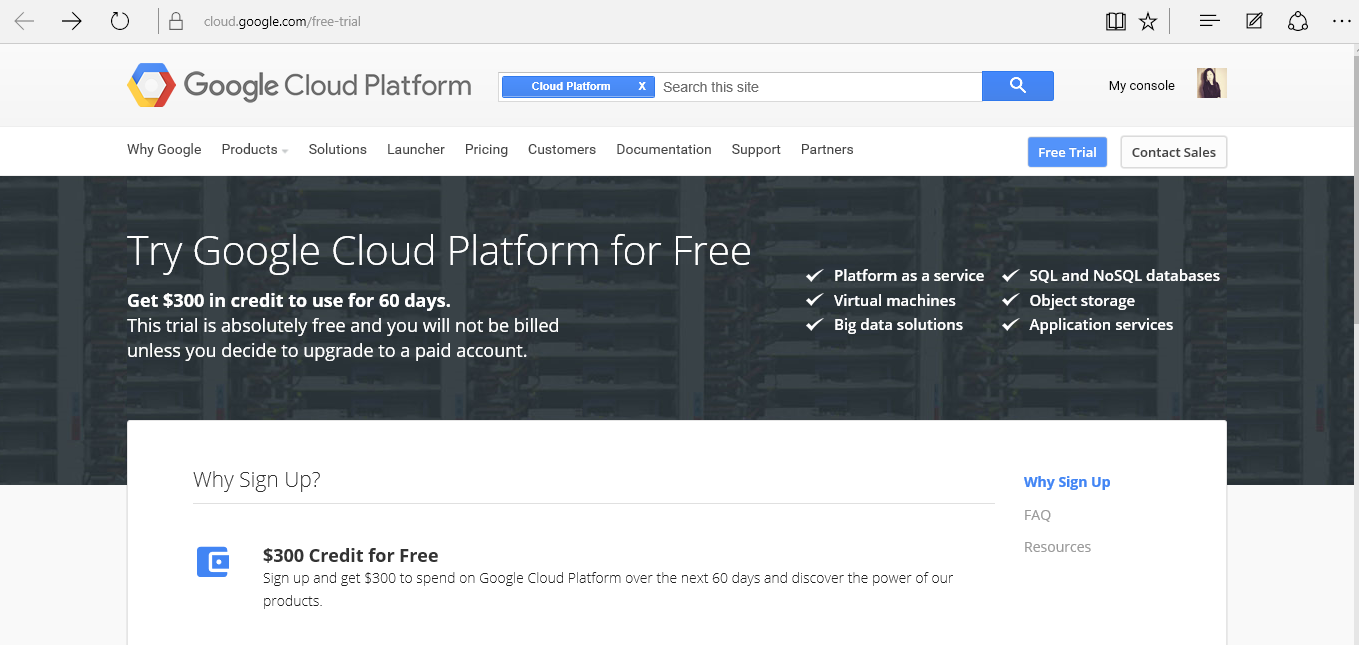
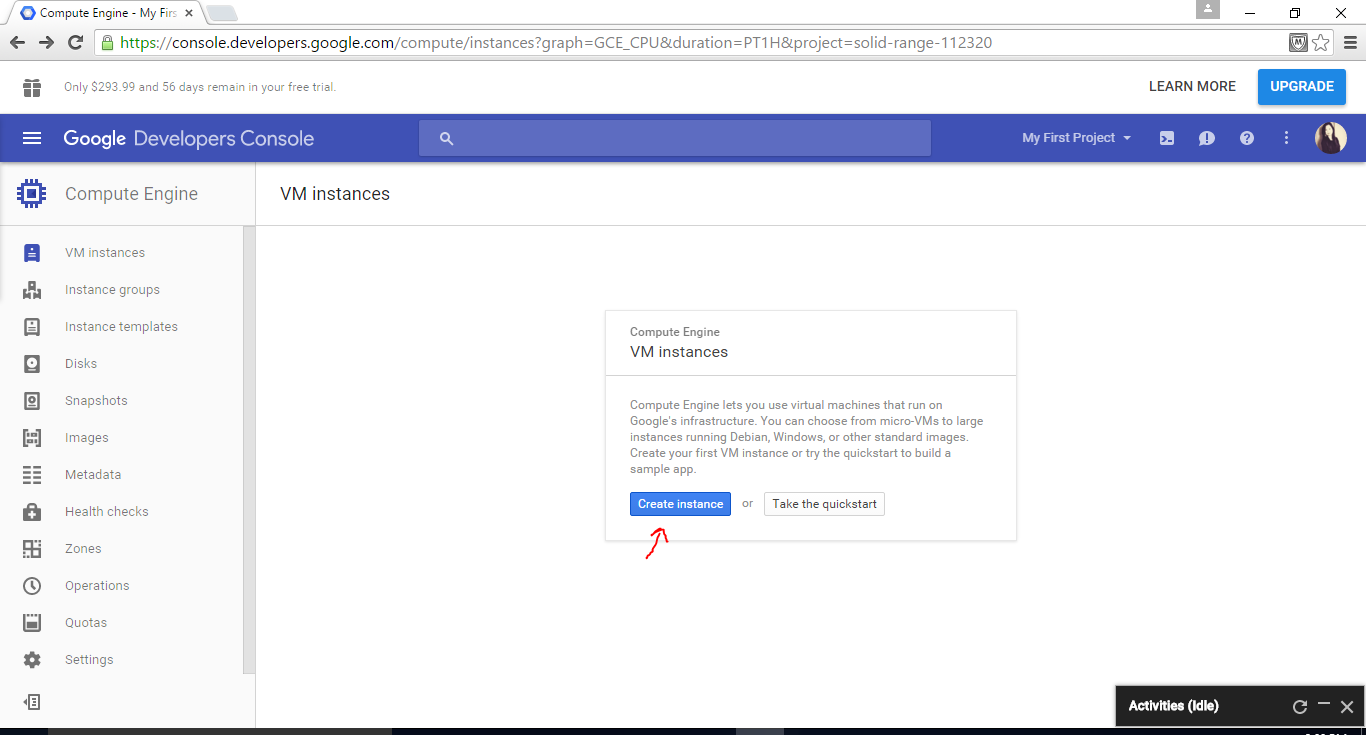
**Name: Nagma Nishat**

