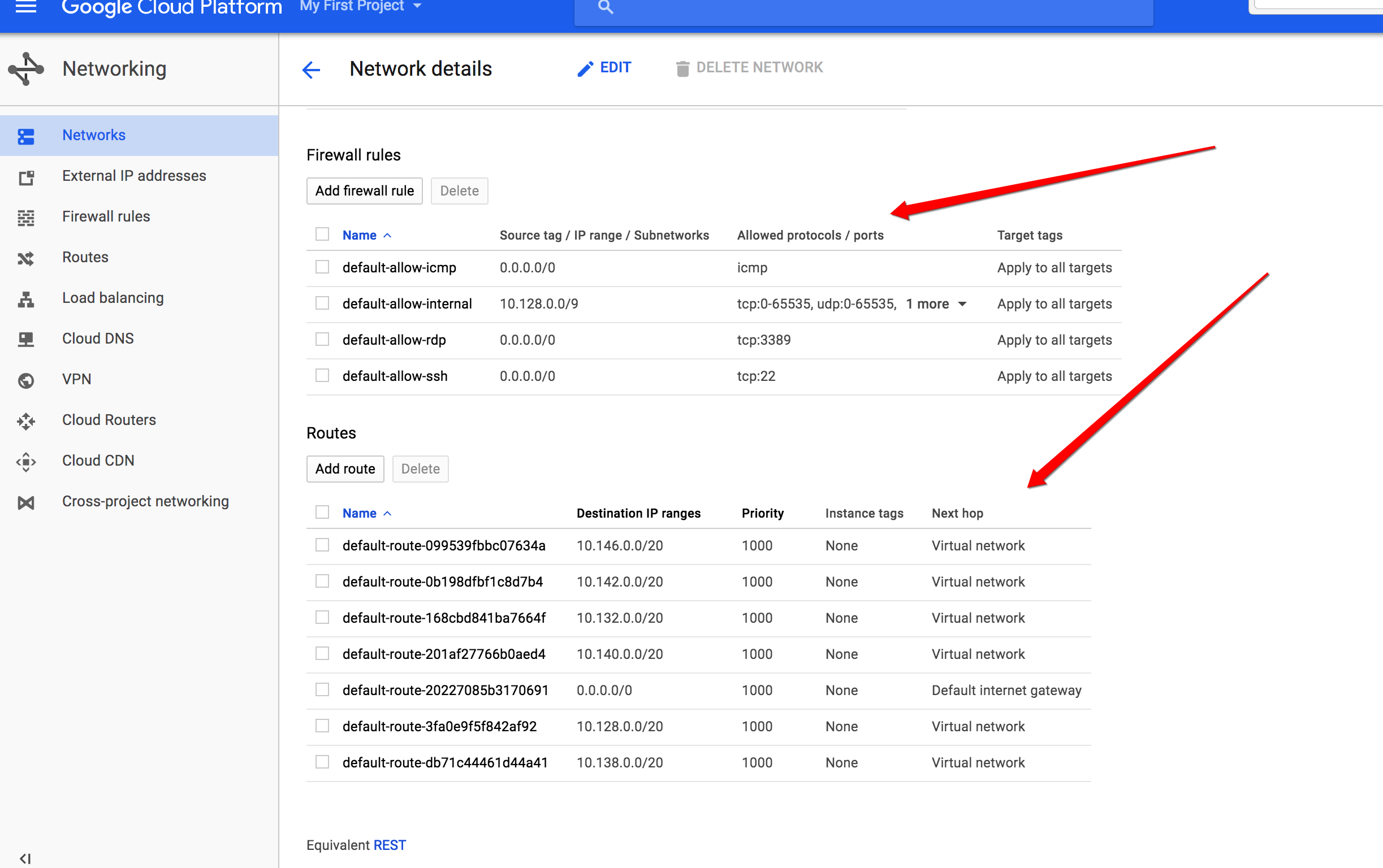
****

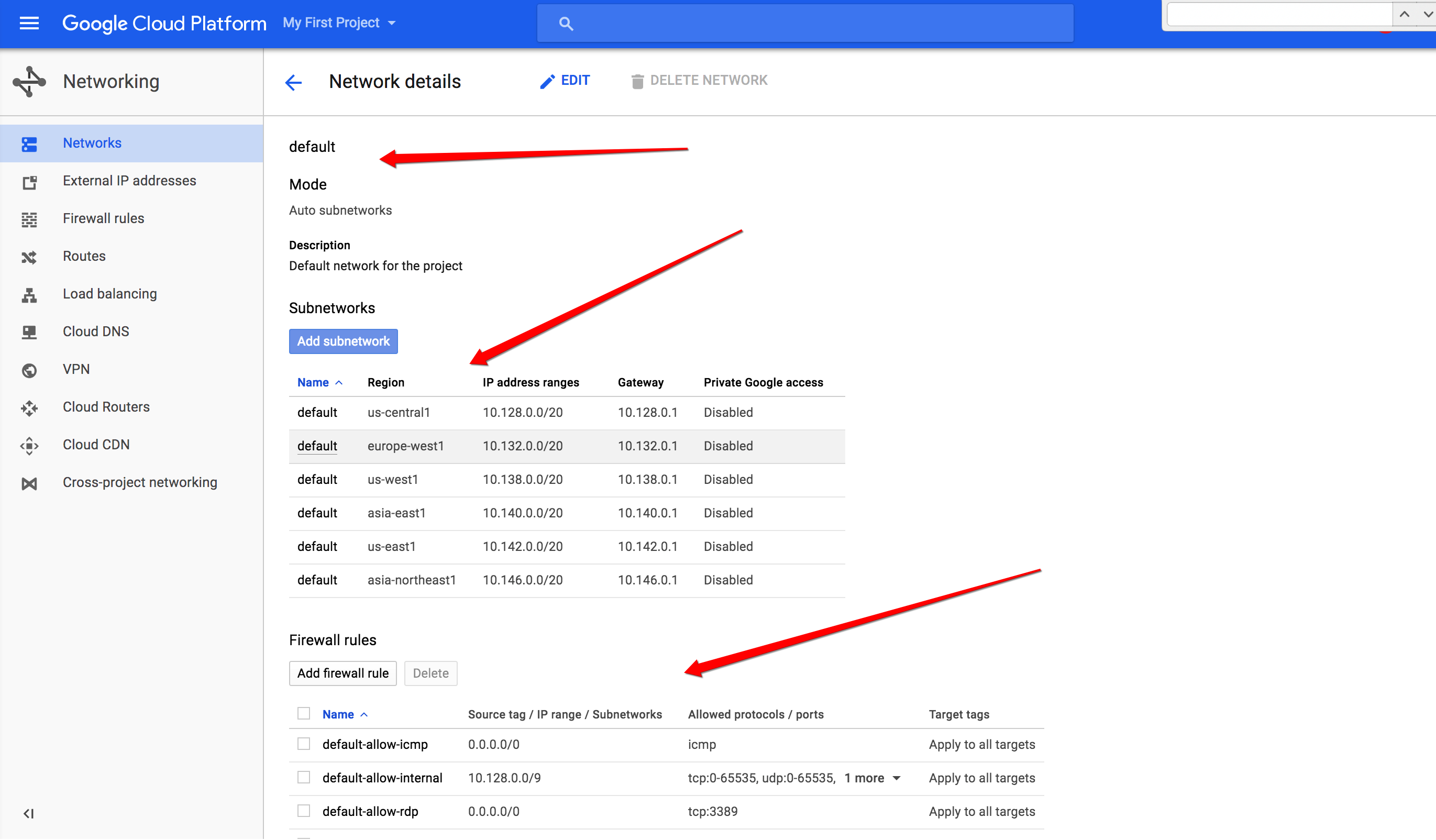
**Step 5: It will give monthly estimate**

