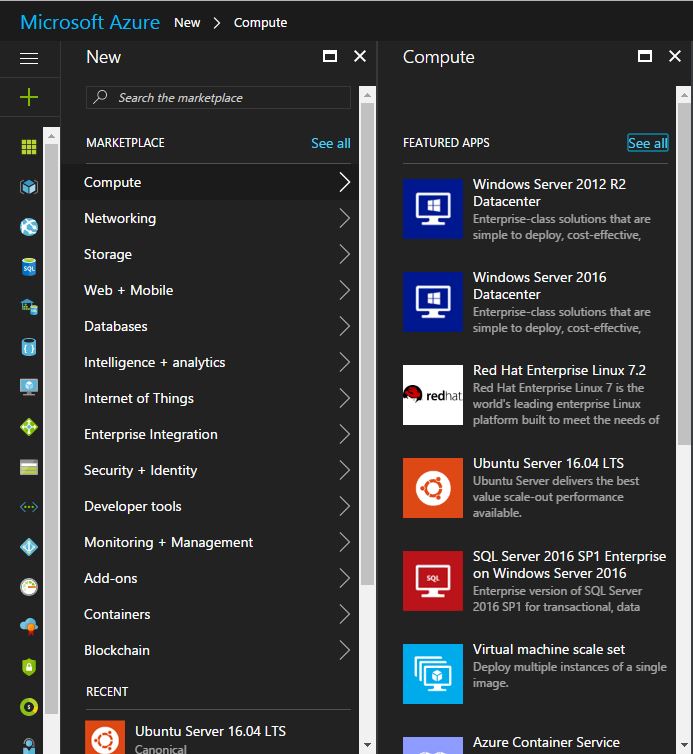
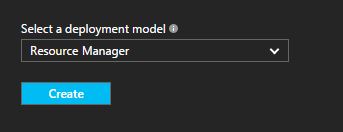
**Deploying VM in Azure:**

First Choose **Compute ->** Select a server type.



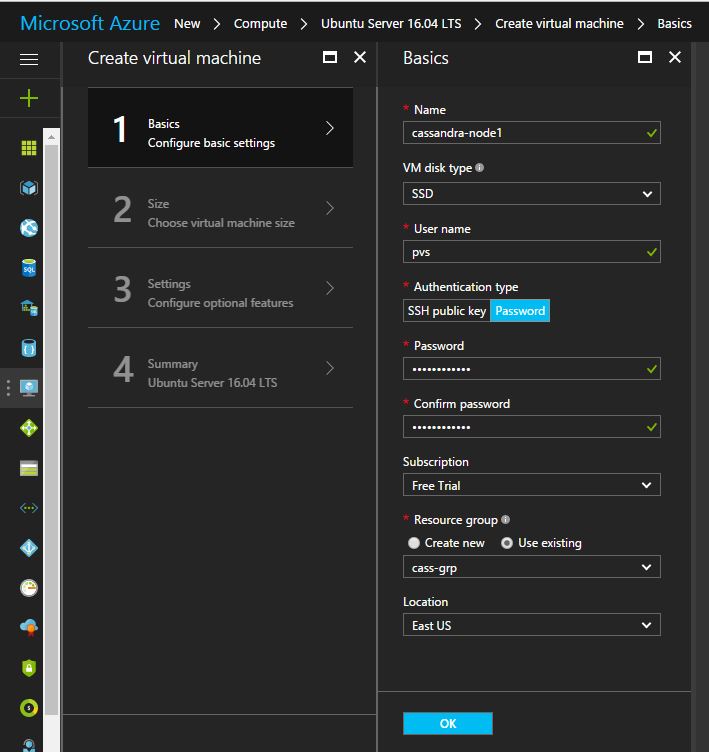
Then select the deployment model.



Then configure the server details.

It has 4 steps.

