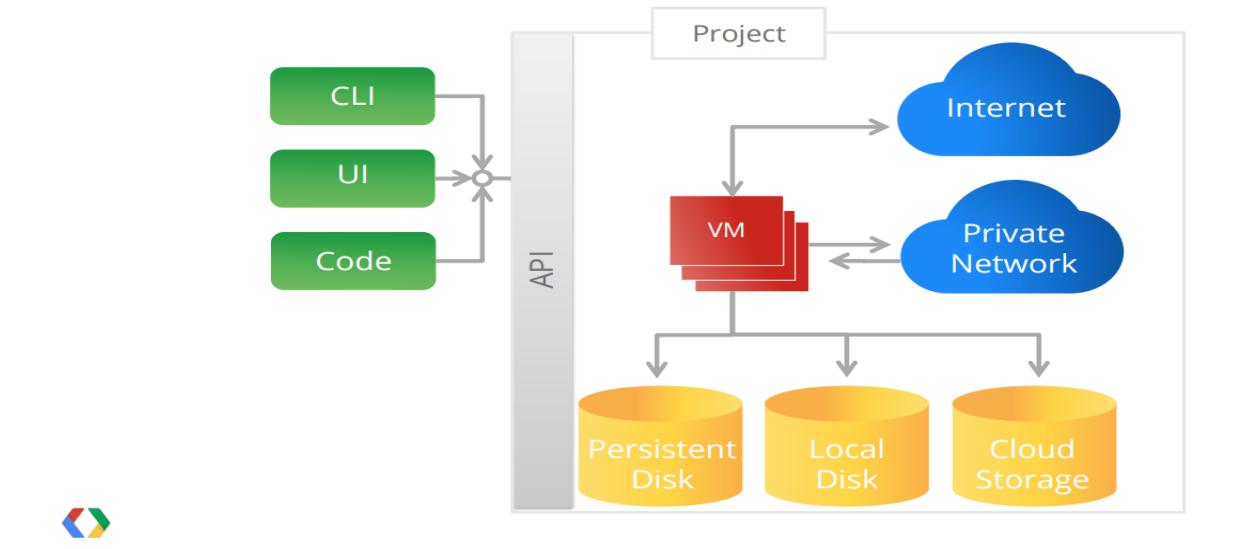


Google's Compute Engine Architecture:

System Components



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.

Google Cloud Platform

Why Google Products Solutions Launcher Pricing Customers Documentation Support Partners TRY IT FREE CONTACT SALES

COMPUTE ENGINE

Scalable, High-Performance Virtual Machines

TRY IT FREE

High-Performance, Scalable VMs

Google Compute Engine delivers virtual machines running in Google's innovative data centers and worldwide fiber network. Compute Engine's tooling and workflow support **enable scaling from single instances to global, load-balanced cloud computing.**

Compute Engine's **VMs boot quickly**, come with persistent disk storage, and deliver consistent performance. Our virtual servers are available in many configurations including predefined sizes or the option to **create Custom Machine Types optimized for your specific needs**. Flexible pricing and automatic sustained use discounts make Compute Engine the leader in price/performance.

Industry Leading Price & Performance

Step 2: Go to google dashboard and select compute Engine

Google Cloud Platform My First Project

Home Dashboard CUSTOMIZE

Dashboard

Project info
My First Project
Project ID: core-hash-161017
#828087224327
Manage project settings

Resources
Compute Engine 8 instances

Trace
No trace data from the past 7 days
Get started with Stackdriver Trace

Getting Started
APIs Enable APIs and get credentials like keys

Compute Engine
CPU (%)
There is no data for this chart
Go to the Compute Engine dashboard

Google Cloud Platform status
Google Cloud Storage incident #16037
Elevated Latency and Error Rates For GCS in Europe
Began at 2017-03-12 (12:08:00)
Google Cloud Dataflow incident #17001
Dataflow Job Log visibility issue in Cloud Console
Began at 2017-03-09 (11:58:00)
All times are US/Pacific
Data provided by status.cloud.google.com
Go to Cloud status dashboard