**Cloud Provider Google Compute Engine:**

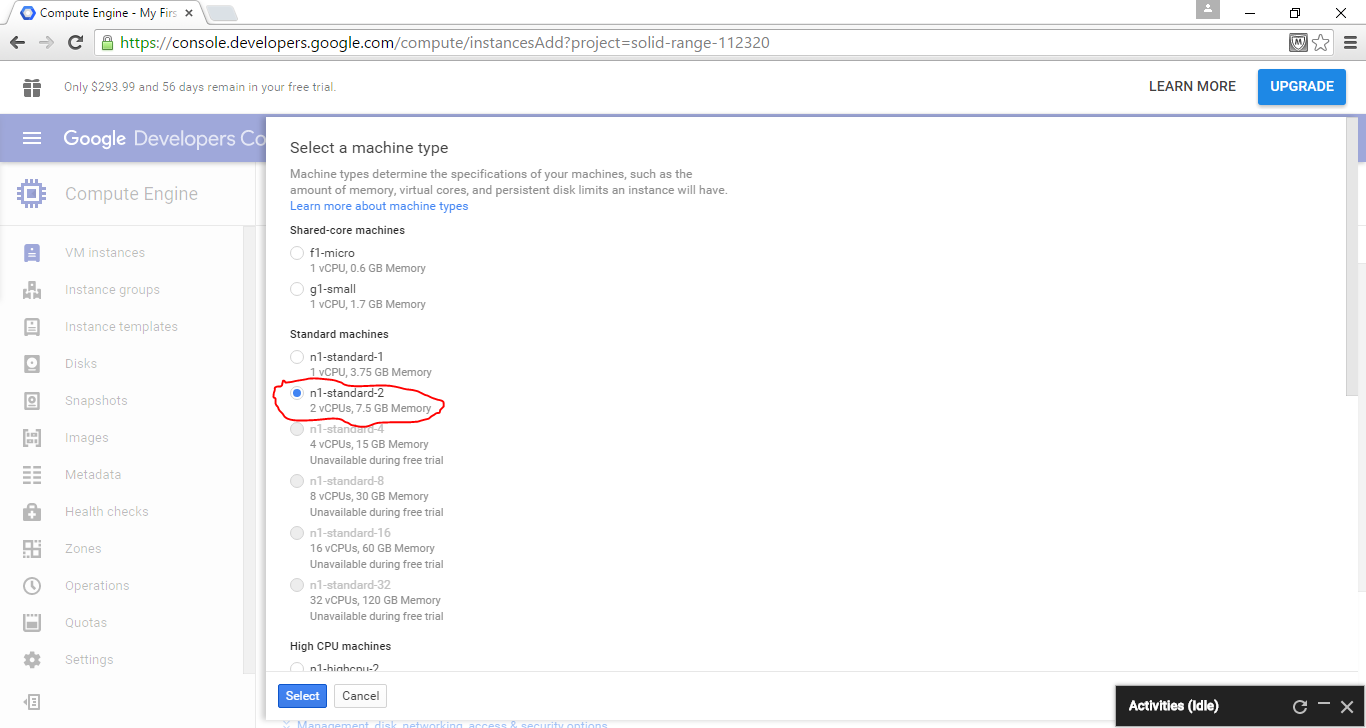
**Sign up for free trail**



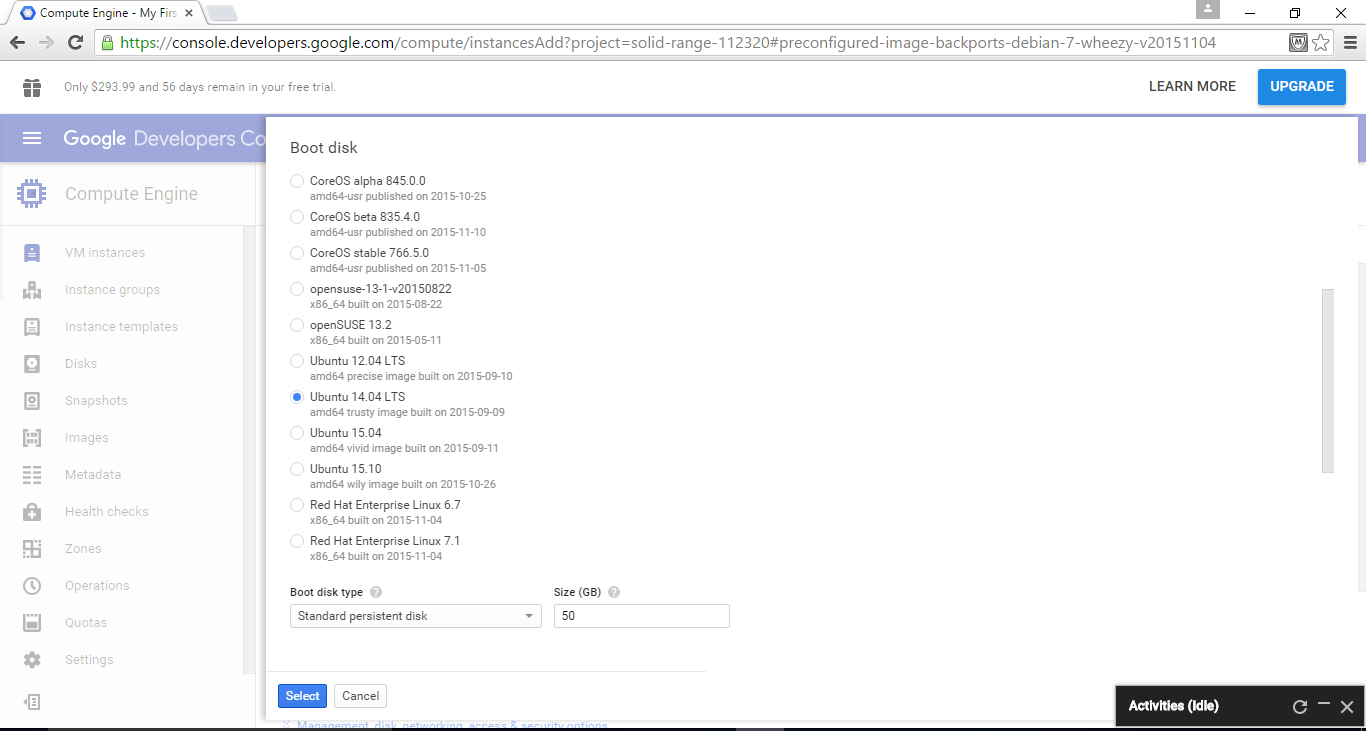
**Create instance**



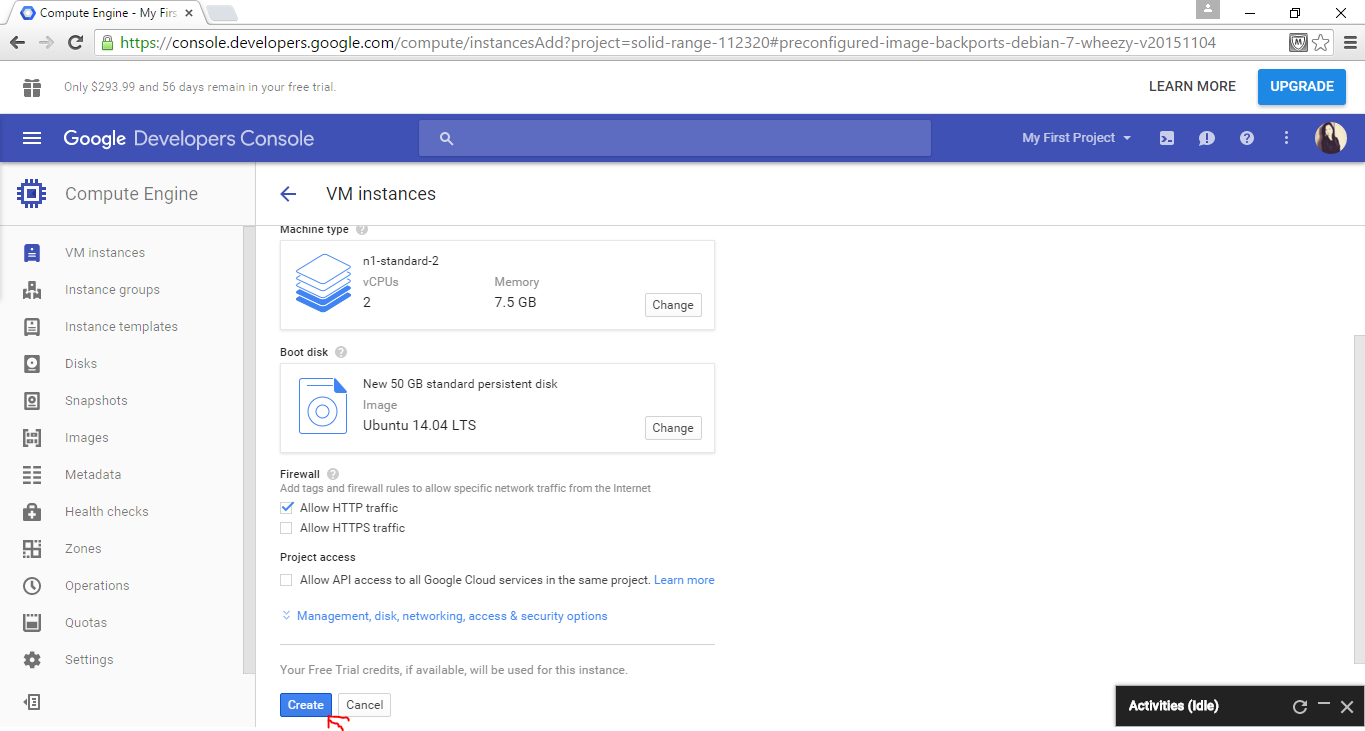
**Machine type:**



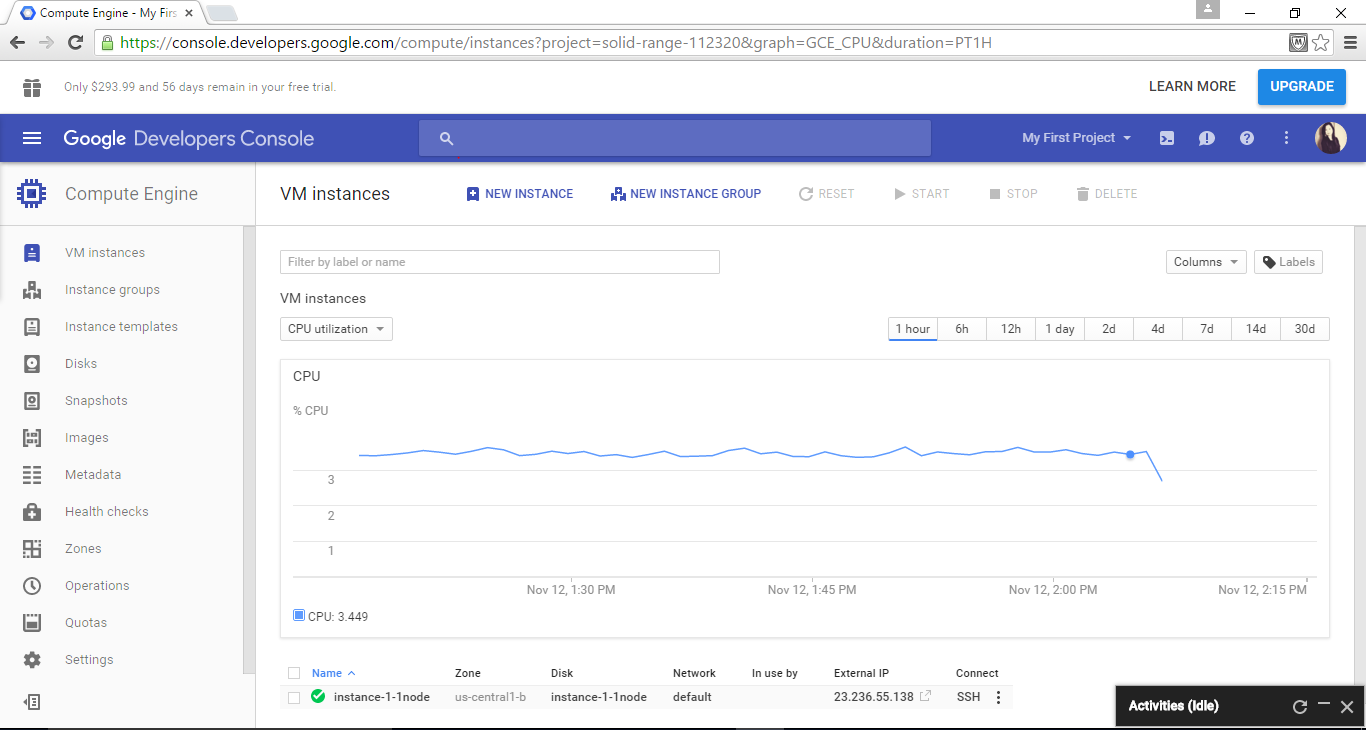
**Boot disk:**



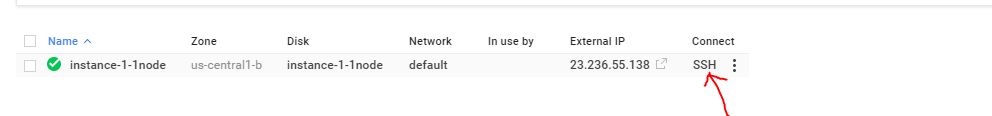
Create instance



Instance created:

