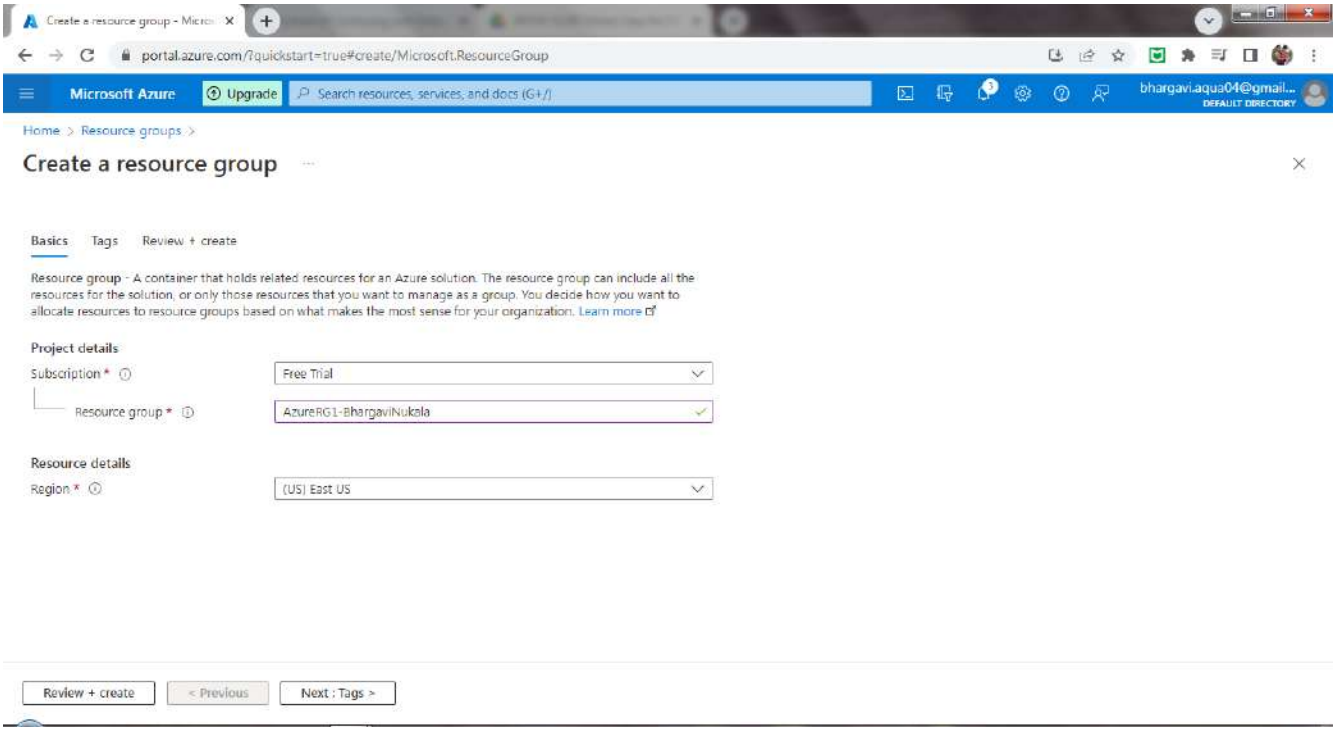
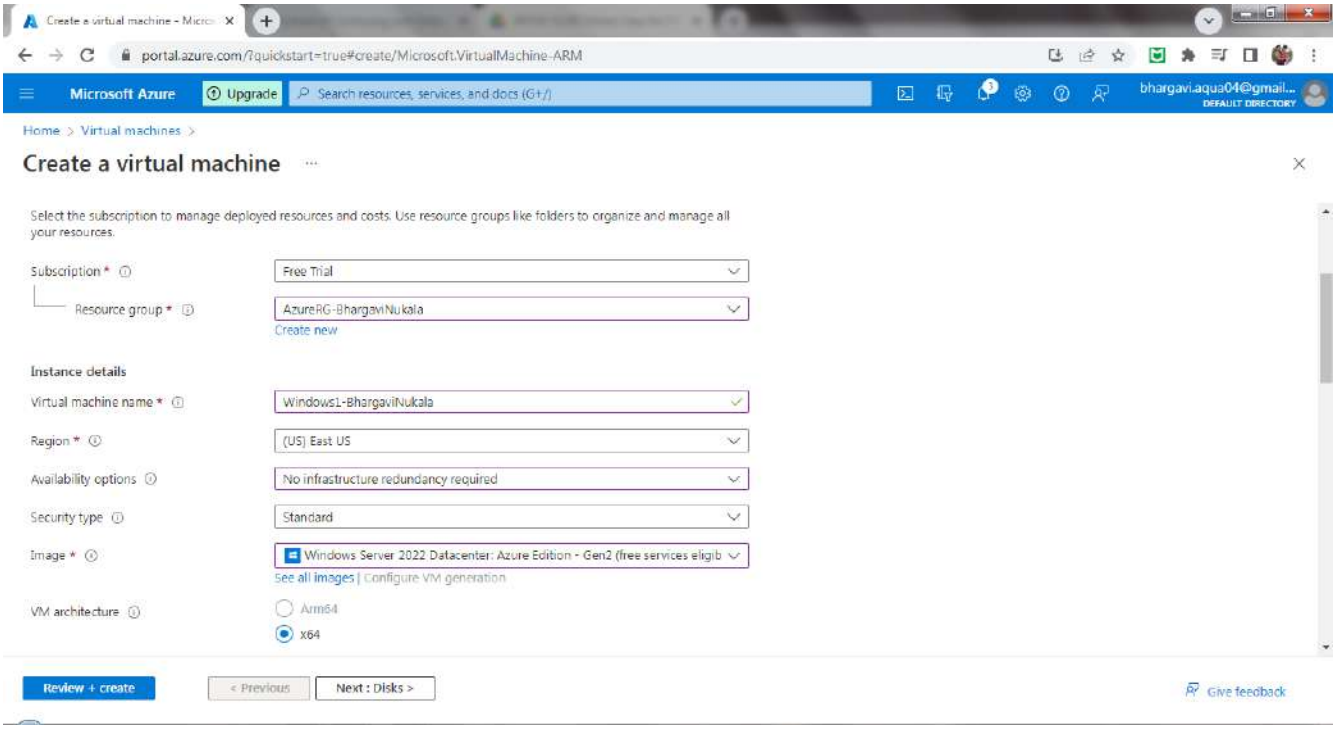
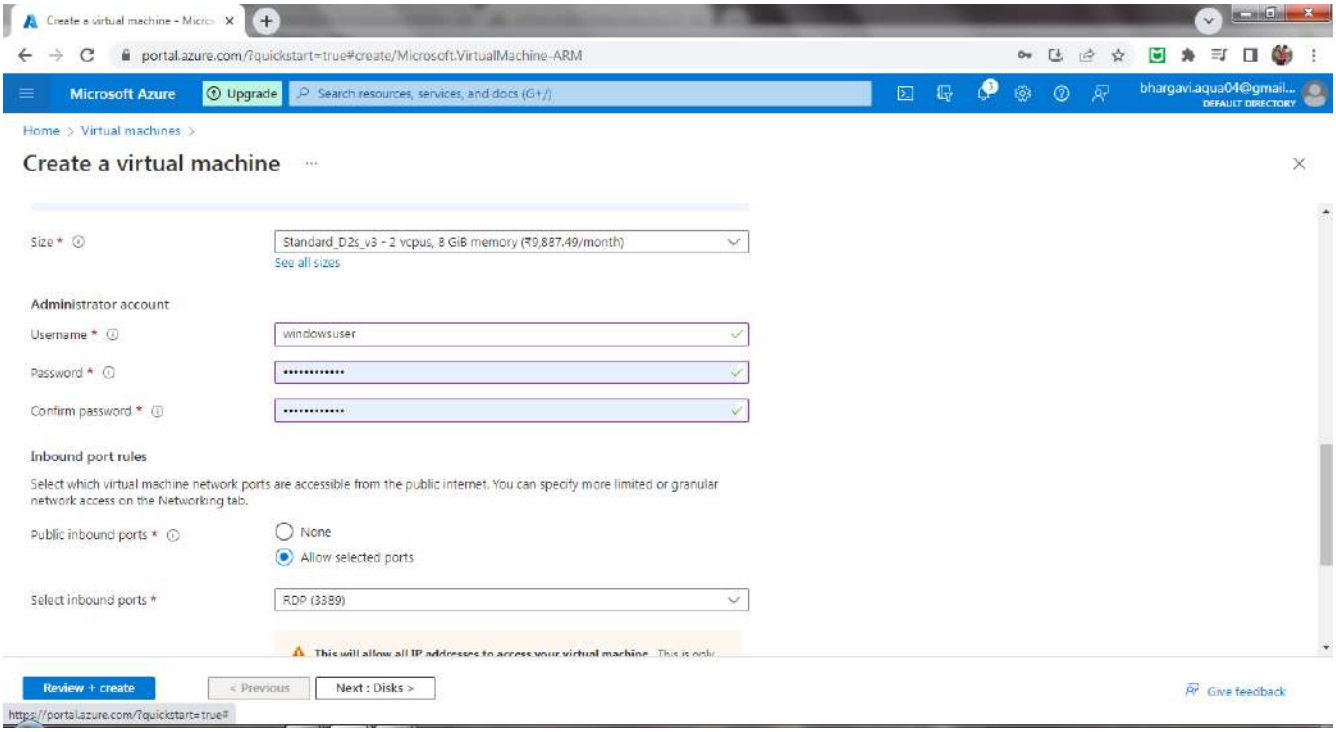
Resource Group: Create a resource group to associate for all resources to be created in Azure



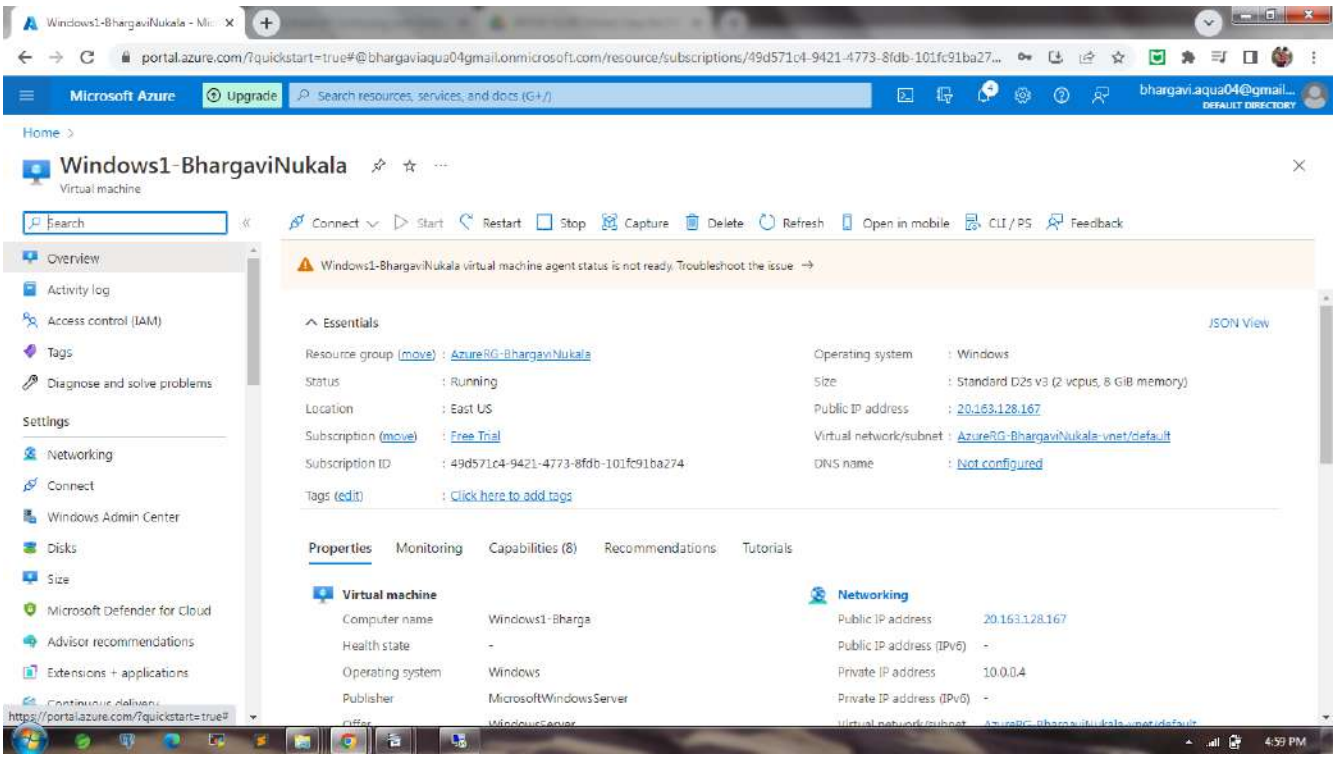
Create a Windows Virtual Machine under above created the ResourceGroup.



We should mention the username and password to connect to VM while creation only.