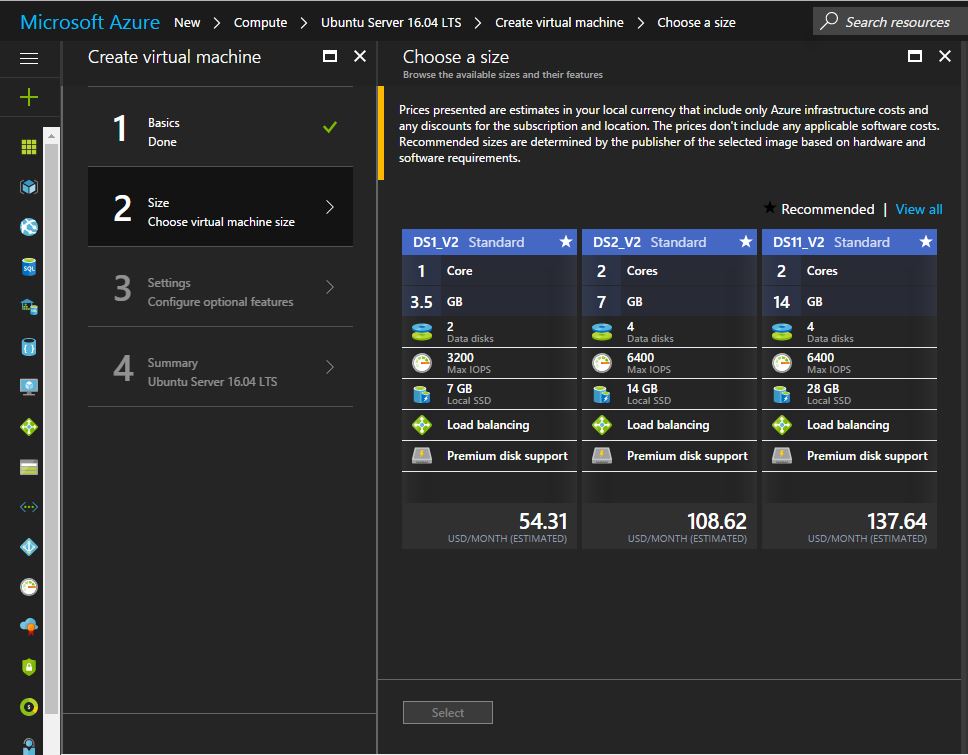
**Step-1:**



Fill out the details like name of the instance, Username for authentication

* For authentication, we can use either SSH Public key or setup a password.
* Choose a resource group if u have any, otherwise create one.
* Select the location of your Instance.

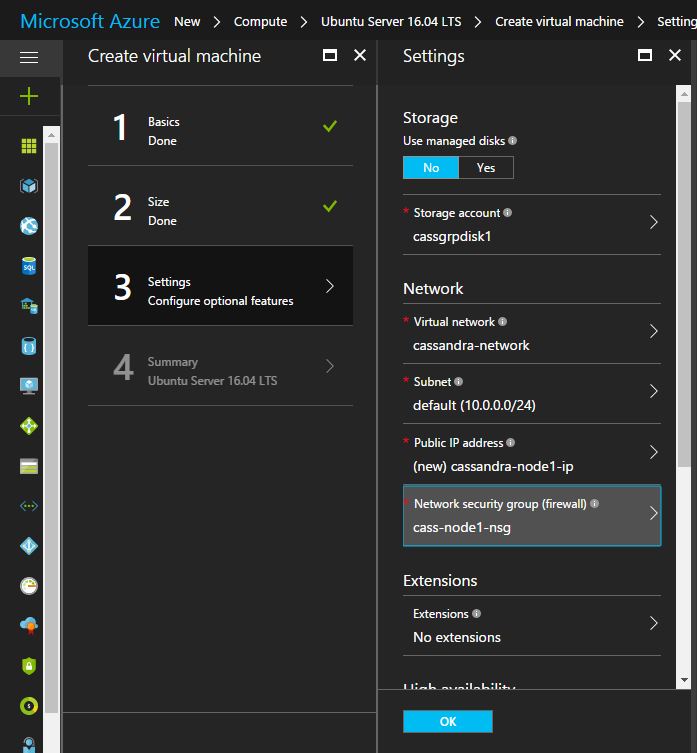
**Step-2:**



Choose an Instance type.