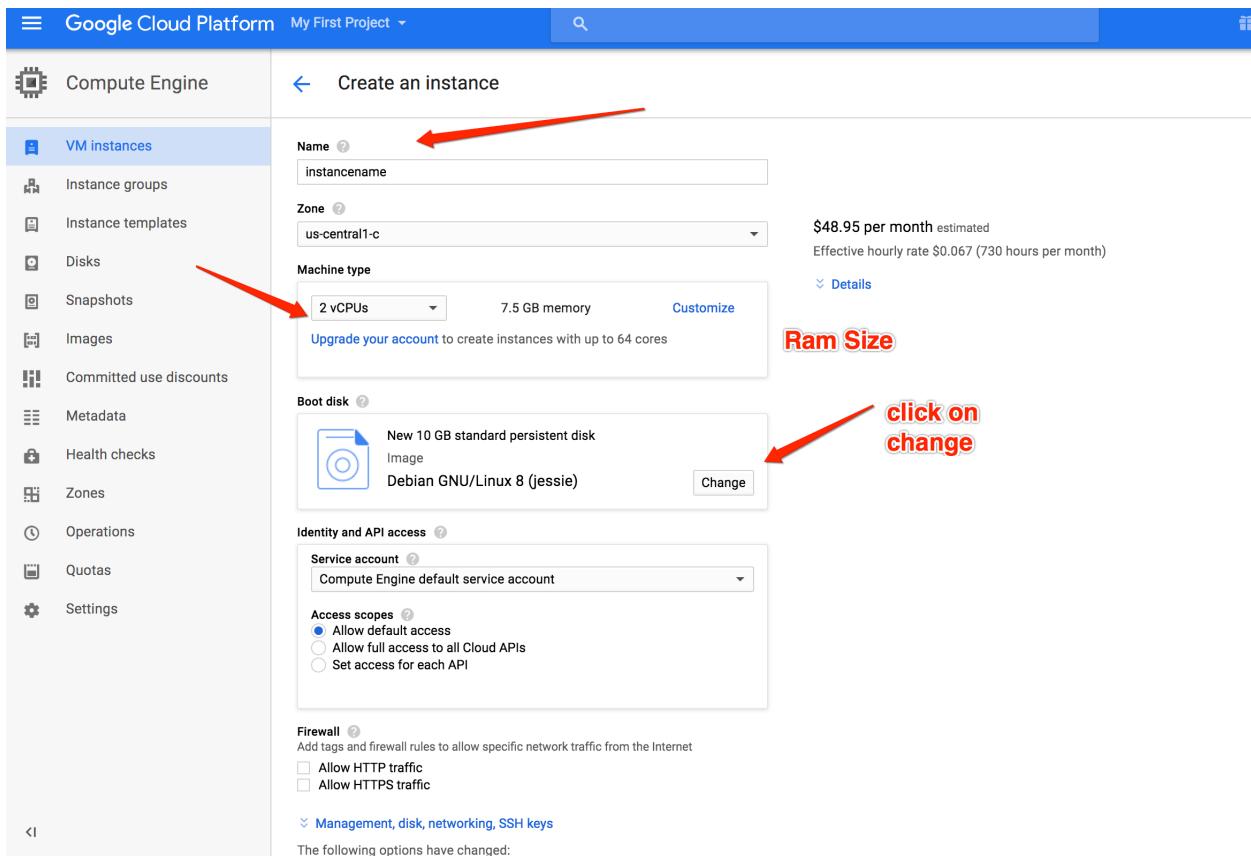
Billing
\$0.00 Approximate charges so far this month
View detailed charges

Error Reporting

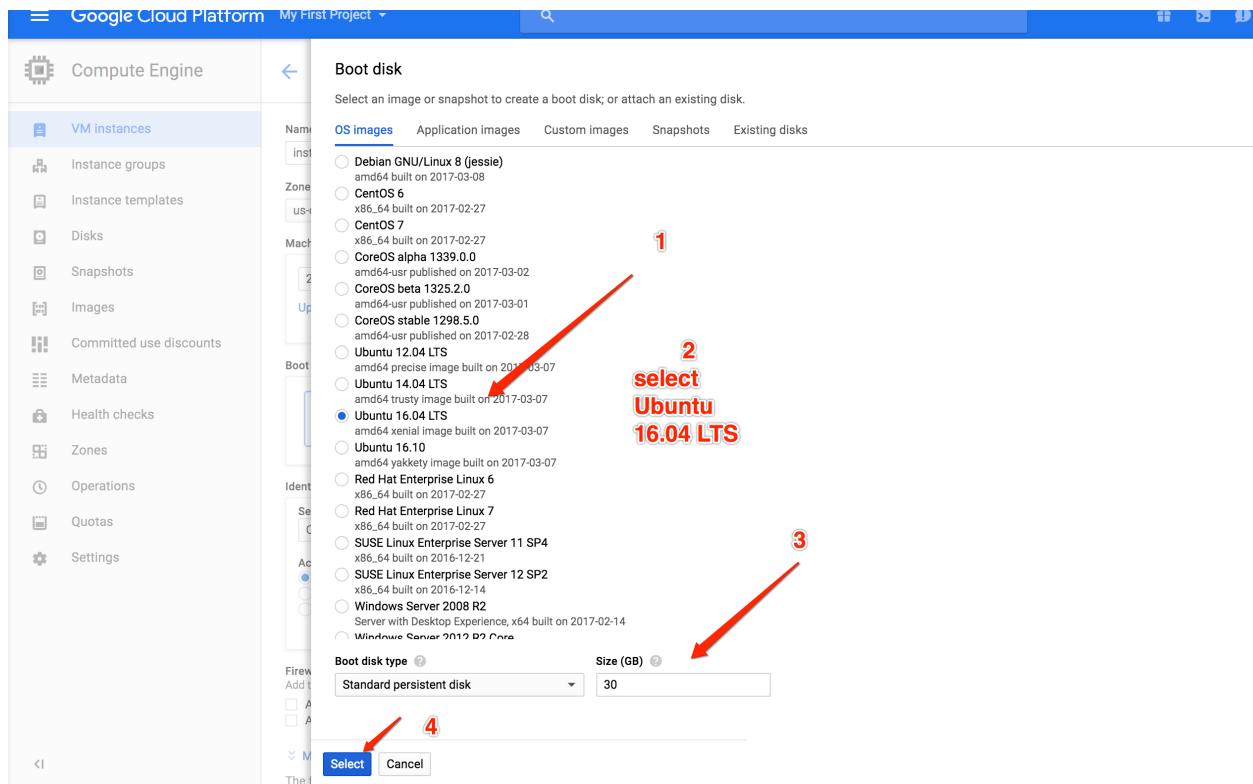
Step 3: Create Instances:

The screenshot shows the Google Cloud Platform interface for Compute Engine VM instances. The left sidebar has 'Compute Engine' selected, and 'VM Instances' is the active sub-item. The main content area displays a brief introduction to Compute Engine and three action buttons: 'Create' (highlighted with a red circle), 'Import', and 'Take the quickstart'.