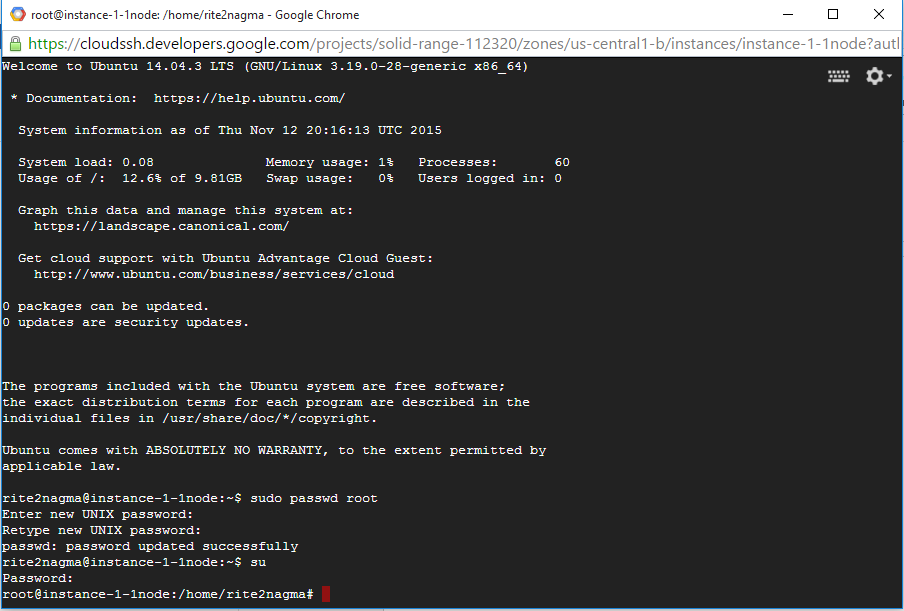


Connect using SSH



**Steps to install Cassandra 3.0.0**

**1. Change root password**  
 sudo passwd root  
 yourpw  
 su  
 yourpw



**2. Install Oracle Java 8 in Ubuntu**   
sudo add-apt-repository ppa:webupd8team/java

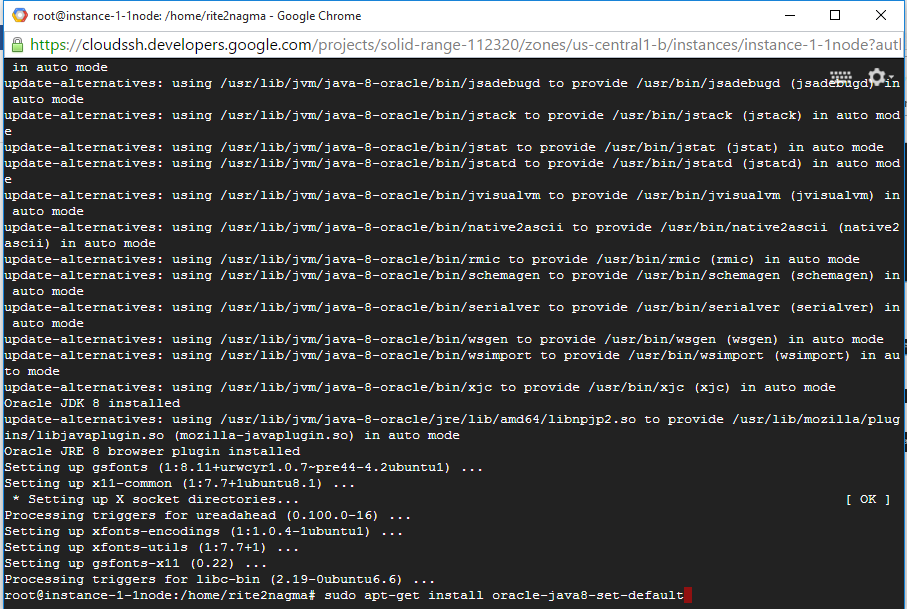
  
sudo apt-get update

  
sudo apt-get install oracle-java8-installer

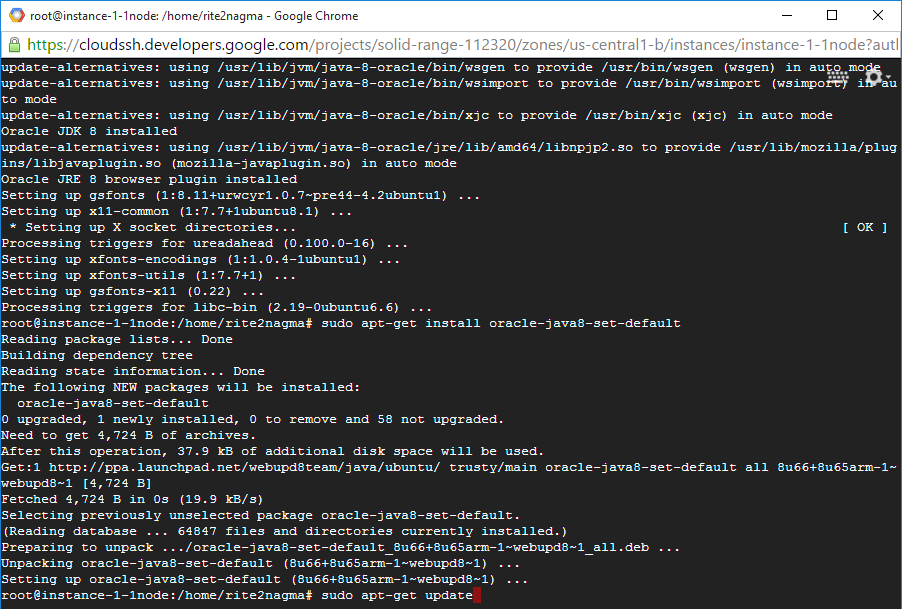


**3. Set Java environment variables**

sudo apt-get install oracle-java8-set-default



sudo apt-get update

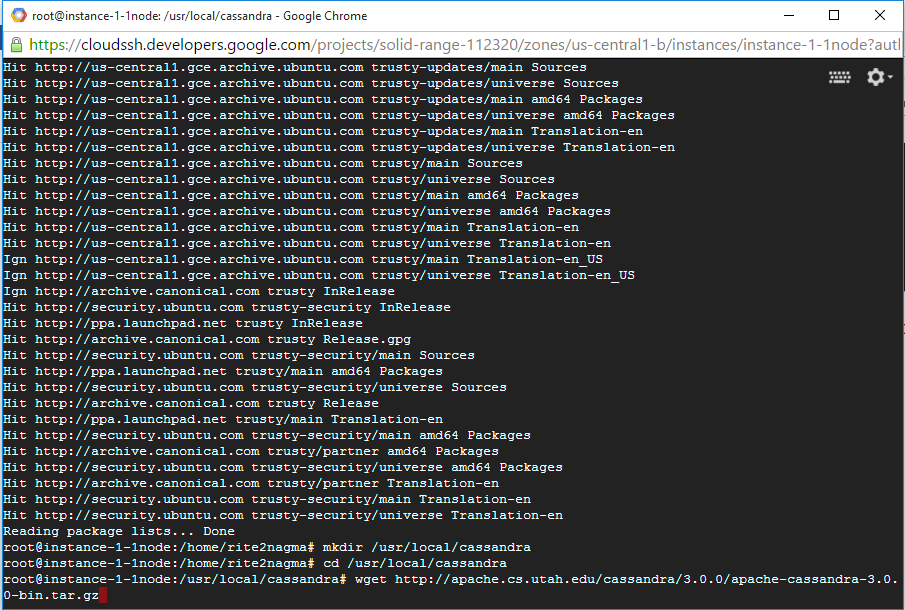


**4. Install cassandra**

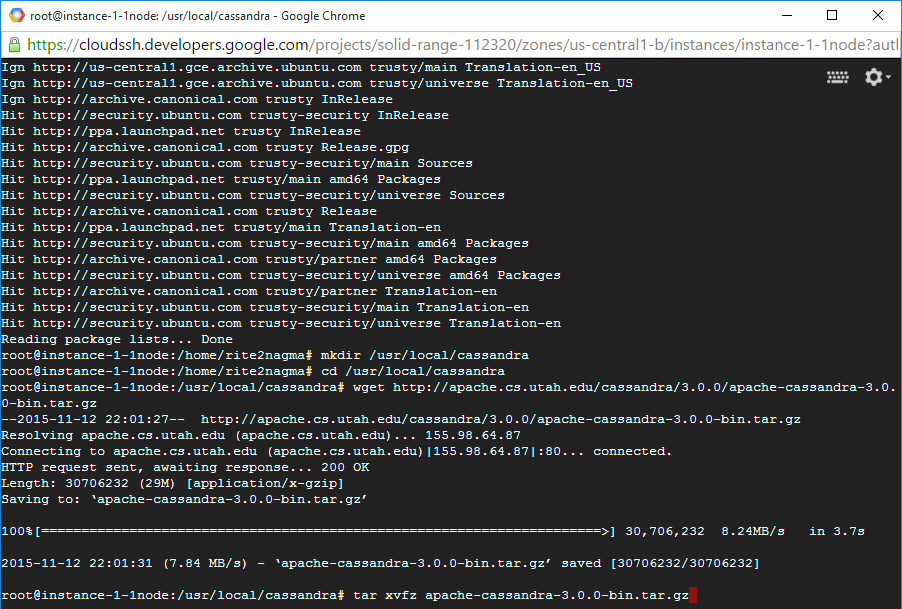
mkdir /usr/local/cassandra

cd /usr/local/cassandra

wget <http://apache.cs.utah.edu/cassandra/3.0.0/apache-cassandra-3.0.0-bin.tar.gz>

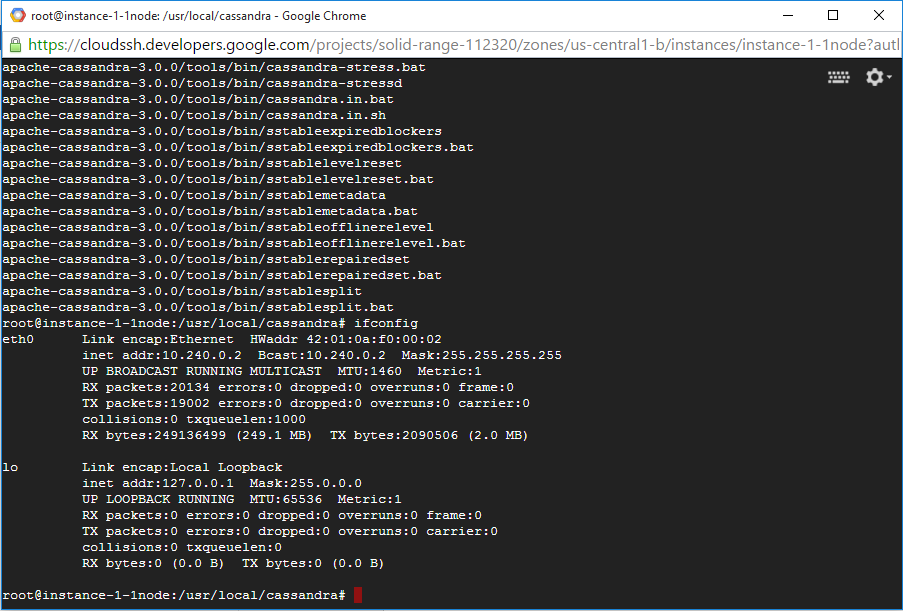


tar xvfz apache-cassandra-3.0.0-bin.tar.gz



**5. check IP addresse of instances**

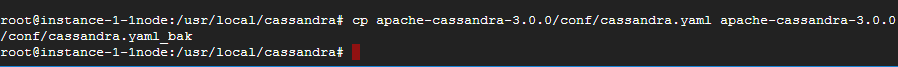
ifconfig



IP: 10.240.0.2

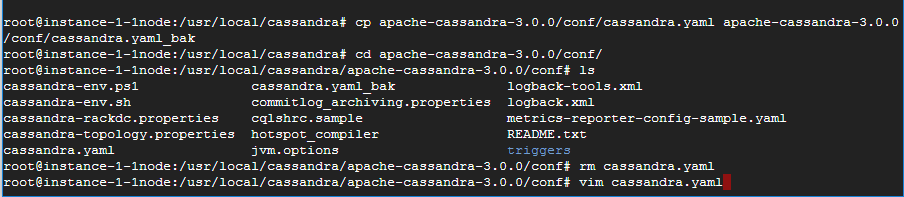
**6. Make a backup for all nodes**

cp apache-cassandra-3.0.0/conf/cassandra.yaml apache-cassandra-3.0.0/conf/cassandra.yaml\_bak



**7. Modify configuration for all nodes.**

nano apache-cassandra-3.0.0/conf/cassandra.yaml



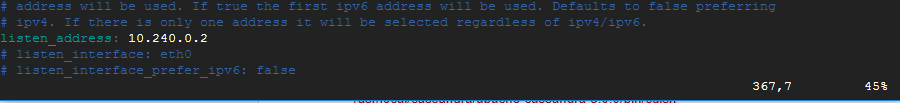
*Note:*

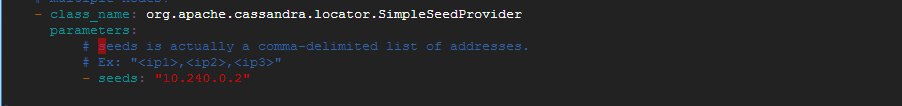
*Find listen\_address: and seeds: (Ctrl+W) and modify it*

*listen\_address should be own IP of each instance.*

listen\_address: 10.240.0.2

seeds:  "10.240.0.2"  
rpc\_address: 0.0.0.0  
broadcast\_rpc\_address: 1.2.3.4



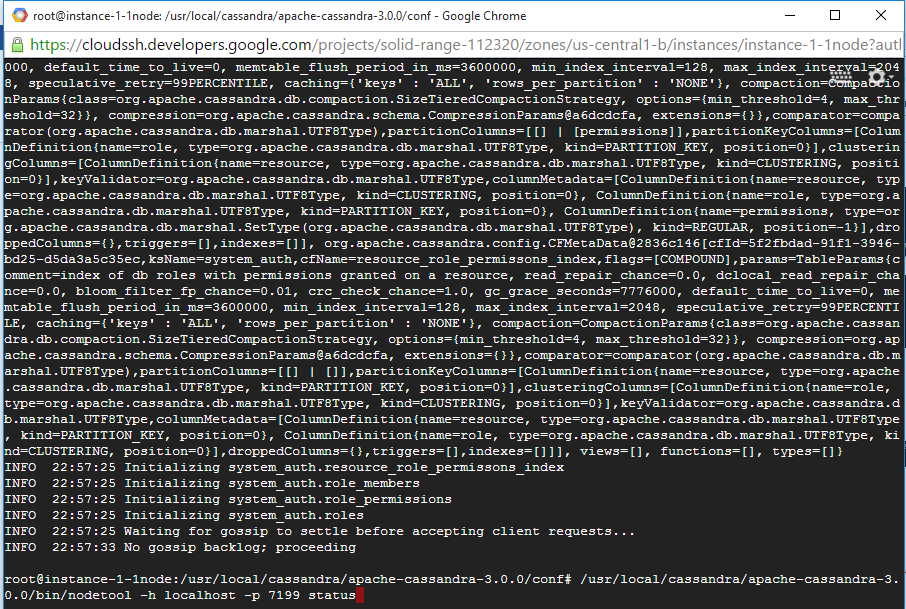


**8. Excute cassandra**

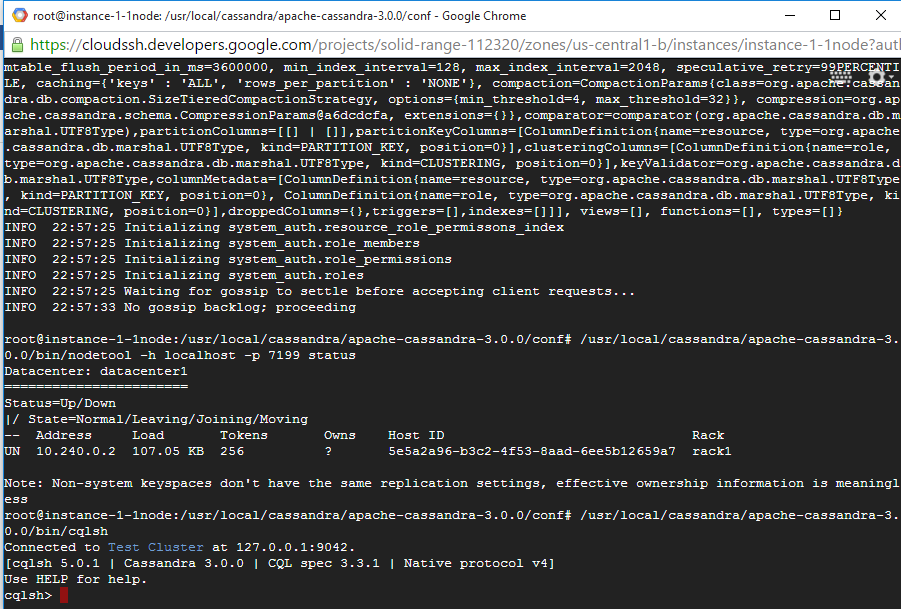
/usr/local/cassandra/apache-cassandra-3.0.0/bin/cassandra



/usr/local/cassandra/apache-cassandra-3.0.0/bin/nodetool -h localhost -p 7199 status



/usr/local/cassandra/apache-cassandra-3.0.0/bin/cqlsh

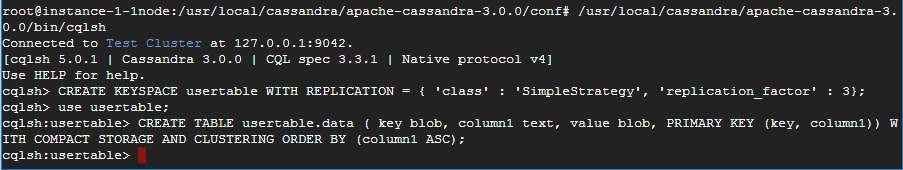


**9. Create keyspace and table for YCSB benchmark**

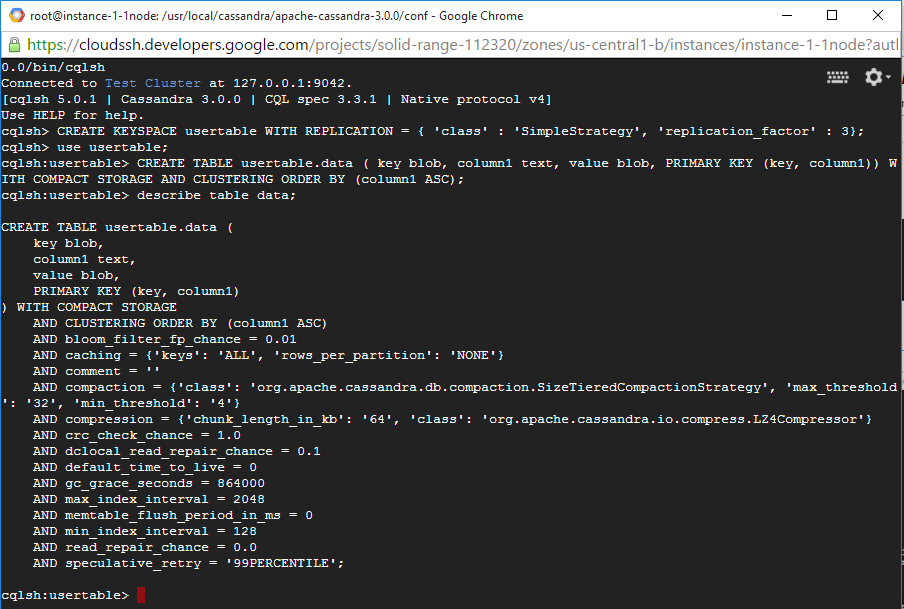
CREATE KEYSPACE usertable WITH REPLICATION = { 'class' : 'SimpleStrategy', 'replication\_factor' : 3};

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);