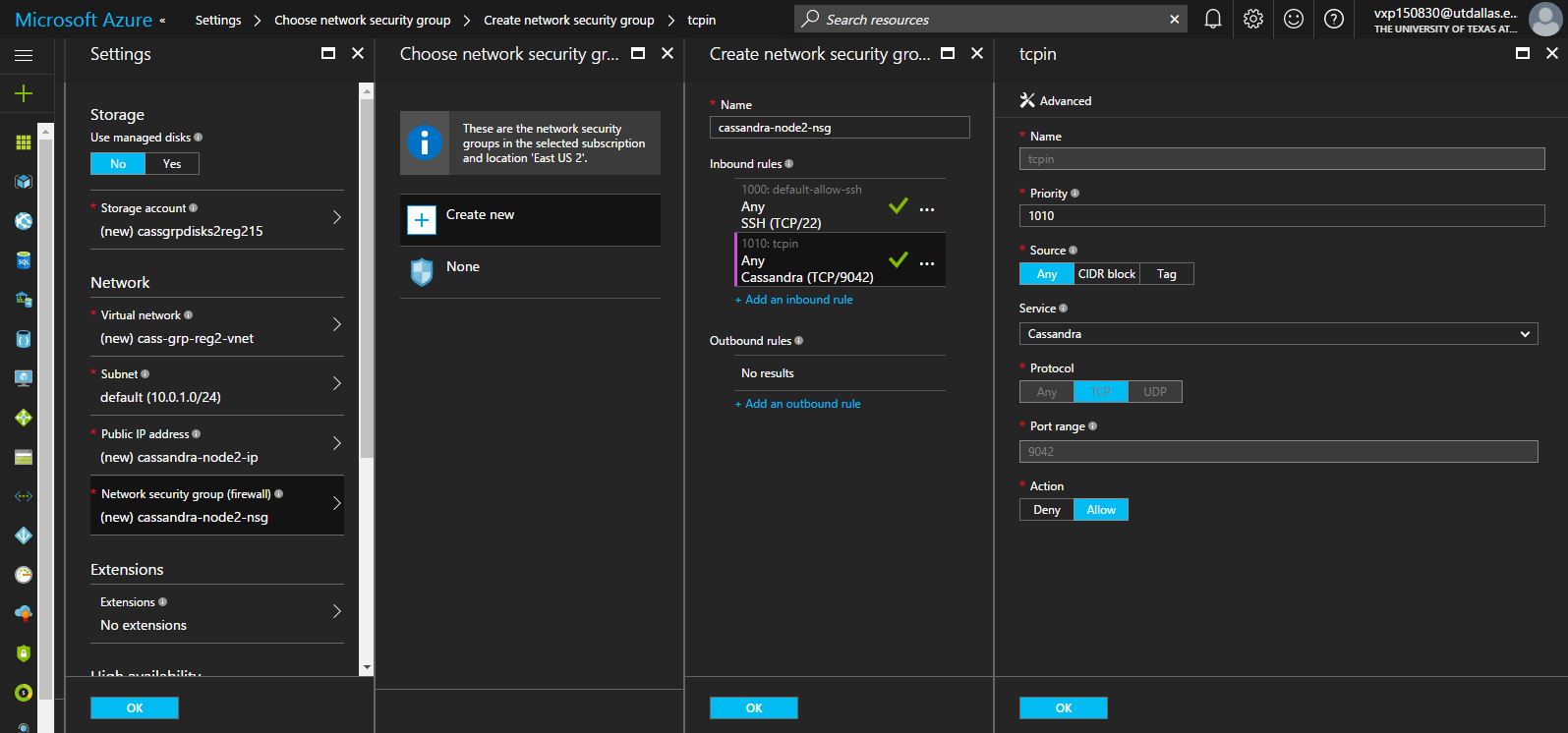
* For a free version of Microsoft Azure we ca use only a total of 4 Cores in any region.