****

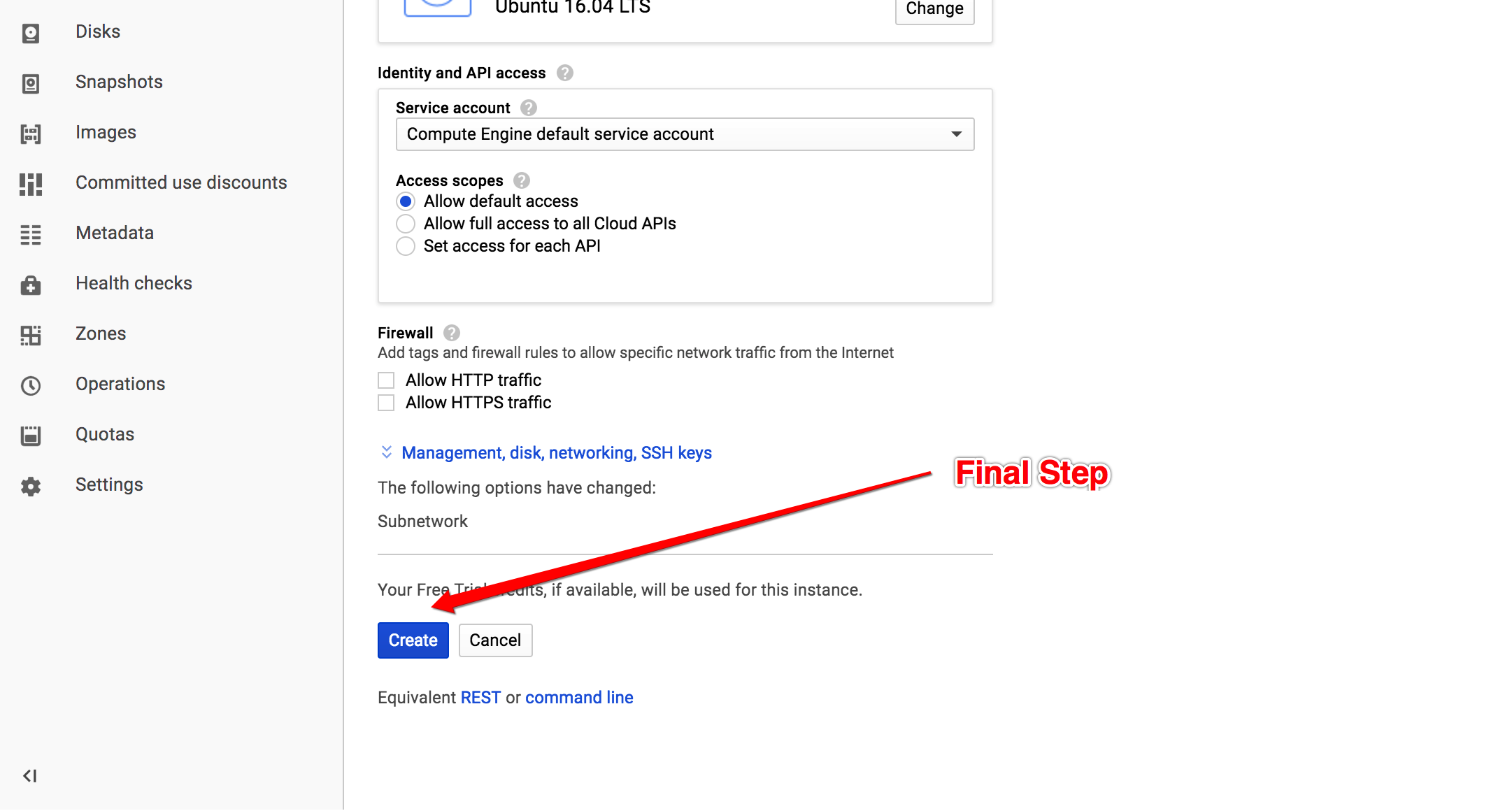
**STEP 5: Inbound/outboud proxy setting**