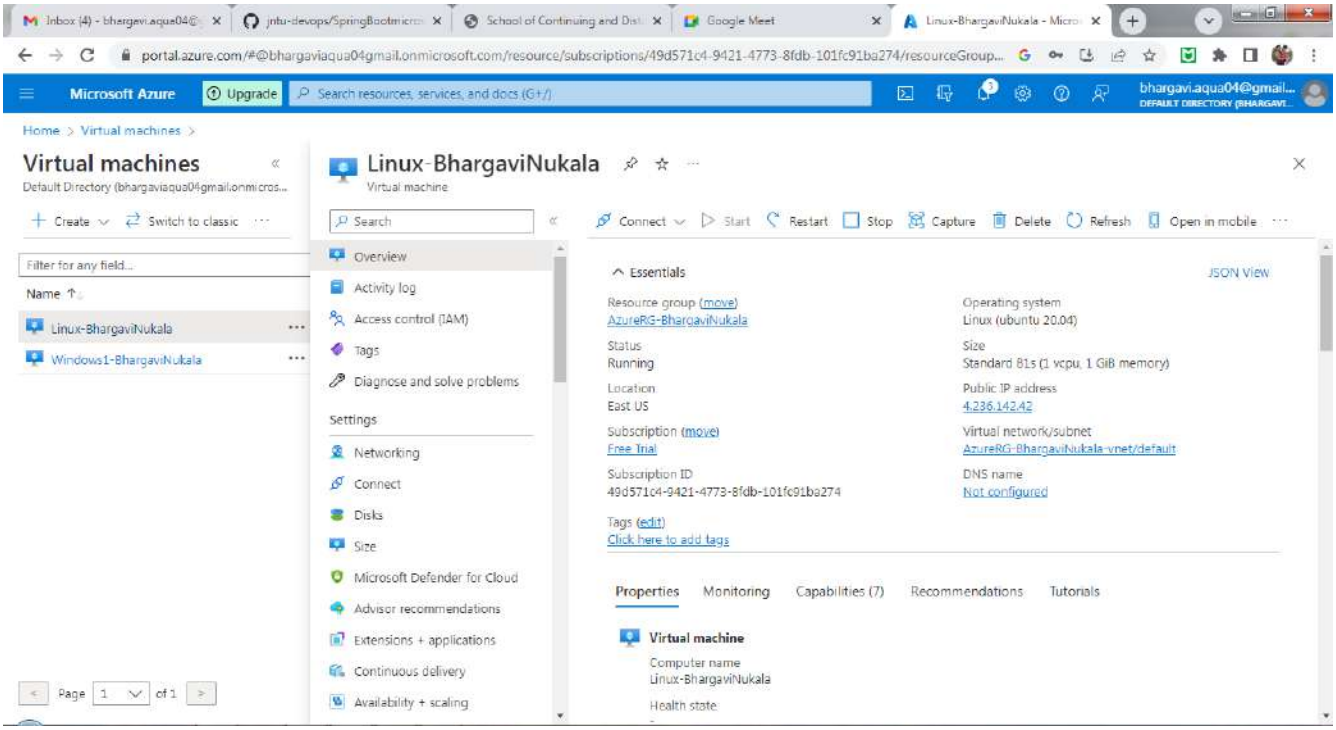


In Networking tab, check the "Delete IP address when VM is deleted" check box and see that Public IP is set.
Virtual Machine is created.

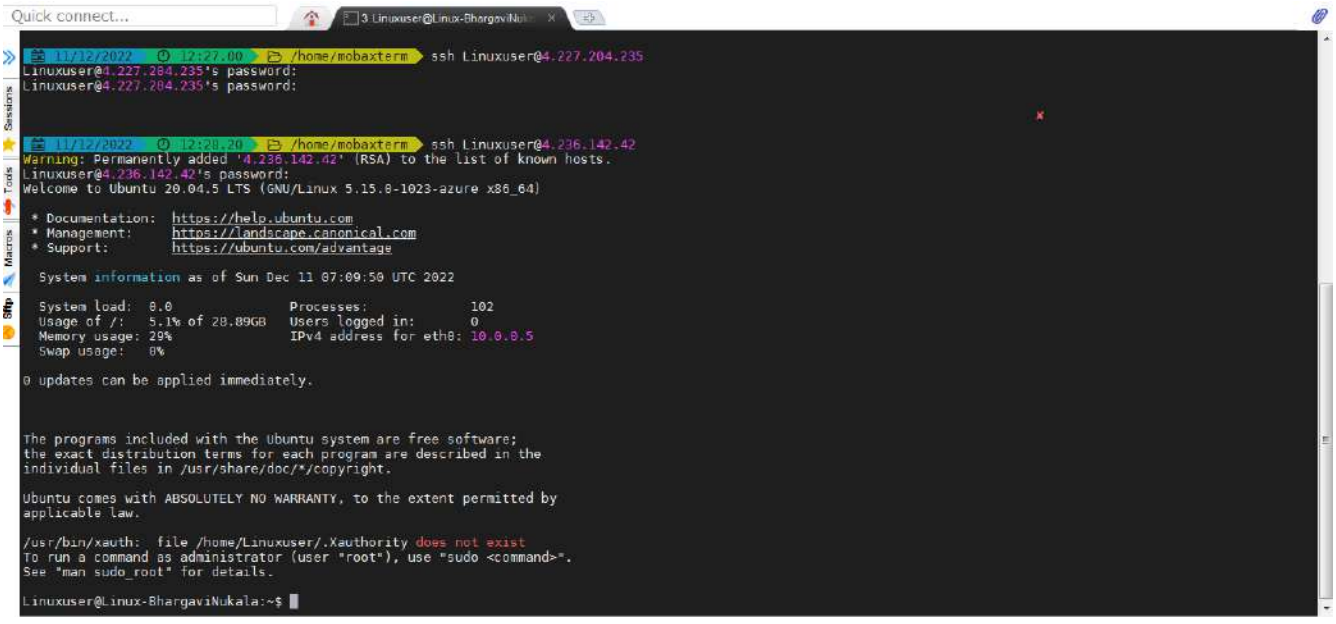


We connect to Windows machine using Remote Desktop Connection.

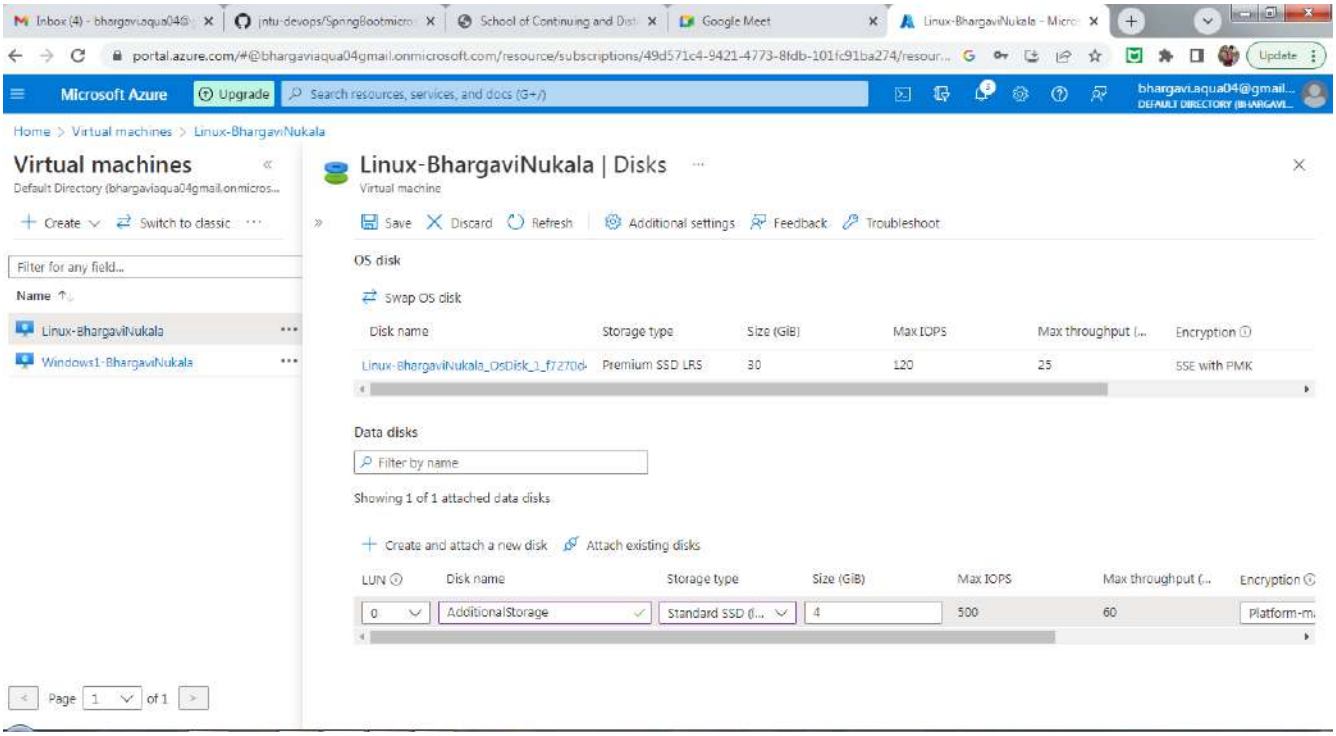
Creation of a Linux Virtual Machine



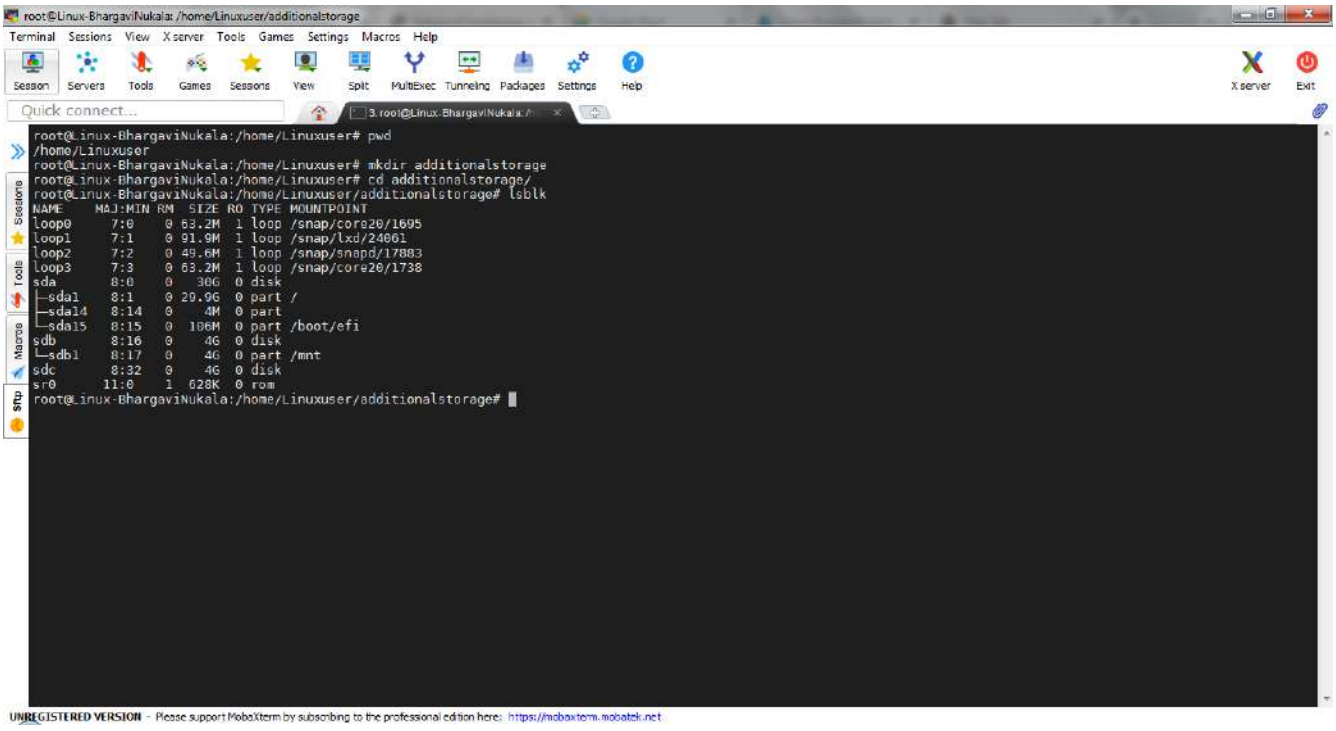
Connect to Linux machine using “ssh username@publicIP”



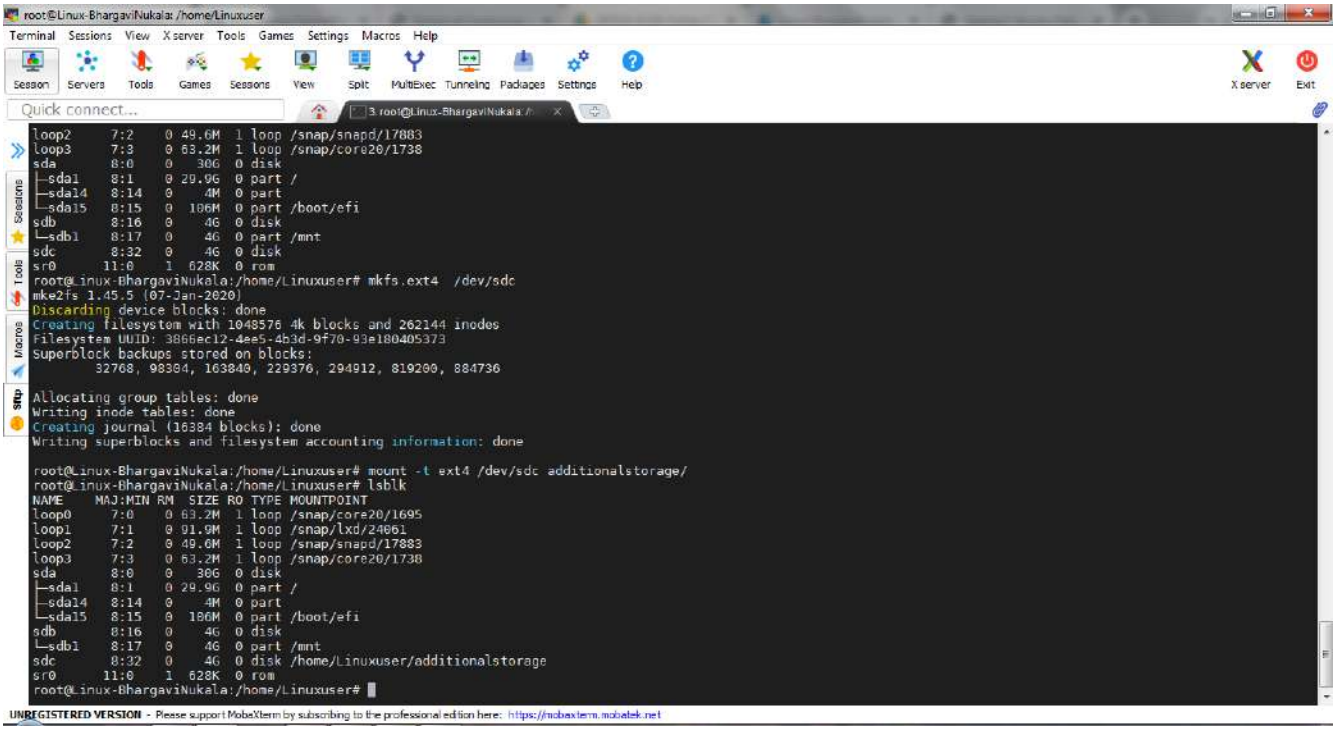
Creating extra storage disks and mount a filesystem
In VM, click on Disks, and create a additional disk. Attach it to the VM and save.