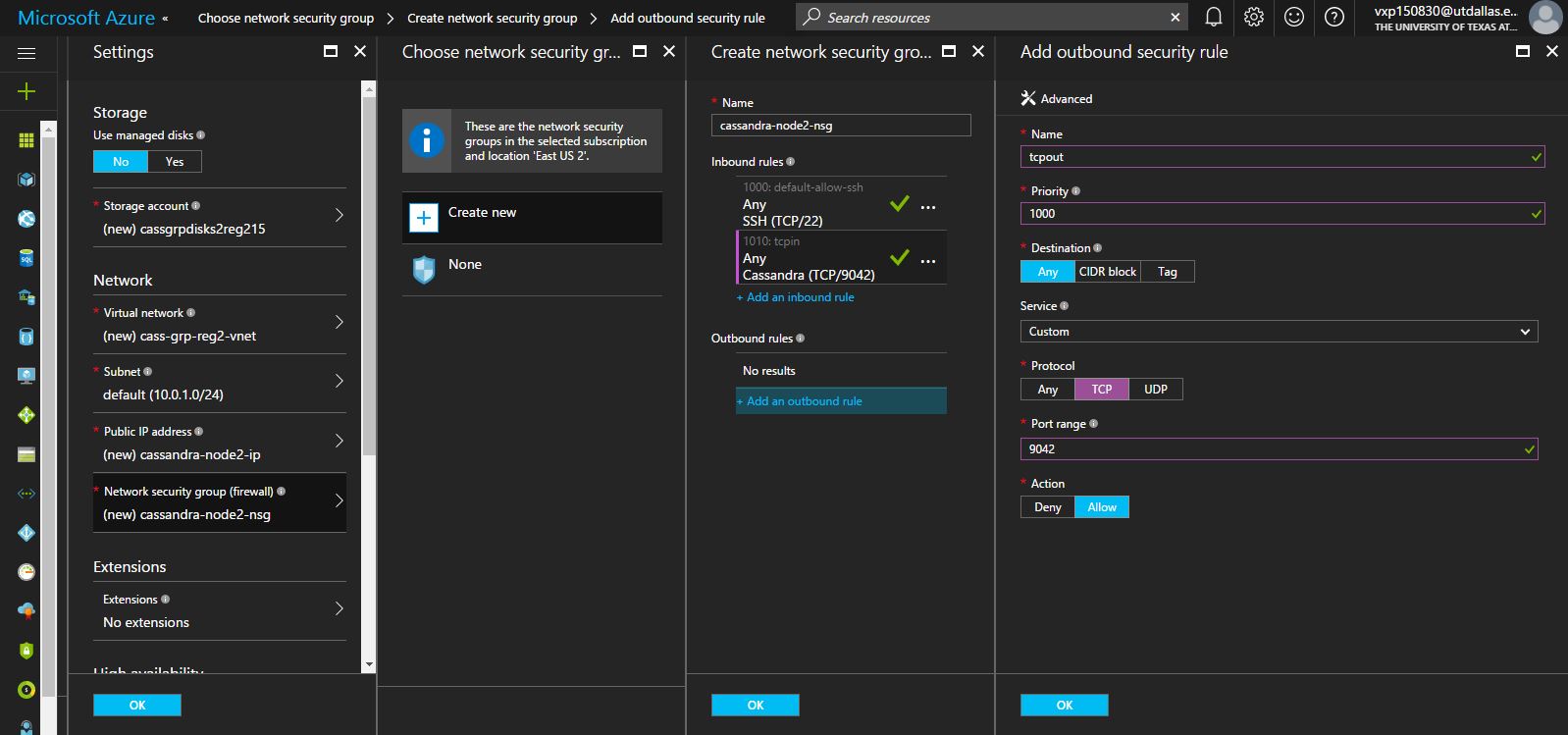
**Step-3:**



Create a storage account -> it tells the details how data is stored and can be removed if necessary.

* Virtual network helps to create a network where we can assign static or dynamin ip’s to the VM.
* Subnet defines the range of the ip’s that can be assigned to the VM.
* Network security group is the place where we will add all the protocols related to that VM.
  + - For a Cassandra VM we need the TCP port 9042 for communication.
    - We need to setup them in the network security group, for Inbound and Outbound rules.





**Step-4:**

It is the review screen where all the details about the VM are displayed before deploying the server.

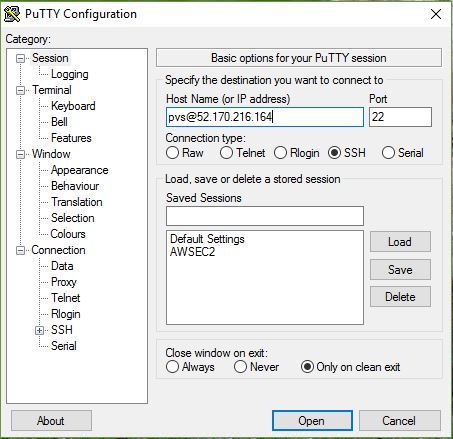
We can review the details and modify, if we need to do any changes before deploying the server.