**9042 port number are required for Cassandra. Selected default firewall setting.**

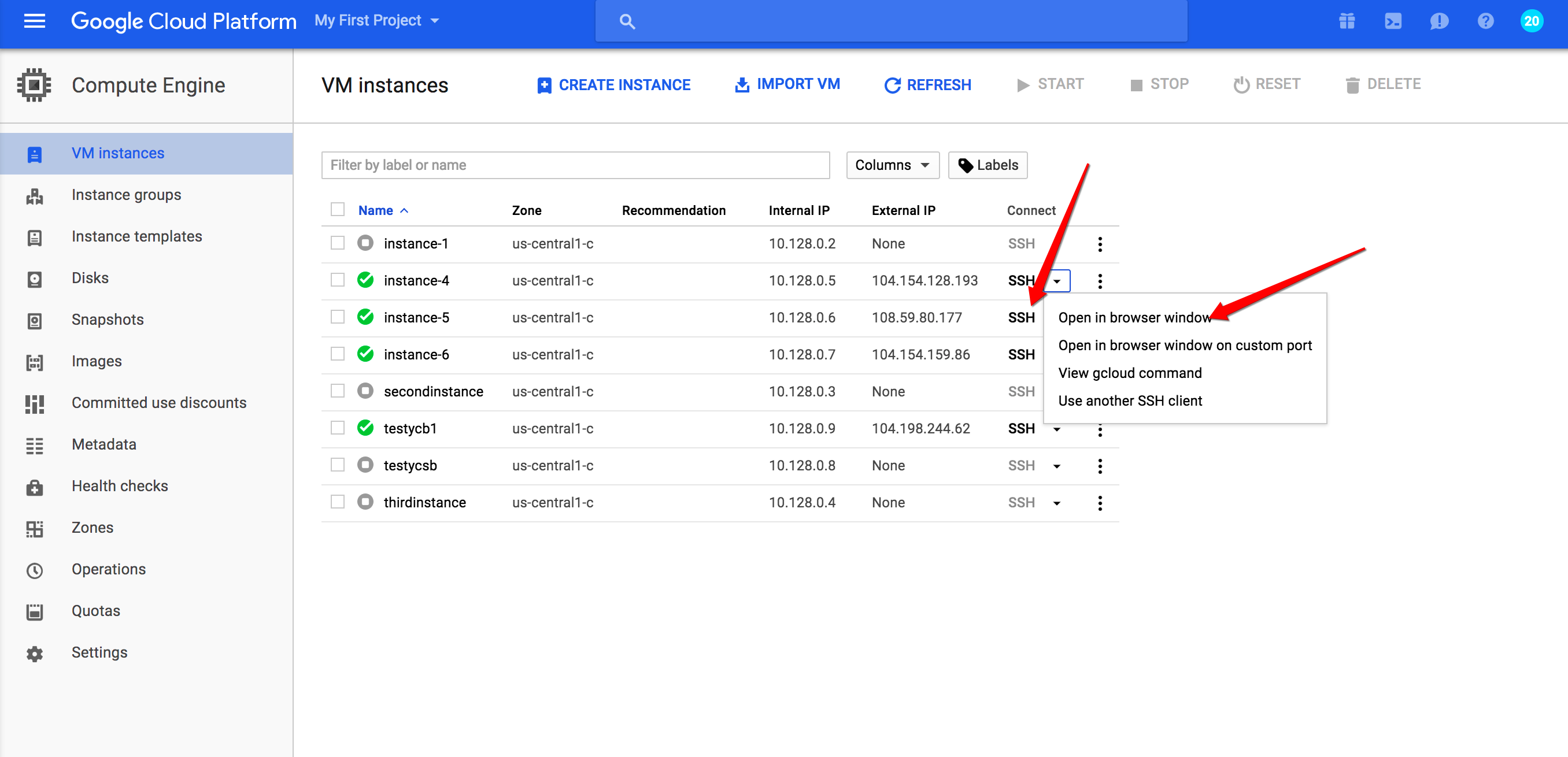
****

****

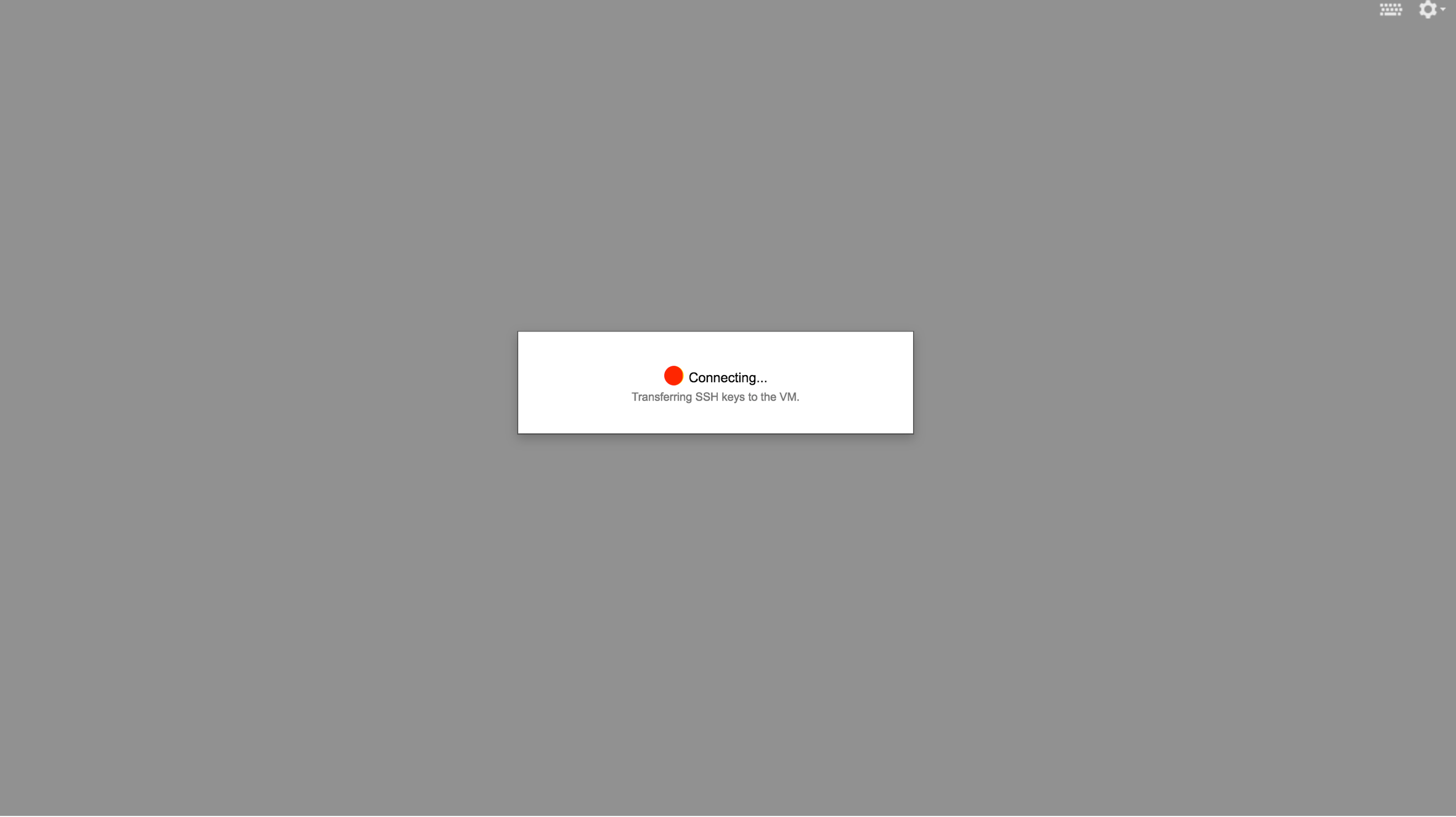
**step 6: After verification create instance**

