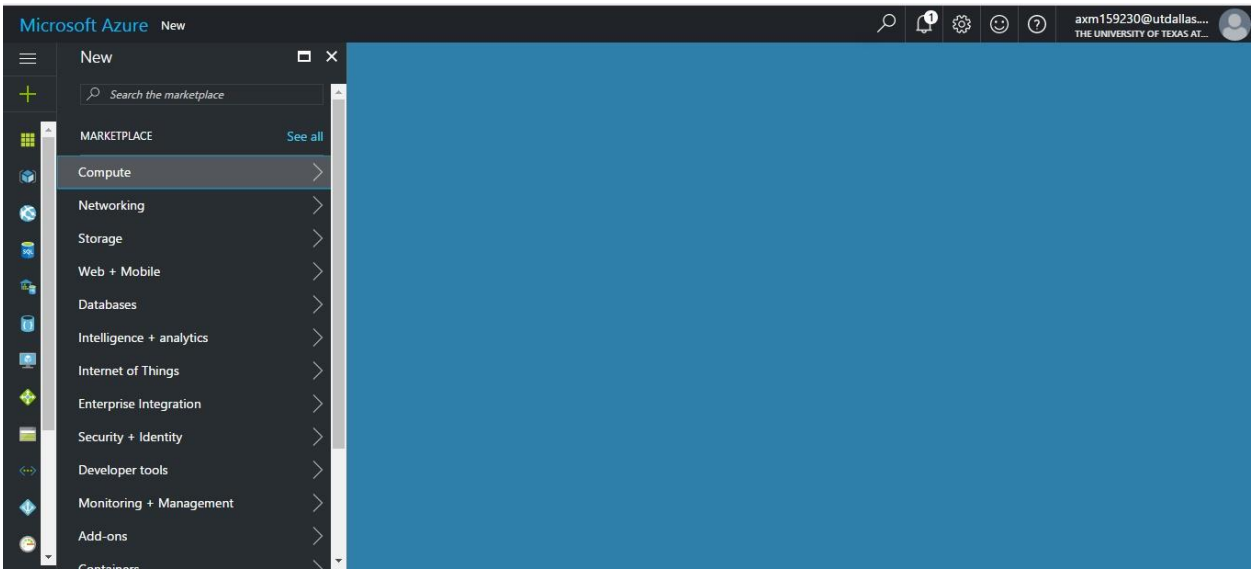
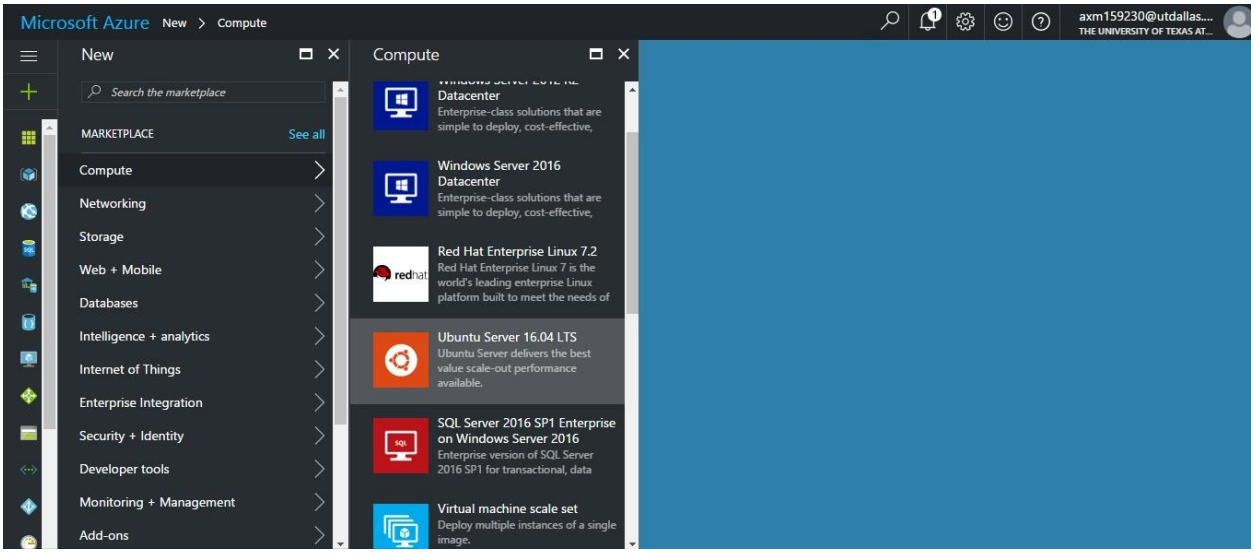
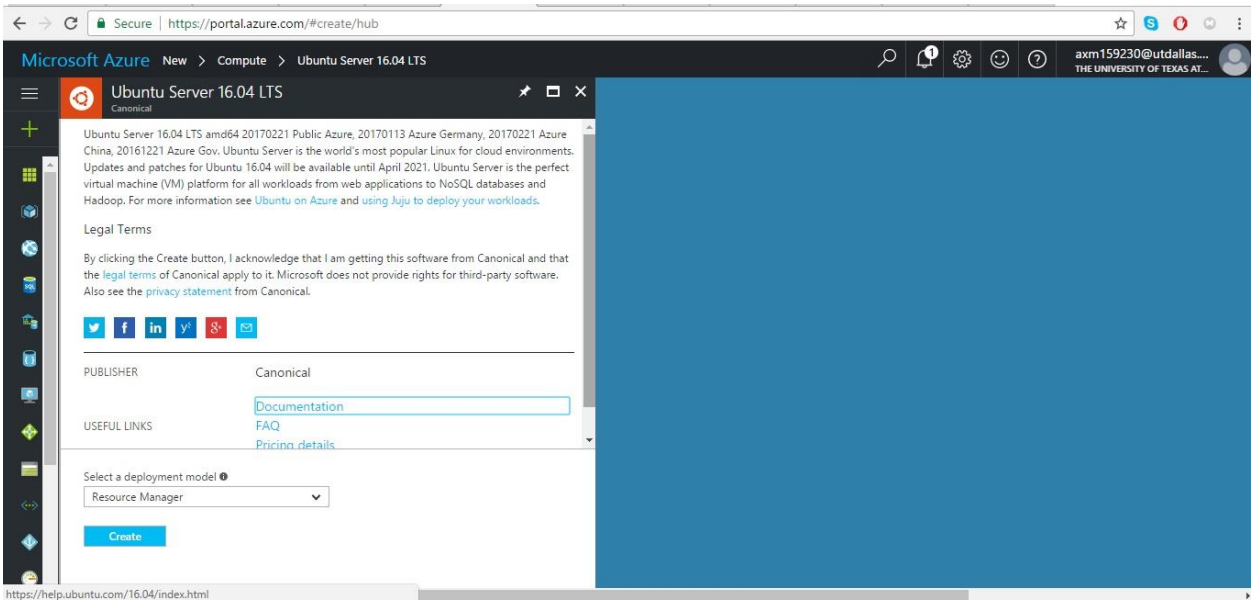
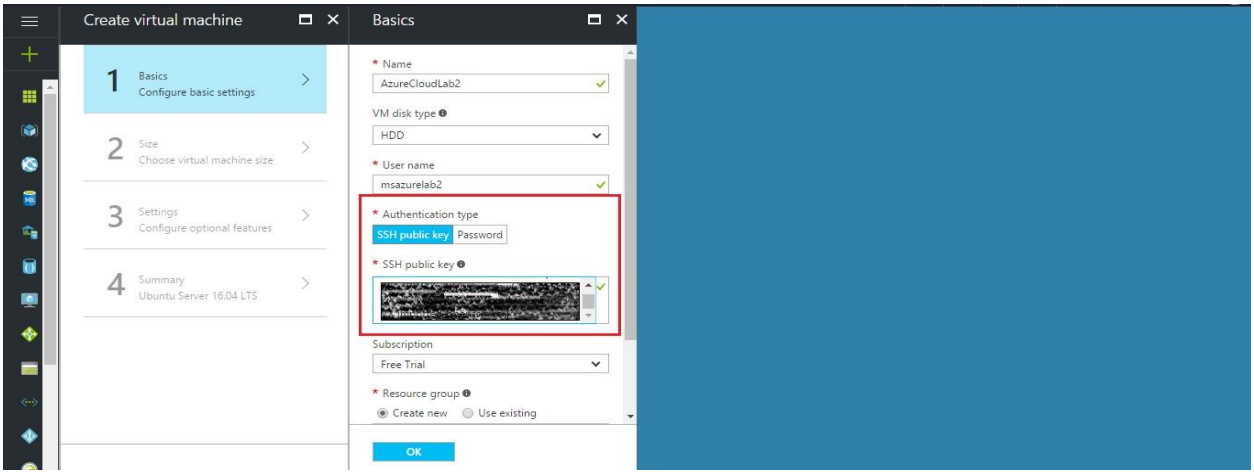
MS Azure

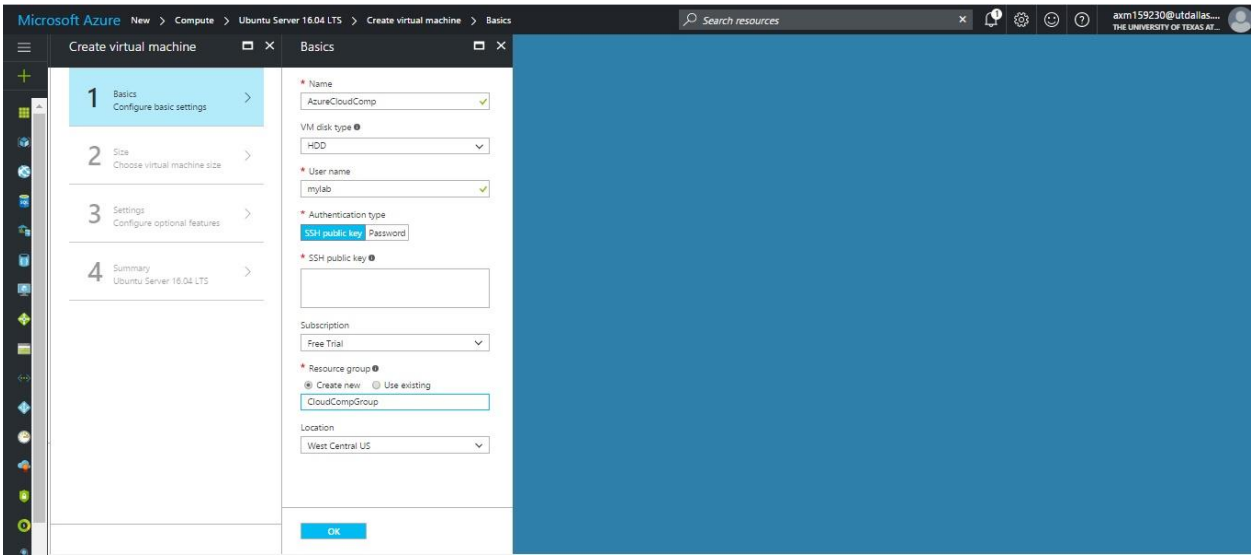
Creating MS Azure instances






Secure | https://portal.azure.com/#create/hub
Microsoft Azure New > Compute > Ubuntu Server 16.04 LTS
axm159230@utdallas....
THE UNIVERSITY OF TEXAS AT...
Ubuntu Server 16.04 LTS
Canonical
Ubuntu Server 16.04 LTS amd64 20170221 Public Azure, 20170113 Azure Germany, 20170221 Azure China, 20161221 Azure Gov. Ubuntu Server is the world's most popular Linux for cloud environments. Updates and patches for Ubuntu 16.04 will be available until April 2021. Ubuntu Server is the perfect virtual machine (VM) platform for all workloads from web applications to NoSQL databases and Hadoop. For more information see Ubuntu on Azure and using Juju to deploy your workloads.
Legal Terms
By clicking the Create button, I acknowledge that I am getting this software from Canonical and that the legal terms of Canonical apply to it. Microsoft does not provide rights for third-party software. Also see the privacy statement from Canonical.
PUBLISHER
Canonical
USEFUL LINKS
Documentation
FAQ
Pricing details
Select a deployment model
Resource Manager
Create
https://help.ubuntu.com/16.04/index.html