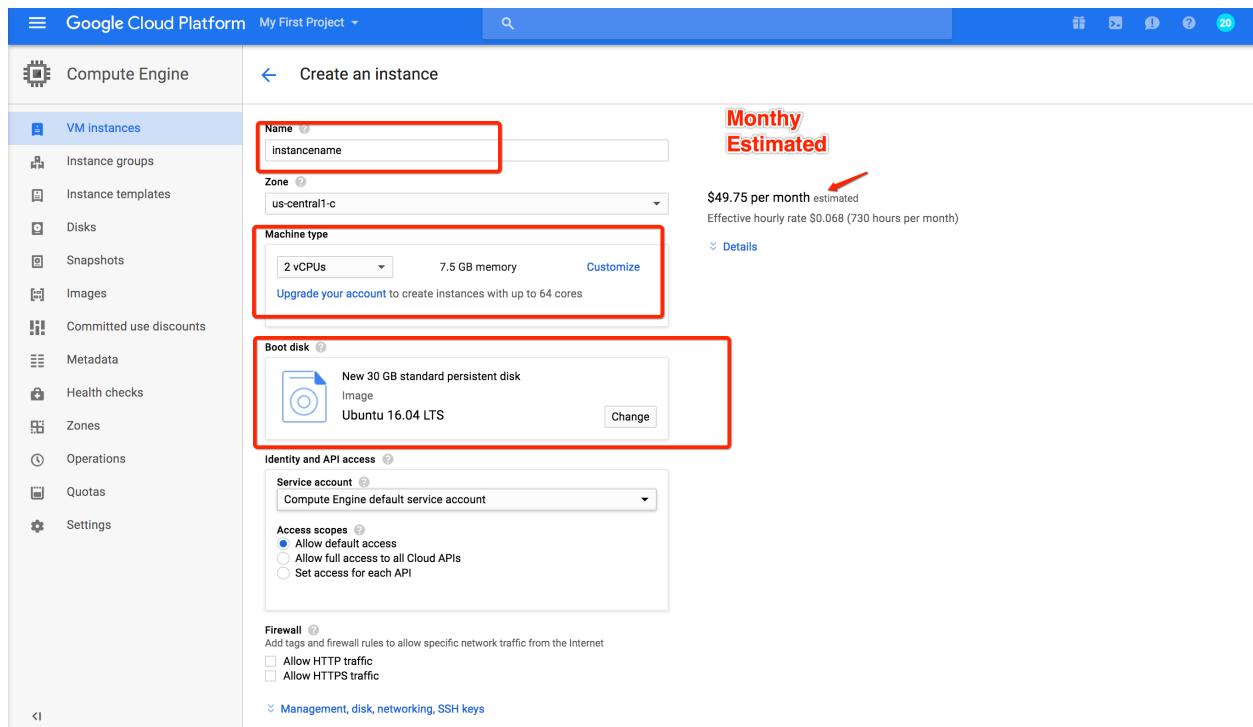
Step 4: Compute Engine select Below Configuration,



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.

The screenshot shows the Google Cloud Platform Network details page for a project named "My First Project". The left sidebar is titled "Networking" and includes options for External IP addresses, Firewall rules, Routes, Load balancing, Cloud DNS, VPN, Cloud Routers, Cloud CDN, and Cross-project networking. The main content area is titled "Network details" and shows two sections: "Firewall rules" and "Routes".

Firewall rules:

Name	Source tag / IP range / Subnetworks	Allowed protocols / ports	Target tags
default-allow-icmp	0.0.0.0/0	icmp	Apply to all targets
default-allow-internal	10.128.0.0/9	tcp:0-65535, udp:0-65535, 1 more	Apply to all targets
default-allow-rdp	0.0.0.0/0	tcp:3389	Apply to all targets
default-allow-ssh	0.0.0.0/0	tcp:22	Apply to all targets

Routes:

Name	Destination IP ranges	Priority	Instance tags	Next hop
default-route-099539fbbc07634a	10.146.0.0/20	1000	None	Virtual network
default-route-0b198dfbf1c8d7b4	10.142.0.0/20	1000	None	Virtual network
default-route-168cbd841ba7664f	10.132.0.0/20	1000	None	Virtual network
default-route-201af27766b0aed4	10.140.0.0/20	1000	None	Virtual network
default-route-20227085b3170691	0.0.0.0/0	1000	None	Default internet gateway
default-route-3fa0e9f5fb42af92	10.128.0.0/20	1000	None	Virtual network
default-route-db71c44461d44a41	10.138.0.0/20	1000	None	Virtual network

Two red arrows point to specific rows in both tables: one to the "default-allow-internal" rule in the Firewall rules section and another to the "default-route-20227085b3170691" route in the Routes section.