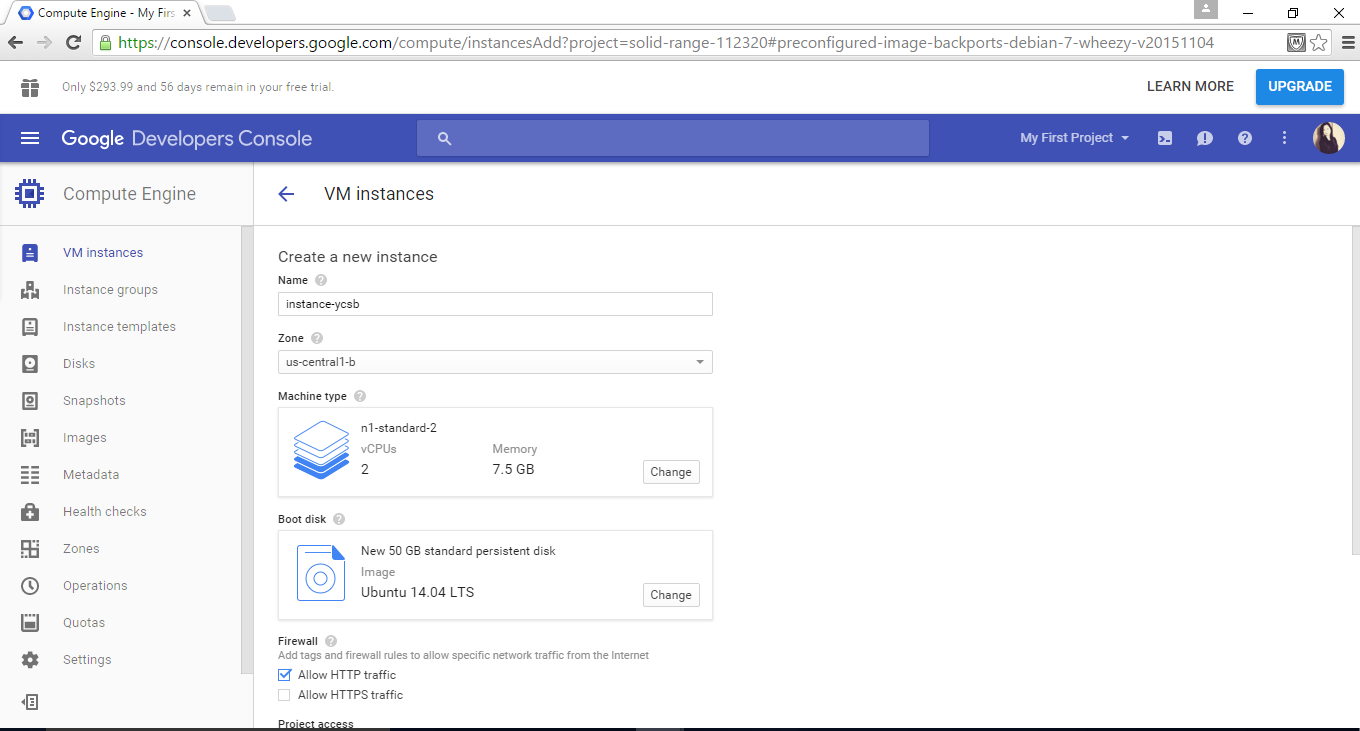


describe table data;

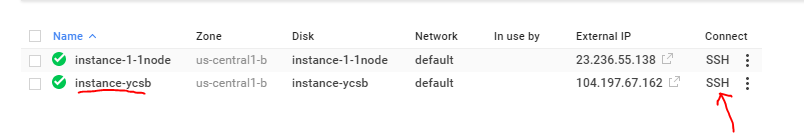


**Steps to install YCSB and benchmark Cassandra:**

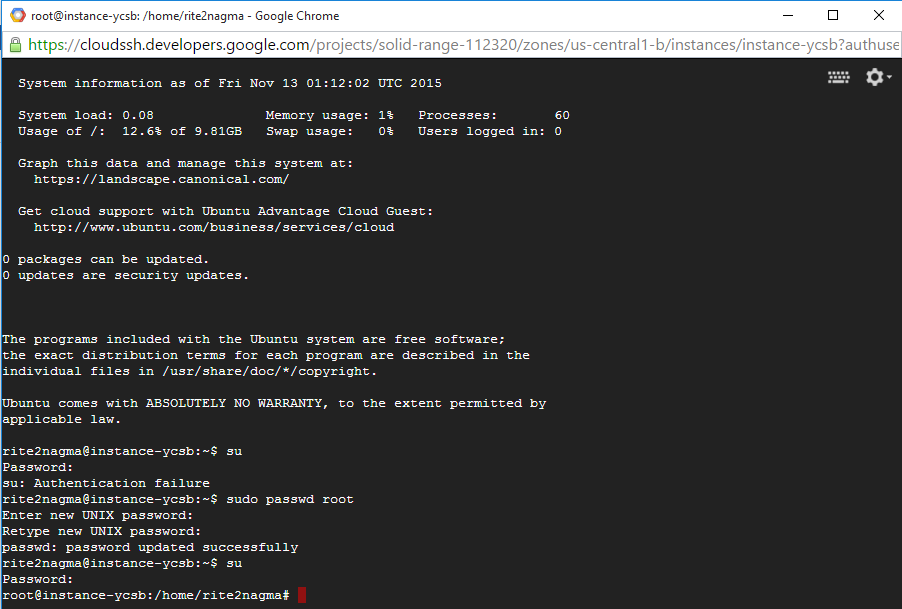
1. **Create new instance for YCSB**

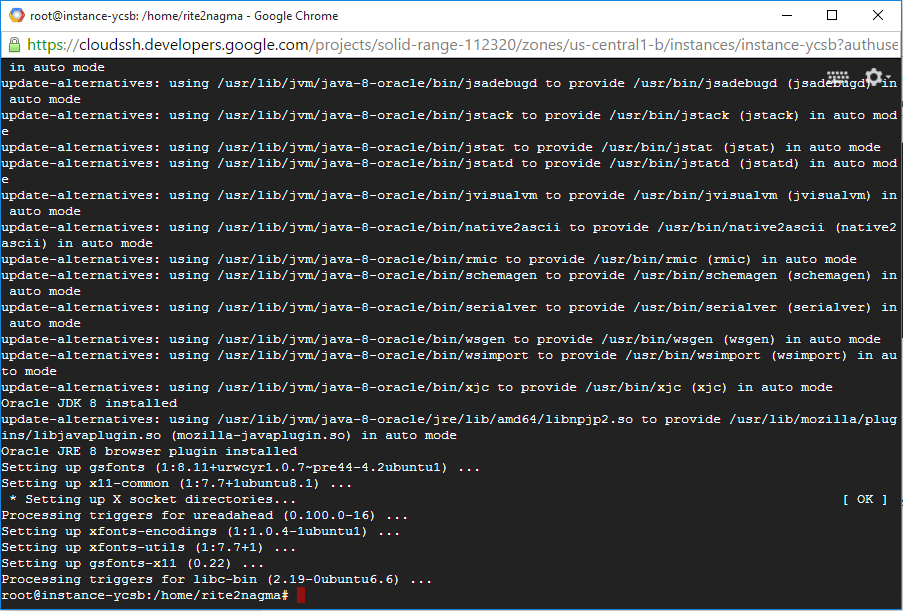


**2.Connect through SSH**



**3. Change root password**  
sudo passwd root  
yourpw  
su  
yourpw

  
**4. Install Oracle Java 8 in Ubuntu**   
sudo add-apt-repository ppa:webupd8team/java  
sudo apt-get update  
sudo apt-get install oracle-java8-installer



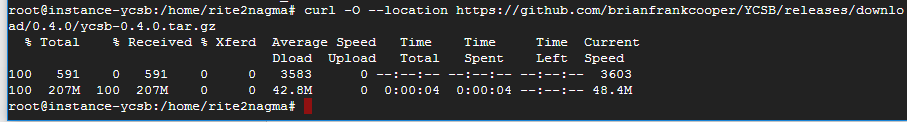
**5. Set Java environment variables**

sudo apt-get install oracle-java8-set-default

sudo apt-get update

**6. Install YCSB**

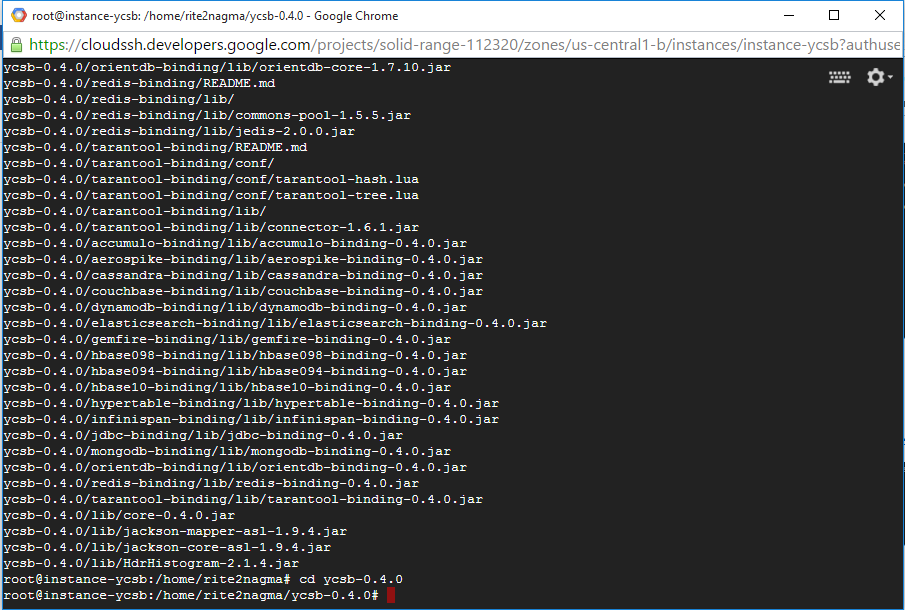
curl -O --location <https://github.com/brianfrankcooper/YCSB/releases/download/0.4.0/ycsb-0.4.0.tar.gz>



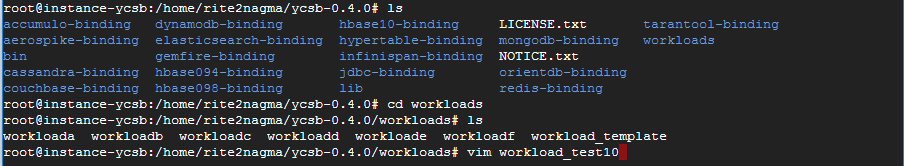
tar xfvz ycsb-0.4.0.tar.gz



cd ycsb-0.4.0



**7. Create workload files in workloads folder**



vim workload\_test10  
  
-------------------------------------------------------------  
recordcount=214748  
operationcount=100000  
workload=com.yahoo.ycsb.workloads.CoreWorkload  
readallfields=true  
readproportion=0.5435  
updateproportion=0.4565  
scanproportion=0  
insertproportion=0  
requestdistribution=zipfian

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

vim workload\_test40  
  
-------------------------------------------------------------  
recordcount=536871  
operationcount=400000  
workload=com.yahoo.ycsb.workloads.CoreWorkload  
readallfields=true  
readproportion=0.5435  
updateproportion=0.4565  
scanproportion=0  
insertproportion=0  
requestdistribution=zipfian

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

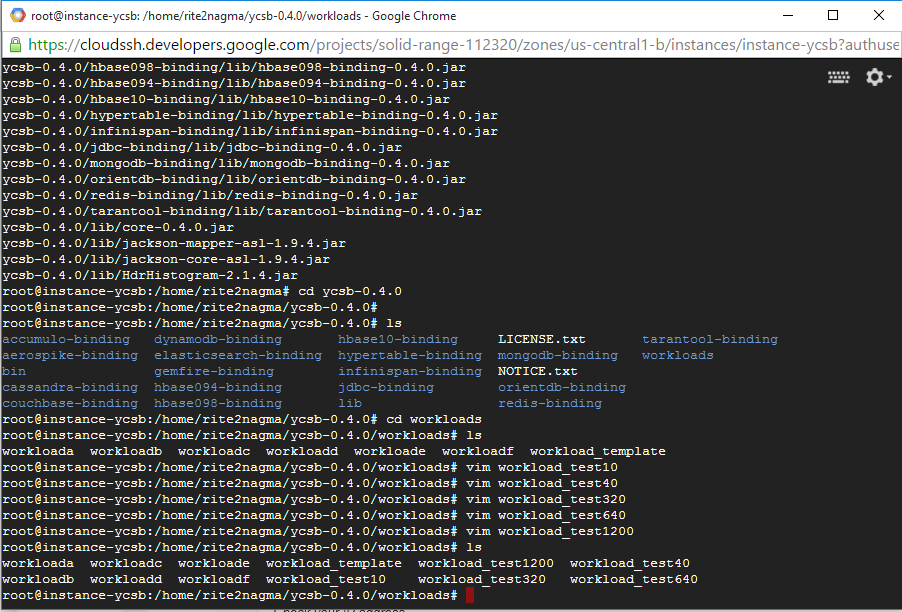
vim workload\_test320  
  
-------------------------------------------------------------  
recordcount=3543348  
operationcount=3200000  
workload=com.yahoo.ycsb.workloads.CoreWorkload  
readallfields=true  
readproportion=0.5435  
updateproportion=0.4565  
scanproportion=0  
insertproportion=0  
requestdistribution=zipfian