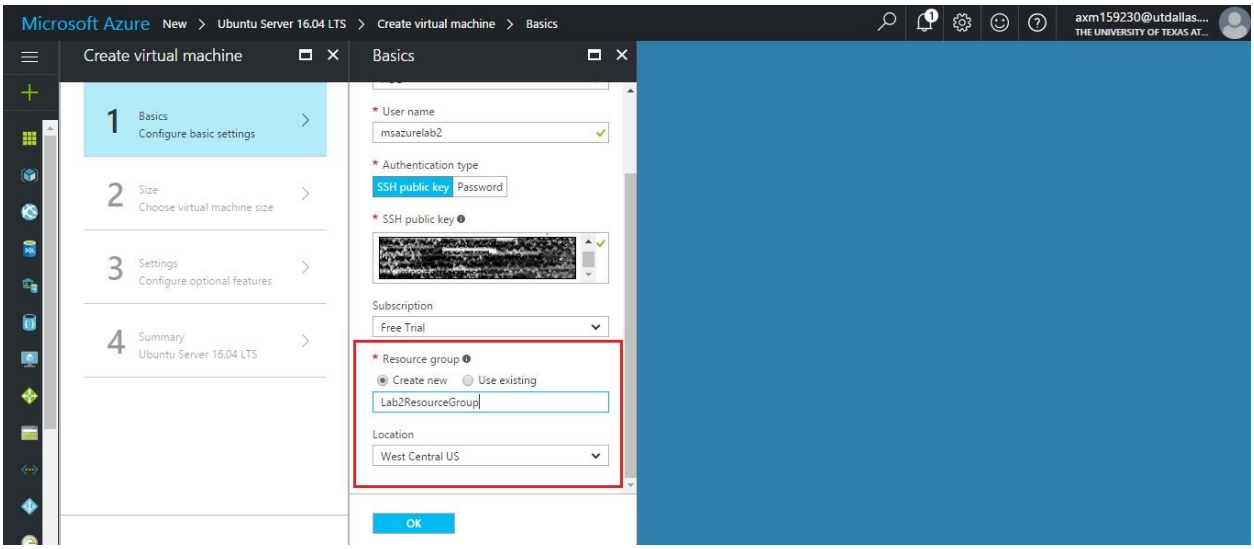

Create virtual machine
1 Basics
Configure basic settings
2 Size
Choose virtual machine size
3 Settings
Configure optional features
4 Summary
Ubuntu Server 16.04 LTS
Basics
* Name
AzureCloudLab2
VM disk type
HDD
* User name
msazurelab2
* Authentication type
SSH public key
Password
* SSH public key
Subscription
Free Trial
* Resource group
Create new
Use existing
OK

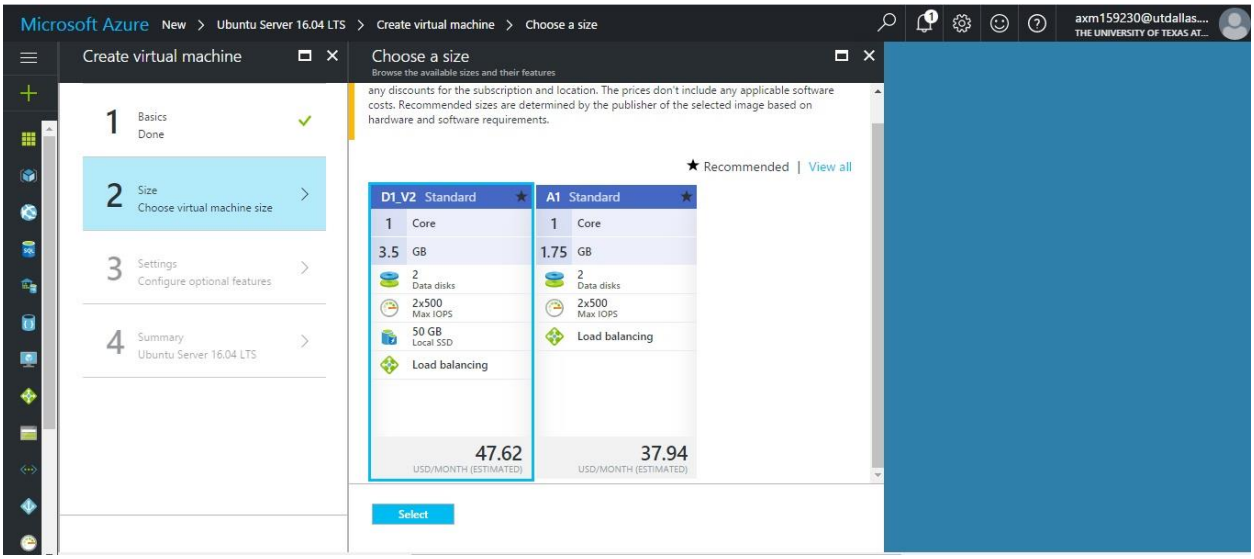

Microsoft Azure New > Compute > Ubuntu Server 16.04 LTS > Create virtual machine > Basics
Search resources
axm159230@utdallas....
THE UNIVERSITY OF TEXAS AT...
Create virtual machine
1 Basics
Configure basic settings
2 Size
Choose virtual machine size
3 Settings
Configure optional features
4 Summary
Ubuntu Server 16.04 LTS
Basics
* Name
AzureCloudComp
VM disk type
HDD
* User name
mylab
* Authentication type
SSH public key
Password
* SSH public key
Subscription
Free Trial
* Resource group
Create new
Use existing
CloudCompGroup
Location
West Central US
OK

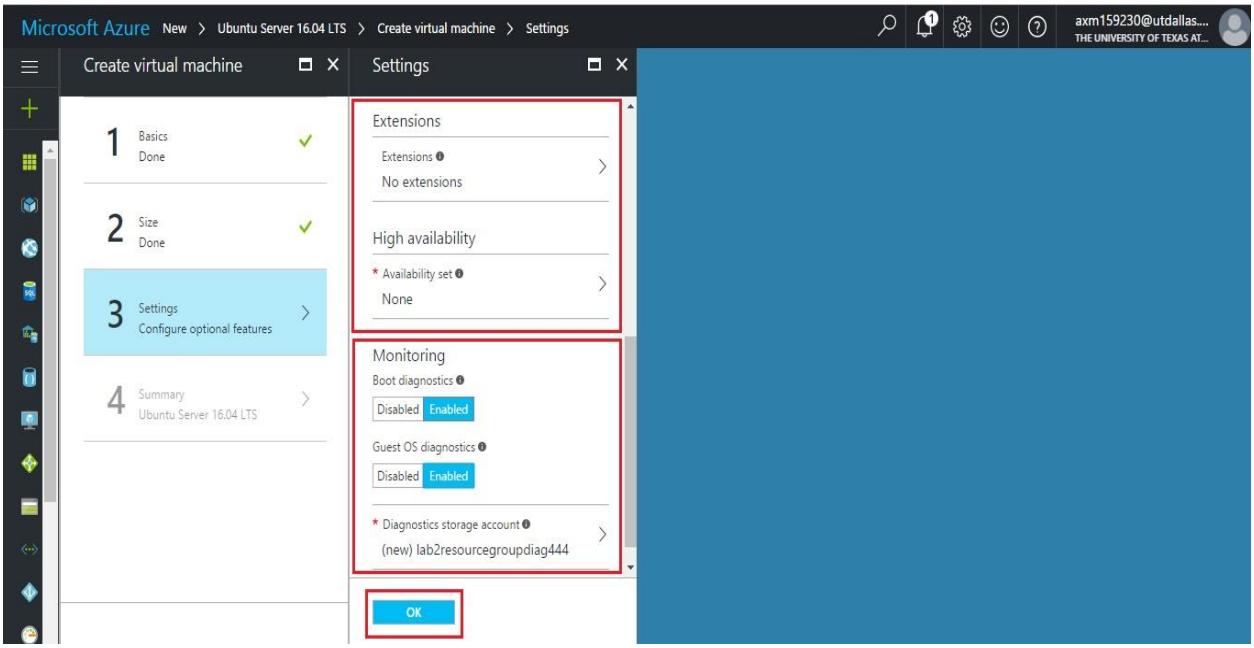
```
Abhiyush@DESKTOP-ODJSA5O MINGW32 ~
$ openssl rsa -in ./[redacted]PrivateKey.key -out [redacted]PrivateKey_rsa
writing RSA key

Abhiyush@DESKTOP-ODJSA5O MINGW32 ~
$ chmod 0600 [redacted]PrivateKey_rsa

Abhiyush@DESKTOP-ODJSA5O MINGW32 ~
$
```