Google Cloud Platform My First Project

Networking

Network details

default

Mode: Auto subnetworks

Description: Default network for the project

Subnetworks

Name	Region	IP address ranges	Gateway	Private Google access
default	us-central1	10.128.0.0/20	10.128.0.1	Disabled
default	europe-west1	10.132.0.0/20	10.132.0.1	Disabled
default	us-west1	10.138.0.0/20	10.138.0.1	Disabled
default	asia-east1	10.140.0.0/20	10.140.0.1	Disabled
default	us-east1	10.142.0.0/20	10.142.0.1	Disabled
default	asia-northeast1	10.146.0.0/20	10.146.0.1	Disabled

Firewall rules

Name	Source tag / IP range / Subnetworks	Allowed protocols / ports	Target tags
default-allow-icmp	0.0.0.0/0	icmp	Apply to all targets
default-allow-internal	10.128.0.0/9	tcp:0-65535, udp:0-65535, 1 more	Apply to all targets
default-allow-rdp	0.0.0.0/0	tcp:3389	Apply to all targets

step 6: After verification create instance

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, disk, networking, SSH keys

The following options have changed:

Subnetwork

Your Free Trial credits, if available, will be used for this instance.

Create **Cancel**

Equivalent REST or command line

Final Step

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

Google Cloud Platform My First Project

Compute Engine

VM instances

CREATE INSTANCE IMPORT VM REFRESH START STOP RESET DELETE

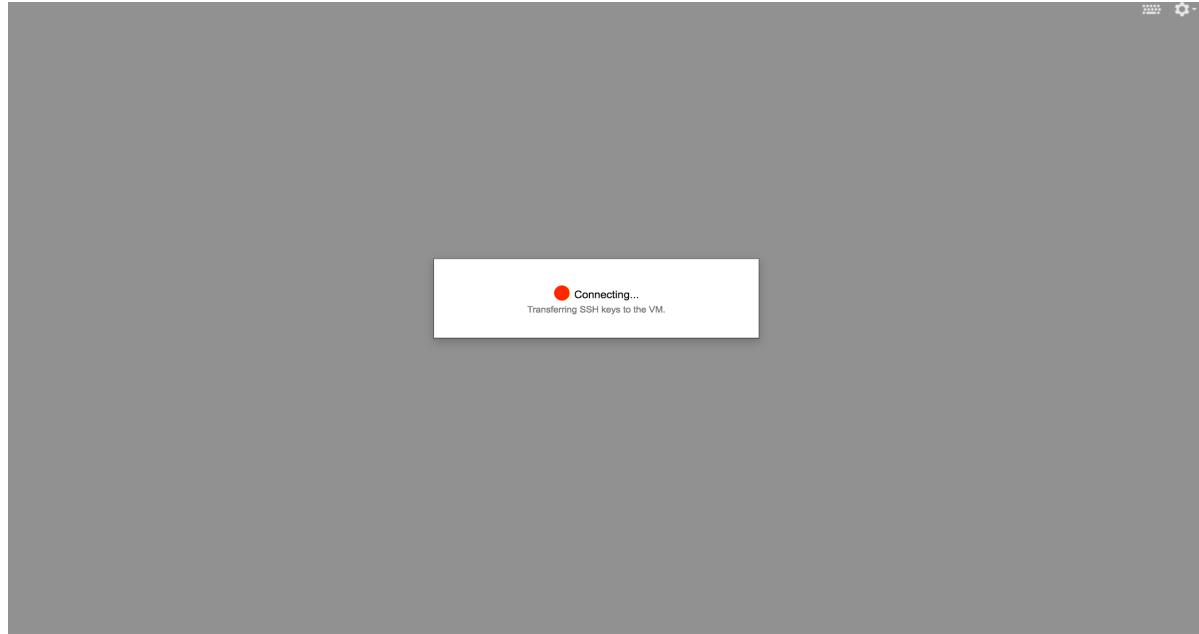
VM instances

Filter by label or name

Columns Labels

Name	Zone	Recommendation	Internal IP	External IP	Connect
Instance-1	us-central1-c		10.128.0.2	None	SSH
Instance-4	us-central1-c		10.128.0.5	104.154.128.193	SSH
Instance-5	us-central1-c		10.128.0.6	108.59.80.177	SSH
Instance-6	us-central1-c		10.128.0.7	104.154.159.86	SSH
secondinstance	us-central1-c		10.128.0.3	None	SSH
testycb1	us-central1-c		10.128.0.9	104.198.244.62	SSH
testycsb	us-central1-c		10.128.0.8	None	SSH
thirdinstance	us-central1-c		10.128.0.4	None	SSH

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



```
Connected, host fingerprint: ssh-rsa 2048 10:01:F7:40:A8:2F:B9:3D:42:C7:73:48:D4:D6:A8:05:14:F3:A1:8B
Welcome to Ubuntu 14.04.5 LTS (GNU/Linux 4.4.0-66-generic x86_64)

 * Documentation: https://help.ubuntu.com/
 
 System information as of Thu Mar  9 18:07:49 UTC 2017

 System load: 0.08           Memory usage: 1%   Processes:      77
 Usage of /: 10.5% of 9.81GB Swap usage: 0%   Users logged in: 0

 Graph this data and manage this system at:
 https://landscape.canonical.com/
 
 Get cloud support with Ubuntu Advantage Cloud Guest:
 http://www.ubuntu.com/business/services/cloud

0 packages can be updated.
0 updates are security updates.

Your Hardware Enablement Stack (HWE) is supported until April 2019.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

kunal12899@instance-1:~$
```