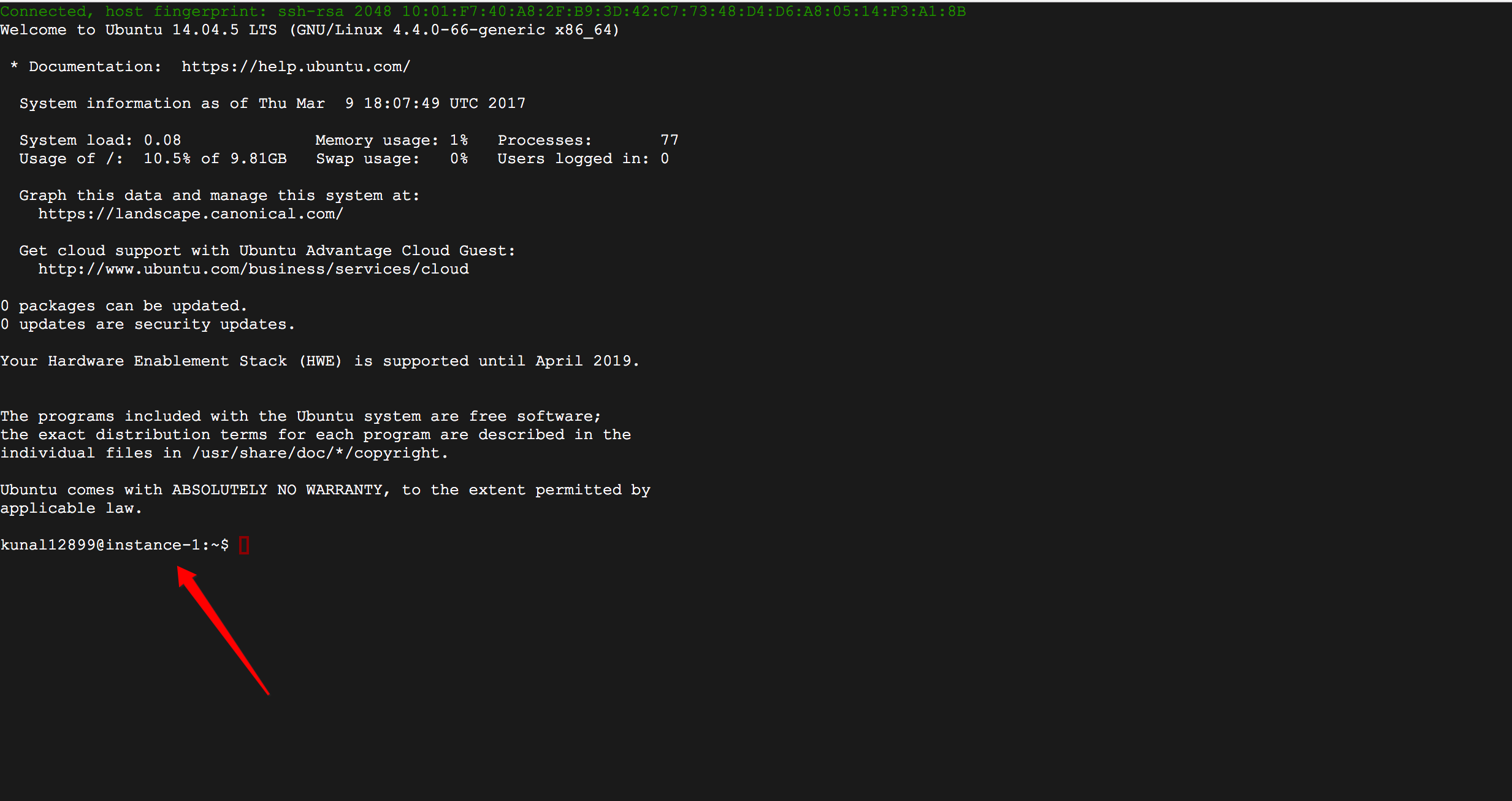
****

**Step 7: Once Instance is created select ssh in open browser mode**

****

**Step 8: Below screen will come then after few minutes console will open**

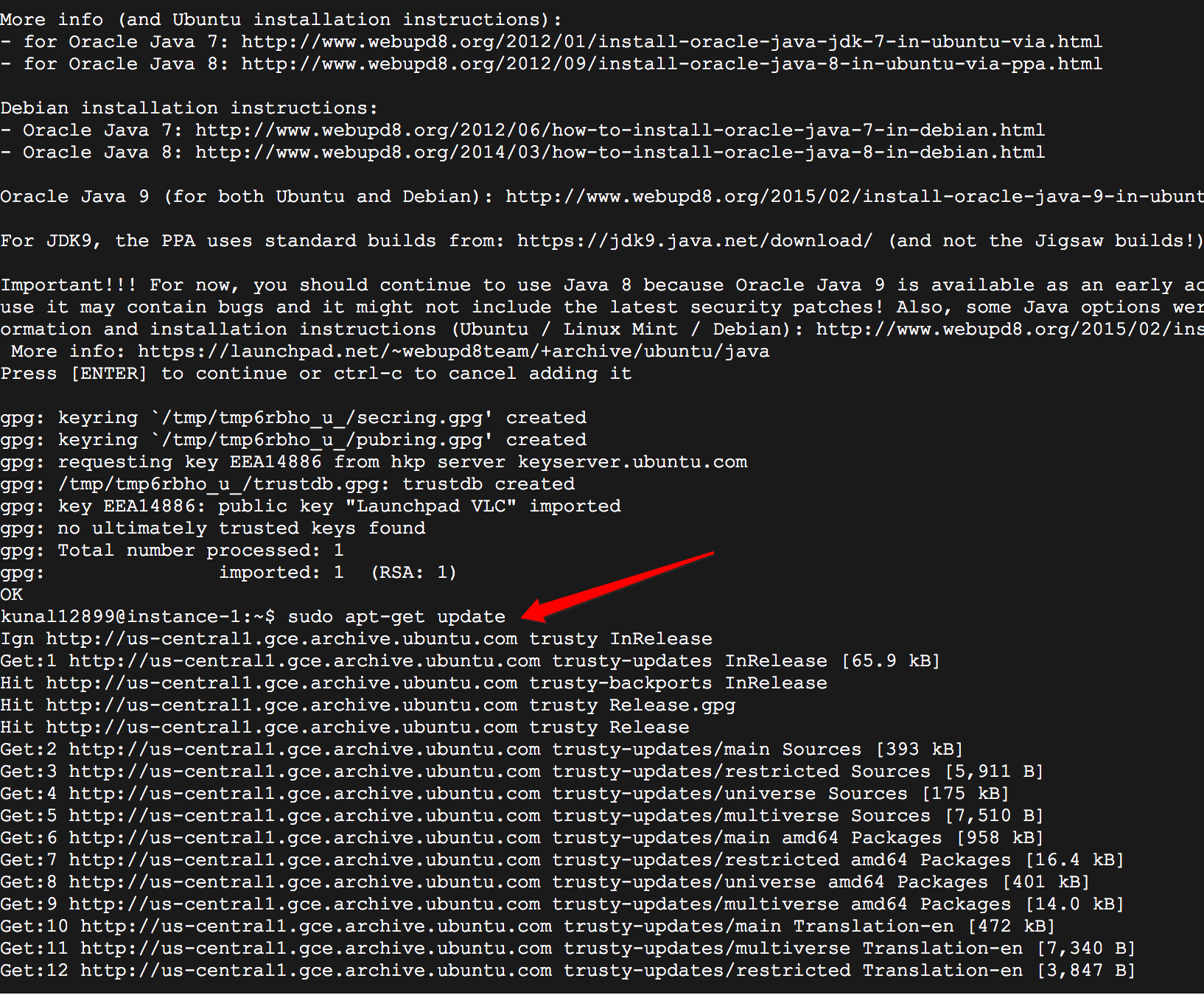
****

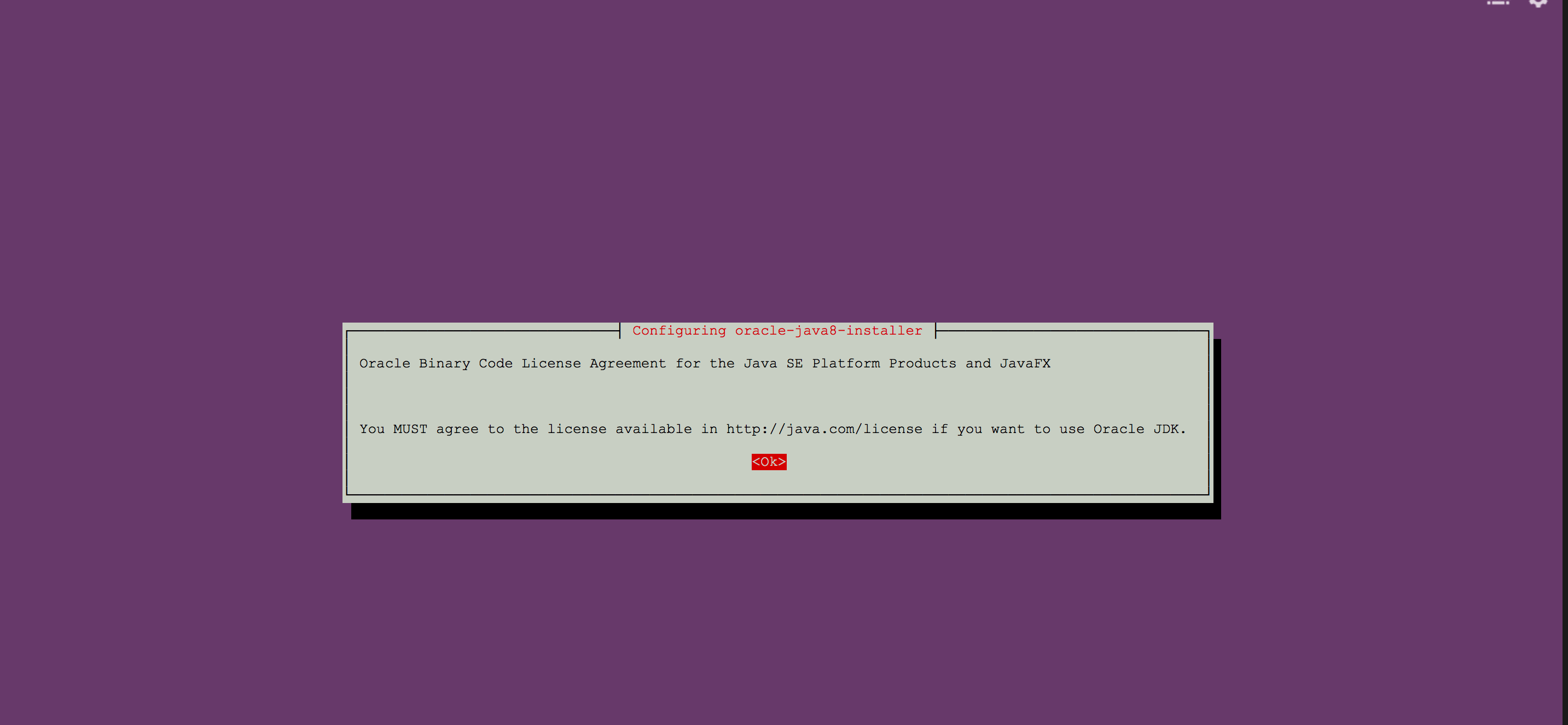
****

**Step 9: Same thing has to be done on 1 node , 3 node , 6 node configuration and YCSB node also :**

1. **Change root password**  
   sudo passwd root  
   <enter password>  
   su  
   <password>
2. **Install Oracle Java 8 in Ubuntu via PPA**  
   sudo add-apt-repository ppa:webupd8team/java  
   sudo apt-get update  
   sudo apt-get install oracle-java8-installer
3. **Set Java environment variables**  
   sudo apt-get install oracle-java8-set-default  
   sudo apt-get update

**Below are some screen print for important steps while following above command:**

****

****

**Step 10: Install Cassandra on all the 1 node, 3 node and 6 node configuration:**

**Below commands need to download and extract.