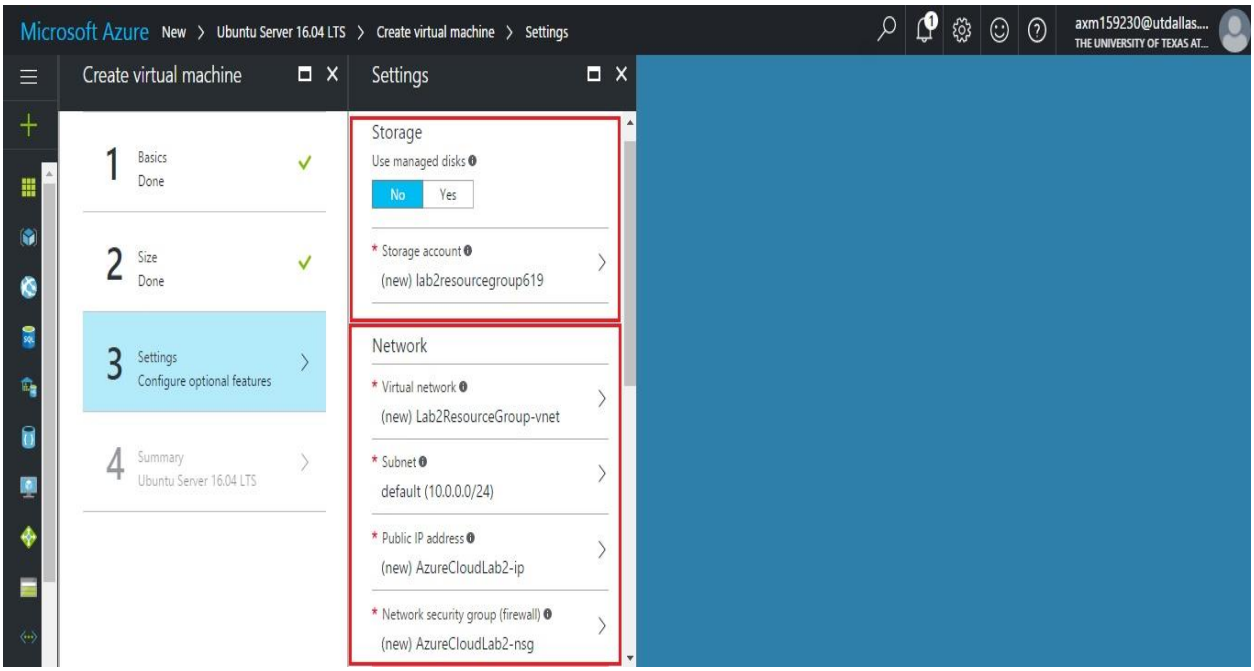
```
Press [ENTER] to continue or ctrl-c to cancel adding it

gpg: keyring `/tmp/tmpjwcfbnso/secring.gpg' created
gpg: keyring `/tmp/tmpjwcfbnso/pubring.gpg' created
gpg: requesting key EEA14886 from hkp server keyserver.ubuntu.com
gpg: /tmp/tmpjwcfbnso/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
```

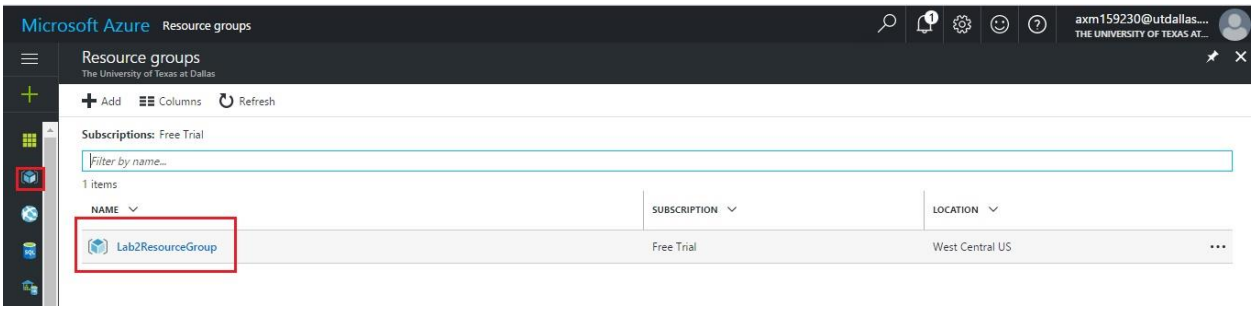
```
Abhiyush@DESKTOP-ODJSA5O MINGW32 ~
$ openssl.exe req -x509 -nodes -days 365 -newkey rsa:2048 -keyout AzurePrivateKey.key -out AzureCert.pem
Generating a 2048 bit RSA private key
.....................................................+++
..+++
writing new private key to 'AzurePrivateKey.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:Texas
Locality Name (eg, city) []:Dallas
Organization Name (eg, company) [Internet Widgits Pty Ltd]:University of Texas at Dallas
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:Abhi
Email Address []:abhiyush.mittal@utdallas.edu
```

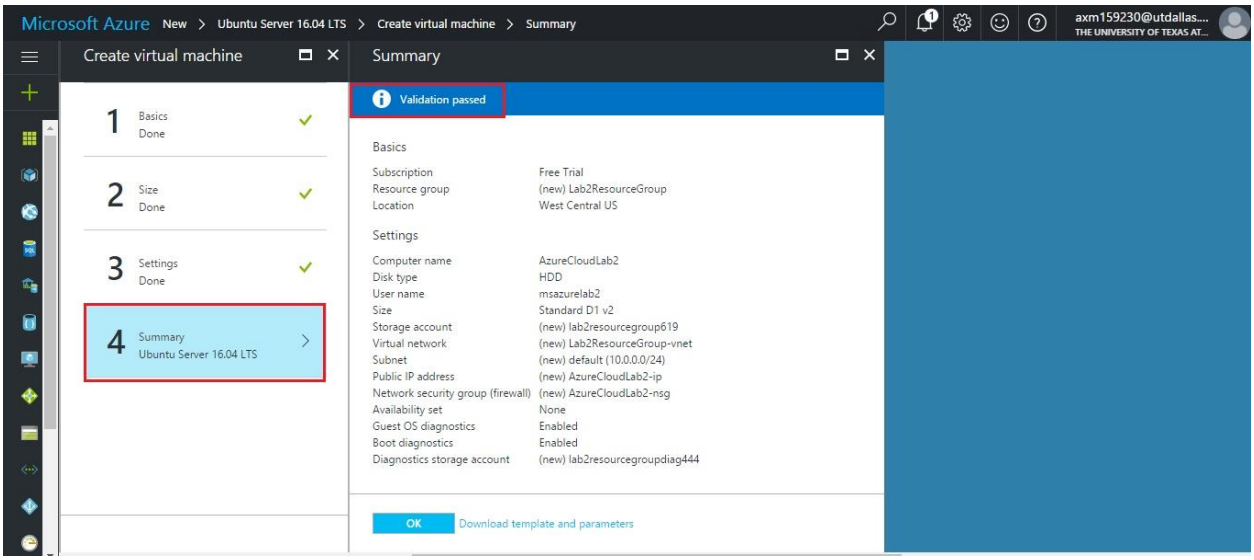

Microsoft Azure
New > Ubuntu Server 16.04 LTS > Create virtual machine > Basics
axm159230@utdallas....
THE UNIVERSITY OF TEXAS AT...
Create virtual machine
1 Basics
Configure basic settings
2 Size
Choose virtual machine size
3 Settings
Configure optional features
4 Summary
Ubuntu Server 16.04 LTS
Basics
* User name
msazurelab2
* Authentication type
SSH public key
Password
* SSH public key
Subscription
Free Trial
* Resource group
Create new
Use existing
Lab2ResourceGroup
Location
West Central US
OK


Microsoft Azure
New > Ubuntu Server 16.04 LTS > Create virtual machine > Choose a size
axm159230@utdallas....
THE UNIVERSITY OF TEXAS AT...
Create virtual machine
1 Basics
Done
2 Size
Choose virtual machine size
3 Settings
Configure optional features
4 Summary
Ubuntu Server 16.04 LTS
Choose a size
Browse the available sizes and their features
any discounts for the subscription and location. The prices don't include any applicable software costs. Recommended sizes are determined by the publisher of the selected image based on hardware and software requirements.
★ Recommended | View all
D1_V2 Standard
1 Core
3.5 GB
2 Data disks
2x500 Max IOPS
50 GB Local SSD
Load balancing
47.62
USD/MONTH (ESTIMATED)
A1 Standard
1 Core
1.75 GB
2 Data disks
2x500 Max IOPS
Load balancing
37.94
USD/MONTH (ESTIMATED)
Select

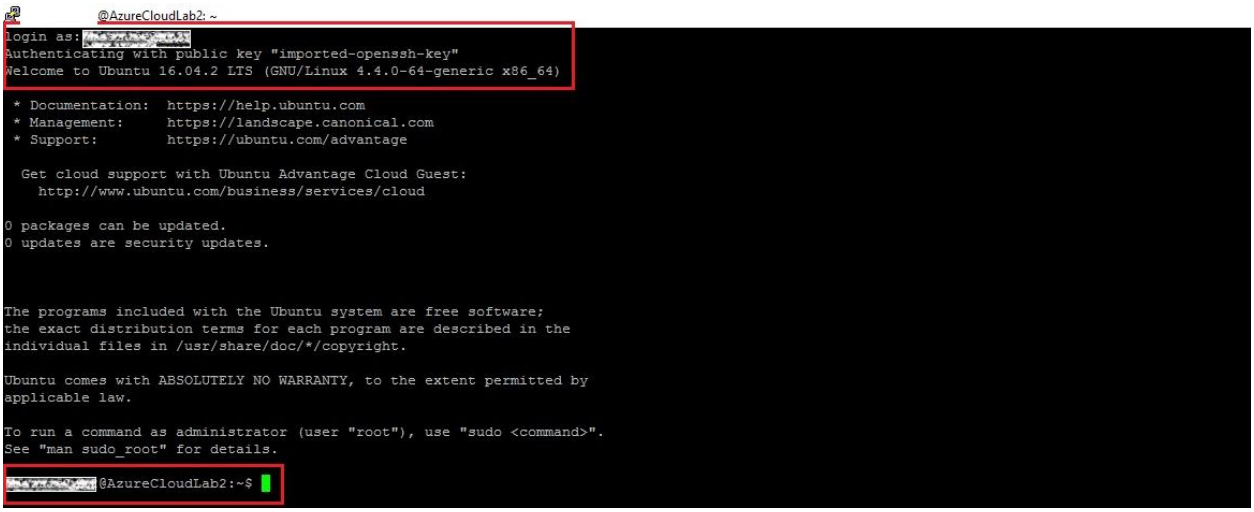

Microsoft Azure
New > Ubuntu Server 16.04 LTS > Create virtual machine > Settings
axm159230@utdallas....
THE UNIVERSITY OF TEXAS AT...
Create virtual machine
1 Basics
Done
2 Size
Done
3 Settings
Configure optional features
4 Summary
Ubuntu Server 16.04 LTS
Settings
Extensions
Extensions
No extensions
High availability
* Availability set
None
Monitoring
Boot diagnostics
Disabled Enabled
Guest OS diagnostics
Disabled Enabled
* Diagnostics storage account
(new) lab2resourcegroupdiag444
OK