***Note: for my instance, I have chosen to setup a password for authentication.***

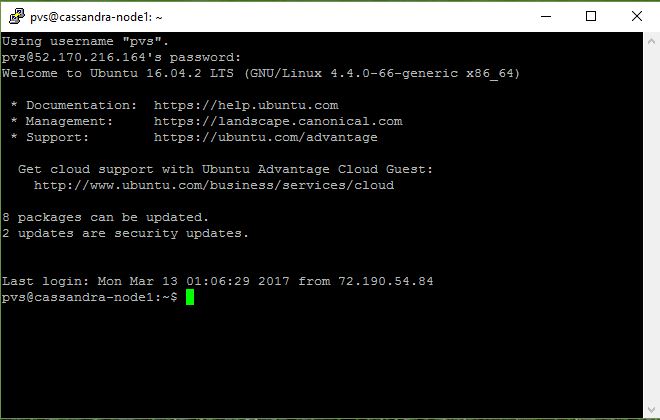
**Logging into the Server:**

Open Putty.

Then enter the Ip address of the server.



In the Login Screen Enter Username and Password:



**Setting Up Cassandra:**

Now we need to setup Cassandra in each VM, so that each instance can act like a node.

Oracle Java 8 and JNI are prerequisites for Cassandra v3.

* Install Oracle Java 8 using:

# To install java:

*sudo apt-add-repository ppa:webupd8team/java*

*sudo apt-get update*

*sudo apt-get install oracle-java8-installer*

# Check that java is properly installed:

*java -version*

* Install JNA using:

*sudo apt-get install libjna-java -y*

* Install Cassandra:

# This will set up the PPAs for Cassandra and the keys for verification.

*echo "deb http://www.apache.org/dist/cassandra/debian 30x main" | sudo tee -a /etc/apt/sources.list.d/cassandra.sources.list*

*echo "deb-src http://www.apache.org/dist/cassandra/debian 30x main" | sudo tee -a /etc/apt/sources.list.d/cassandra.sources.list*

*gpg --keyserver pgp.mit.edu --recv-keys F758CE318D77295D*

*gpg --export --armor F758CE318D77295D | sudo apt-key add -*

*gpg --keyserver pgp.mit.edu --recv-keys 2B5C1B00*

*gpg --export --armor 2B5C1B00 | sudo apt-key add -*

*gpg --keyserver pgp.mit.edu --recv-keys 0353B12C*

*gpg --export --armor 0353B12C | sudo apt-key add –*

Now Install Cassandra:

*sudo apt-get update*

*sudo apt-get install Cassandra*

# To check if cassandra is running:

*sudo service cassandra status*

# To check the nodes

*sudo nodetool status*

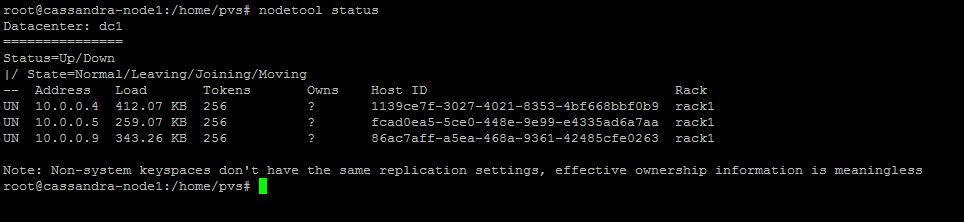
* configuring Cassandra for Multi-Node setup:
  + - Stop Cassandra using: *sudo service cassandra stop.*
    - Find your ethernet card interface ID using *ifconfig*, it should be eth(x).
    - Edit Cassandra's configuration cassandra.yaml: *sudo vim /etc/cassandra/cassandra.yaml*
      * Change the cluster name.
      * Add the IP addresses of the seed nodes.
      * Comment out the listen\_address.
      * Add the listen interface.
      * Start the RPC service.
      * Set the RPC interface.
      * Set the endpoint snitch.
      * By editing the file: sudo vim /etc/cassandra/cassandra.yaml (Shown Below)
* cluster\_name: 'Pvs Cluster'
* seeds: "10.0.0.4,10.0.0.5"
* # listen\_address:
* listen\_interface: eth0
* start\_rpc: **true**
* # rpc\_address:
* rpc\_interface: eth0
* endpoint\_snitch: GossipingPropertyFileSnitch

Delete all Cassandra system configurations: *sudo rm -rf /var/lib/cassandra/data/system/.*

Start Cassandra: *sudo service cassandra start*.