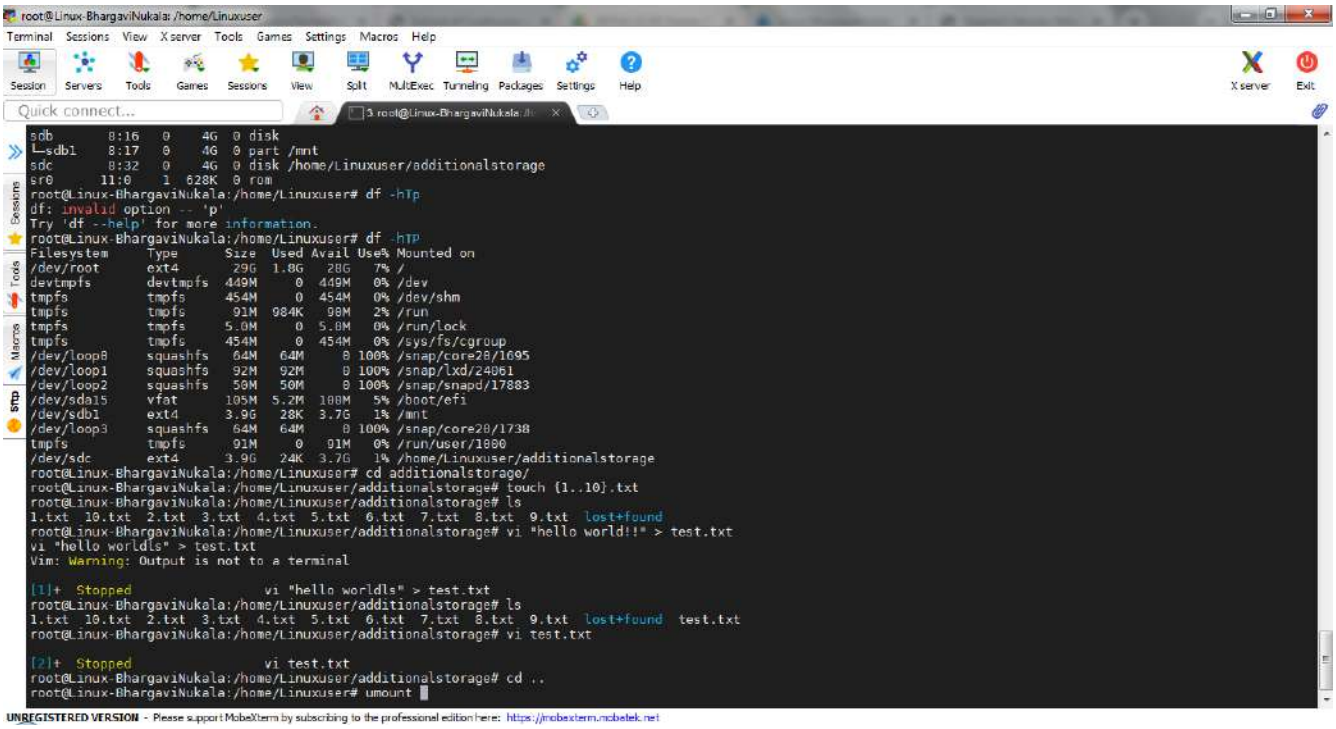
Additional Storage “sdc” avaiable on Linux VM and shows not mounted



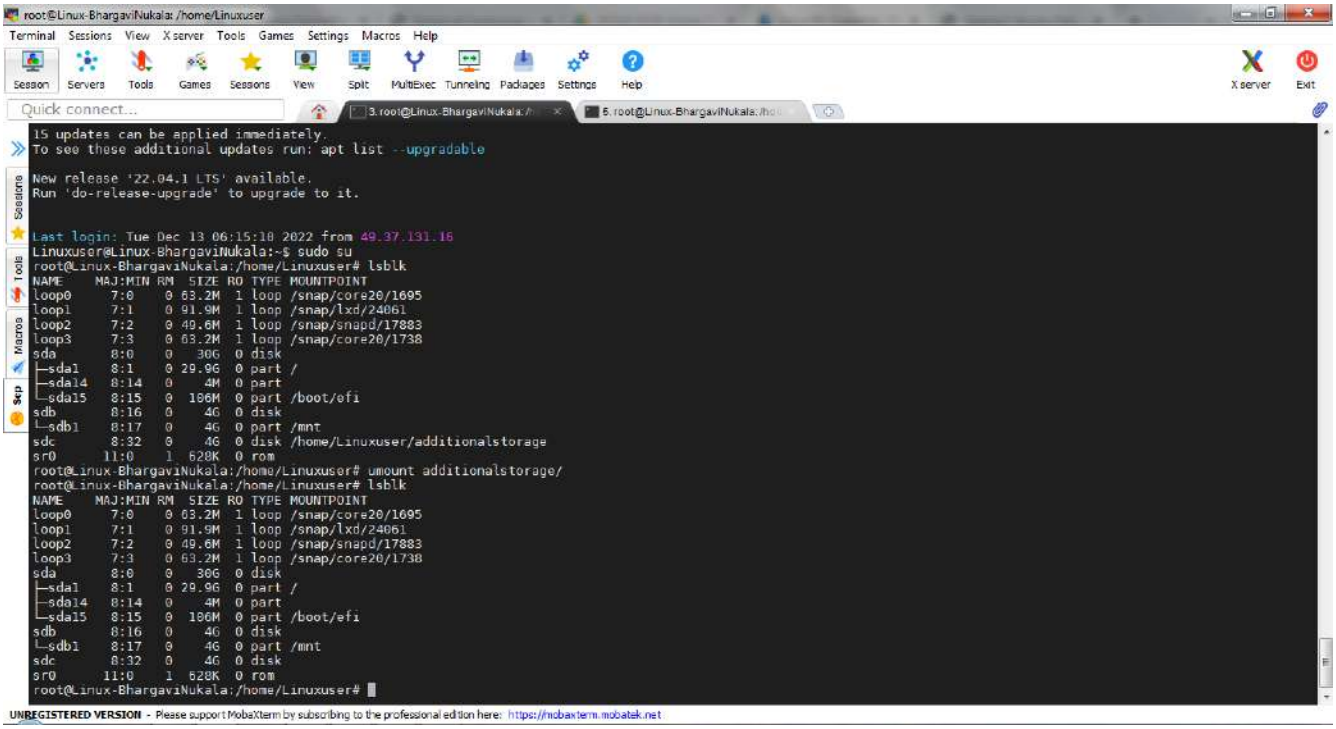
Now mount the storage using commands
$mkdir additionalstorage
$mkfs .ext4 /dev/sdc
$mount -t ext4/dev/sdc additionalstorage/



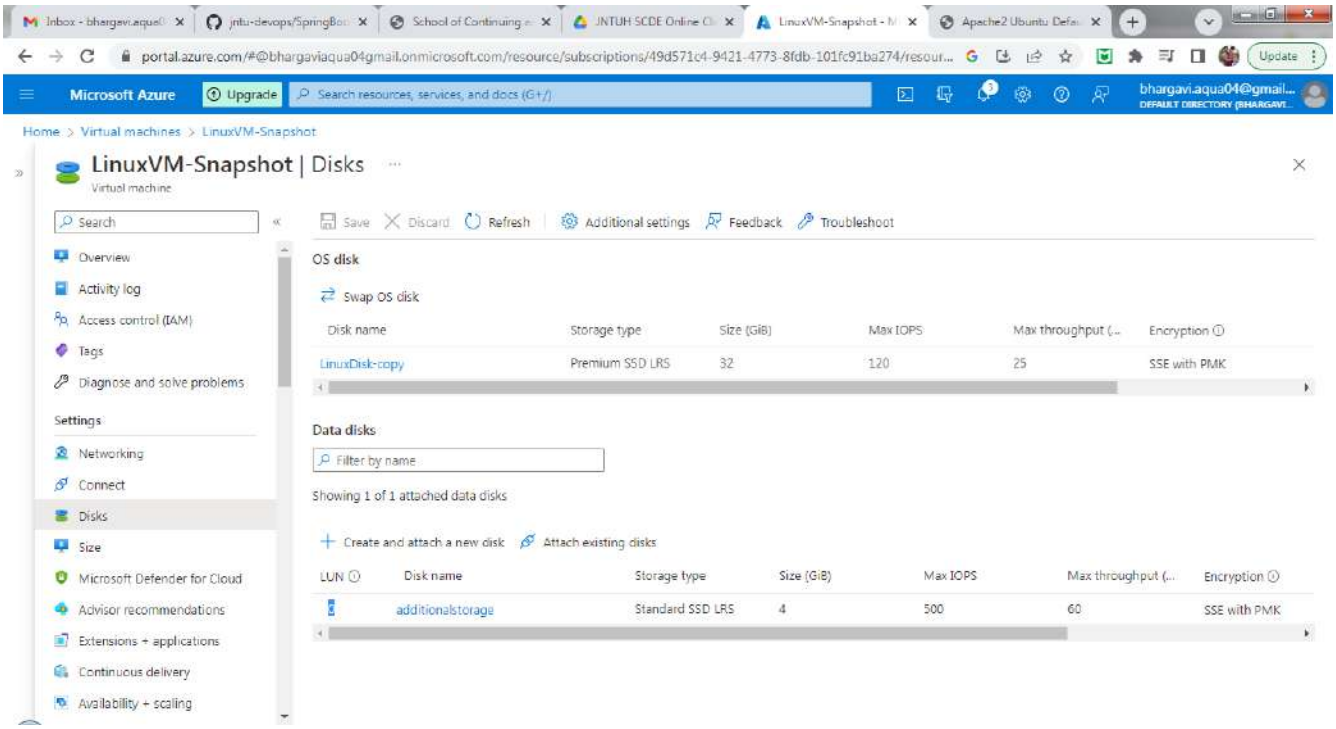
Create 10 files in the disk using the command
$touch {1..10}.txt



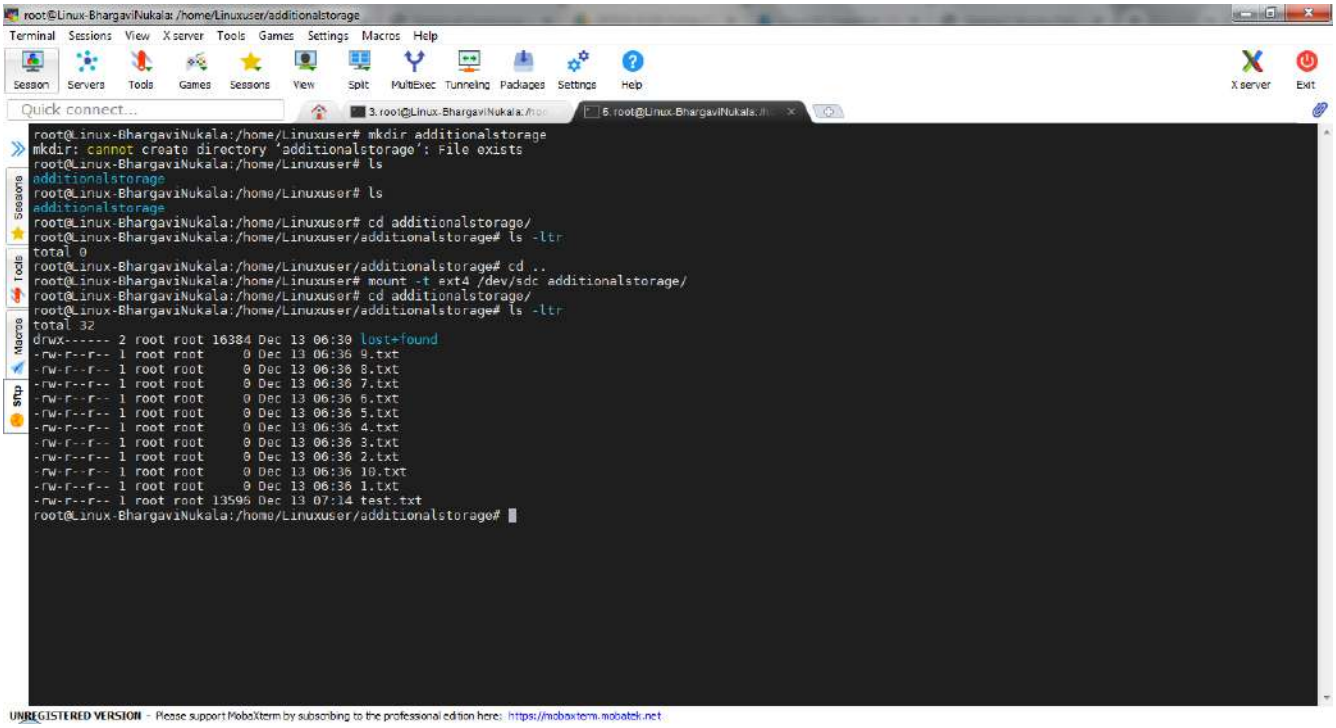
Unmount the additional Storage using the command
$umount additionalstorage/



Now attach the same disk to another VM



Connect to the VM and check the disk is already mounted and the additionalStorage is already available on the disk. Also the 10 files which we created in previous VM are also available in this VM.