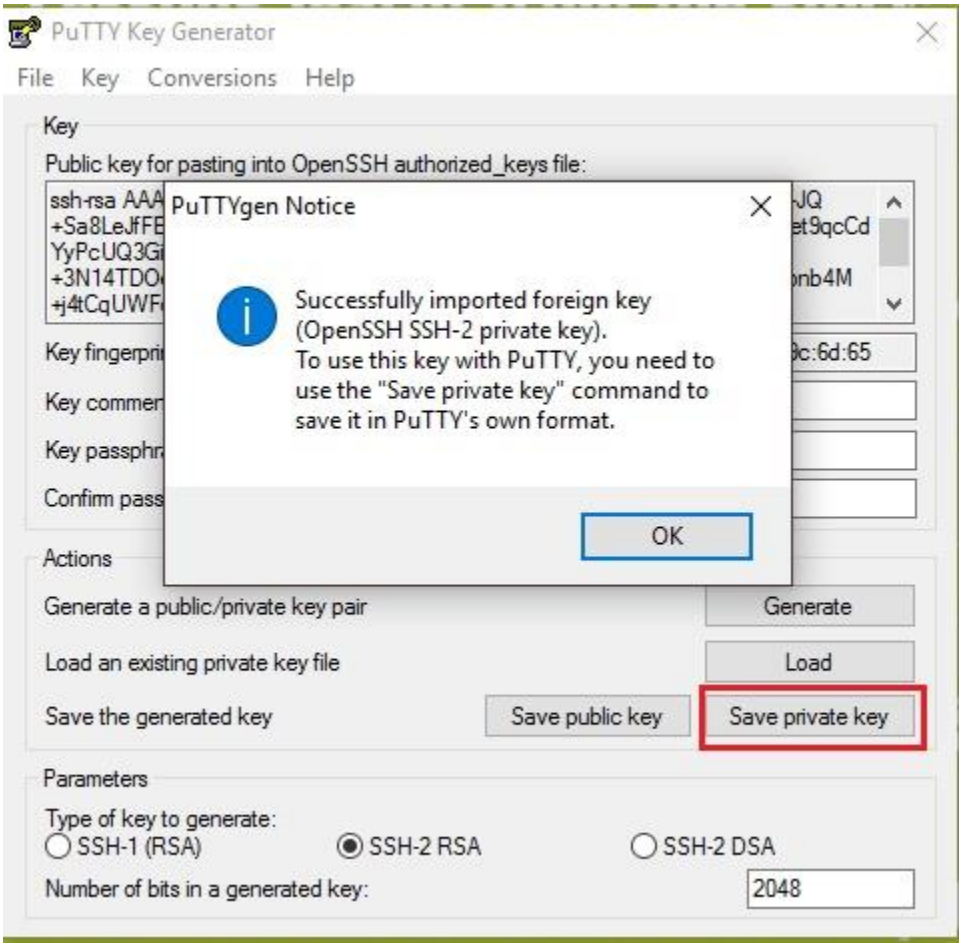

Microsoft Azure
New > Ubuntu Server 16.04 LTS > Create virtual machine > Settings
axm159230@utdallas....
THE UNIVERSITY OF TEXAS AT...
Create virtual machine
1 Basics
Done
2 Size
Done
3 Settings
Configure optional features
4 Summary
Ubuntu Server 16.04 LTS
Settings
Storage
Use managed disks
No Yes
* Storage account
(new) lab2resourcegroup619
Network
* Virtual network
(new) Lab2ResourceGroup-vnet
* Subnet
default (10.0.0.0/24)
* Public IP address
(new) AzureCloudLab2-ip
* Network security group (firewall)
(new) AzureCloudLab2-nsg


Microsoft Azure
Resource groups
axm159230@utdallas....
THE UNIVERSITY OF TEXAS AT...
Resource groups
The University of Texas at Dallas
Add
Columns
Refresh
Subscriptions: Free Trial
Filter by name...
1 items
NAME
SUBSCRIPTION
LOCATION
Lab2ResourceGroup
Free Trial
West Central US


Microsoft Azure
New > Ubuntu Server 16.04 LTS > Create virtual machine > Summary
axm159230@utdallas....
THE UNIVERSITY OF TEXAS AT...
Create virtual machine
1 Basics
Done
2 Size
Done
3 Settings
Done
4 Summary
Ubuntu Server 16.04 LTS
Summary
Validation passed
Basics
Subscription Free Trial
Resource group (new) Lab2ResourceGroup
Location West Central US
Settings
Computer name AzureCloudLab2
Disk type HDD
User name msazurelab2
Size Standard D1 v2
Storage account (new) lab2resourcegroup619
Virtual network (new) Lab2ResourceGroup-vnet
Subnet (new) default (10.0.0.0/24)
Public IP address (new) AzureCloudLab2-ip
Network security group (firewall) (new) AzureCloudLab2-nsg
Availability set None
Guest OS diagnostics Enabled
Boot diagnostics Enabled
Diagnostics storage account (new) lab2resourcegroupdiag444
OK
Download template and parameters

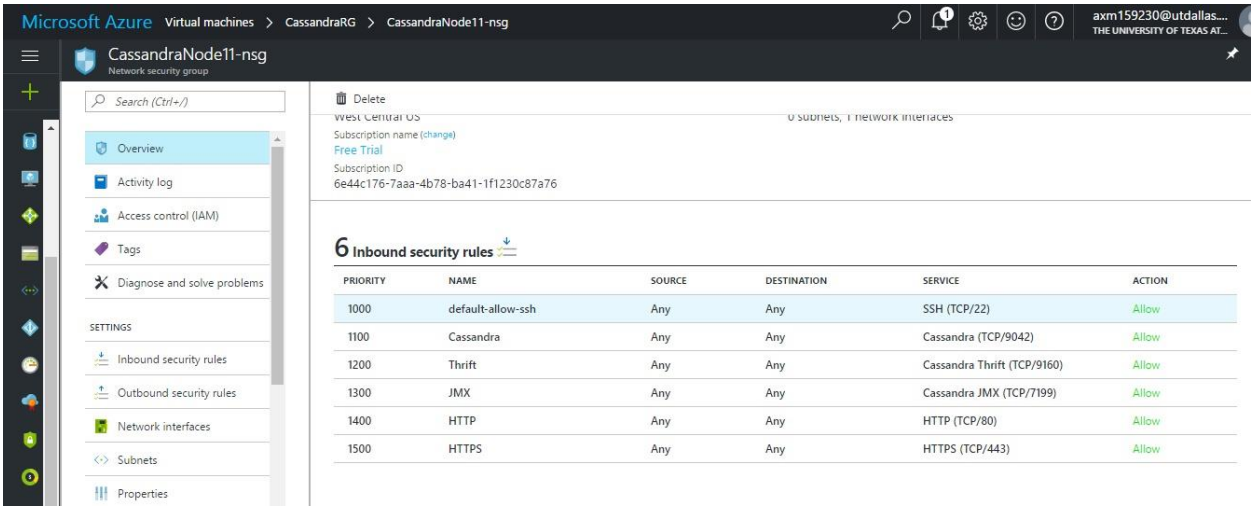
Connecting to instances



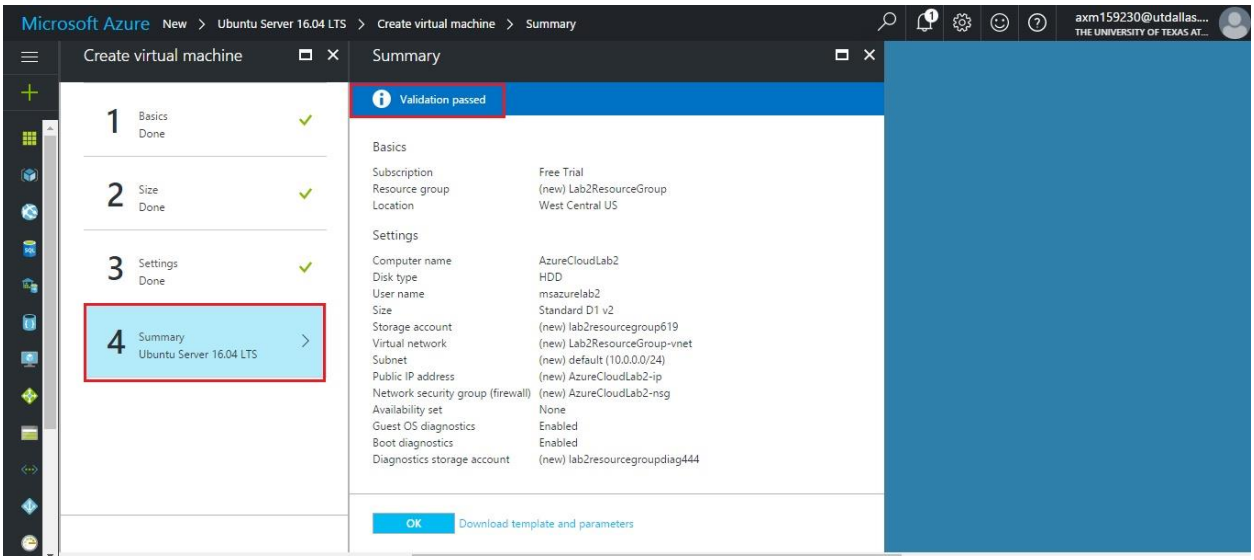


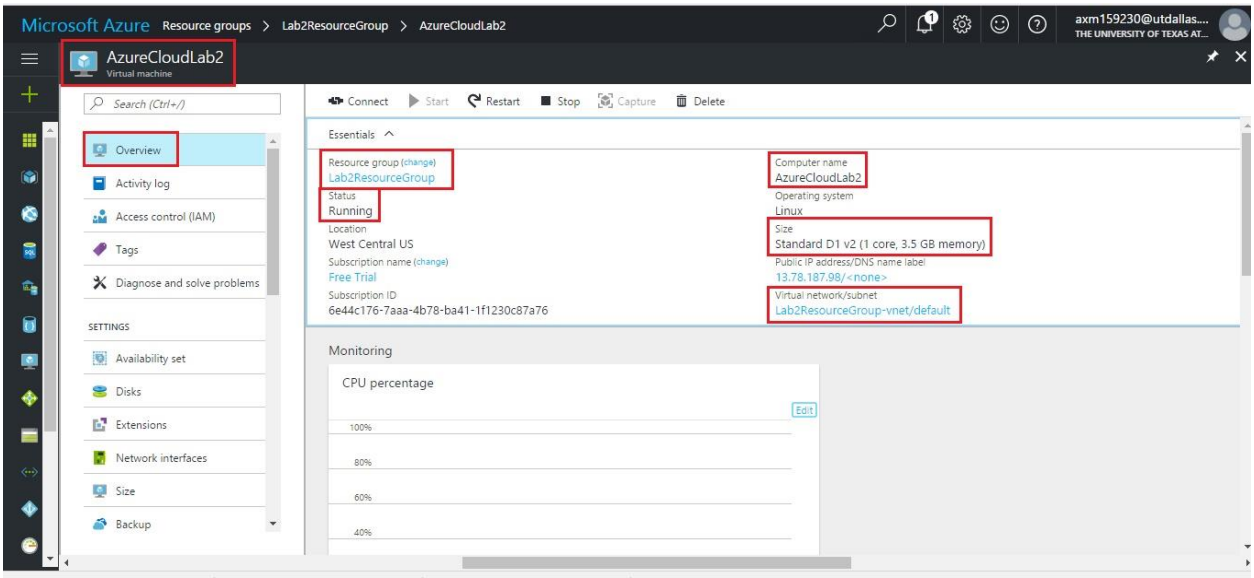
Making instances communicate

Creating Inbound/Outbound rules.