**

**wget** [**http://mirror.cc.columbia.edu/pub/software/apache/cassandra/3.0.11/apache-cassandra-3.0.11-bin.tar.gz**](http://mirror.cc.columbia.edu/pub/software/apache/cassandra/3.0.11/apache-cassandra-3.0.11-bin.tar.gz)

**tar xvfz apache-cassandra-3.0.11-bin.tar.gz**

**Now go to (apache-cassandra-3.0.11/conf/) and first take backup of Cassandra.yaml and do below modification:**

**. Make a backup for all nodes**

cp apache-cassandra-3.0.11/conf/cassandra.yaml apache-cassandra-3.0.11/conf/cassandra.yaml\_bak

seeds: "<internal node ip>,<other node ip within same configuration >,< other node ip within same configuration >"

# listen\_address: **comment this**

listen\_interface: ens4

**start\_rpc: true**

# rpc\_address: **comment this**

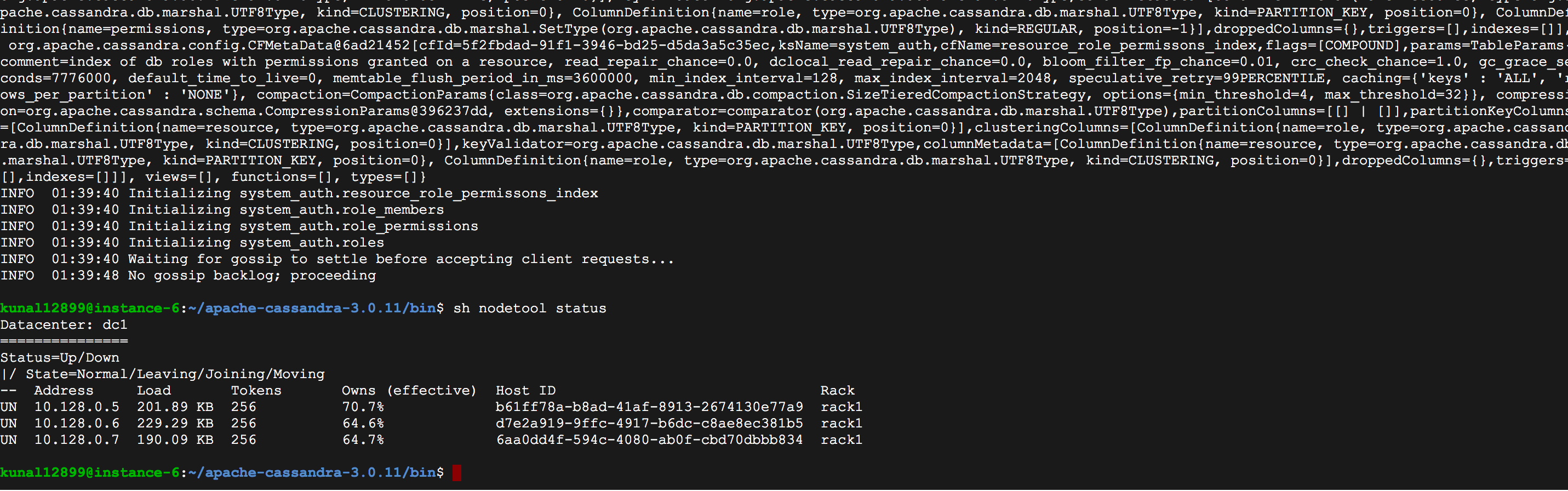
rpc\_interface: ens4(uncomment and make this changes)