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

a) Change root password

```
sudo passwd root
<enter password>
su
<password>
```

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
```

c) 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:

```

More info (and Ubuntu installation instructions):
- for Oracle Java 7: http://www.webupd8.org/2012/01/install-oracle-java-jdk-7-in-ubuntu-via.html
- for Oracle Java 8: http://www.webupd8.org/2012/09/install-oracle-java-8-in-ubuntu-via-ppa.html

Debian installation instructions:
- Oracle Java 7: http://www.webupd8.org/2012/06/how-to-install-oracle-java-7-in-debian.html
- Oracle Java 8: http://www.webupd8.org/2014/03/how-to-install-oracle-java-8-in-debian.html

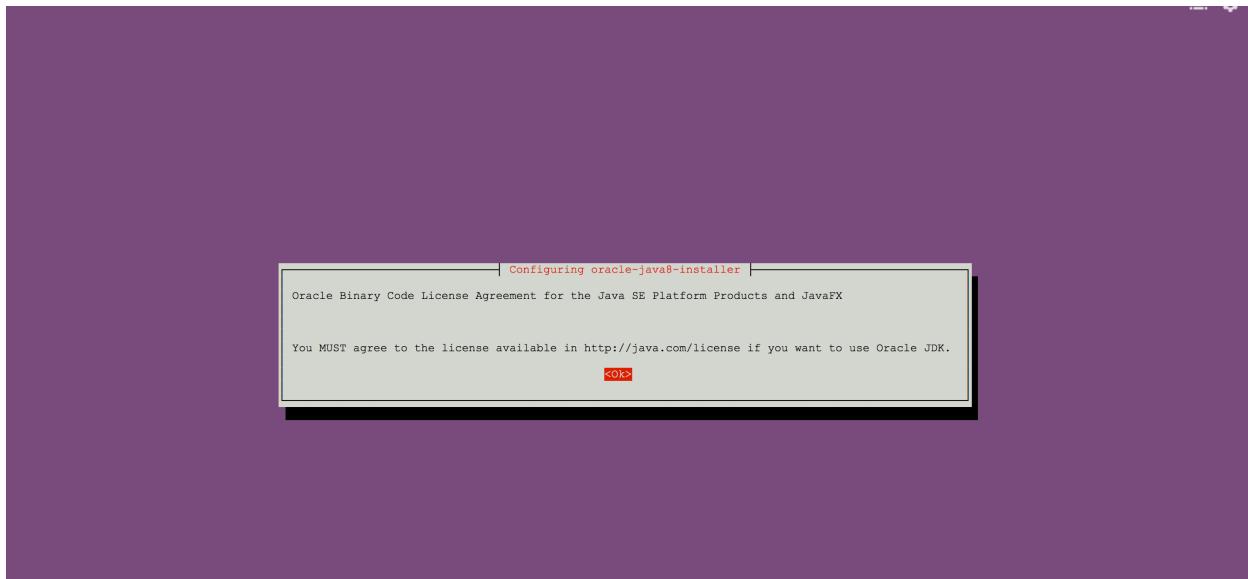
Oracle Java 9 (for both Ubuntu and Debian): http://www.webupd8.org/2015/02/install-oracle-java-9-in-ubuntu

For JDK9, the PPA uses standard builds from: https://jdk9.java.net/download/ (and not the Jigsaw builds!)

Important!!! For now, you should continue to use Java 8 because Oracle Java 9 is available as an early ad
use it may contain bugs and it might not include the latest security patches! Also, some Java options were
information and installation instructions (Ubuntu / Linux Mint / Debian): http://www.webupd8.org/2015/02/ins
More info: https://launchpad.net/~webupd8team/+archive/ubuntu/java
Press [ENTER] to continue or ctrl-c to cancel adding it

gpg: keyring `/tmp/tmp6rbho_u/_secring.gpg' created
gpg: keyring `/tmp/tmp6rbho_u/_pubring.gpg' created
gpg: requesting key EEA14886 from hkp server keyserver.ubuntu.com
gpg: /tmp/tmp6rbho_u/_trustdb.gpg: trustdb created
gpg: key EEA14886: public key "Launchpad VLC" imported
gpg: no ultimately trusted keys found
gpg: Total number processed: 1
gpg:                         imported: 1  (RSA: 1)
OK
kunal12899@instance-1:~$ sudo apt-get update
Ign http://us-central1.gce.archive.ubuntu.com trusty InRelease
Get:1 http://us-central1.gce.archive.ubuntu.com trusty-updates InRelease [65.9 kB]
Hit http://us-central1.gce.archive.ubuntu.com trusty-backports InRelease
Hit http://us-central1.gce.archive.ubuntu.com trusty Release.gpg
Hit http://us-central1.gce.archive.ubuntu.com trusty Release
Get:2 http://us-central1.gce.archive.ubuntu.com trusty-updates/main Sources [393 kB]
Get:3 http://us-central1.gce.archive.ubuntu.com trusty-updates/restricted Sources [5,911 B]
Get:4 http://us-central1.gce.archive.ubuntu.com trusty-updates/universe Sources [175 kB]
Get:5 http://us-central1.gce.archive.ubuntu.com trusty-updates/multiverse Sources [7,510 B]
Get:6 http://us-central1.gce.archive.ubuntu.com trusty-updates/main amd64 Packages [958 kB]
Get:7 http://us-central1.gce.archive.ubuntu.com trusty-updates/restricted amd64 Packages [16.4 kB]
Get:8 http://us-central1.gce.archive.ubuntu.com trusty-updates/universe amd64 Packages [401 kB]
Get:9 http://us-central1.gce.archive.ubuntu.com trusty-updates/multiverse amd64 Packages [14.0 kB]
Get:10 http://us-central1.gce.archive.ubuntu.com trusty-updates/main Translation-en [472 kB]
Get:11 http://us-central1.gce.archive.ubuntu.com trusty-updates/multiverse Translation-en [7,340 B]
Get:12 http://us-central1.gce.archive.ubuntu.com trusty-updates/restricted Translation-en [3,847 B]

```



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
```

```
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