VM Instance specification





Installing Cassandra

- Install Oracle Java 8 in Ubuntu or Linux Mint via PPA

  sudo add-apt-repository ppa:webupd8team/java

  sudo apt-get update

  sudo apt-get install oracle-java8-installer

- Set Java environment variables

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

  sudo apt-get update

- Follow below commands:

```
echo "deb http://www.apache.org/dist/cassandra/debian 39x main" | sudo tee -a /etc/apt/sources.list.d/cassandra.sources.list
```

```
curl https://www.apache.org/dist/cassandra/KEYS | sudo apt-key add -
```

```
sudo apt-key adv --keyserver pool.sks-keyservers.net --recv-key A278B781FE4B2BDA
```

```
sudo apt-get update
```

```
sudo apt-get install cassandra
```

```
sudo systemctl enable cassandra
```

```
sudo systemctl start cassandra
```

```
sudo systemctl stop cassandra
```

```
nodetool status
```

Altering cassandra.yaml file

- Open the cassandra.yaml file in "\etc\cassandra" folder
    - **sudo vi \etc\cassandra\cassandra.yaml**
- Change the following values in file:
    - listen_address : <node IP address> eg:10.0.0.4
    - seeds : <ip addresses of communicating nodes>eg "10.0.0.4, 10.0.0.5, 10.0.0.6"
    - rpc_address : 0.0.0.0
    - rpc_start : true
    - broadcast_rpc_address : 1.2.3.4
    - **:wq** (to save the file)

- Note: After making the changes stop cassandra and then start again.

Creating a Table and Keyspace

```
cqlsh 10.0.0.4
```

```
cqlsh> CREATE KEYSPACE usertable WITH REPLICATION = {'class': 'NetworkTopologyStrategy', 'datacenter1': 1};
```

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

Installing YCSB

- Install Oracle Java 8 in Ubuntu or Linux Mint via PPA

  sudo add-apt-repository ppa:webupd8team/java

  sudo apt-get update

  sudo apt-get install oracle-java8-installer

- Set Java environment variables

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

  sudo apt-get update

- Install YCSB

  curl -O --location https://github.com/brianfrankcooper/YCSB/releases/download/0.3.0/ycsb-0_.3.0.tar.gz

  tar xfvz ycsb-0.3.0.tar.gz

  cd ycsb-0.3.0

- Create workload files in workload folder

  sudo vi workload_test10 (similarly for 40,80,160,320,640 users)

  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

- Create a folder for results

  mkdir results_Cassandra

- Load data

  ./bin/ycsb load cassandra-10 -P workloads/workload_test10 -p hosts=<IP Address> -threads 10 -p columnfamily=data -s > results_Cassandra/load_10

- Run benchmark

  ./bin/ycsb run cassandra-10 -P workloads/workload_test10 -p hosts=<IP Address> -threads 10 -p columnfamily=data -s > results_Cassandra/run_10

- Clear data for next benchmark by using cqlsh (on Cassandra node)

  truncate table usertable.data;

- Check whether all data is cleared (on another Cassandra node).

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

  alter table data with GC_GRACE_SECONDS = 1;

  Note: Create table in any one of the communicating nodes and truncate the table before running any benchmark

- Repeat benchmark with different configurations.

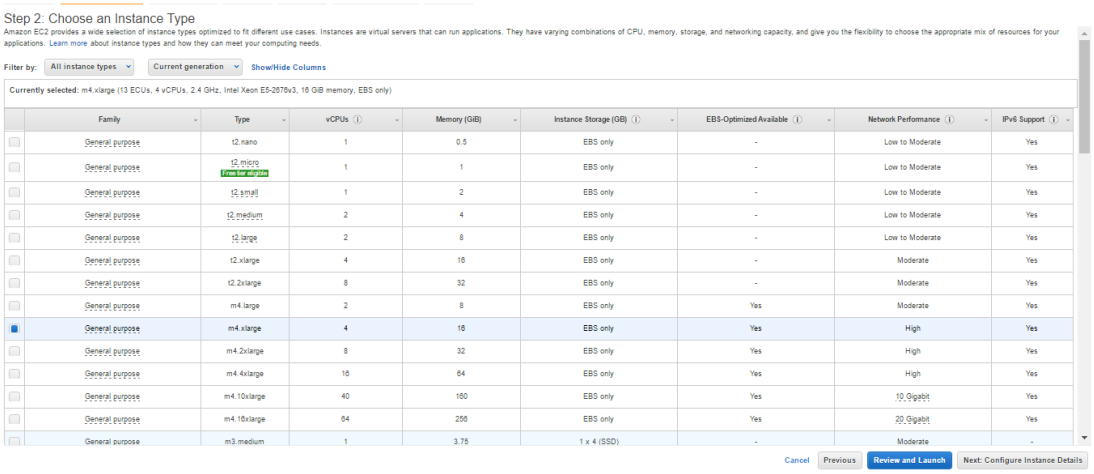
  Note: benchmark for 40 concurrent users

  ./bin/ycsb load cassandra-10 -P workloads/workload_test40 -p hosts=<IP Address> -threads 40 -p columnfamily=data -s > results_Cassandra/load_40

Amazon AWS

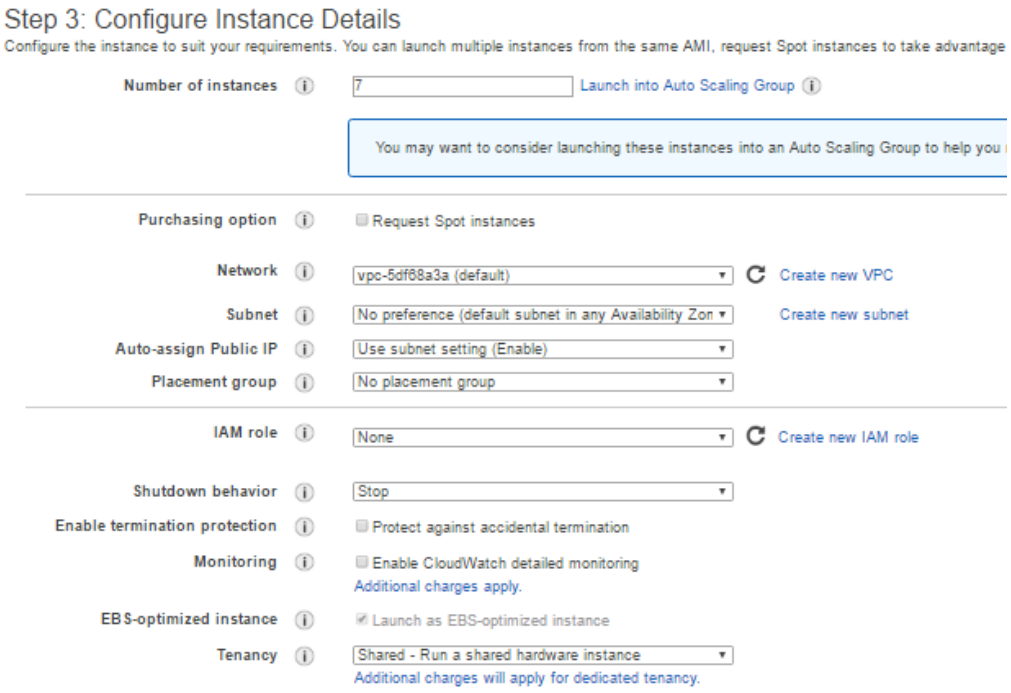
Steps Followed:

1. Creating instances:
   a. Go to AWS EC2 console and click Launch Instance
   b. Now choose the AMI of your choice (we choose Ubuntu Server 16.04 LTS (HVM), SSD Volume Type - ami-a58d0dc5)
   c. Choose the type of instance (we choose m4. xlarge)

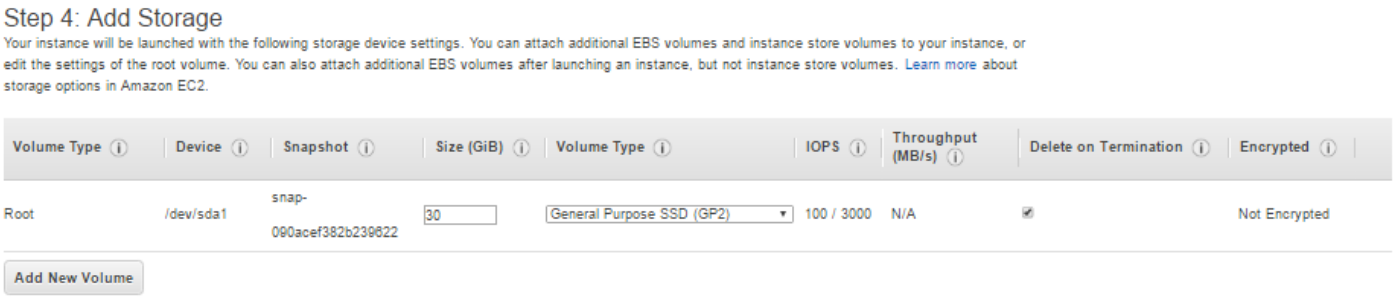


   d. Configure the instance details:
      1. Configuration 1: Cassandra cluster with 1 node: use 2 instances (1 for Cassandra and 1 for YCSB)
      2. Configuration 2: Cassandra cluster with 3 nodes: use 4 instances (3 for Cassandra and 1 for YCSB)
      3. Configuration 3: Cassandra cluster with 6 nodes: use 7 instances (6 for Cassandra and 1 for YCSB)

(NOTE: snip below shows configuration 3 i.e. we will launch 7 instances)



e. Add Storage (we choose 30 GiB)



Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. Learn more about storage options in Amazon EC2.