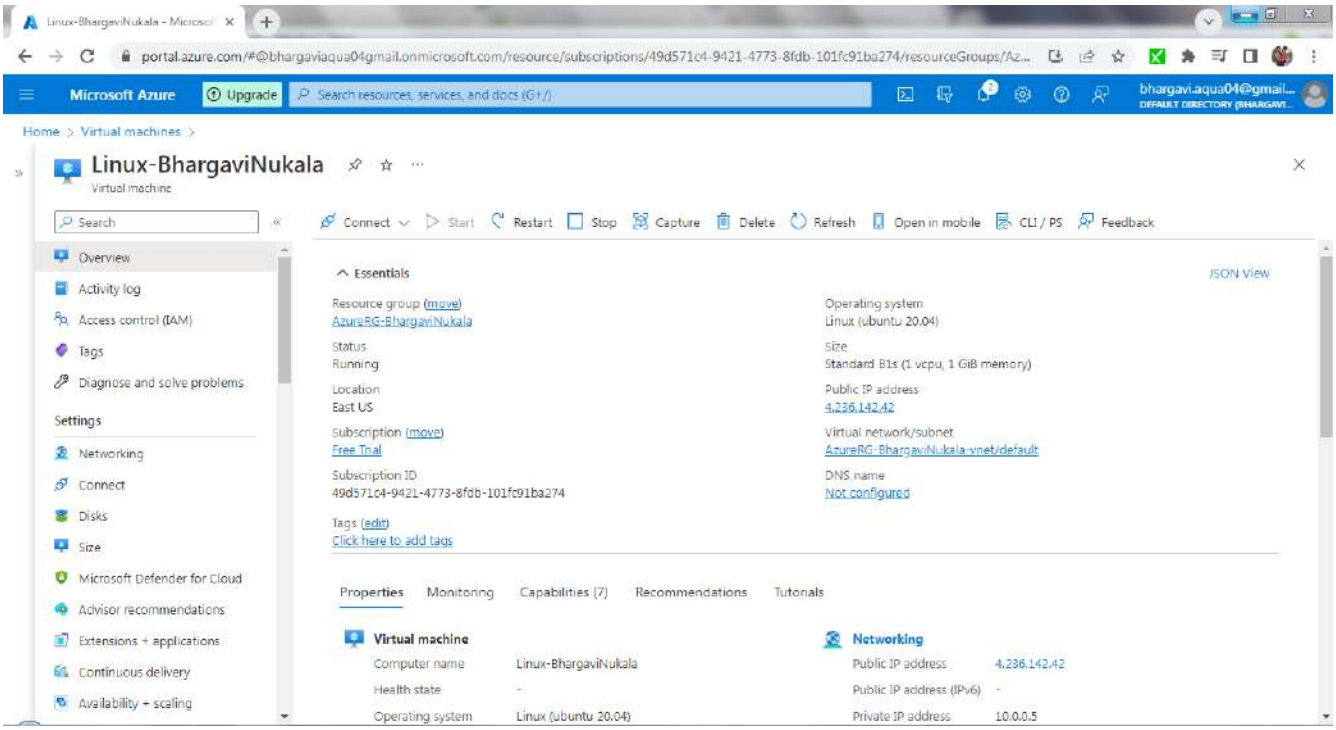


Snapshot:

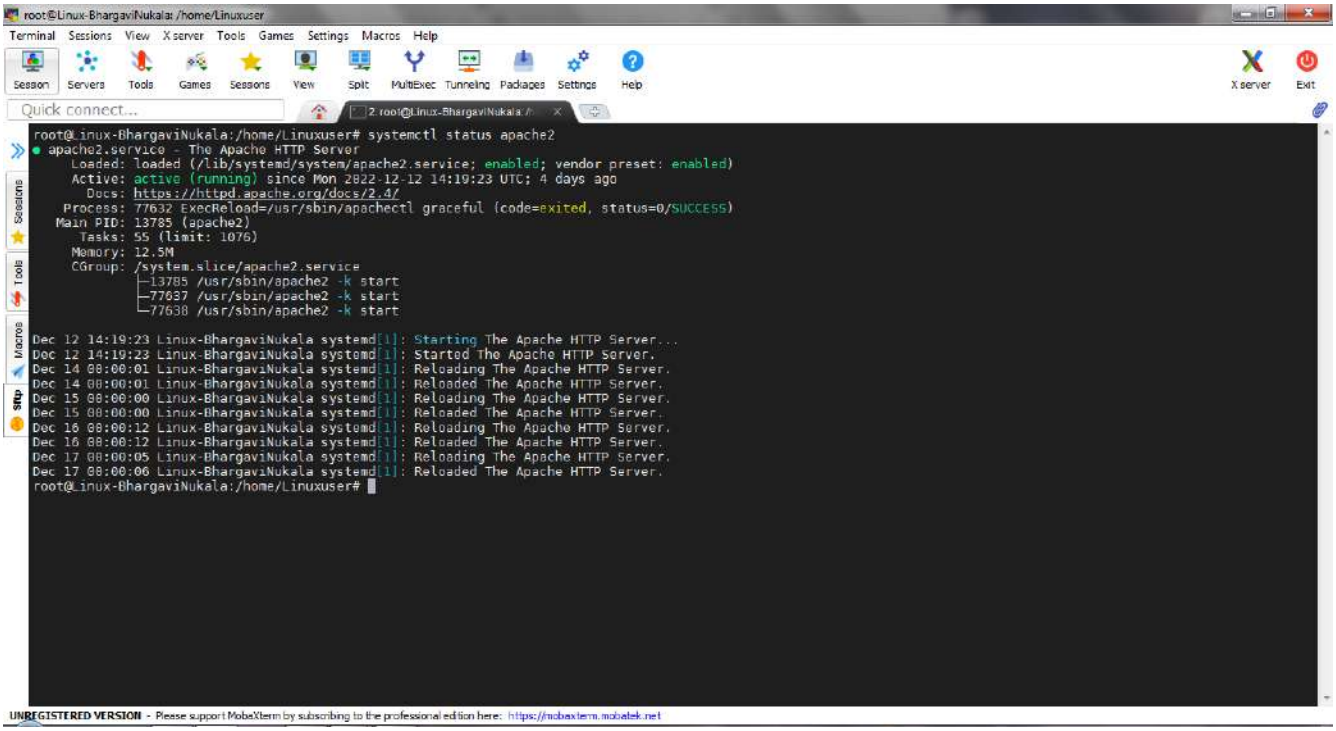
Create a Linux VM and install apache2.

$apt update

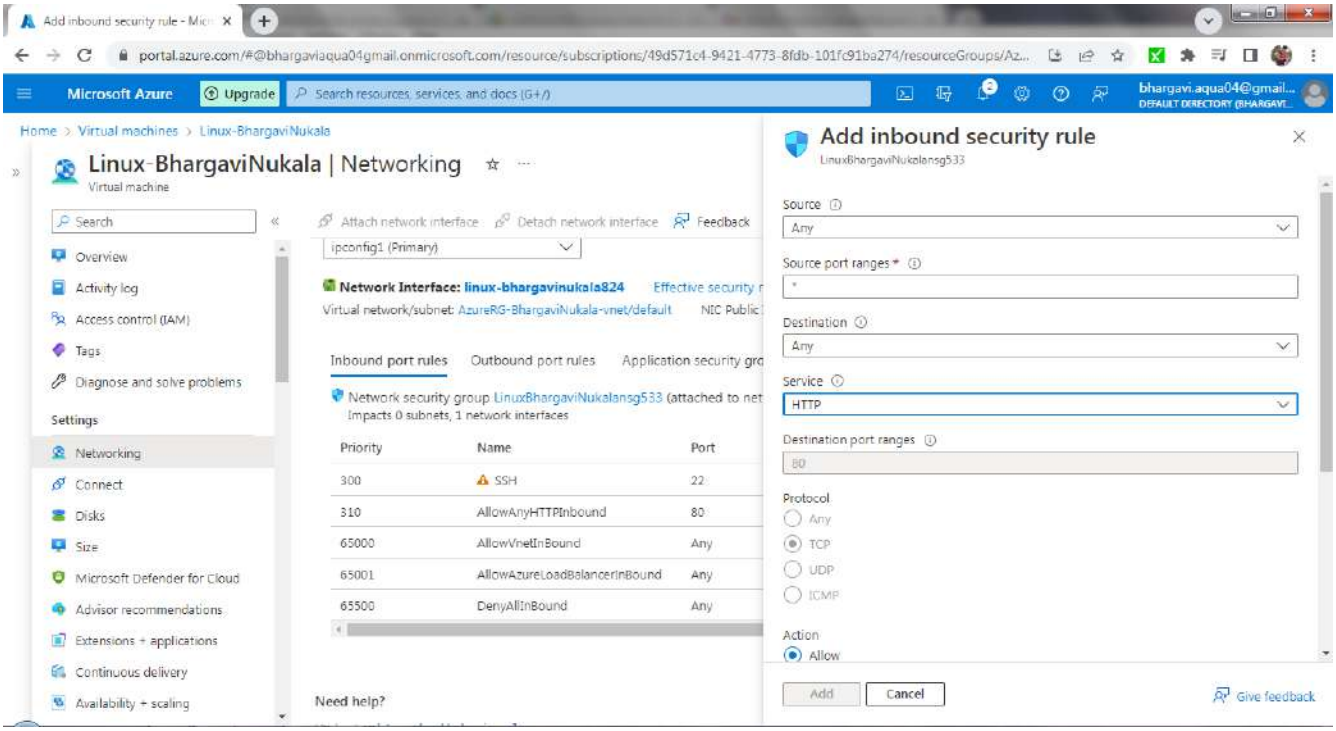
$apt install apache2



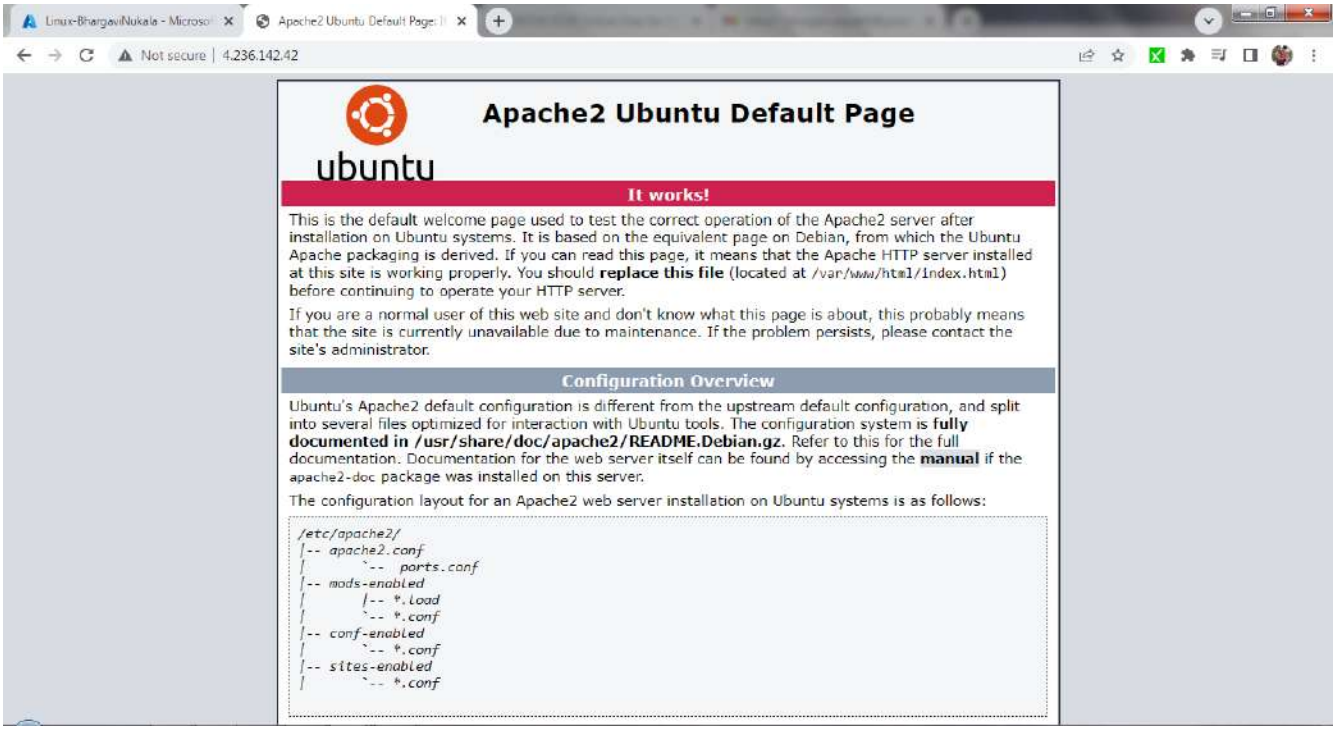
Check the status of apache2 running
$systemctl status apache2



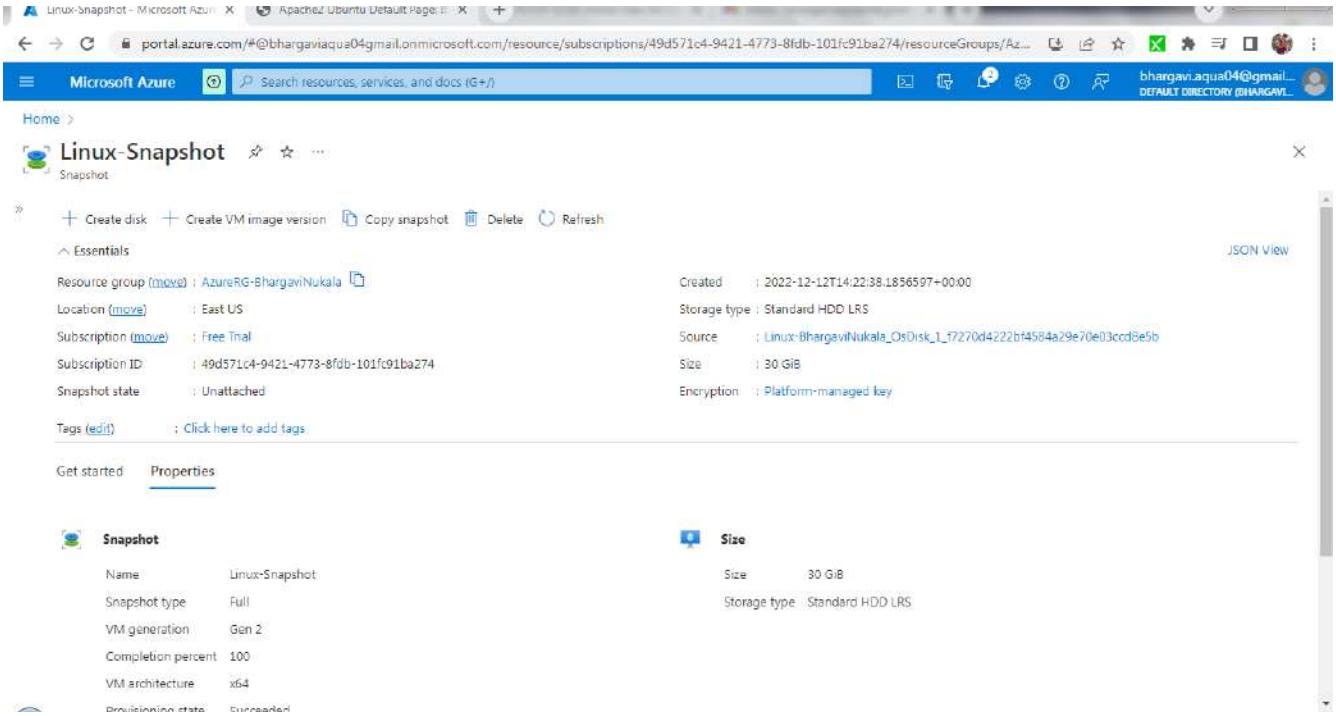
Add the Inbound rule for HTTP-80 port in Networking tab of VM