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

vim workload\_test640

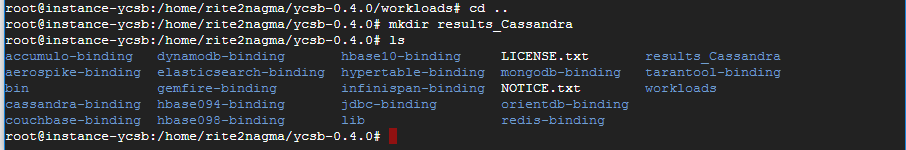
-------------------------------------------------------------  
recordcount=6120328  
operationcount=6400000  
workload=com.yahoo.ycsb.workloads.CoreWorkload  
readallfields=true  
readproportion=0.5435  
updateproportion=0.4565  
scanproportion=0  
insertproportion=0  
requestdistribution=zipfian

------------------------------------------------------------  
  
vim workload\_test1200  
  
------------------------------------------------------------  
recordcount=10630044  
operationcount=12000000  
workload=com.yahoo.ycsb.workloads.CoreWorkload  
readallfields=true  
readproportion=0.5435  
updateproportion=0.4565  
scanproportion=0  
insertproportion=0  
requestdistribution=zipfian  
------------------------------------------------------------



**8. Create a folder for results**

mkdir results\_Cassandra



*//Note: benchmark for 10 concurrent users*

**9. Load data**

./bin/ycsb load cassandra-10 -P workloads/workload\_test10 -p hosts=10.240.0.2 -threads 10 -p columnfamily=data -s > results\_Cassandra/load\_10



*Note:*

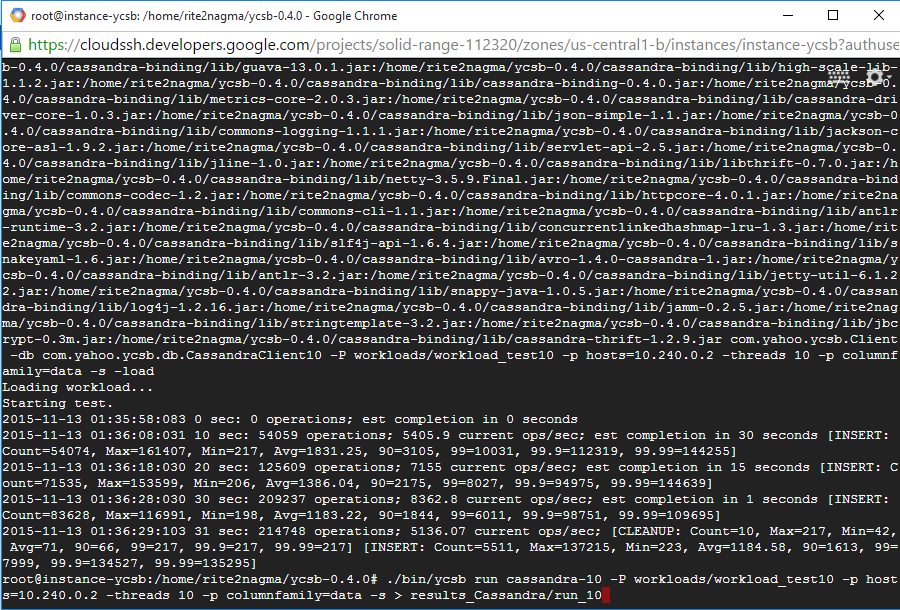
*Check your IP address and run just one commend for each experiment.*

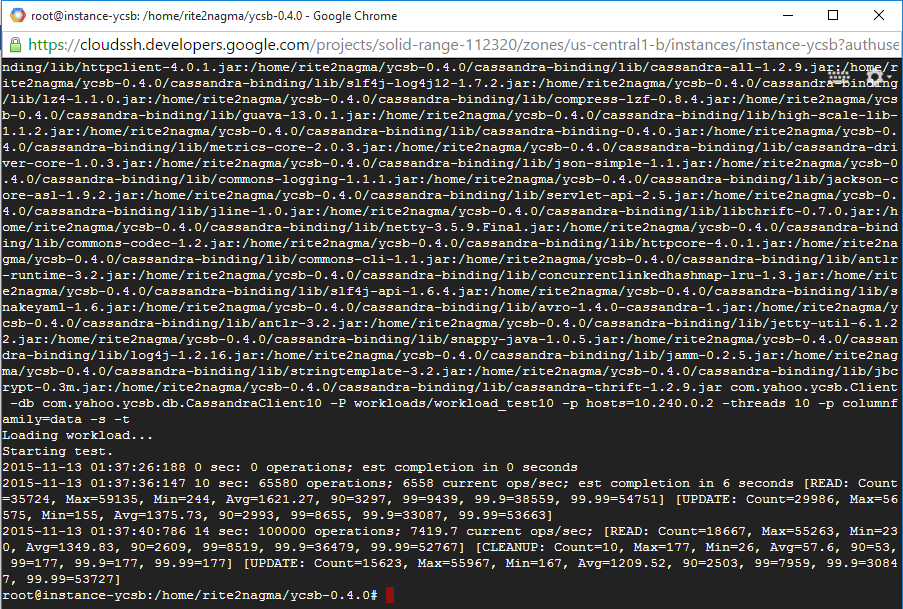
**10. Run benchmark**

./bin/ycsb run cassandra-10 -P workloads/workload\_test10 -p hosts=10.240.0.2 -threads 10 -p columnfamily=data -s > results\_Cassandra/run\_10

*Note:*

*Check your IP address*

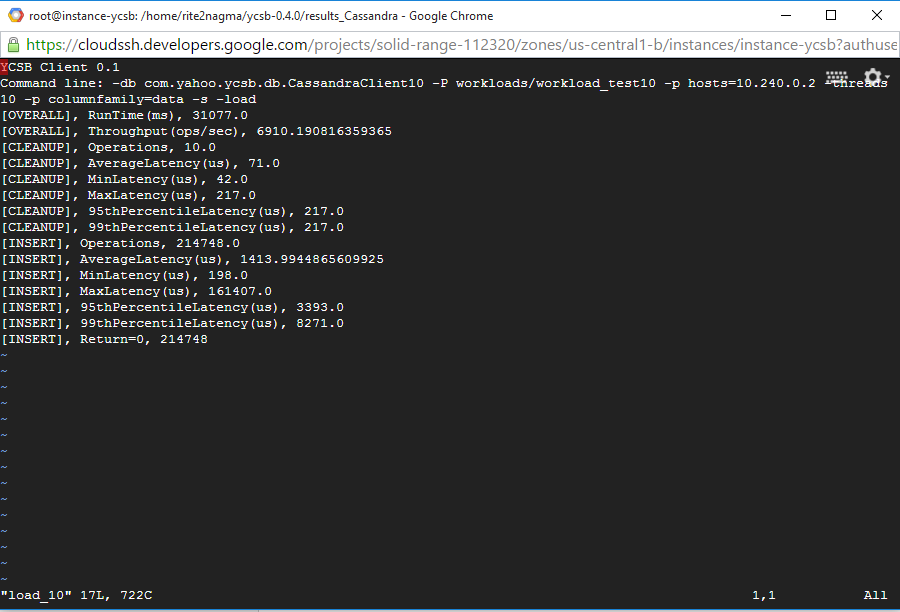




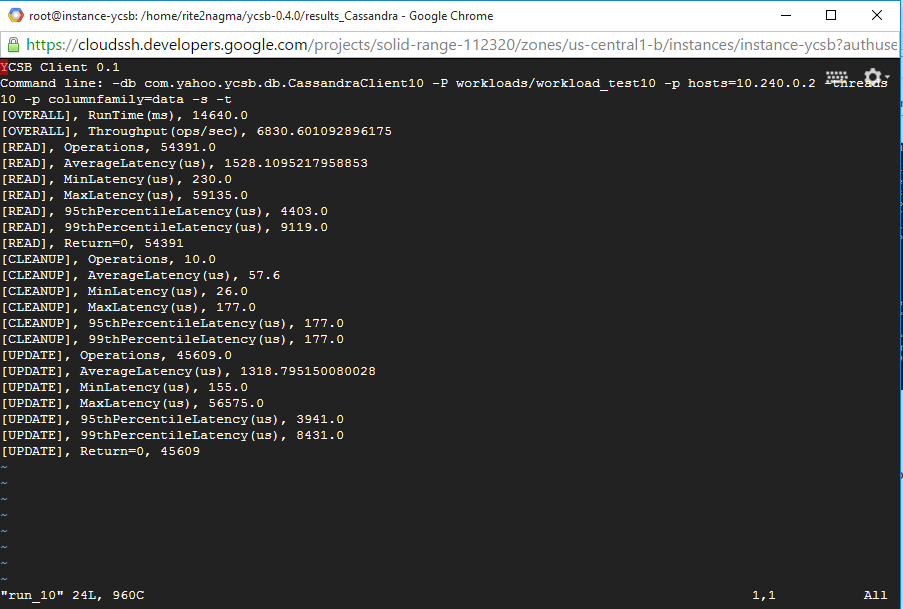
**load\_10 and run\_10 files are generated in directory results\_Cassandra**



**Load\_10 file**



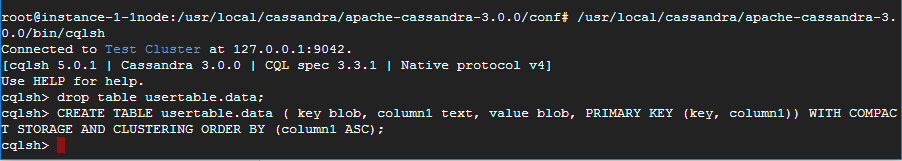
**Run\_10 file**



**11. Clean loaded data for next benchmark**

/usr/local/cassandra/apache-cassandra-3.0.0/bin/cqlsh

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



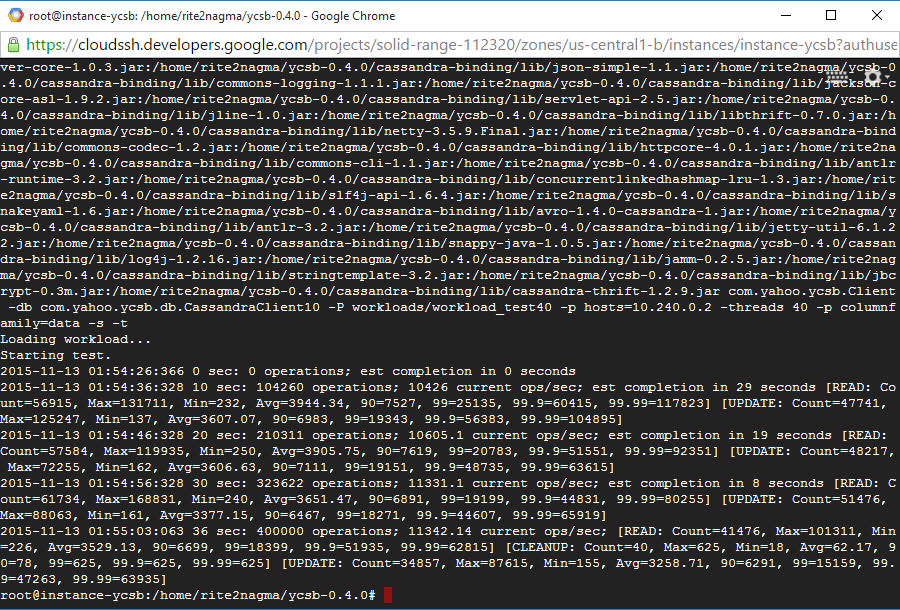
**11. Repeat the above steps for benchmark with different configurations.**

//Note: benchmark for 40 concurrent users

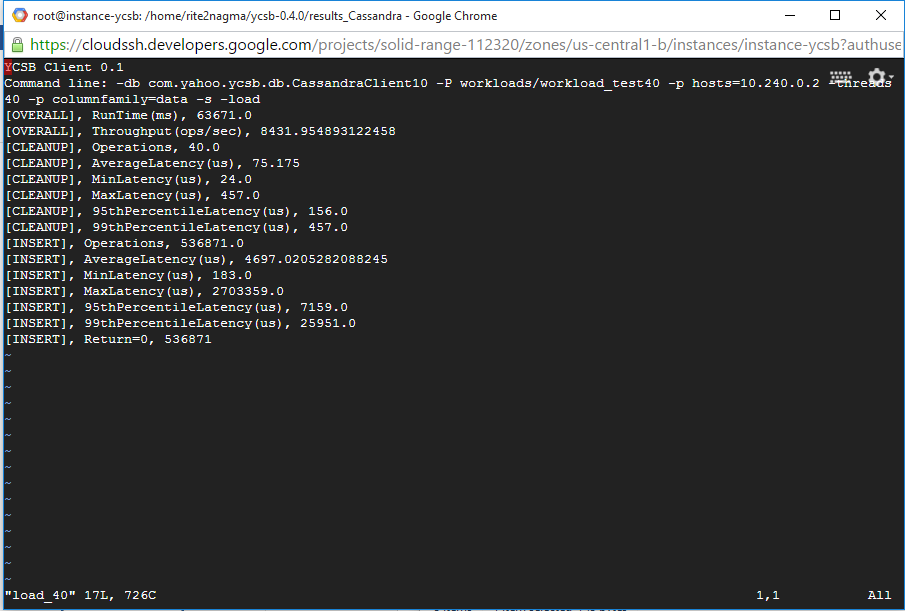
./bin/ycsb load cassandra-10 -P workloads/workload\_test40 -p hosts=10.240.0.2 -threads 40 -p columnfamily=data -s > results\_Cassandra/load\_40