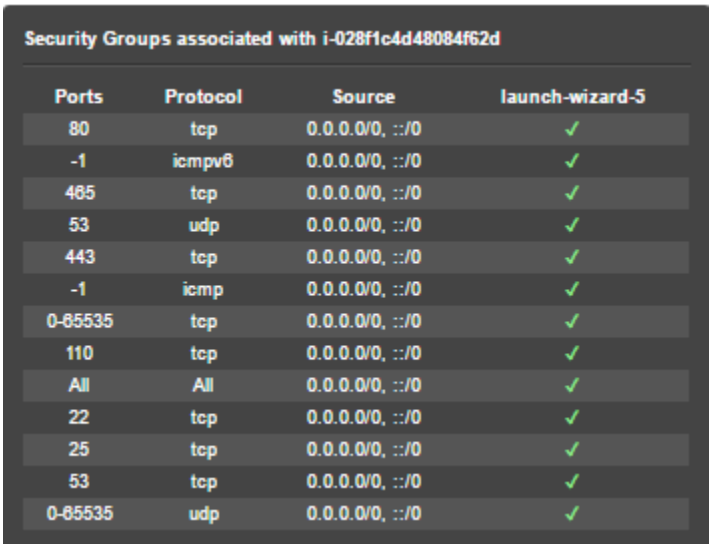
Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/sda1	snap-090acef382b239822	30	General Purpose SSD (GP2)	100 / 3000	N/A	☑	Not Encrypted

Add New Volume

f. Connect the created instances via Putty using the Public IP for each respective instance and Secure Key Pair that we have used during the creation of the instance.

g. Inbound/Outbound Rules: (we choose it as anywhere so that we can connect freely without any restrictions, you may configure your security group features as per your requirements)

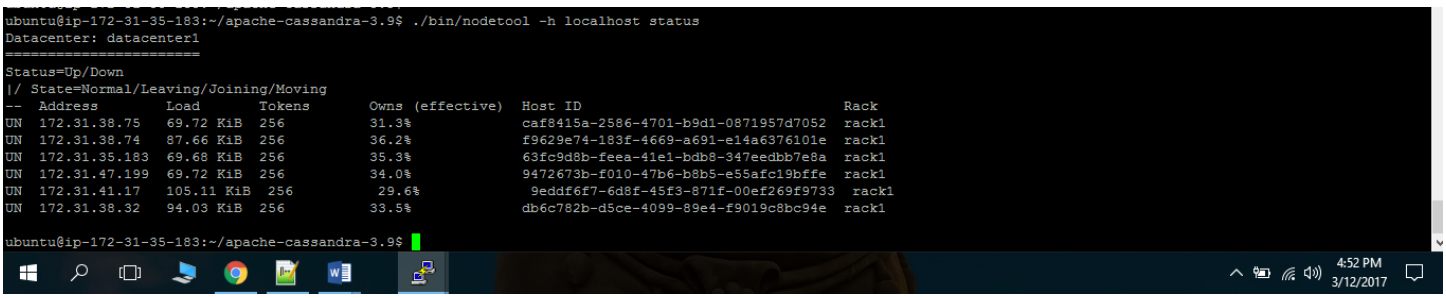
(NOTE : snip below shows Security group rules associated with one of our instance)



Security Groups associated with i-028f1c4d48084f62d

Ports	Protocol	Source	launch-wizard-5
80	tcp	0.0.0.0/0, ::/0	✓
-1	icmpv6	0.0.0.0/0, ::/0	✓
465	tcp	0.0.0.0/0, ::/0	✓
53	udp	0.0.0.0/0, ::/0	✓
443	tcp	0.0.0.0/0, ::/0	✓
-1	icmp	0.0.0.0/0, ::/0	✓
0-65535	tcp	0.0.0.0/0, ::/0	✓
110	tcp	0.0.0.0/0, ::/0	✓
All	All	0.0.0.0/0, ::/0	✓
22	tcp	0.0.0.0/0, ::/0	✓
25	tcp	0.0.0.0/0, ::/0	✓
53	tcp	0.0.0.0/0, ::/0	✓
0-65535	udp	0.0.0.0/0, ::/0	✓

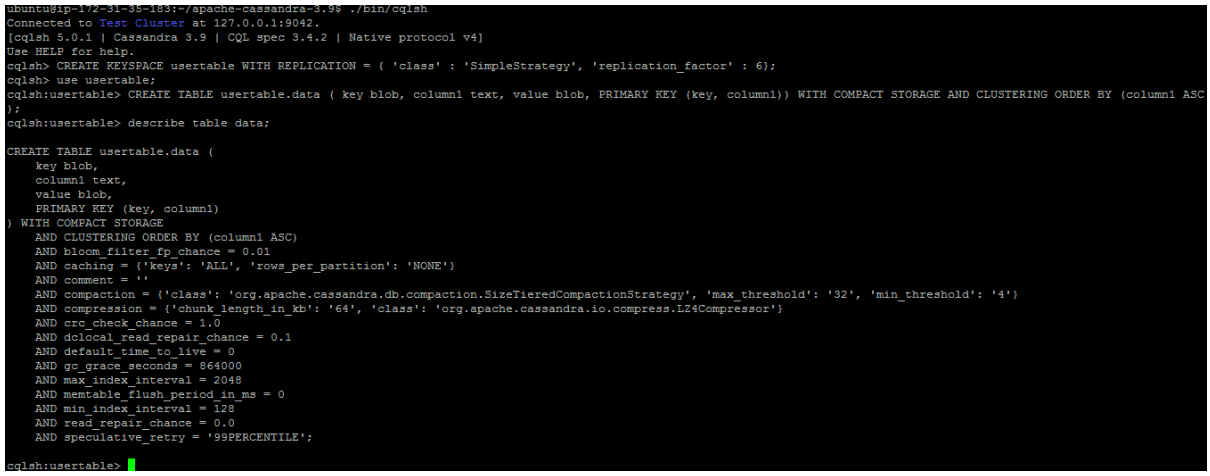
h. Working on instance via Putty (shows all nodes that are working with Cassandra)



```
ubuntu@ip-172-31-35-183:~/apache-cassandra-3.9$ ./bin/nodetool -h localhost status
Datacenter: datacenter1
=======================
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
--  Address         Load       Tokens       Owns (effective)  Host ID                               Rack
UN  172.31.38.75    69.72 KiB  256          31.3%             caf8415a-2586-4701-b9d1-0871957d7052  rack1
UN  172.31.38.74    87.66 KiB  256          36.2%             f9629e74-183f-4669-a691-e14a6376101e  rack1
UN  172.31.35.183   69.68 KiB  256          35.3%             63fc9d8b-feea-41e1-bdb8-347eedbb7e8a  rack1
UN  172.31.47.199   69.72 KiB  256          34.0%             9472673b-f010-47b6-b8b5-e55afc19bffe  rack1
UN  172.31.41.17    105.11 KiB  256          29.6%             9eddf6f7-6d8f-45f3-871f-00ef269f9733  rack1
UN  172.31.38.32    94.03 KiB  256          33.5%             db6c782b-d5ce-4099-89e4-f9019c8bc94e  rack1

ubuntu@ip-172-31-35-183:~/apache-cassandra-3.9$
```

i. Creating replication tables in Cassandra: (we choose 1 node set replication factor as 1, set it 2 for 2 nodes and 6 for 6 nodes)

(NOTE: snip below shows the table definition)



Steps for Running Cassandra on Ec2 instance:

1. Choose your version of Cassandra (we choose 3.9)

   curl -OL http://www.apache.org/dist/cassandra/3.9/apache-cassandra-3.9-bin.tar.gz
   tar xzf apache-cassandra-3.9-bin.tar.gz

2. Install Java
3. Install Python
4. Run Cassandra through given commands as below :
   cd/apache-cassandra-3.2.1/
   bin/cassandra -R
   /bin/nodetool -h localhost status

YCSB commands are as under:

- ./bin/ycsb load cassandra-10 -P workloads/workload_test10 -p hosts=172.31.44.1 -threads 10 -p columnfamily=data -s > results_Cassandra/load_10
- ./bin/ycsb run cassandra-10 -P workloads/workload_test10 -p hosts=172.31.44.1 -threads 10 -p columnfamily=data -s > results_Cassandra/run_10
- ./bin/ycsb load cassandra-10 -P workloads/workload_test40 -p hosts=172.31.44.1 -threads 10 -p columnfamily=data -s > results_Cassandra/load_40
- ./bin/ycsb run cassandra-10 -P workloads/workload_test40 -p hosts=172.31.44.1 -threads 10 -p columnfamily=data -s > results_Cassandra/run_40
- ./bin/ycsb load cassandra-10 -P workloads/workload_test80 -p hosts=172.31.44.1 -threads 10 -p columnfamily=data -s > results_Cassandra/load_80
- ./bin/ycsb run cassandra-10 -P workloads/workload_test80 -p hosts=172.31.44.1 -threads 10 -p columnfamily=data -s > results_Cassandra/run_80
- ./bin/ycsb load cassandra-10 -P workloads/workload_test160 -p hosts=172.31.44.1 -threads 10 -p columnfamily=data -s > results_Cassandra/load_160
- ./bin/ycsb run cassandra-10 -P workloads/workload_test160 -p hosts=172.31.44.1 -threads 10 -p columnfamily=data -s > results_Cassandra/run_160
- ./bin/ycsb load cassandra-10 -P workloads/workload_test320 -p hosts=172.31.44.1 -threads 10 -p columnfamily=data -s > results_Cassandra/load_320
- ./bin/ycsb run cassandra-10 -P workloads/workload_test320 -p hosts=172.31.44.1 -threads 10 -p columnfamily=data -s > results_Cassandra/run_320
- ./bin/ycsb load cassandra-10 -P workloads/workload_test640 -p hosts=172.31.44.1 -threads 10 -p columnfamily=data -s > results_Cassandra/load_640
- ./bin/ycsb run cassandra-10 -P workloads/workload_test640 -p hosts=172.31.44.1 -threads 10 -p columnfamily=data -s > results_Cassandra/run_640

Benchmark Observations (AWS EC2)

#1 nodes

concurrent users	load		run	
	RunTime(ms)	Throughput(ops/sec)	RunTime(ms)	Throughput(ops/sec)
10	19672	12990.54494	8886	11253.65744
40	37447	15211.3654	26430	15134.31706
80	70700	13937.36917	63522	12594.0619
160	130384	13855.76451	158556	10091.07192
320	247471	13770.2357	291366	10982.75022
640	480565	13411.57804	2380461	2688.554864

3 nodes

concurrent users	load		run	
	RunTime(ms)	Throughput(ops/sec)	RunTime(ms)	Throughput(ops/sec)
10	28137	9082.347087	13161	7598.206823
40	49548	11496.32679	36362	11000.49502
80	90774	10855.22286	81743	9786.770733
160	177056	10203.38198	176318	9074.513096
320	355224	9593.197532	371858	8605.435408
640	689910	9341.993883	1037578	6168.210968