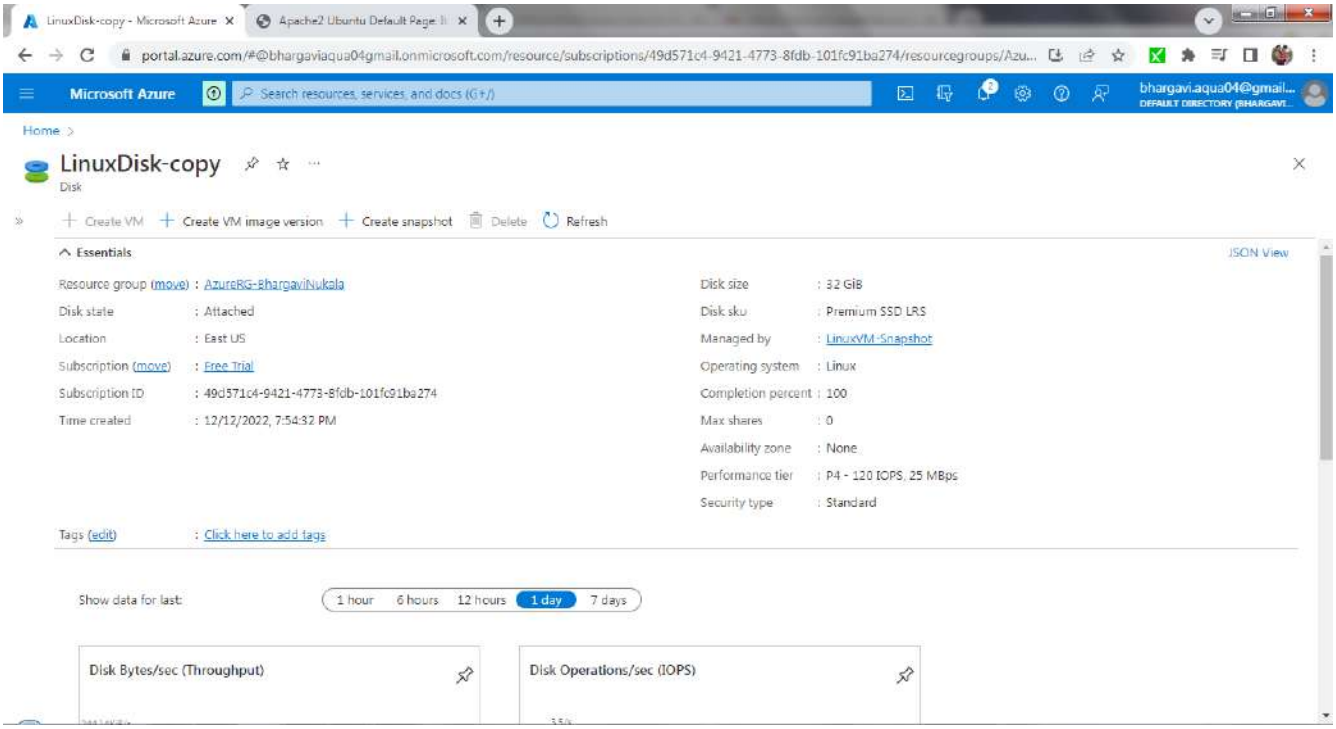
Access the webpage from browser using public IP of VM



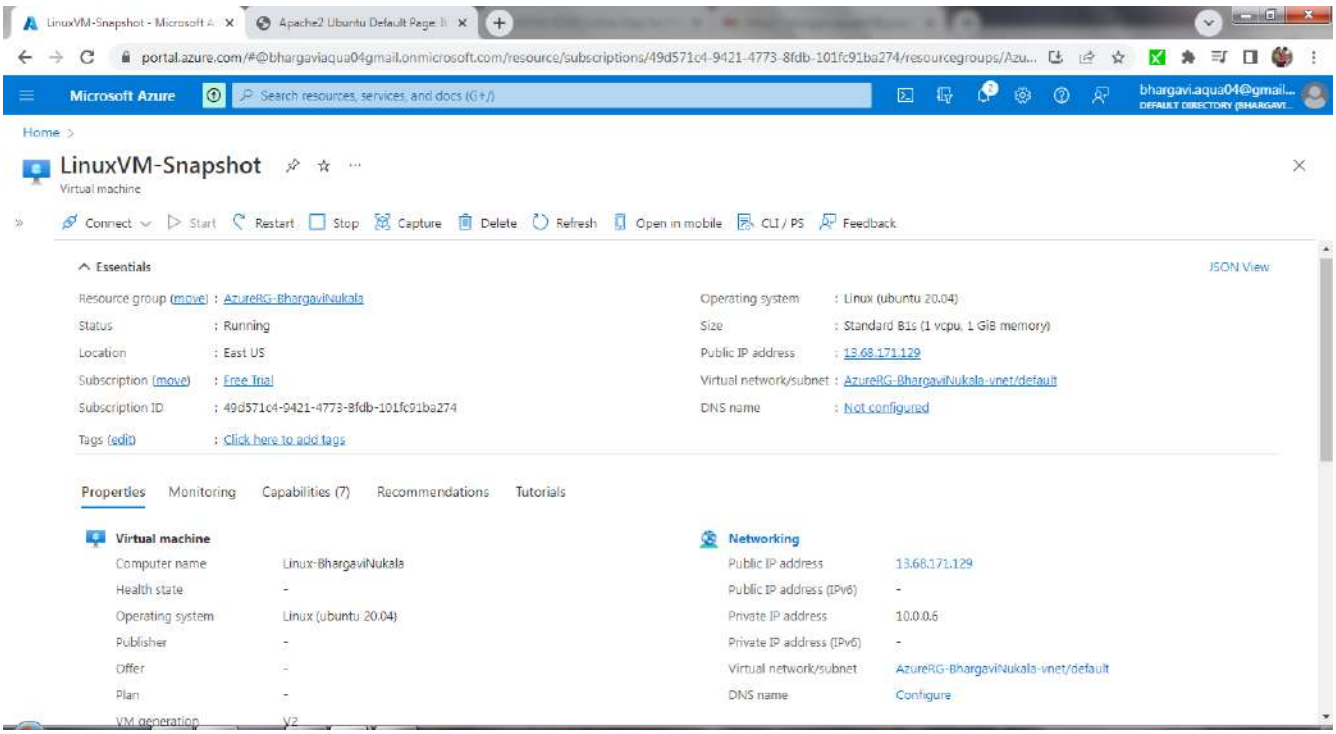
Go to VM->Disks->Create Snapshot of the disk



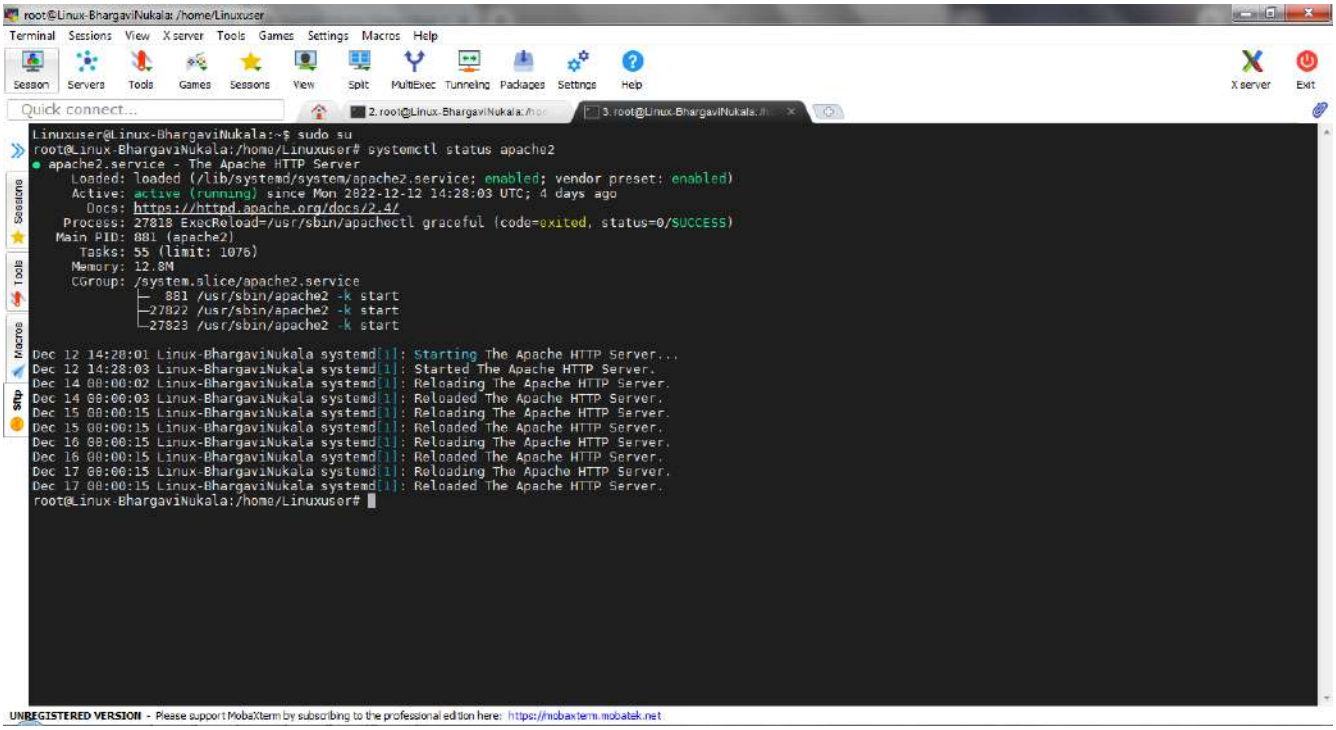
Now create a disk from the created snapshot
Goto Created Snapshot->Create Disk



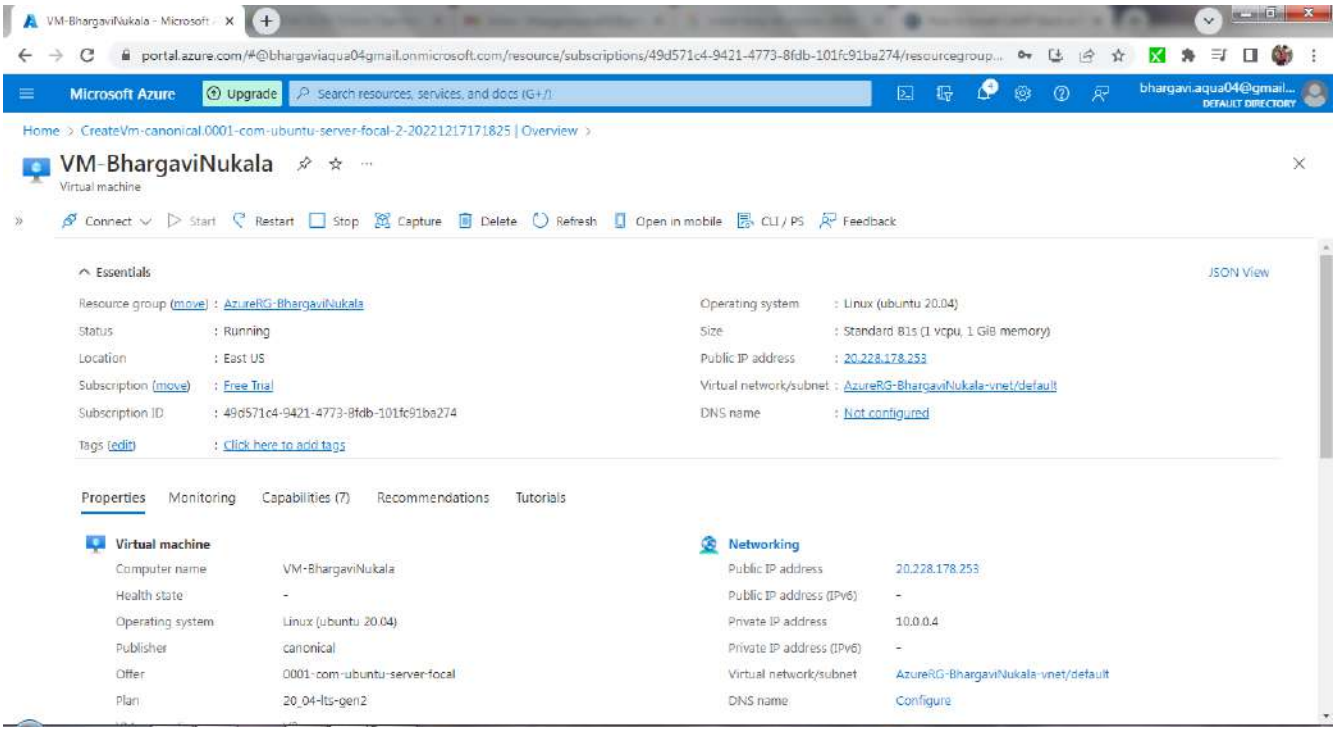
From the newly created disk-> create a VM
Select the above created disk in OS dropdown box