comment broadcast address:

endpoint\_snitch: GossipingPropertyFileSnitch

**We can check the stauts if all node are connect are or not using below command**

**Sh nodetool status**

****

**Once every node is connected then we can create table using below command**

**sh cqlsh 10.128.0.5**

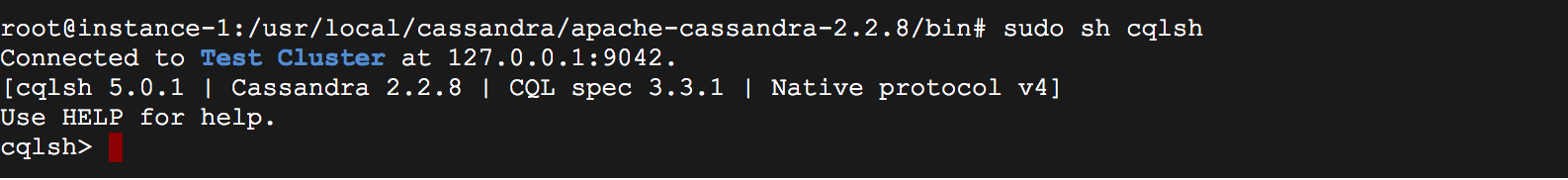
Create keyspace and table for YCSB benchmark

CREATE KEYSPACE usertable WITH REPLICATION = { 'class' : 'SimpleStrategy', 'replication\_factor' : 3};

*use usertable;*

CREATE TABLE usertable.data ( key blob, column1 text, value blob, PRIMARY KEY (key, column1)) WITH COMPACT STORAGE AND CLUSTERING ORDER BY (column1 ASC);

describe table data;

****

Now we can check every node if that table is created or not . now once every node in the cluster is having running Cassandra then we can start YCSB process in other testing node.

**STEP 12: Create a test node with same configuration and follow below steps:**

**Step a: Change root password**sudo passwd root  
<enter password>  
su  
<password> **Step b: Install Oracle Java 8 in Ubuntu via PPA**sudo add-apt-repository ppa:webupd8team/java  
sudo apt-get update  
sudo apt-get install oracle-java8-installer **Step c: Set Java environment variables**sudo apt-get install oracle-java8-set-default  
sudo apt-get update

**Step d: Install YCSB:**

curl -O --location https://github.com/brianfrankcooper/YCSB/releases/download/0.4.0/ycsb-0.4.0.tar.gz

tar xfvz ycsb-0.4.0.tar.gz  
cd ycsb-0.4.0

**Step e:**

Create workload files in workload folder

nano workload\_test10  
-------------------------------------------------------------  
recordcount=255550  
operationcount=100000  
workload=com.yahoo.ycsb.workloads.CoreWorkload  
readallfields=true  
readproportion=63.25892119  
updateproportion=21.64707002  
scanproportion=0  
insertproportion=15.09400879  
requestdistribution=uniform

------------------------------------------------------------

nano workload\_test40  
-------------------------------------------------------------  
recordcount=569620  
operationcount=400000  
workload=com.yahoo.ycsb.workloads.CoreWorkload  
readallfields=true  
readproportion=63.25892119  
updateproportion=21.64707002  
scanproportion=0  
insertproportion=15.09400879  
requestdistribution=uniform

------------------------------------------------------------

nano workload\_test80   
-------------------------------------------------------------  
recordcount=985372  
operationcount=800000  
workload=com.yahoo.ycsb.workloads.CoreWorkload  
readallfields=true  
readproportion=63.25892119  
updateproportion=21.64707002  
scanproportion=0  
insertproportion=15.09400879  
requestdistribution=uniform

------------------------------------------------------------

nano workload\_test160

-------------------------------------------------------------  
recordcount=1806570  
operationcount=1600000  
workload=com.yahoo.ycsb.workloads.CoreWorkload  
readallfields=true  
readproportion=63.25892119  
updateproportion=21.64707002  
scanproportion=0  
insertproportion=15.09400879  
requestdistribution=uniform

------------------------------------------------------------  
  
nano workload\_test320  
------------------------------------------------------------  
recordcount=3407734  
operationcount=3200000  
workload=com.yahoo.ycsb.workloads.CoreWorkload  
readallfields=true  
readproportion=63.25892119  
updateproportion=21.64707002  
scanproportion=0  
insertproportion=15.09400879  
requestdistribution=uniform  
------------------------------------------------------------

nano workload\_test640  
------------------------------------------------------------  
recordcount=6445135  
operationcount=6400000  
workload=com.yahoo.ycsb.workloads.CoreWorkload  
readallfields=true  
readproportion=63.25892119  
updateproportion=21.64707002  
scanproportion=0  
insertproportion=15.09400879  
requestdistribution=uniform  
  
------------------------------------------------------------

**Step f: Create a folder for results**

* mkdir results\_Cassandra

**Step g: Load data**

./bin/ycsb load cassandra-10 -P workloads/workload\_test10 -p hosts=<IPAddress> -threads 10 -p columnfamily=data -s > results\_Cassandra/load\_10

**Step h: Run benchmark**

./bin/ycsb run cassandra-10 -P workloads/workload\_test10 -p hosts=<IPAddress> -threads 10 -p columnfamily=data -s > results\_Cassandra/run\_10

every time change three thing input file name in above case it is **workload\_test10**