```

apache.cassandra.db.marshal.UTF8Type, kind=CLUSTERING, position=0), ColumnDefinition(name=role, type=org.apache.cassandra.db.marshal.UTF8Type, kind=PARTITION_KEY, position=0), ColumnDefinition(name=permissions, type=org.apache.cassandra.db.marshal.SetType(org.apache.cassandra.db.marshal.UTF8Type), kind=REGULAR, position=1), droppedColumns=[], triggers=[], indexes=[]),
org.apache.cassandra.config.CFMetadata@ed2452{cfid=5ffbbad-91f1-3d6-pd2-d0da3a5dec, kName=system_auth, cName=resource_role_permissions_index, flags=[COMPOUND], params=tableParams,
comment='Index of db system_auth permissions grants', read_repair_chance=0.0, gc_grace=604800000, min_index=0, max_index=28, max_index_in_partition=28, max_threshold=2048, cumulative_threshold=10240, strategy=PERCFILL, strategy_options={key='ALL', 'L', 'O', 'P', 'R', 'S', 'T', 'W', 'X'}, per_partition='NONE', compaction=CompactionParams(class=org.apache.cassandra.db.compaction.CompactionParams@8396c37d4, extensions={}), comparator=comparator(org.apache.cassandra.db.marshal.UTF8Type), kind=PARTITION_KEY, position=0)}, clusteringColumns=[ColumnDefinition(name=role, type=org.apache.cassandra.db.marshal.UTF8Type, kind=CLUSTERING, position=0)], keyValidator=org.apache.cassandra.db.marshal.UTF8Type, columnMetadata=[ColumnDefinition(name=resource, type=org.apache.cassandra.db.marshal.UTF8Type, kind=CLUSTERING, position=0)], droppedColumns=[], triggers=[], indexes=[], views=[], functions=[], types=[]}
INFO 01:39:40 Initializing system.auth.resource_role_permissions_index
INFO 01:39:40 Initializing system.auth.role_members
INFO 01:39:40 Initializing system.auth.role_permissions
INFO 01:39:40 Initializing system.auth.roles
INFO 01:39:40 Waiting for gossip to settle before accepting client requests...
INFO 01:39:40 No gossip backlog; proceeding

[root@instance-6:~/apache-cassandra-3.0.11/bin$ sh nodetool status
Datacenter: dc1
-----
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
-- Address Load Tokens Owns (effective) Host ID Rack
UN 10.128.0.5 201.89 KB 256 70.7% b61ff78a-b8ad-41af-8913-2674130e77a9 rack1
UN 10.128.0.6 229.29 KB 256 64.6% d7e2a919-9ffc-4917-b6dc-cbae8ec381b5 rack1
UN 10.128.0.7 190.09 KB 256 64.7% 6aa0dd4f-594c-4080-ab0f-cbd70dbbb834 rack1
[root@instance-6:~/apache-cassandra-3.0.11/bin$ ]

```

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 userstable WITH REPLICATION = { 'class' : 'SimpleStrategy',
'replication_factor' : 3};
```

use userstable;

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

describe table data;

```

root@instance-1:/usr/local/cassandra/apache-cassandra-2.2.8/bin# sudo sh cqlsh
Connected to Test Cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 2.2.8 | CQL spec 3.3.1 | Native protocol v4]
Use HELP for help.
cqlsh> 
```