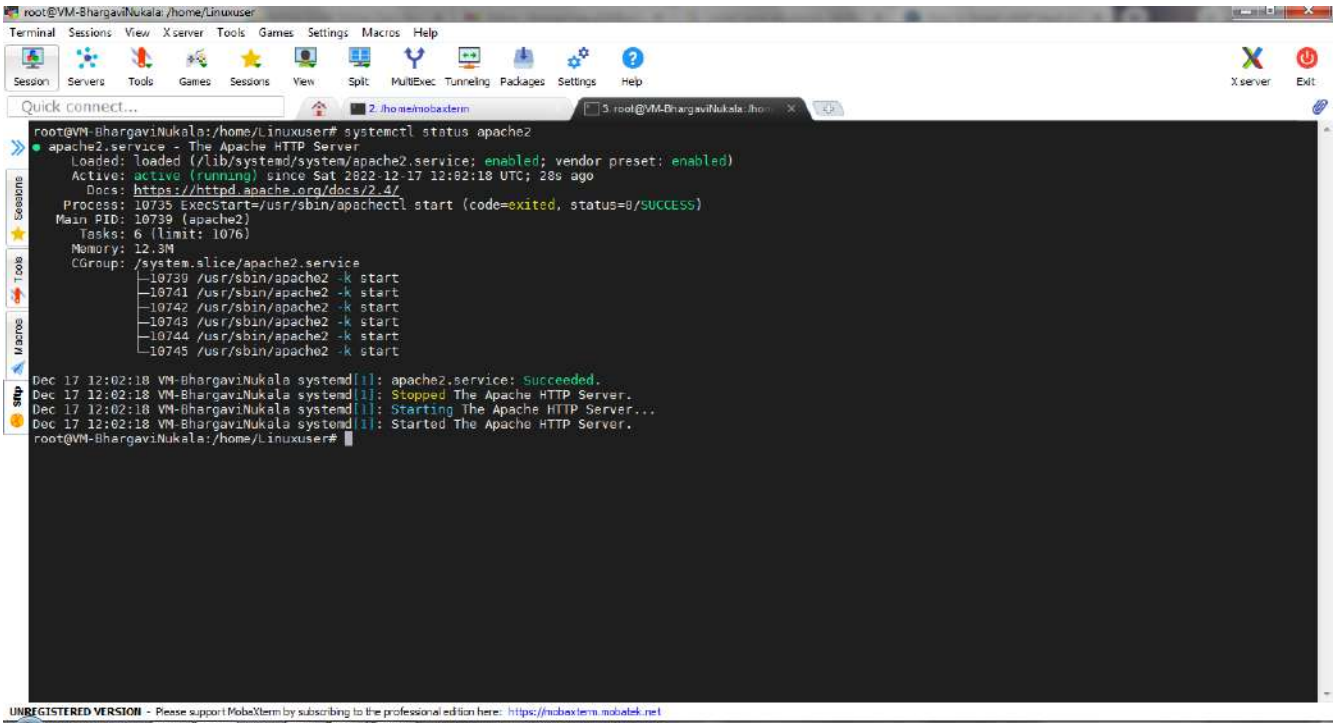
After the new VM is created from the snapshot, check the status of apache2.
We can see that apache2 is already installed.



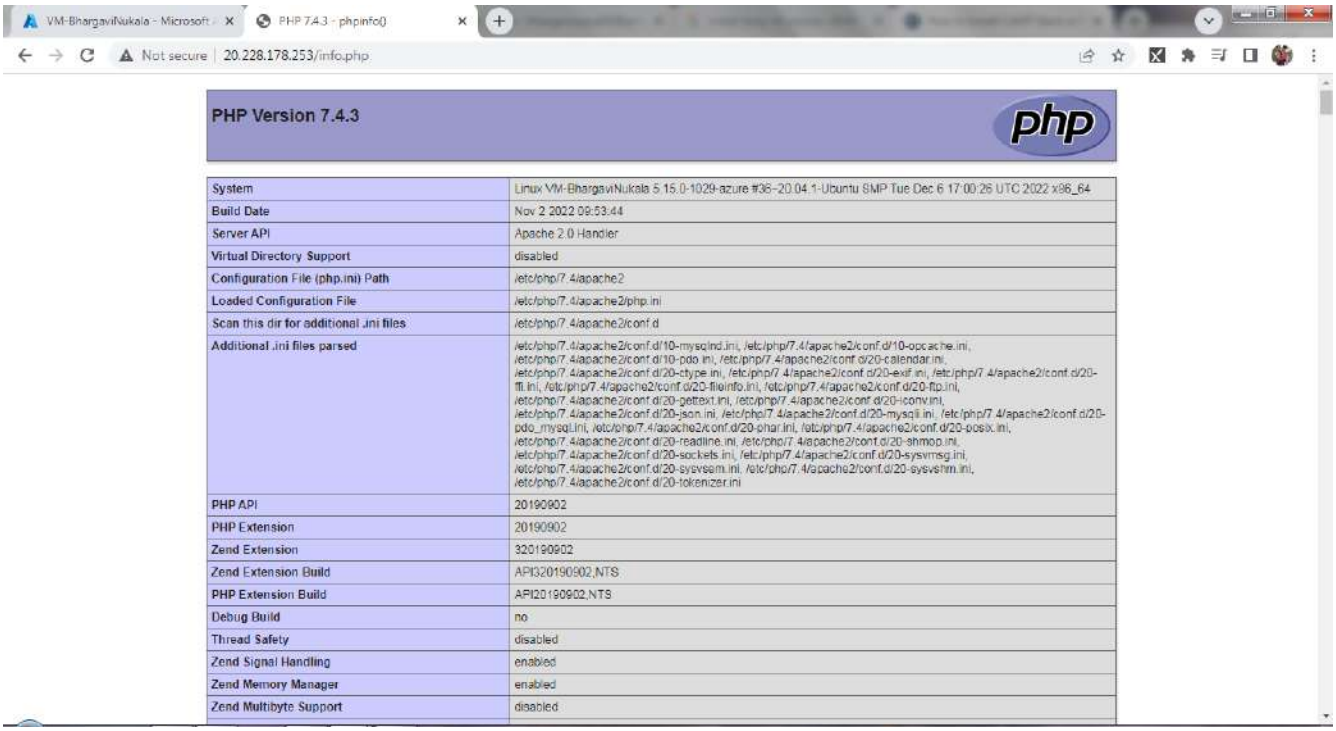
Machine Image:
Create an Ubuntu machine and install apache2 and php as mentioned in the link https://www.tecmint.com/install-lamp-with-phpmyadmin-in-ubuntu-18-04/



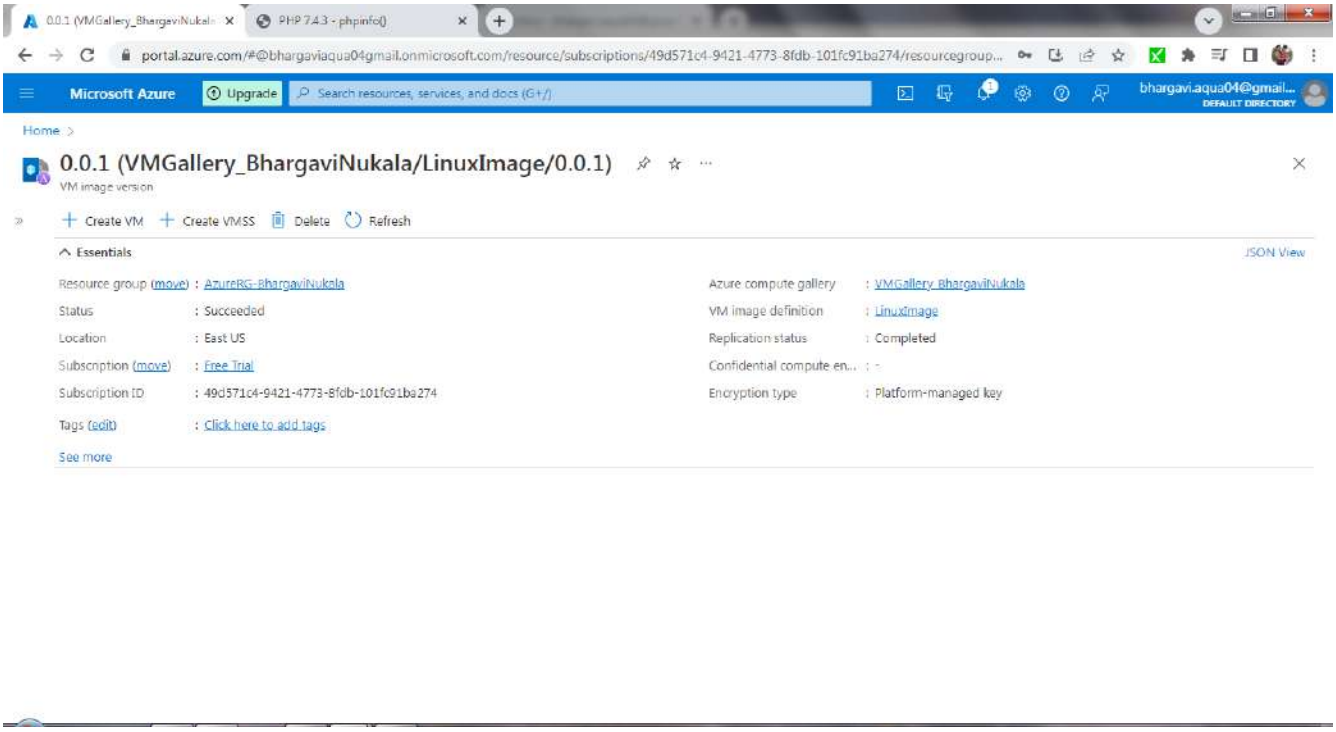
Install apache2 and php



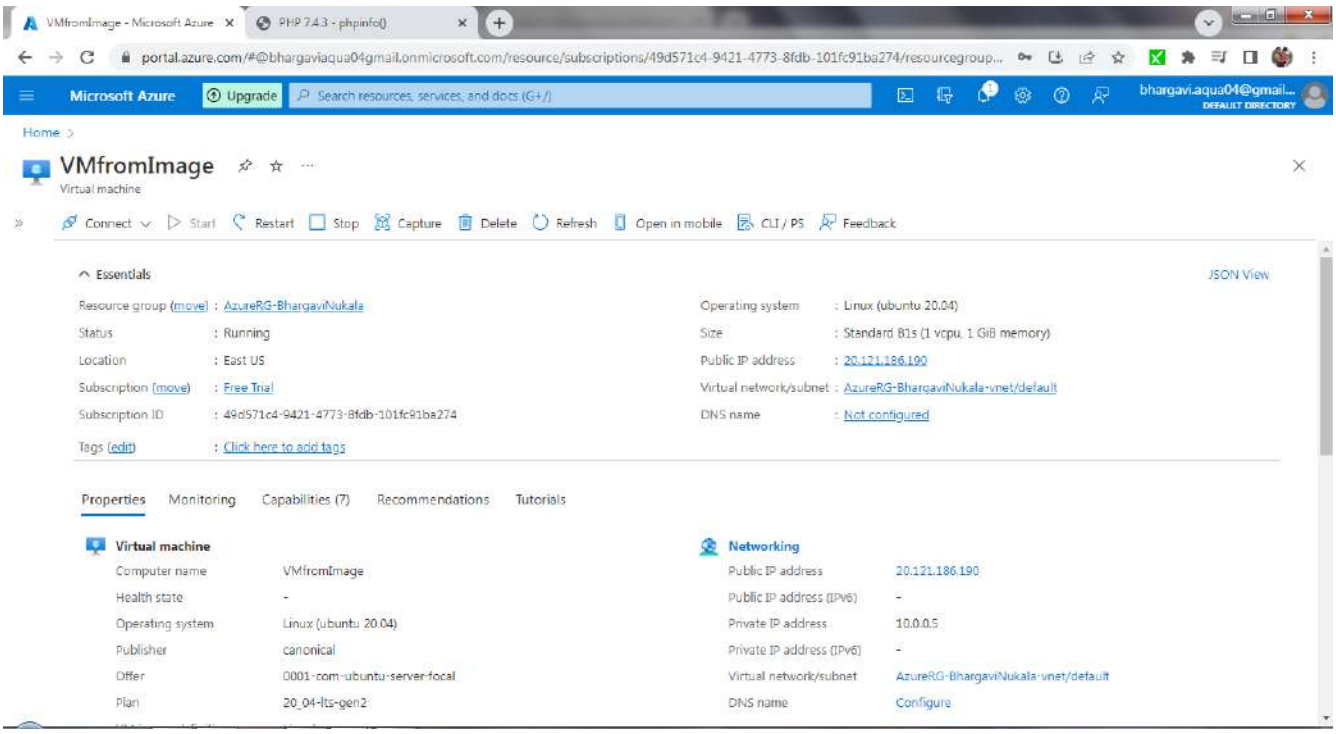
Access the webpage of VM using public IP/info.php