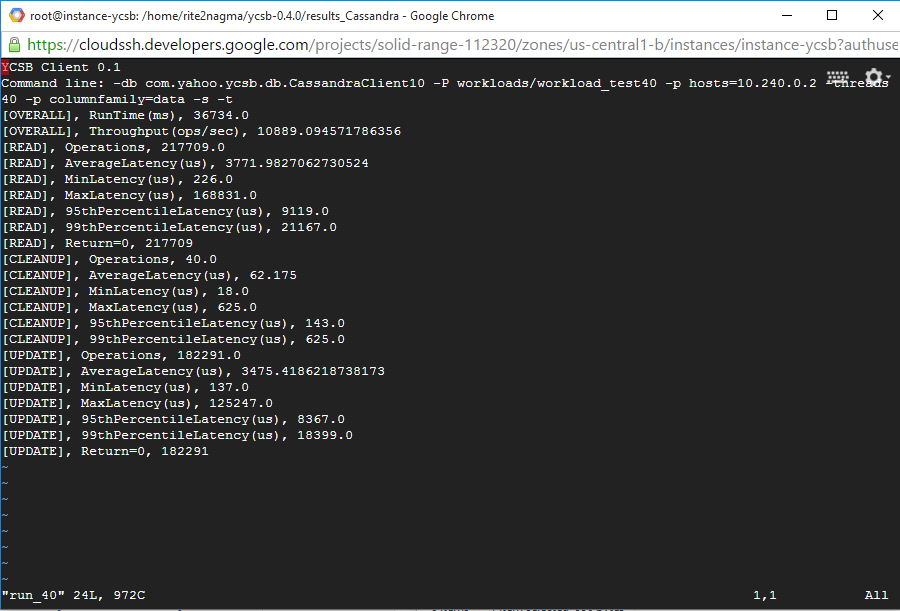


./bin/ycsb run cassandra-10 -P workloads/workload\_test40 -p hosts=10.240.0.2 -threads 40 -p columnfamily=data -s > results\_Cassandra/run\_40



**Load\_40 and run\_40 files generated:**



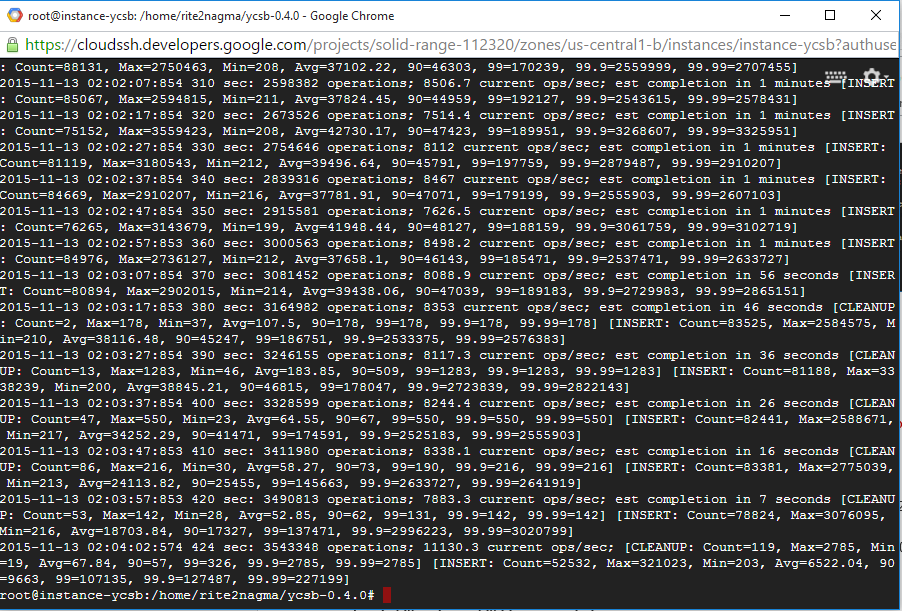


**Clean loaded data for next benchmark**

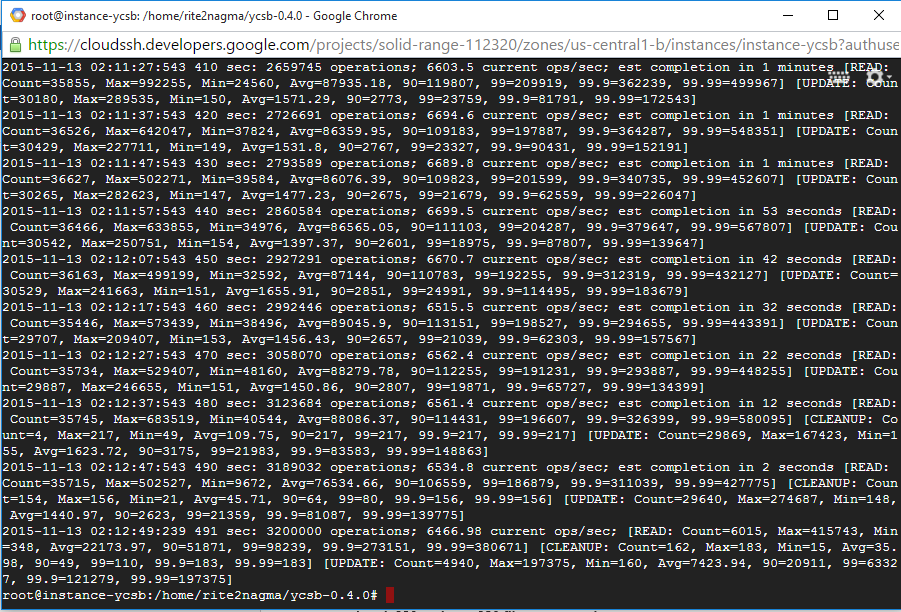
drop table usertable.data;  
CREATE TABLE usertable.data ( key blob, column1 text, value blob, PRIMARY KEY (key, column1)) WITH COMPACT STORAGE AND CLUSTERING ORDER BY (column1 ASC);

//Note: benchmark for 320 concurrent users

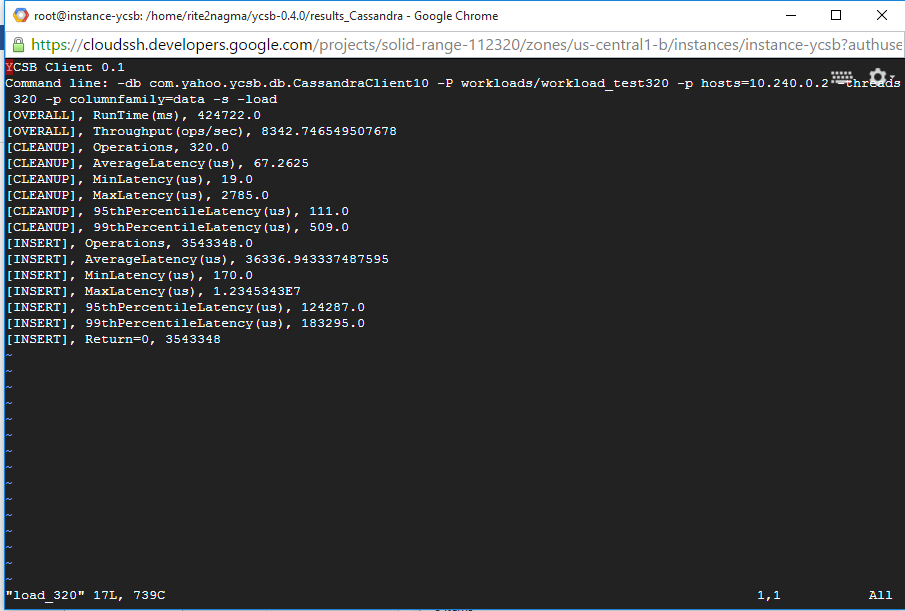
./bin/ycsb load cassandra-10 -P workloads/workload\_test320 -p hosts=10.240.0.2 -threads 320 -p columnfamily=data -s > results\_Cassandra/load\_320

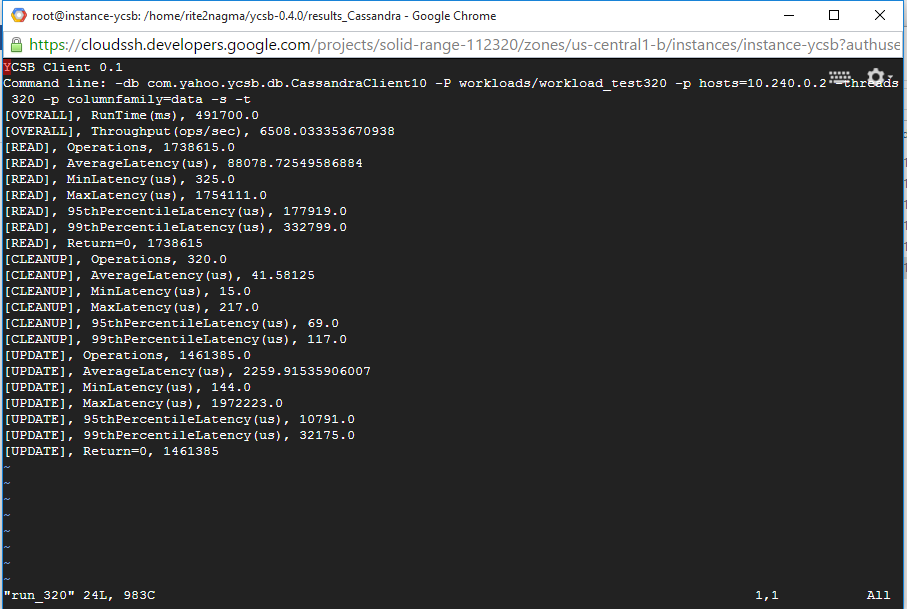


./bin/ycsb run cassandra-10 -P workloads/workload\_test320 -p hosts=10.240.0.2 -threads 320 -p columnfamily=data -s > results\_Cassandra/run\_320



**Load\_320 and run\_320 files generated:**



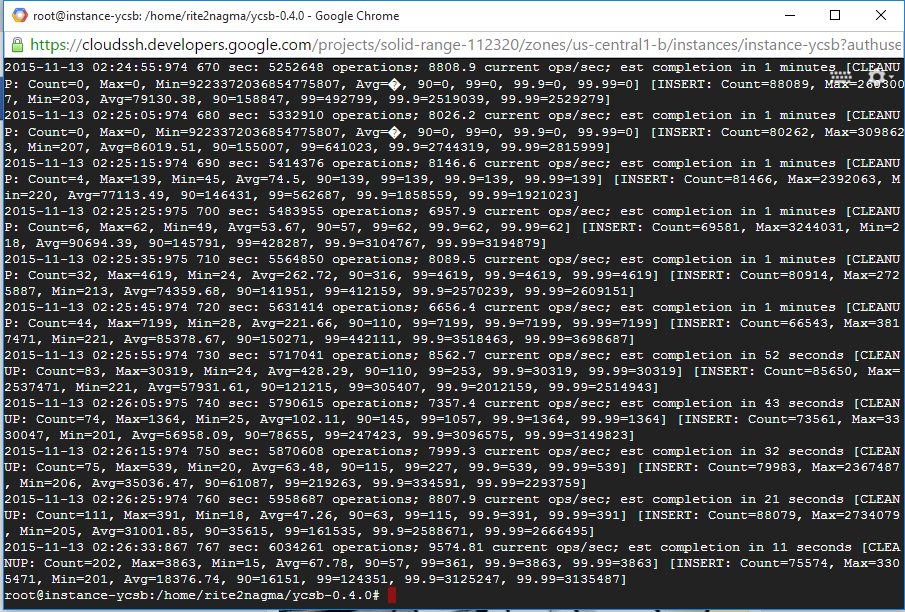


**Clean loaded data for next benchmark**

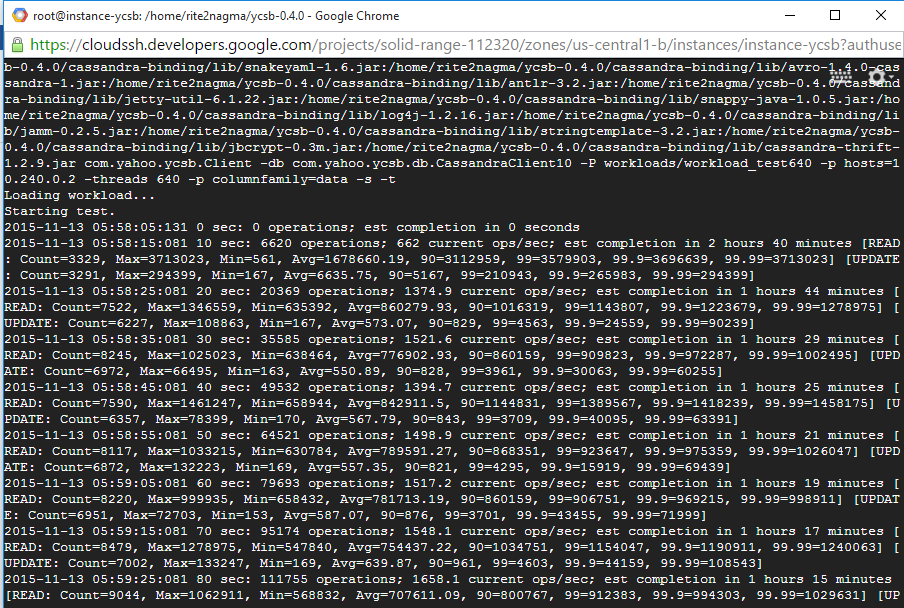
drop table usertable.data;  
CREATE TABLE usertable.data ( key blob, column1 text, value blob, PRIMARY KEY (key, column1)) WITH COMPACT STORAGE AND CLUSTERING ORDER BY (column1 ASC);