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

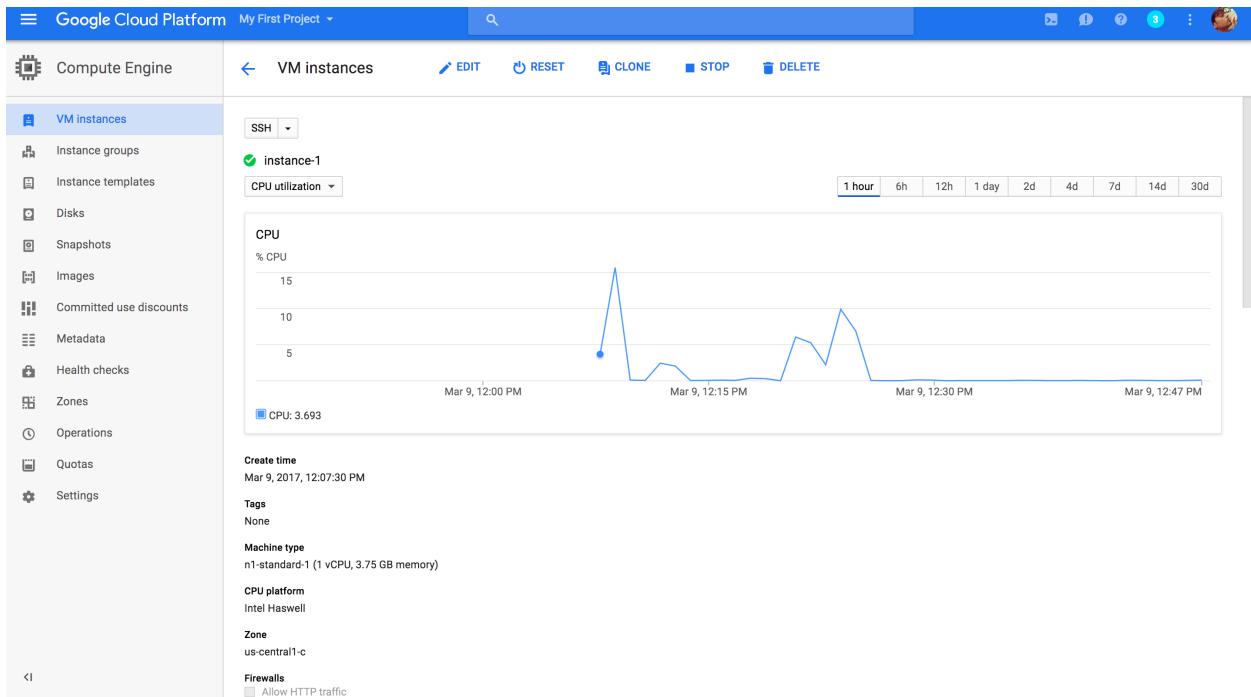
output file name:

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

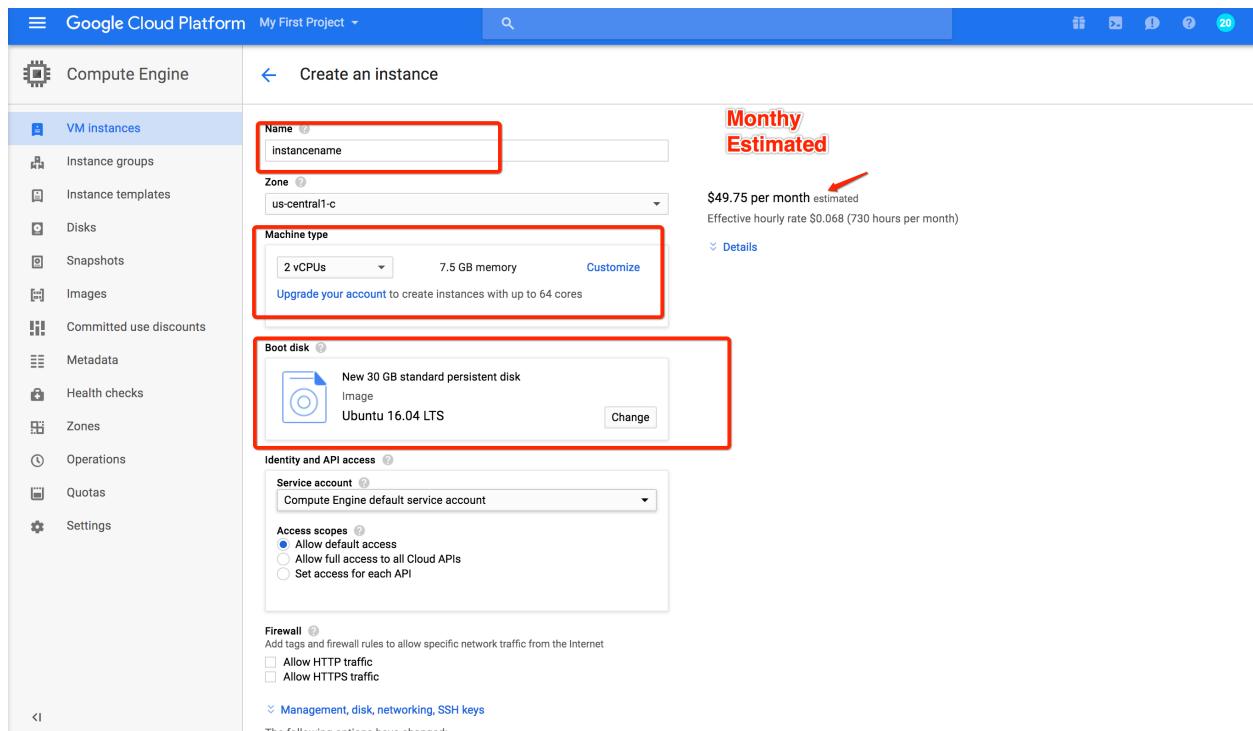
ERROR 22:12:16 Exiting due to error while processing commit log during initialization.