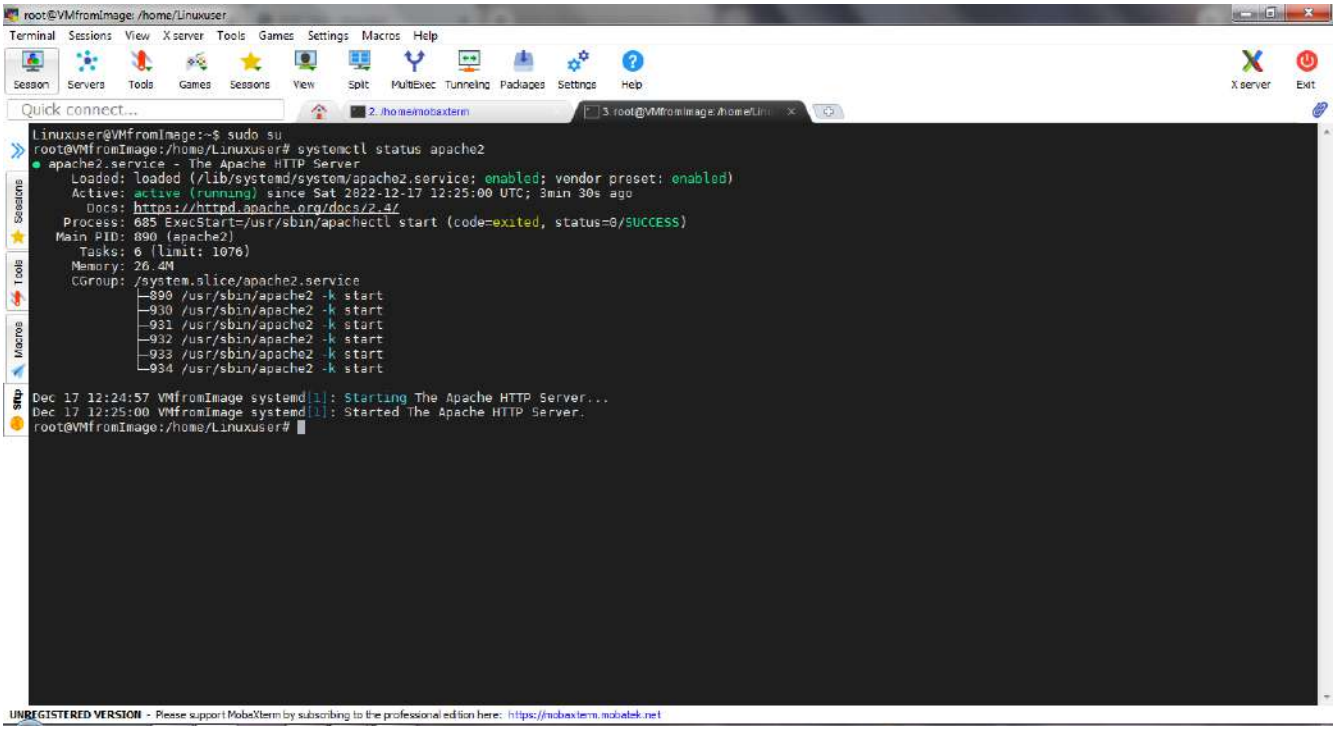
To create an image of the above created VM, goto VM->Capture

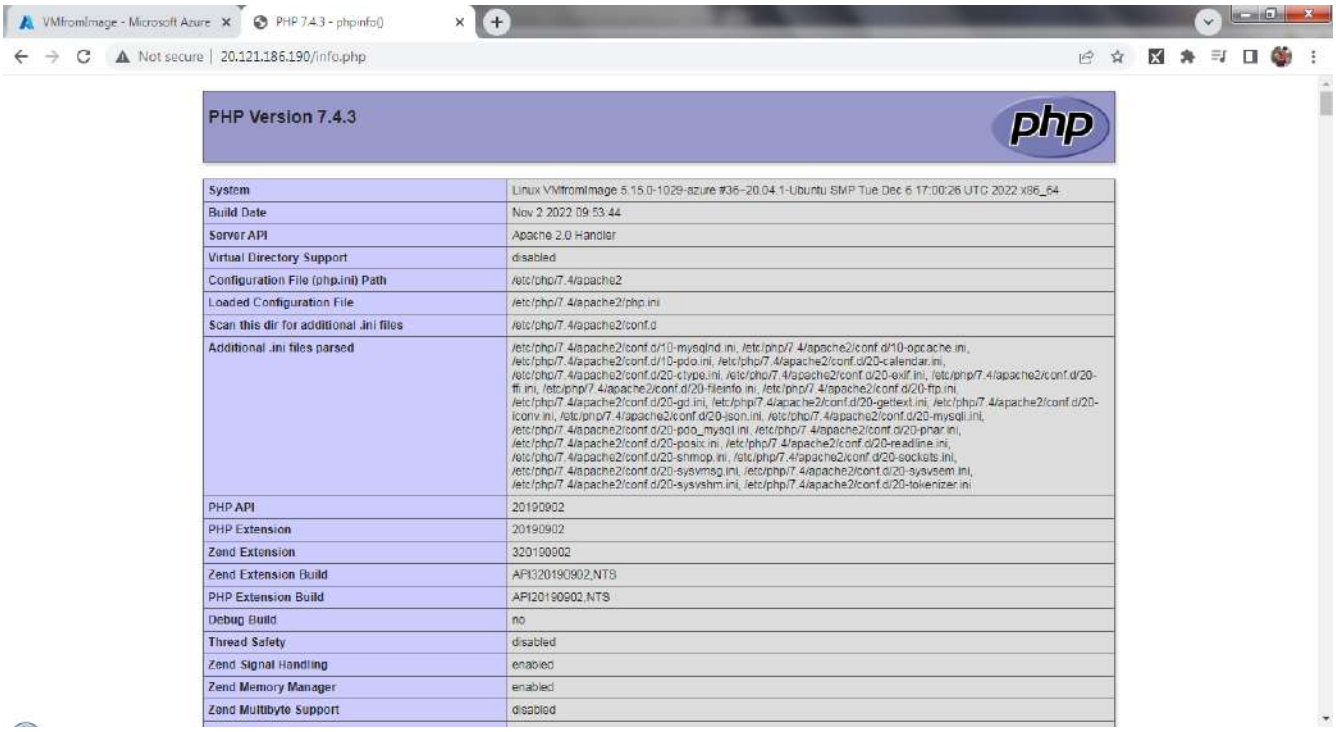


Create a VM from this Image by selecting this image in OS section of VM.



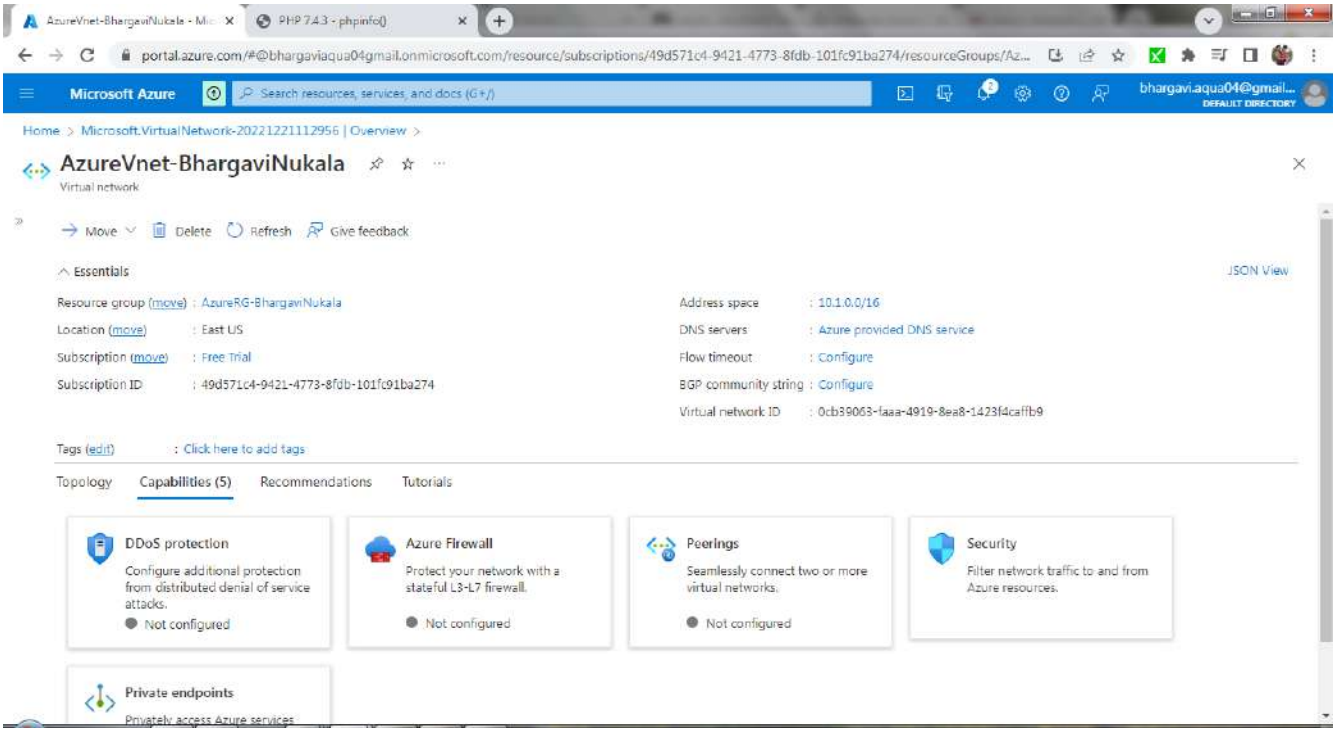
Check that apache2 and php are already installed in this new VM



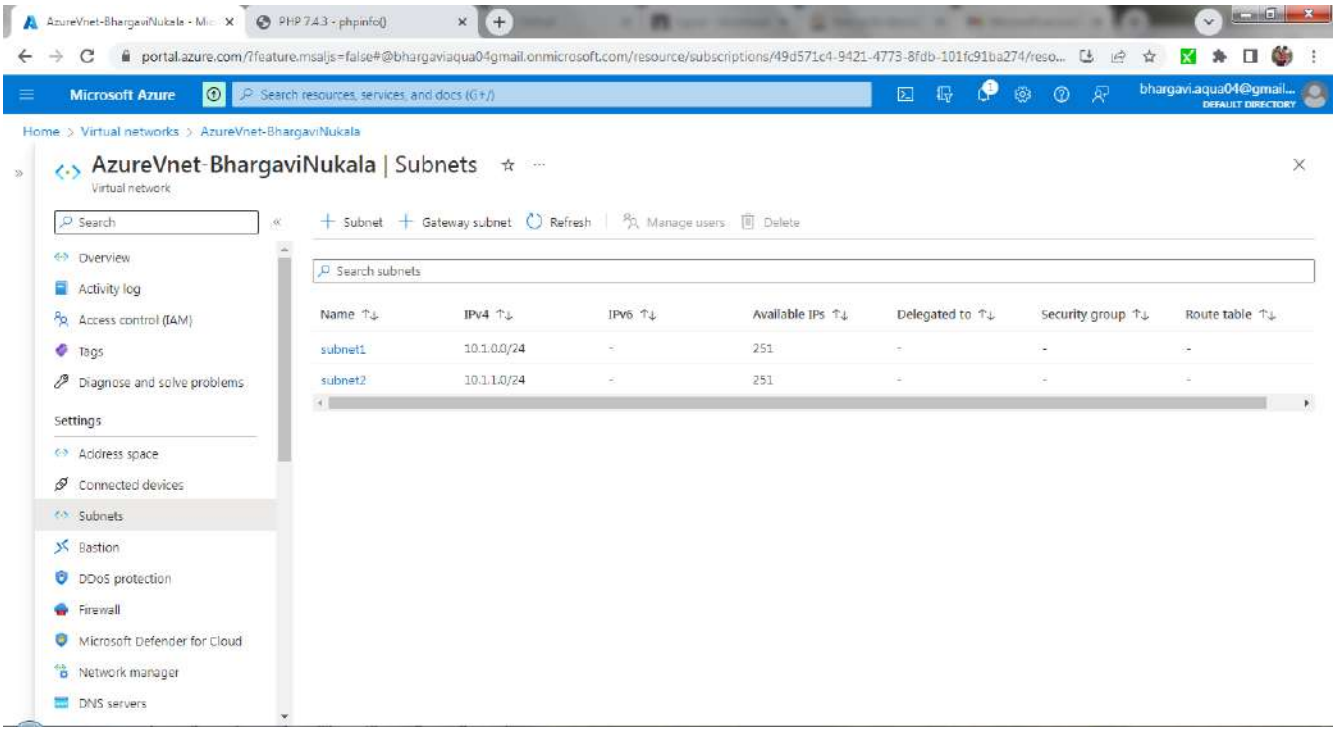


Network Security Group

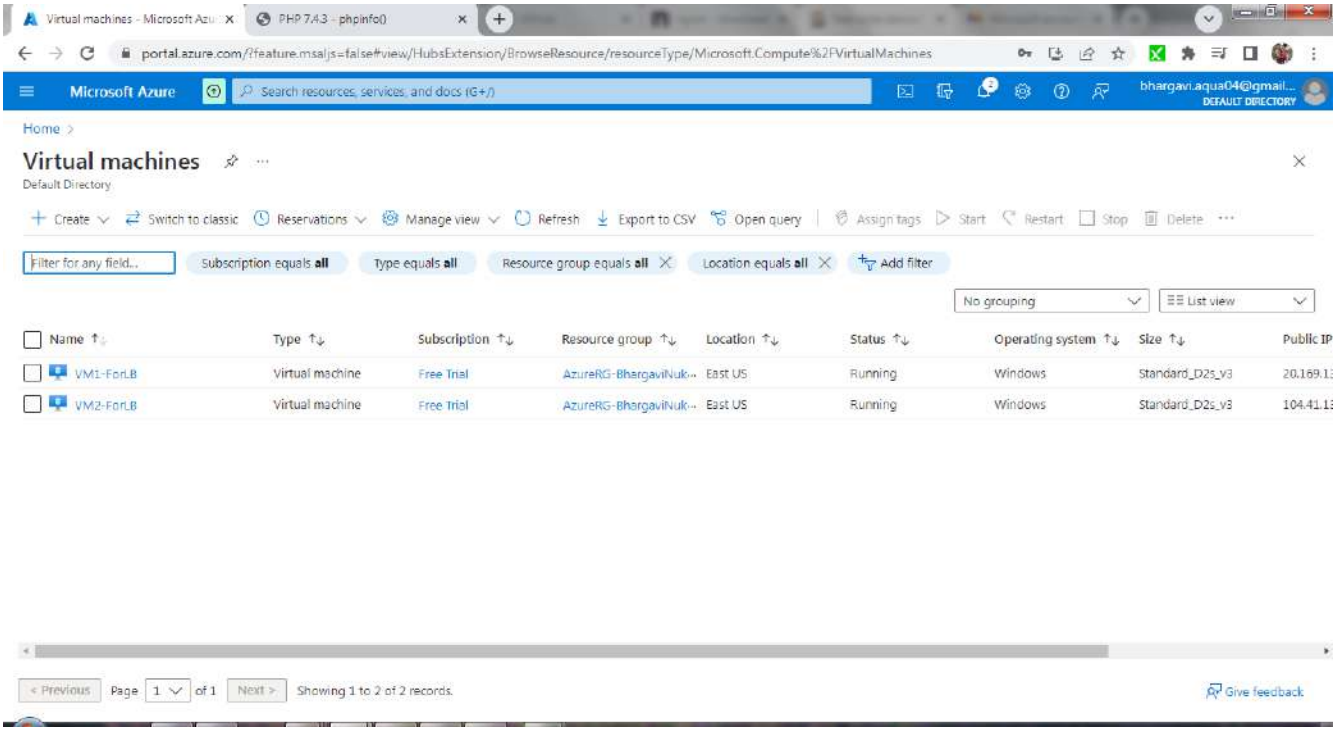
Create a Vnet with two subnets—use Virtual Networks to create a Vnet.