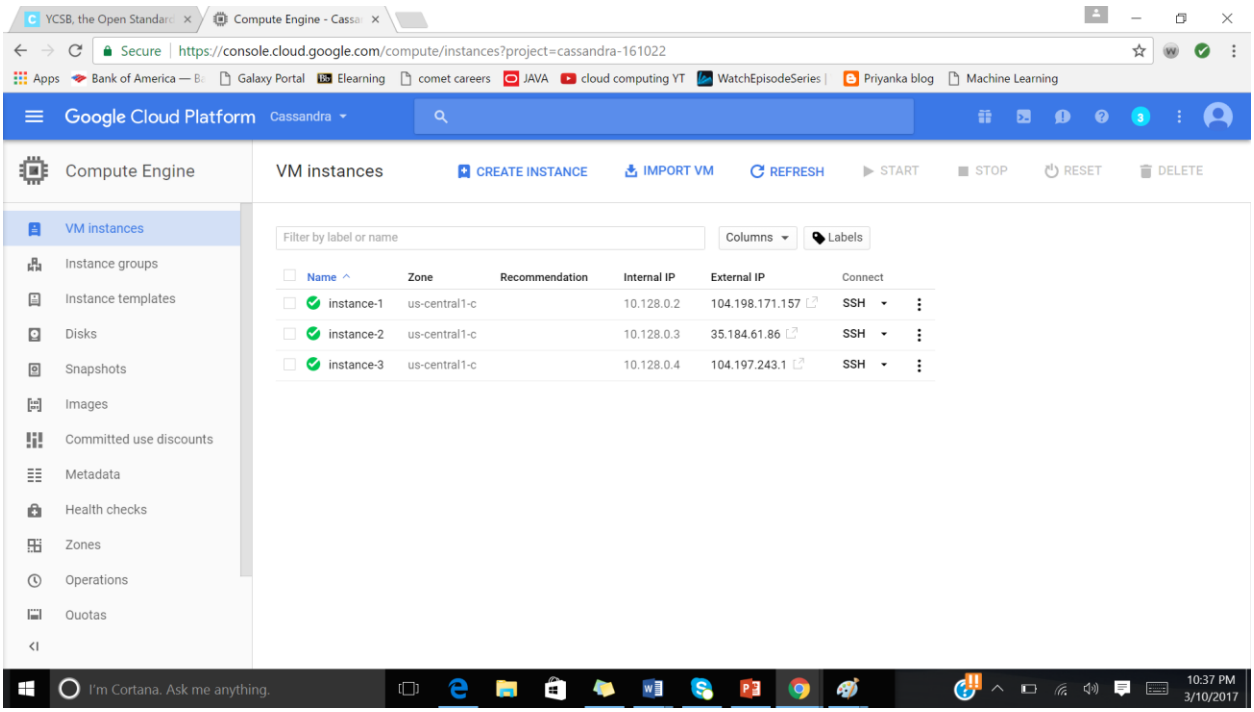
6 nodes

concurrent users	load		run	
	RunTime(ms)	Throughput(ops/sec)	RunTime(ms)	Throughput(ops/sec)
10	38342	6665.014866	15148	6601.531555
40	72479	7859.104016	45702	8752.352195
80	131366	7500.966765	96136	8321.544479
160	240137	7523.080575	214151	7471.363664
320	456853	7459.147691	420728	7605.864121
640	898313	7174.709706	1161273	5511.19332

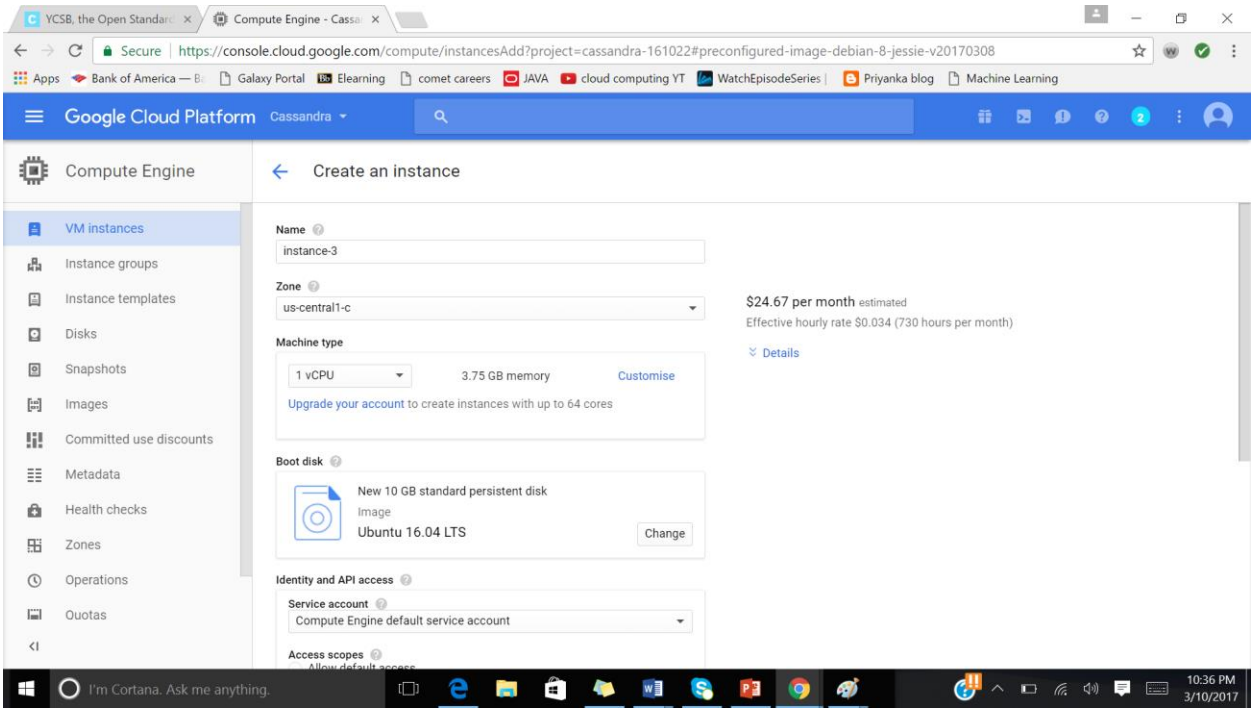
GOOGLE COMPUTE ENGINE

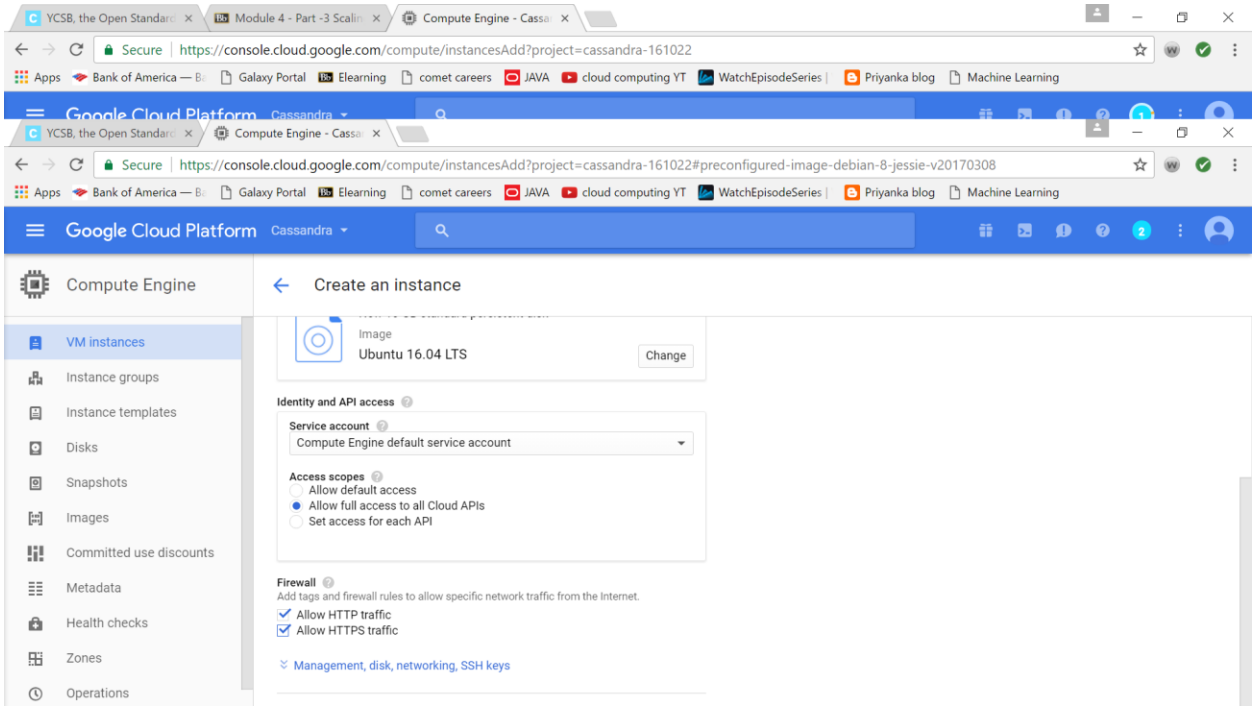
To create instances

1. Click on the **create instance** option on top of the dashboard.



2. Change name of the instance, if needed.
   Select Ubuntu 16.04 LTS on **Boot disk.**
   In firewall option, allow HTTP and HTTPS traffic.