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

```
Apache-cassandra-3.0.11/data/
```

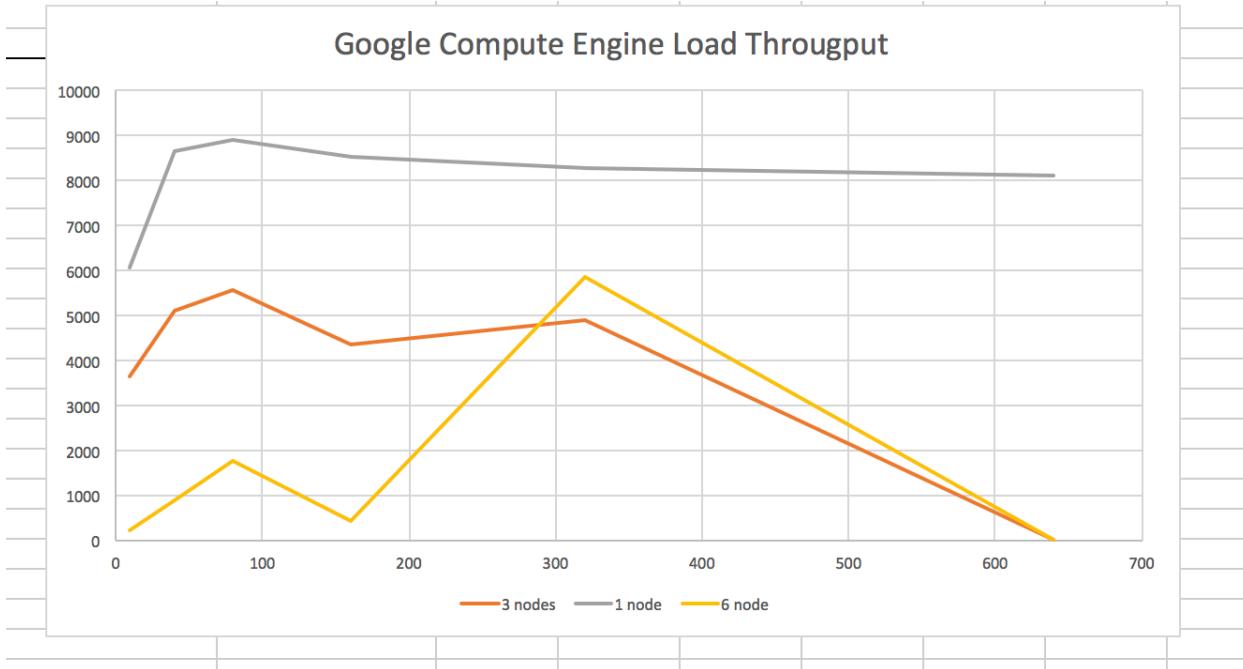
```
rm -rf commitlog/*
```

Cost and Pricing Estimate

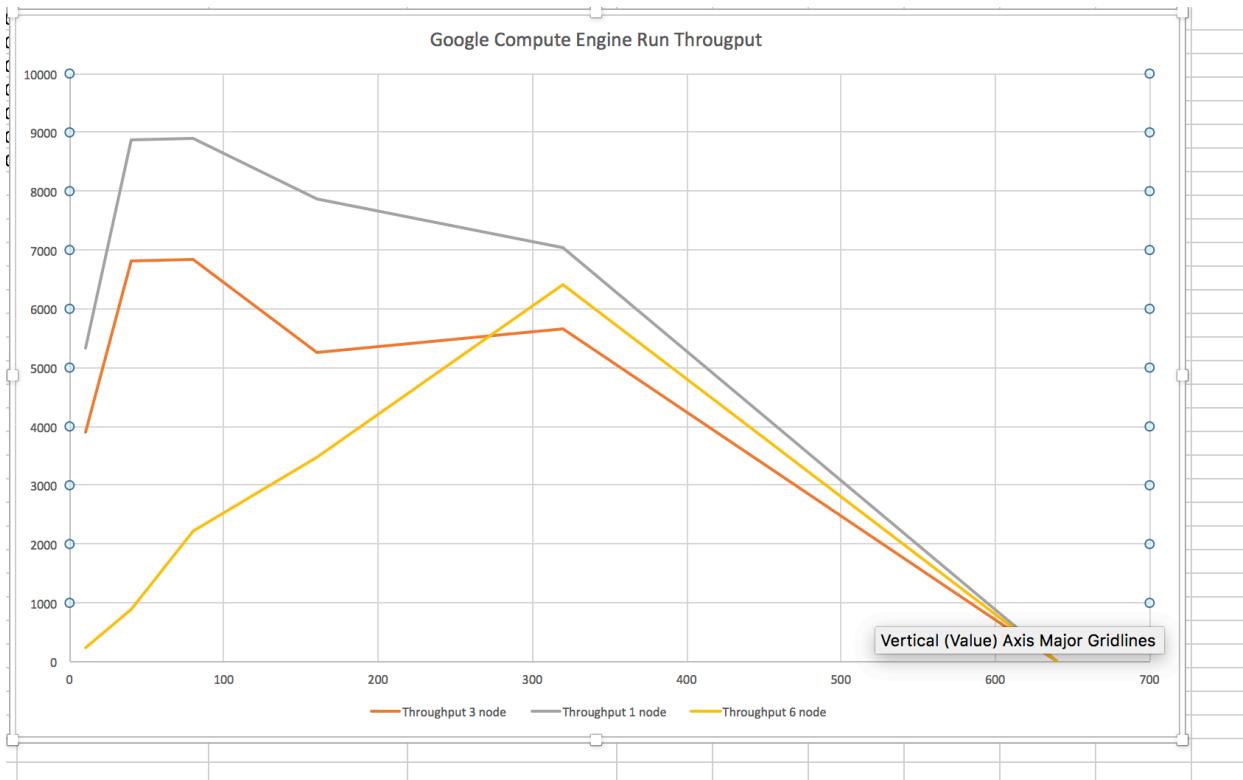


Output:

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