//Note: benchmark for 640 concurrent users

./bin/ycsb load cassandra-10 -P workloads/workload\_test640 -p hosts=10.240.0.2 -threads 640 -p columnfamily=data -s > results\_Cassandra/load\_640



./bin/ycsb run cassandra-10 -P workloads/workload\_test640 -p hosts=10.240.0.2 -threads 640 -p columnfamily=data -s > results\_Cassandra/run\_640



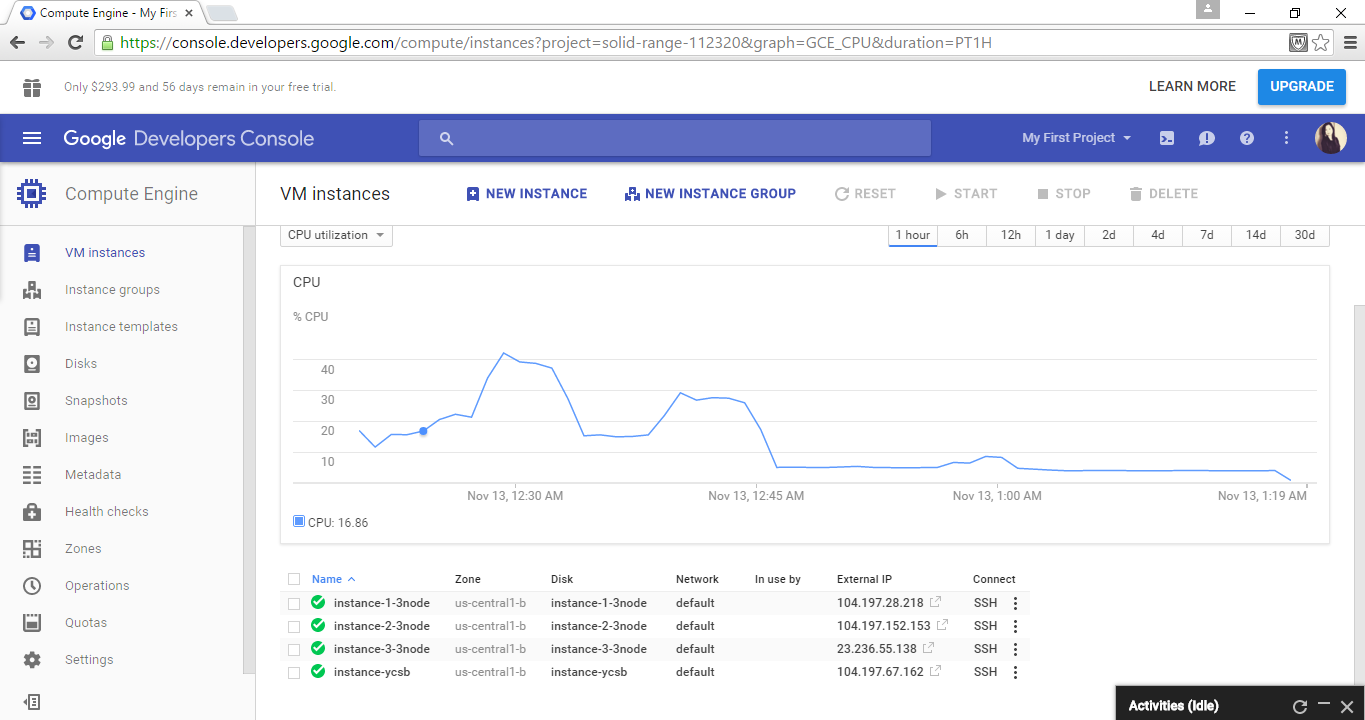
*Benchmark for 640 users taking more time to run. So we cancel the process.*

**Clean loaded data for next benchmark**

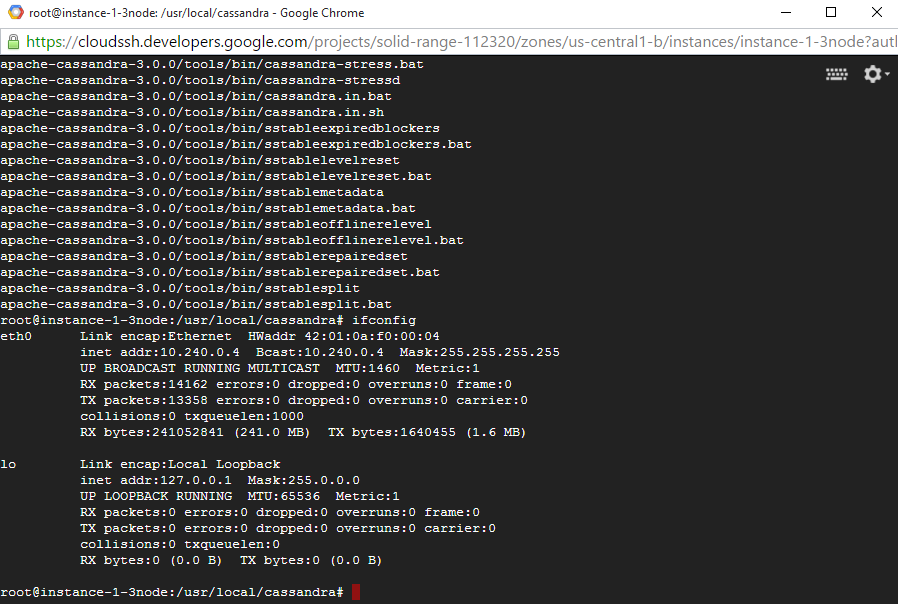
drop table usertable.data;  
CREATE TABLE usertable.data ( key blob, column1 text, value blob, PRIMARY KEY (key, column1)) WITH COMPACT STORAGE AND CLUSTERING ORDER BY (column1 ASC);

**Steps to Benchmark Cassandra with 3 nodes:**

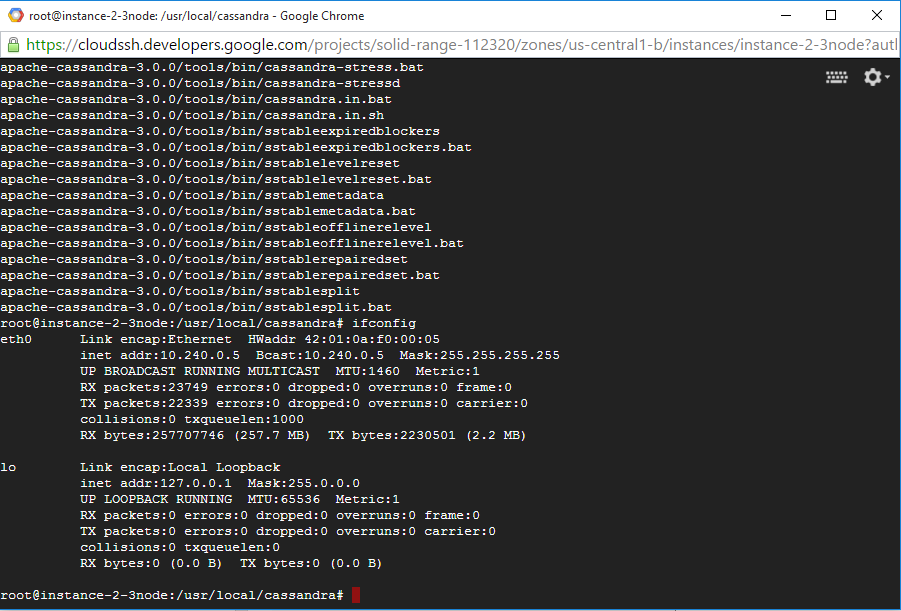
**Create 3 instances for running 3 nodes:**



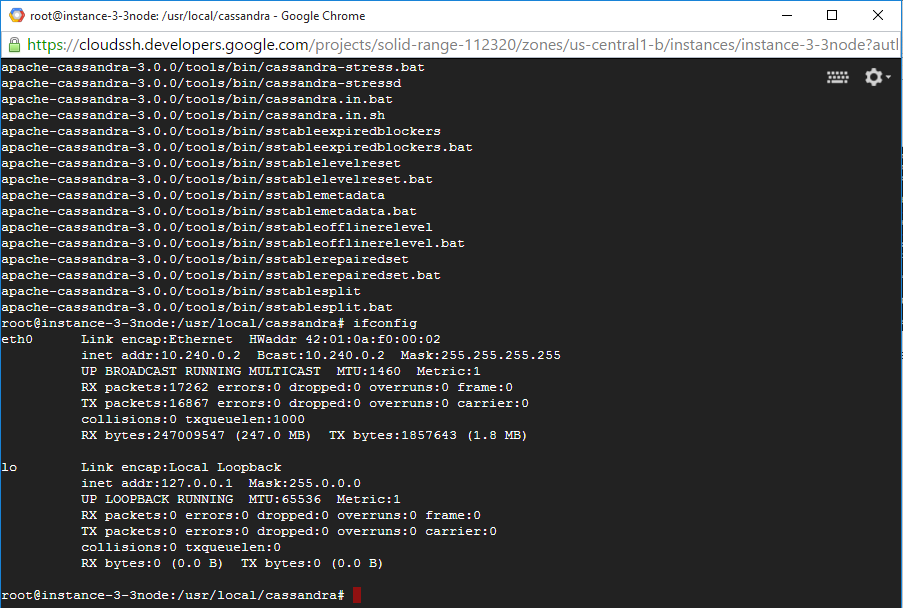
Install Java and Cassandra on all the three instances (follow the same steps as done for 1 node above)



IP for 1st node: 10.240.0.4



IP for 2nd node: 10.240.0.5

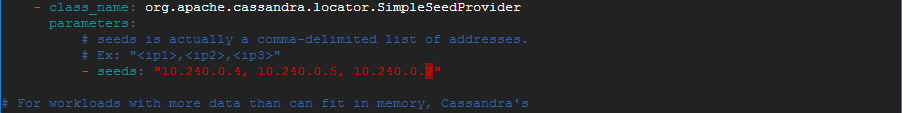


IP 3rd node : 10.240.0.2

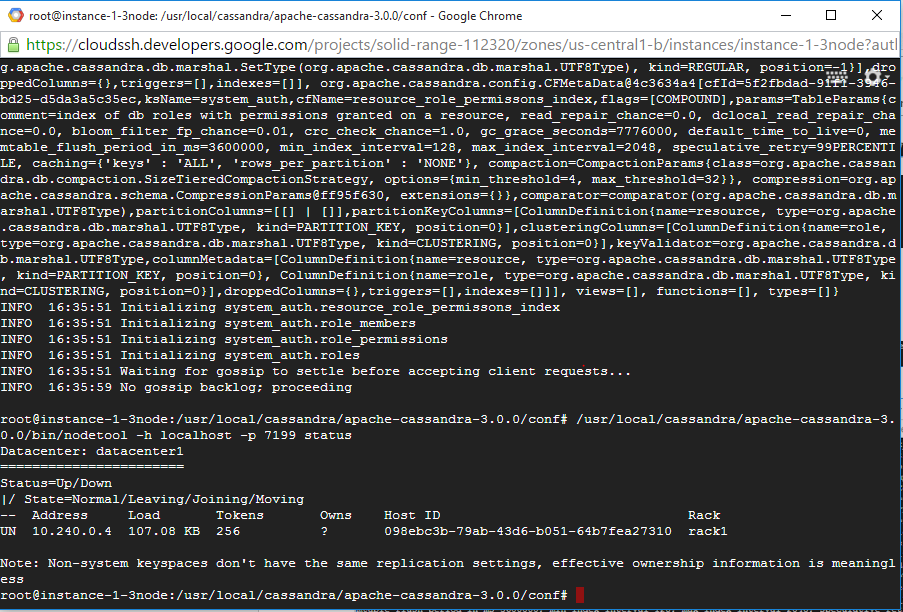
Open file cassandra.yaml in all the three nodes Find listen\_address: and seeds: (Ctrl+W) and modify it.

listen\_address should be own IP of each instance.