**thread**

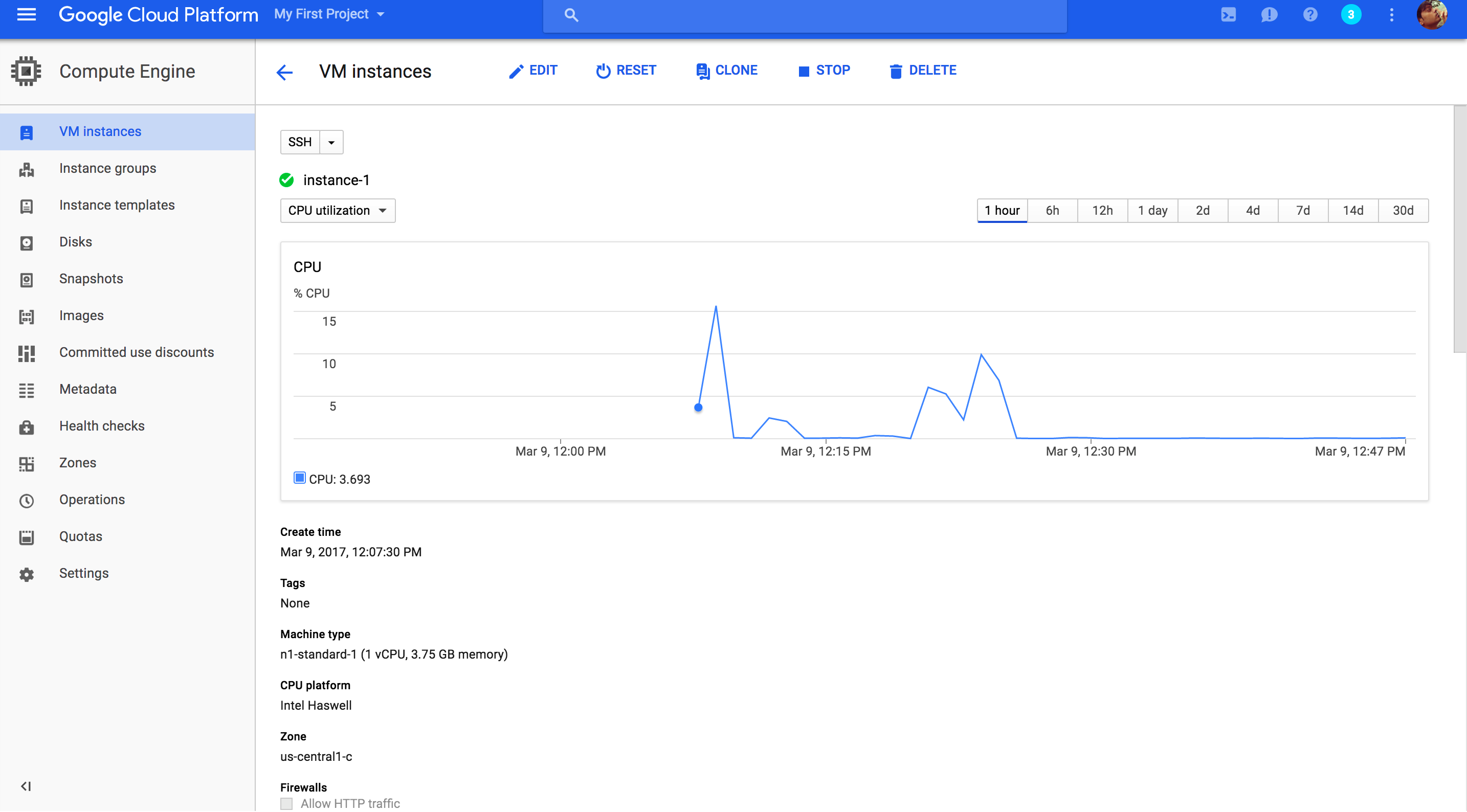
**output file name:**

After each step

**Final step: Clean loaded data for next benchmark ;**

truncate table data;

**You can verify and see the instance cpu utilization:**

****

**ISSUE and Solution**

***Some Issues you might want to know!***

**Issue 1: A table in Cassandra was unable to be dropped or truncated.**

**Possible Reason & Solution :** When you want to drop or truncate a table data in a Cassandra Userspace, then all the nodes in the Cassandra Cluster must be up. This can be checked by using the nodetool command. Make sure you run Cassandra in all the nodes and then use the drop or truncate command.

**Issue 2: 7199 port already in use**

**Possible Reason & Solution :** You can manually kill the Cassandra process in all the nodes using the below command and run it again.

**$ lsof -i:7199**

COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME

ruby 13402 zero 4u IPv4 2847851 0t0 TCP \*:3000 (LISTEN)

**$ kill 13402**

**Issue 3: Not able to start Cassandra once we start after YCSB failure**

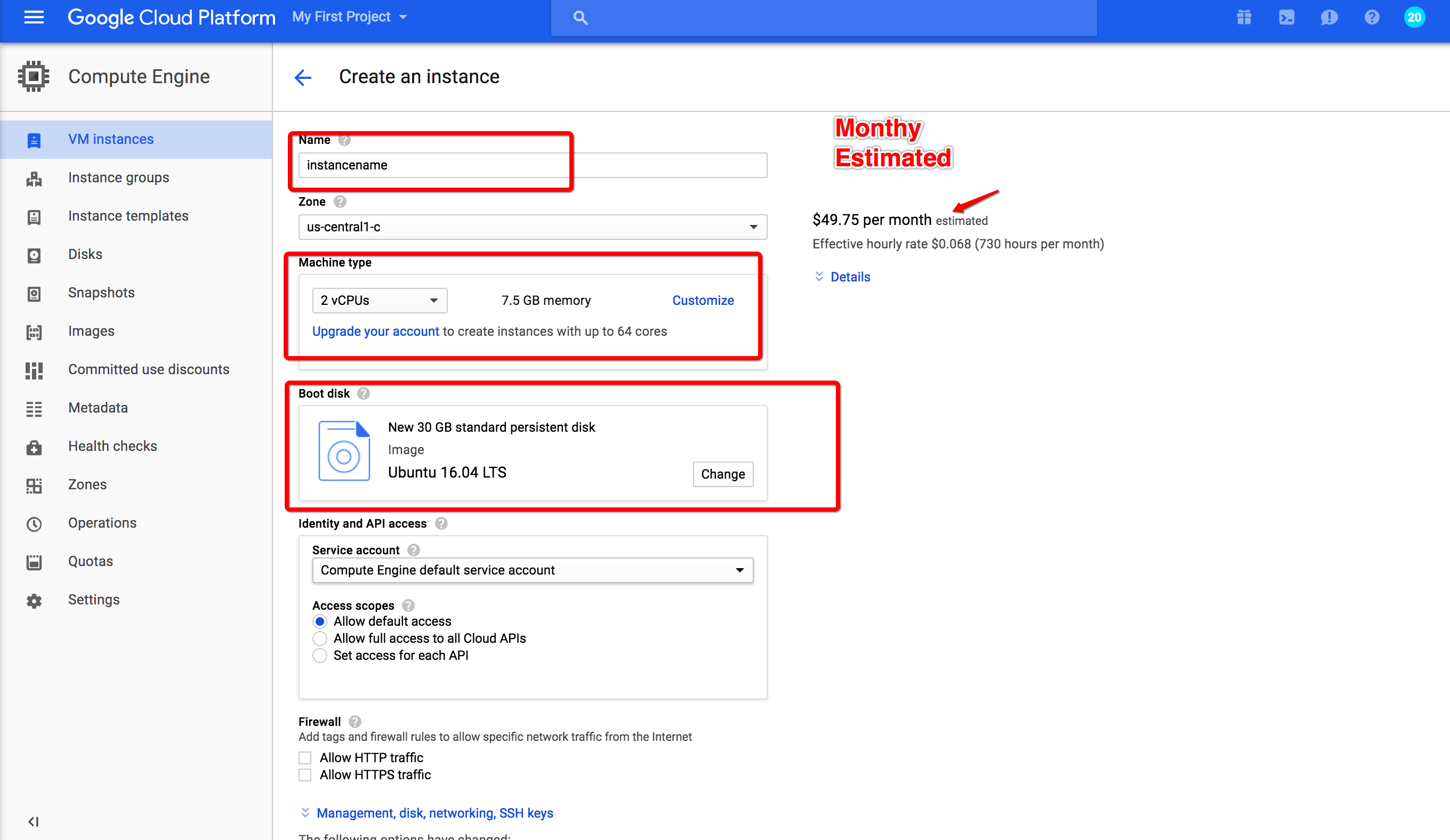
ERROR 22:12:16 Exiting due to error while processing commit log during initialization.