VM Instance configuration

Operating system - Ubuntu 16.04 LTS

Hard Disk size – 10GB standard persistent disk

Memory – 3.75GB

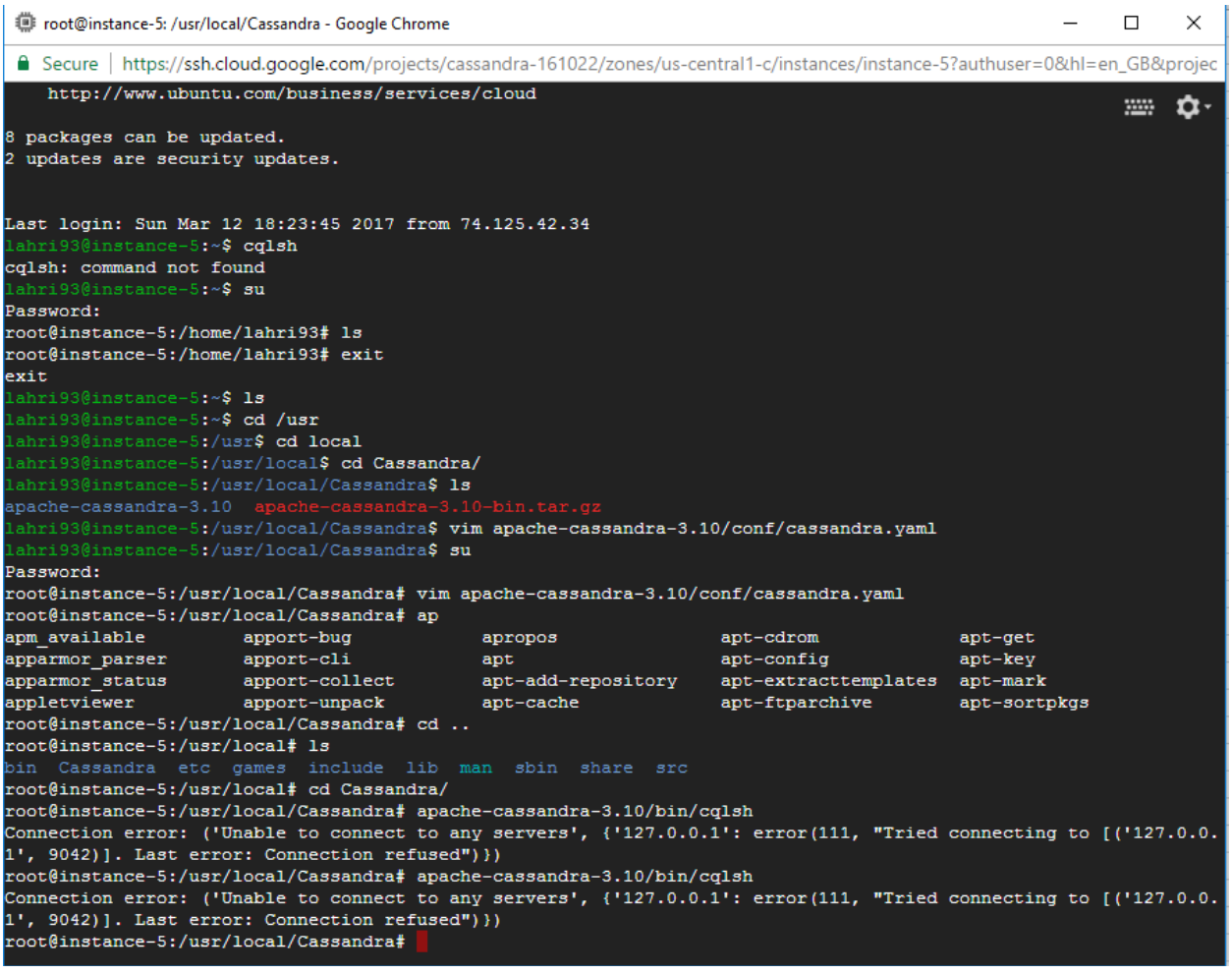
Number of cores – 1

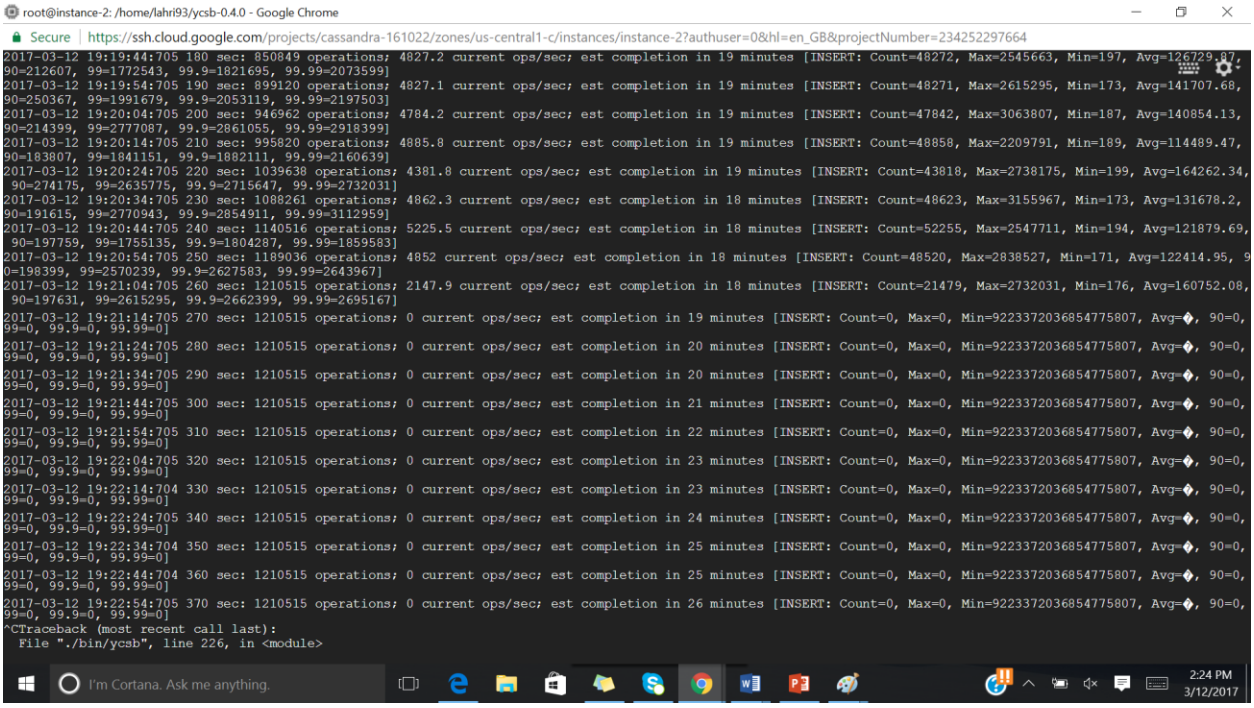
Problems faced

1. We tried working the same set of commands on Cassandra's 3.10 version but found that commands didn't worked out and same set of commands worked fine on Cassandra's 3.9 version
2. Cassandra YCSB are case sensitive for e.g :

   ./bin/ycsb load cassandra-10 **-P** workloads/workload_test10 -p hosts=172.31.44.1 -threads 10 -p columnfamily=data -s > results_Cassandra/load_10

   Highlighted P doesn't work if it is small case. We have to debug the command and came to conclusion that commands are case sensitive.
3. Changing the Cassandra configuration file totally varies on what version of Cassandra you are using we have to look for online help to get through the .yaml file configuration
4. YCSB commands executed in different manner ; might be because of network bandwidth
5. Given set of YCSB not compatible with current version of YSCB i.e. yscb-0.12.0 hence we have to switchback to ycsb-0.3.0
6. Java.net.SocketException – Broken pipe: Running the workload for 640 concurrent users crashed the system because the node with given configurations could not support that huge amount of data
7. Cannot connect to server using cqlsh – (10.0.0.4, 9160) connection not allowed: Proper Inbound Rules were not set which was not allowing cqlsh to connect to the required server.
   Set the proper Inbound Rules allowing following services (Cassandra, port: 9042; Thrift, port: 9160; Cassandra (JMX), port: 7199)




root@instance-2: /home/lahri93/ycsb-0.4.0 - Google Chrome
Secure | https://ssh.cloud.google.com/projects/cassandra-161022/zones/us-central1-c/instances/instance-2?authuser=0&hl=en_GB&projectNumber=234252297664
2017-03-12 19:19:44:705 180 sec: 850849 operations; 4827.2 current ops/sec; est completion in 19 minutes [INSERT: Count=48272, Max=2545663, Min=197, Avg=126729.87, 90=212607, 99=1772543, 99.9=1821695, 99.99=2073599]
2017-03-12 19:19:54:705 190 sec: 899120 operations; 4827.1 current ops/sec; est completion in 19 minutes [INSERT: Count=48271, Max=2615295, Min=173, Avg=141707.68, 90=250367, 99=1991679, 99.9=2053119, 99.99=2197503]
2017-03-12 19:20:04:705 200 sec: 946962 operations; 4784.2 current ops/sec; est completion in 19 minutes [INSERT: Count=47842, Max=3063807, Min=187, Avg=140854.13, 90=214399, 99=2777087, 99.9=2861055, 99.99=2918399]
2017-03-12 19:20:14:705 210 sec: 995820 operations; 4885.8 current ops/sec; est completion in 19 minutes [INSERT: Count=48858, Max=2209791, Min=189, Avg=114489.47, 90=183807, 99=1841151, 99.9=1882111, 99.99=2160639]
2017-03-12 19:20:24:705 220 sec: 1039638 operations; 4381.8 current ops/sec; est completion in 19 minutes [INSERT: Count=43818, Max=2738175, Min=199, Avg=164262.34, 90=274175, 99=2635775, 99.9=2715647, 99.99=2732031]
2017-03-12 19:20:34:705 230 sec: 1088261 operations; 4862.3 current ops/sec; est completion in 18 minutes [INSERT: Count=48623, Max=3155967, Min=173, Avg=131678.2, 90=191615, 99=2770943, 99.9=2854911, 99.99=3112959]
2017-03-12 19:20:44:705 240 sec: 1140516 operations; 5225.5 current ops/sec; est completion in 18 minutes [INSERT: Count=52255, Max=2547711, Min=194, Avg=121879.69, 90=197759, 99=1755135, 99.9=1804287, 99.99=1859583]
2017-03-12 19:20:54:705 250 sec: 1189036 operations; 4852 current ops/sec; est completion in 18 minutes [INSERT: Count=48520, Max=2838527, Min=171, Avg=122414.95, 90=198399, 99=2570239, 99.9=2627583, 99.99=2643967]
2017-03-12 19:21:04:705 260 sec: 1210515 operations; 2147.9 current ops/sec; est completion in 18 minutes [INSERT: Count=21479, Max=2732031, Min=176, Avg=160752.08, 90=197631, 99=2615295, 99.9=2662399, 99.99=2695167]
2017-03-12 19:21:14:705 270 sec: 1210515 operations; 0 current ops/sec; est completion in 19 minutes [INSERT: Count=0, Max=0, Min=9223372036854775807, Avg=, 90=0, 99=0, 99.9=0, 99.99=0]
2017-03-12 19:21:24:705 280 sec: 1210515 operations; 0 current ops/sec; est completion in 20 minutes [INSERT: Count=0, Max=0, Min=9223372036854775807, Avg=, 90=0, 99=0, 99.9=0, 99.99=0]
2017-03-12 19:21:34:705 290 sec: 1210515 operations; 0 current ops/sec; est completion in 20 minutes [INSERT: Count=0, Max=0, Min=9223372036854775807, Avg=, 90=0, 99=0, 99.9=0, 99.99=0]
2017-03-12 19:21:44:705 300 sec: 1210515 operations; 0 current ops/sec; est completion in 21 minutes [INSERT: Count=0, Max=0, Min=9223372036854775807, Avg=, 90=0, 99=0, 99.9=0, 99.99=0]
2017-03-12 19:21:54:705 310 sec: 1210515 operations; 0 current ops/sec; est completion in 22 minutes [INSERT: Count=0, Max=0, Min=9223372036854775807, Avg=, 90=0, 99=0, 99.9=0, 99.99=0]
2017-03-12 19:22:04:705 320 sec: 1210515 operations; 0 current ops/sec; est completion in 23 minutes [INSERT: Count=0, Max=0, Min=9223372036854775807, Avg=, 90=0, 99=0, 99.9=0, 99.99=0]
2017-03-12 19:22:14:704 330 sec: 1210515 operations; 0 current ops/sec; est completion in 23 minutes [INSERT: Count=0, Max=0, Min=9223372036854775807, Avg=, 90=0, 99=0, 99.9=0, 99.99=0]
2017-03-12 19:22:24:705 340 sec: 1210515 operations; 0 current ops/sec; est completion in 24 minutes [INSERT: Count=0, Max=0, Min=9223372036854775807, Avg=, 90=0, 99=0, 99.9=0, 99.99=0]
2017-03-12 19:22:34:704 350 sec: 1210515 operations; 0 current ops/sec; est completion in 25 minutes [INSERT: Count=0, Max=0, Min=9223372036854775807, Avg=, 90=0, 99=0, 99.9=0, 99.99=0]
2017-03-12 19:22:44:704 360 sec: 1210515 operations; 0 current ops/sec; est completion in 25 minutes [INSERT: Count=0, Max=0, Min=9223372036854775807, Avg=, 90=0, 99=0, 99.9=0, 99.99=0]
2017-03-12 19:22:54:705 370 sec: 1210515 operations; 0 current ops/sec; est completion in 26 minutes [INSERT: Count=0, Max=0, Min=9223372036854775807, Avg=, 90=0, 99=0, 99.9=0, 99.99=0]
^CTraceback (most recent call last):
File "./bin/ycsb", line 226, in <module>
I'm Cortana. Ask me anything.
2:24 PM
3/12/2017