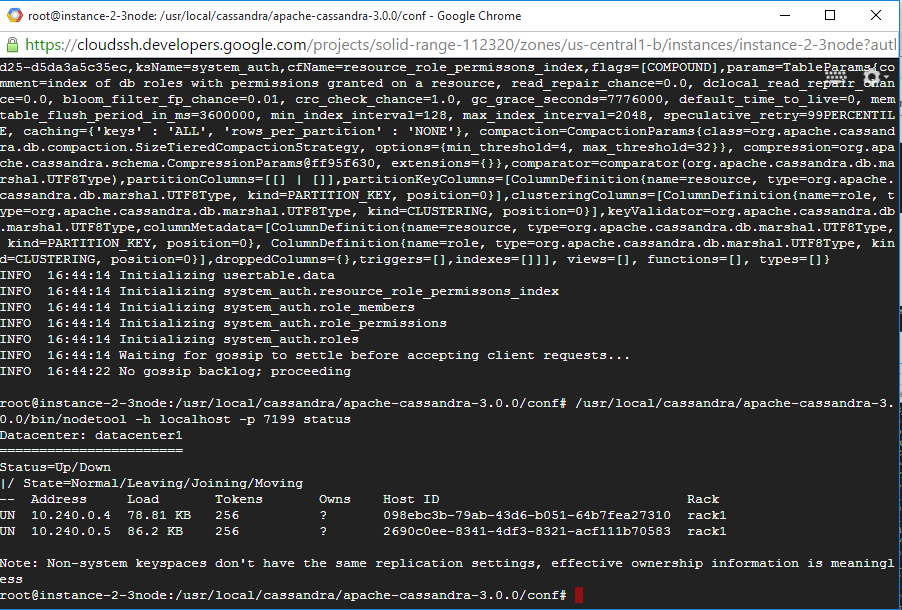
Seeds should be IP of all the three nodes i.e. “10.240.0.4, 10.240.0.5, 10.240.0.2”



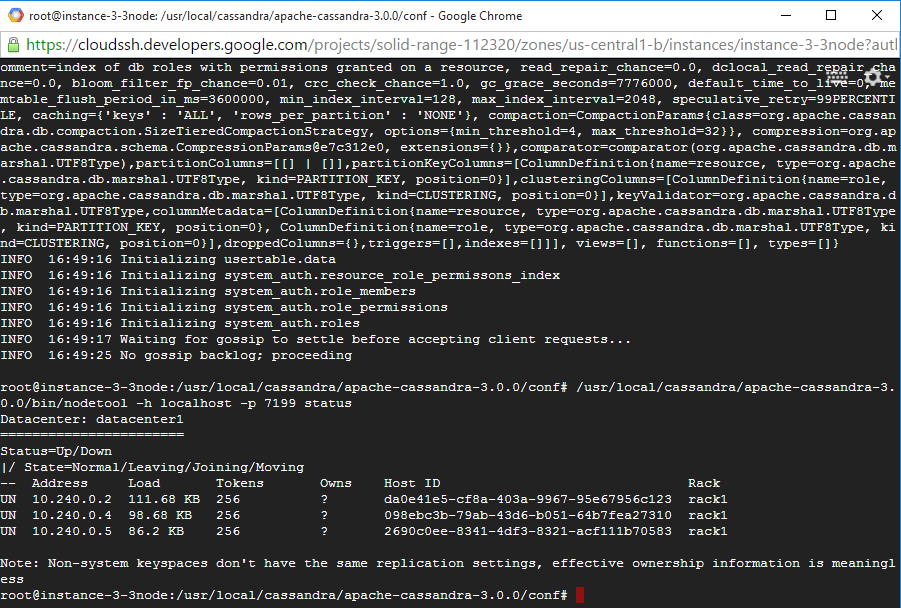
Cassandra installed for node 1:



Cassandra installed for node 2:



Cassandra installed for node 3:

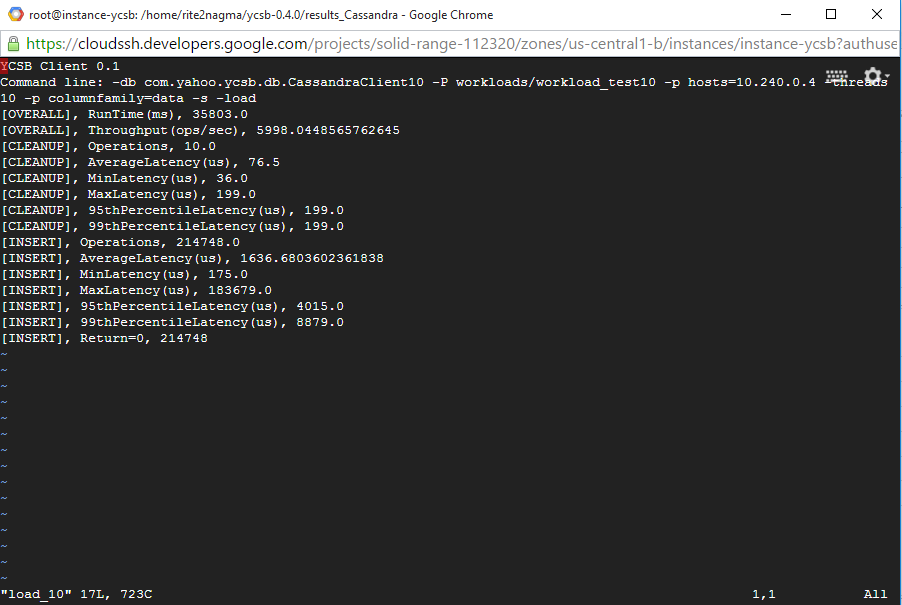


**Run YCSB and benchmark Cassandra for 3 node:**

**Load data**

./bin/ycsb load cassandra-10 -P workloads/workload\_test10 -p hosts=10.240.0.4 -threads 10 -p columnfamily=data -s > results\_Cassandra/load\_10

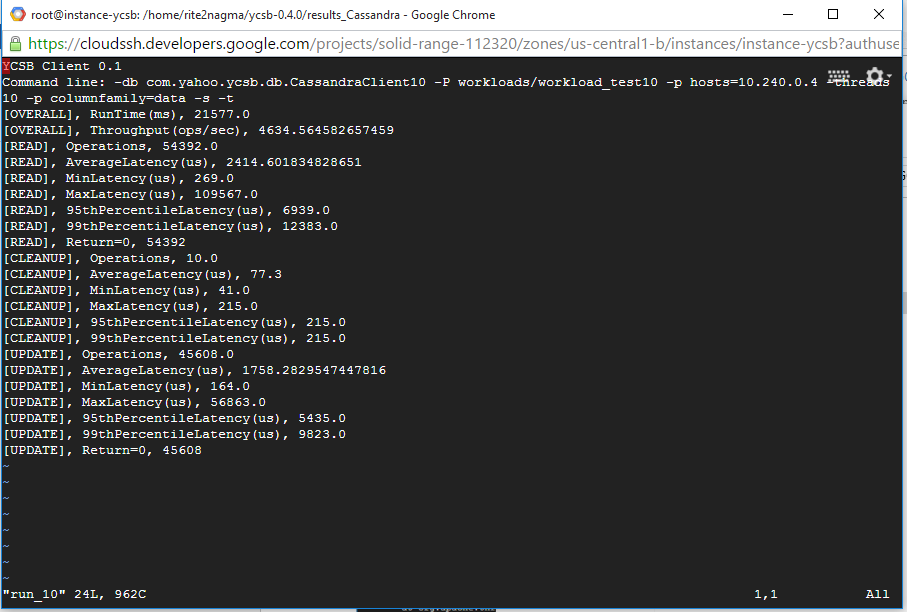
**Load\_10 file generated:**



**10. Run benchmark for 10 concurrent users**

./bin/ycsb run cassandra-10 -P workloads/workload\_test10 -p hosts=10.240.0.4 -threads 10 -p columnfamily=data -s > results\_Cassandra/run\_10

**Run\_10 file generated:**



*Note:*

*Check your IP address*

**Clean loaded data for next benchmark**

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

**Repeat benchmark with different configurations.**

*//Note: benchmark for 40 concurrent users*

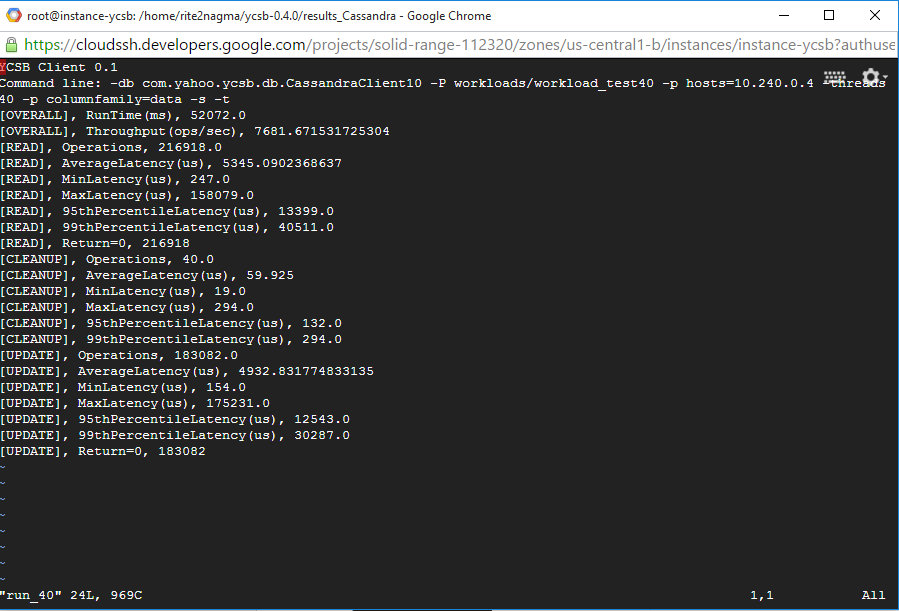
./bin/ycsb load cassandra-10 -P workloads/workload\_test40 -p hosts=10.240.0.4 -threads 40 -p columnfamily=data -s > results\_Cassandra/load\_40

**Load\_40 file generated:**



./bin/ycsb run cassandra-10 -P workloads/workload\_test40 -p hosts=10.240.0.4 -threads 40 -p columnfamily=data -s > results\_Cassandra/run\_40

**Run\_40 file generated:**

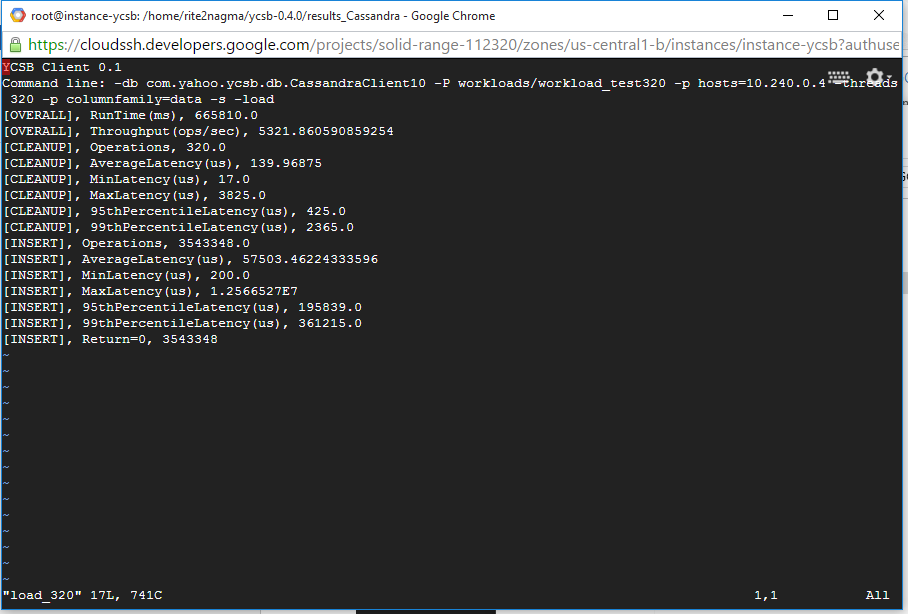


drop table data;  
CREATE TABLE usertable.data ( key blob, column1 text, value blob, PRIMARY KEY (key, column1)) WITH COMPACT STORAGE AND CLUSTERING ORDER BY (column1 ASC);

*//Note: benchmark for 320 concurrent users*

./bin/ycsb load cassandra-10 -P workloads/workload\_test320 -p hosts=10.240.0.4 -threads 320 -p columnfamily=data -s > results\_Cassandra/load\_320

**Load\_320 file generated:**



./bin/ycsb run cassandra-10 -P workloads/workload\_test320 -p hosts=10.240.0.4 -threads 320 -p columnfamily=data -s > results\_Cassandra/run\_320

**Run\_320 file generated:**



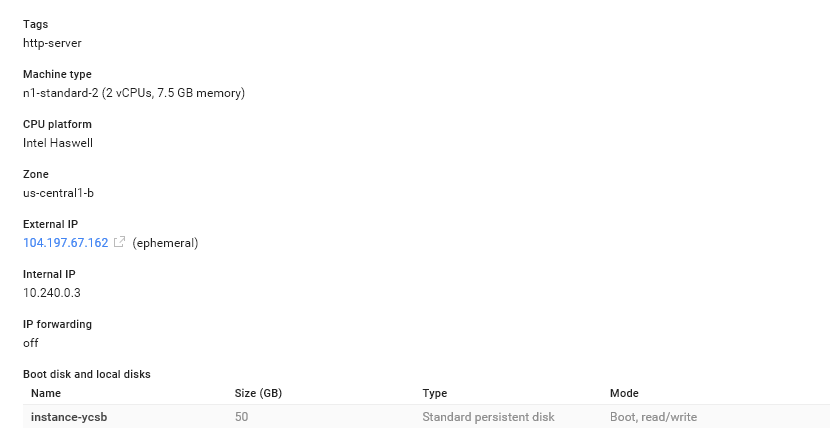
drop table data;  
CREATE TABLE usertable.data ( key blob, column1 text, value blob, PRIMARY KEY (key, column1)) WITH COMPACT STORAGE AND CLUSTERING ORDER BY (column1 ASC);

*//Note: benchmark for 640 concurrent users*

./bin/ycsb load cassandra-10 -P workloads/workload\_test640 -p hosts=10.240.0.4 -threads 640 -p columnfamily=data -s > results\_Cassandra/load\_640

**Load 640 taking long time to run, so we cancel the process.**

**VM Instance specification:**



**Machine type:** n1-standard-2 (2 vCPUs, 7.5 GB memory)

**Boot disk and local disks:** 50GB

**Price: $51.10 per month for 1 node**

**$153.30 per month for 3 nodes**