**Solution: Delete all files from commitlog directory in the data folder**

Apache-cassandra-3.0.11/data/

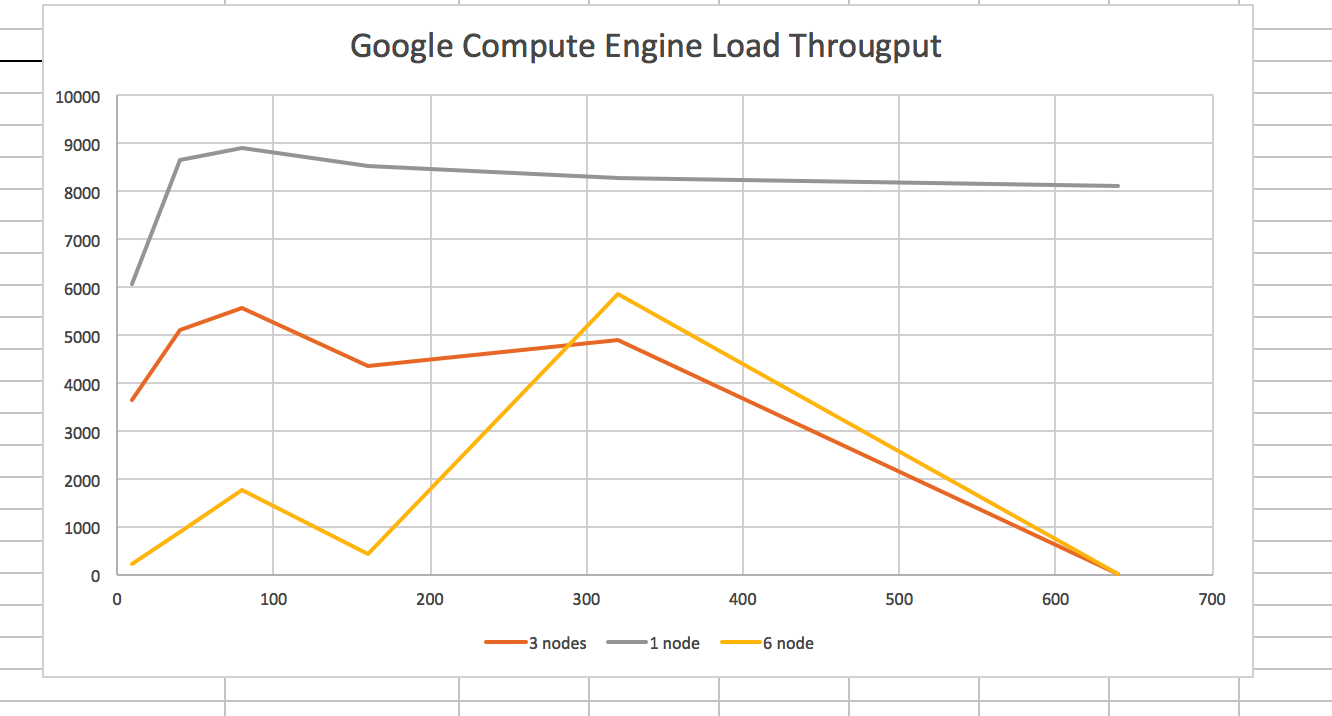
rm –rf commitlog/\*

**Cost and Pricing Estimate**

****

Output:

**Graph Analysis of GCE Load Throughput**



**Graph Analysis of GCE Run Throughput**



**Important Note:**

For Node 1 and Node 3

worked till loading test640 and run of the test is taking forever to over process 20000 operations against total of 6400000

both cassandra and banchmark instances are on the same region cloud.

**For Node 6:**

Benchmark instance working really fast. worked all the way till loading test640 case and after loading almost 6000000 operations out of 6445135 (95% of total counts).

6 instances of cassandra: 3 on same region cloud ; 2 on different one and remaining 1 and the benchmark instance on an another region cloud

**commands :**

./bin/ycsb load cassandra-10 -P workload\_test10 -p hosts=10.128.0.5 -threads 10 -p columnfamily=data -s > results\_Cassandra/load\_10

./bin/ycsb run cassandra-10 -P workload\_test10 -p hosts=10.128.0.5 -threads 10 -p columnfamily=data -s > results\_Cassandra/run\_10

./bin/ycsb load cassandra-10 -P workload\_test40 -p hosts=10.128.0.5 -threads 40 -p columnfamily=data -s > results\_Cassandra/load\_40

./bin/ycsb run cassandra-10 -P workload\_test40 -p hosts=10.128.0.5 -threads 40 -p columnfamily=data -s > results\_Cassandra/run\_40

./bin/ycsb load cassandra-10 -P workload\_test80 -p hosts=10.128.0.5 -threads 80 -p columnfamily=data -s > results\_Cassandra/load\_80

./bin/ycsb run cassandra-10 -P workload\_test80 -p hosts=10.128.0.5 -threads 80 -p columnfamily=data -s > results\_Cassandra/run\_80

./bin/ycsb load cassandra-10 -P workload\_test160 -p hosts=10.128.0.5 -threads 160 -p columnfamily=data -s > results\_Cassandra/load\_160

./bin/ycsb run cassandra-10 -P workload\_test160 -p hosts=10.128.0.5 -threads 160 -p columnfamily=data -s > results\_Cassandra/run\_160

./bin/ycsb load cassandra-10 -P workload\_test320 -p hosts=10.128.0.5 -threads 320 -p columnfamily=data -s > results\_Cassandra/load\_320

./bin/ycsb run cassandra-10 -P workload\_test320 -p hosts=10.128.0.5 -threads 320 -p columnfamily=data -s > results\_Cassandra/run\_320

./bin/ycsb load cassandra-10 -P workload\_test640 -p hosts=10.128.0.5 -threads 640 -p columnfamily=data -s > results\_Cassandra/load\_640

./bin/ycsb run cassandra-10 -P workload\_test640 -p hosts=10.128.0.5 -threads 640 -p columnfamily=data -s > results\_Cassandra/run\_640

delete table