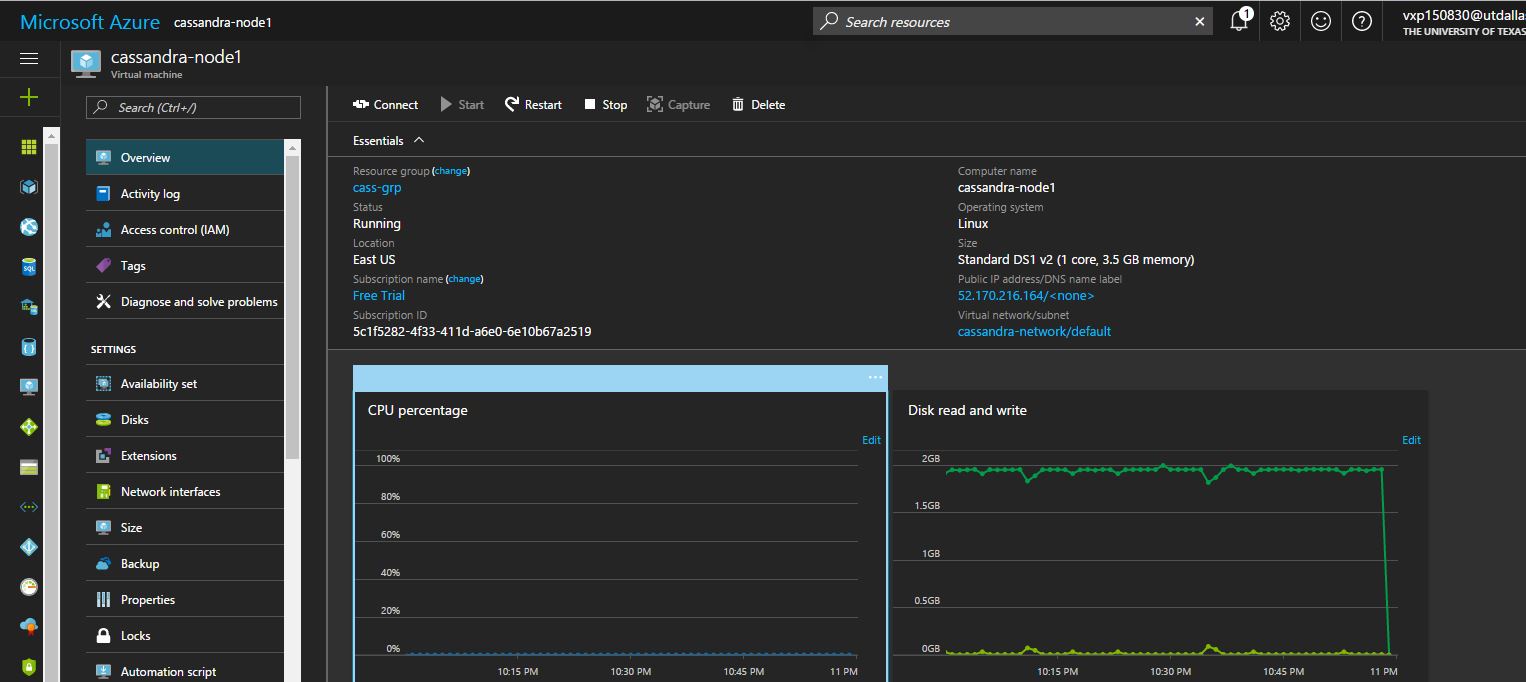
check the nodes using: *sudo nodetool status.*

The output of the nodetool status gives the list of nodes present in the cluster.



VM Instance specification:

Once the VM Instance is launched it looks like this.



*Specifications:*

* 1 core
* 3.5 GB RAM
* 2 Data Disks
* 3200 Max IOPS
* 7 GB Local SSD

Pricing: 54.31 USD/month

**Issues Faced:**

1) Microsoft Azure doesn’t provide much flexibility for free users.

The maximum number of cores that can be run in a region is 4. So, if we choose a 2-core instance, we can create more than two instances in one region. So, for a 3-node cluster we need to use 2 regions and for a 6-node cluster we need to use 3 regions. For benchmarking, we need an extra instance, which is going to be in another region. Setup communication between instances of different regions and the Interface is clumsy.

*Solution*: what we decided is that we choose VM’s with one core so that we can run 1-node and 3-node cluster.

2) Instance capacity:

Azure VM with 1-core and 2-core cannot run 640 concurrent users with a 1-node Cassandra configuration.

*Solution*: So, the maximum we can compare is up to 320 users for a single node Cassandra configuration.

3) The free tier of Azure has lot of restrictions which is very difficult to use even for the benchmarking purpose.