In the Vnet, create 2 subnets



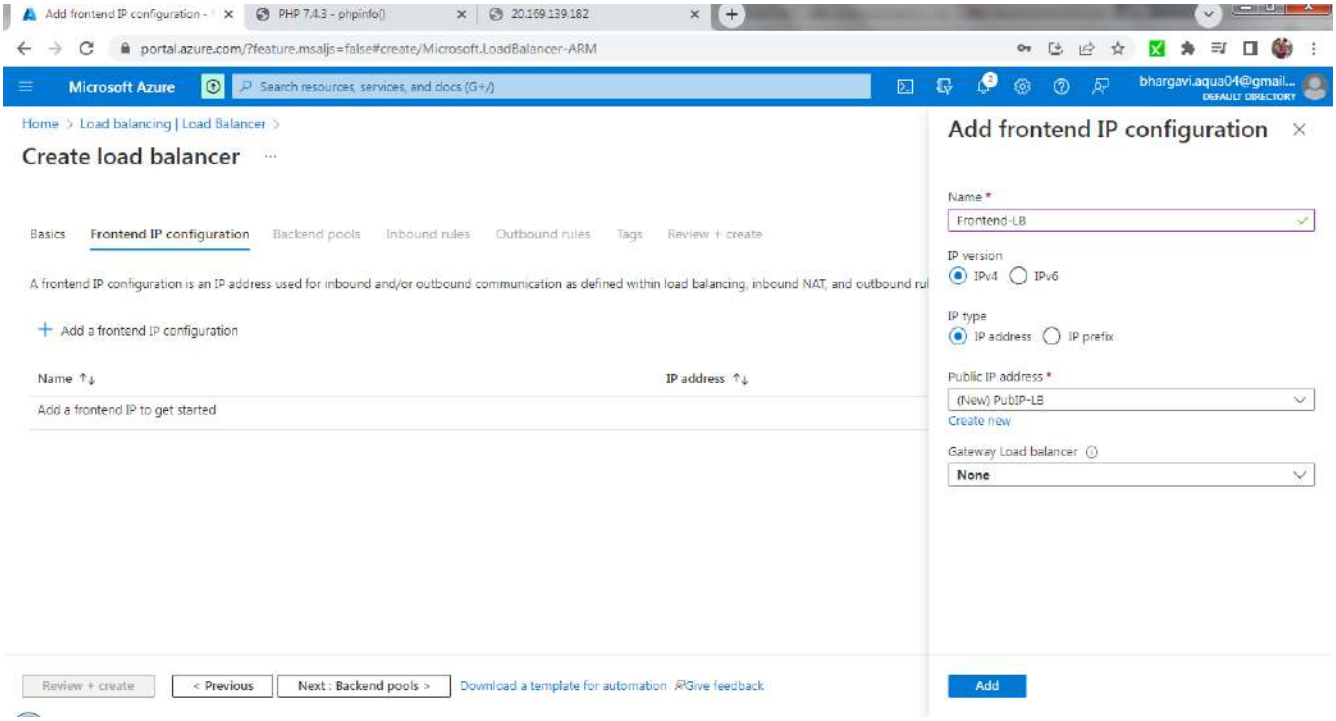
Now create one VM in each subnet.

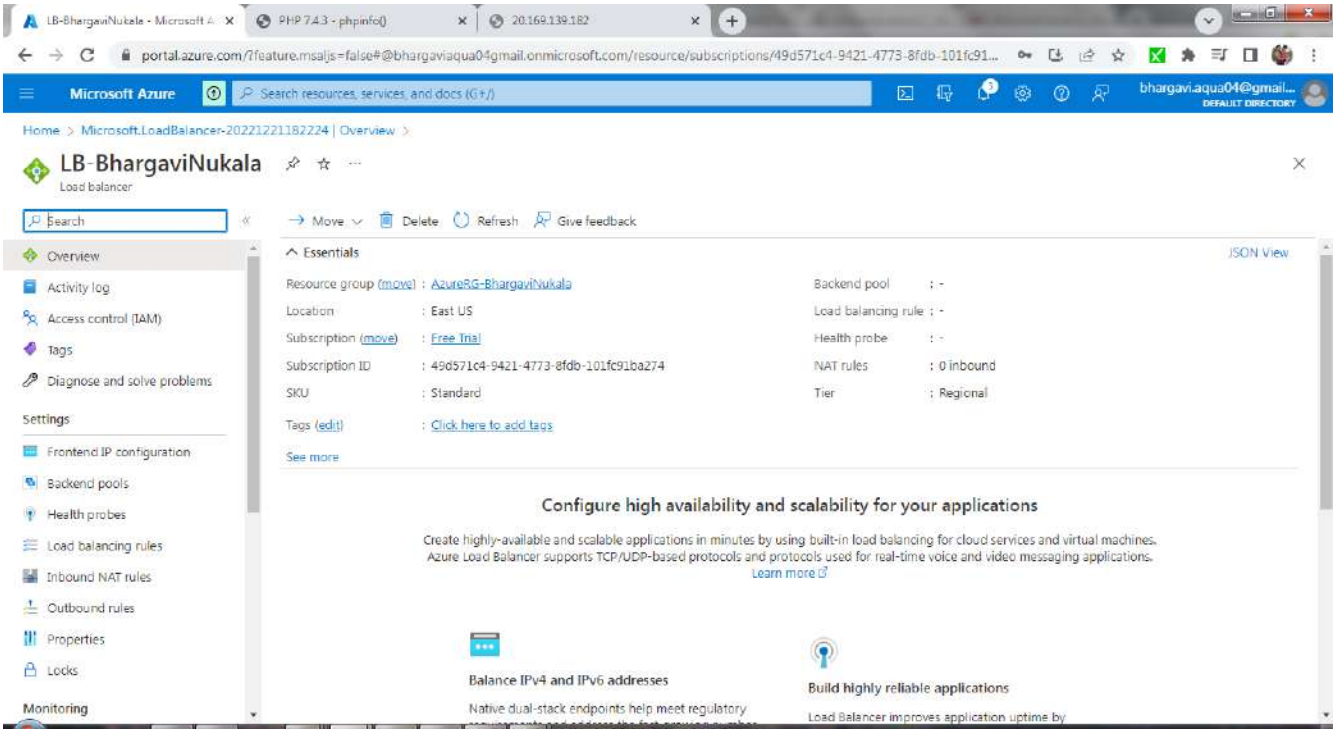


Setup IIS server on both VM s. As we have created Vnet with default Inbound and Outbound rules, with default rules, only Inbound traffic from same Vnet are allowed and from outside is not allowed.

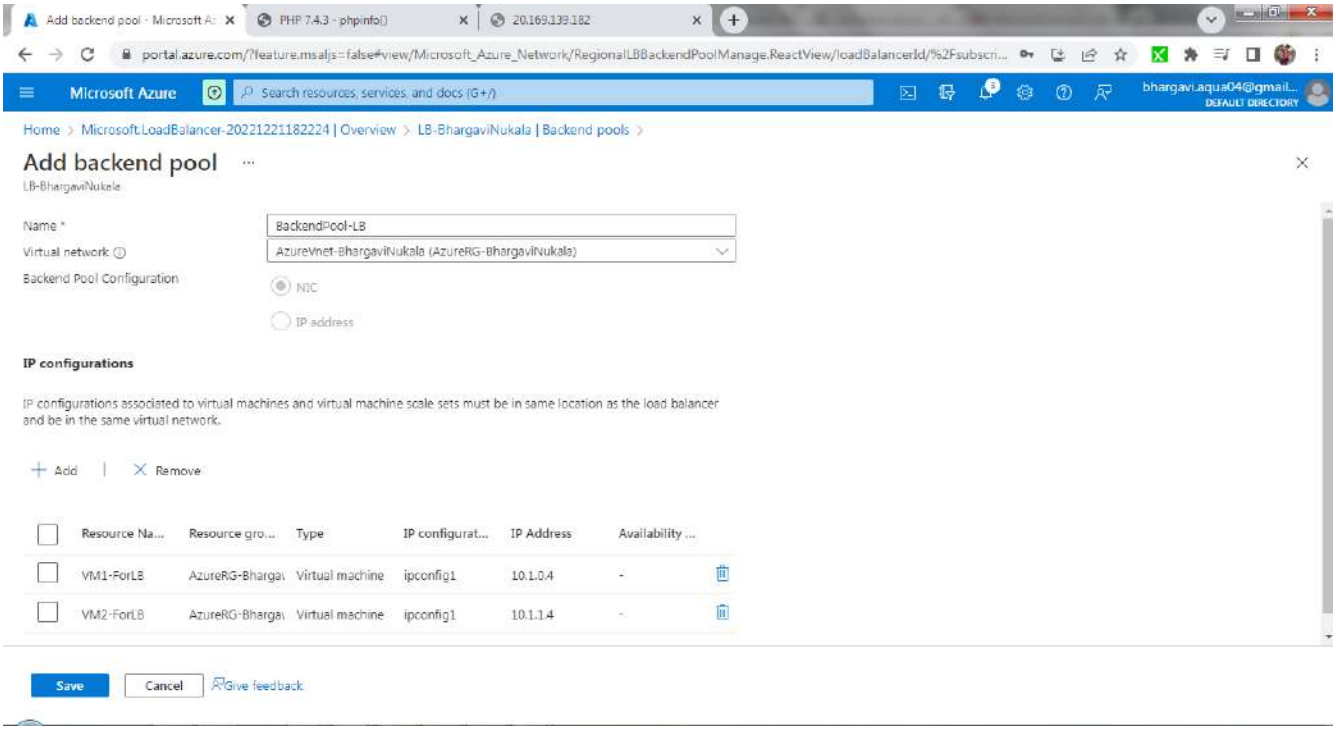
Load Balancer:

Create a Load Balancer by adding a new Frontend Configuration and adding a new Public IP.

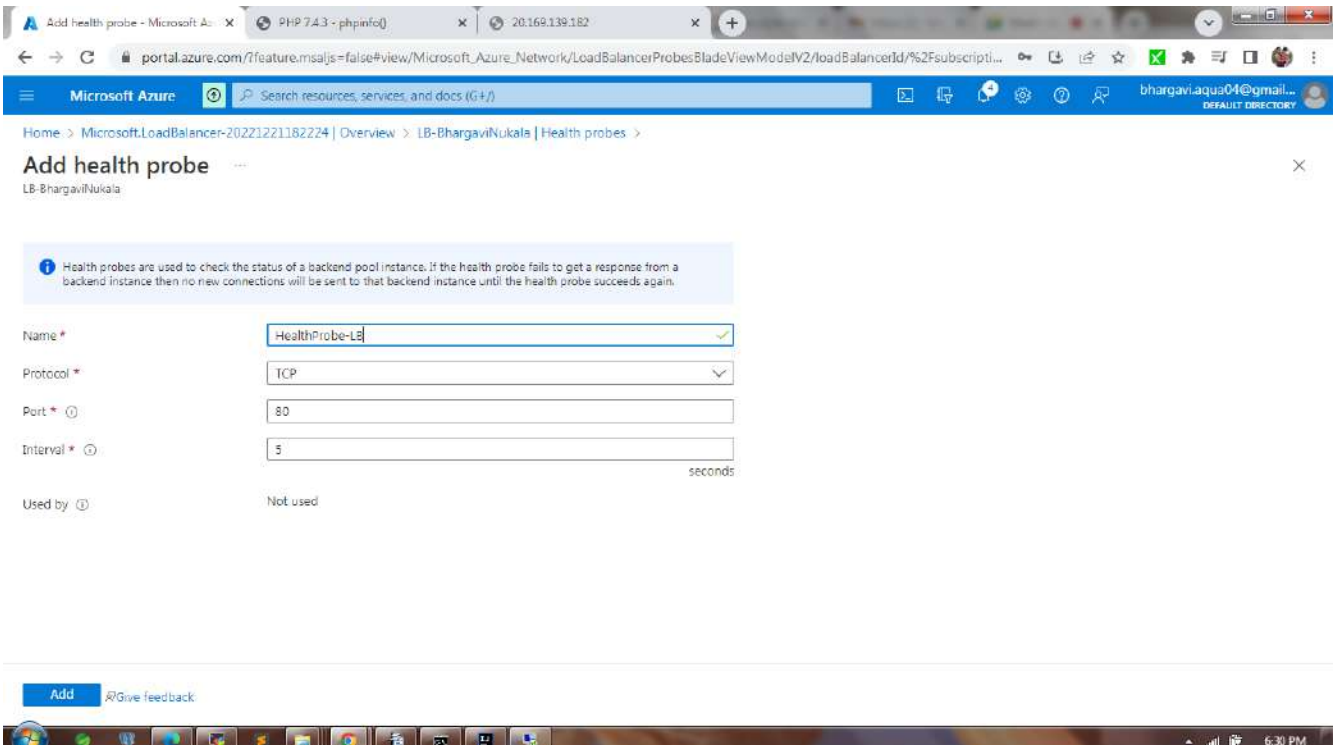




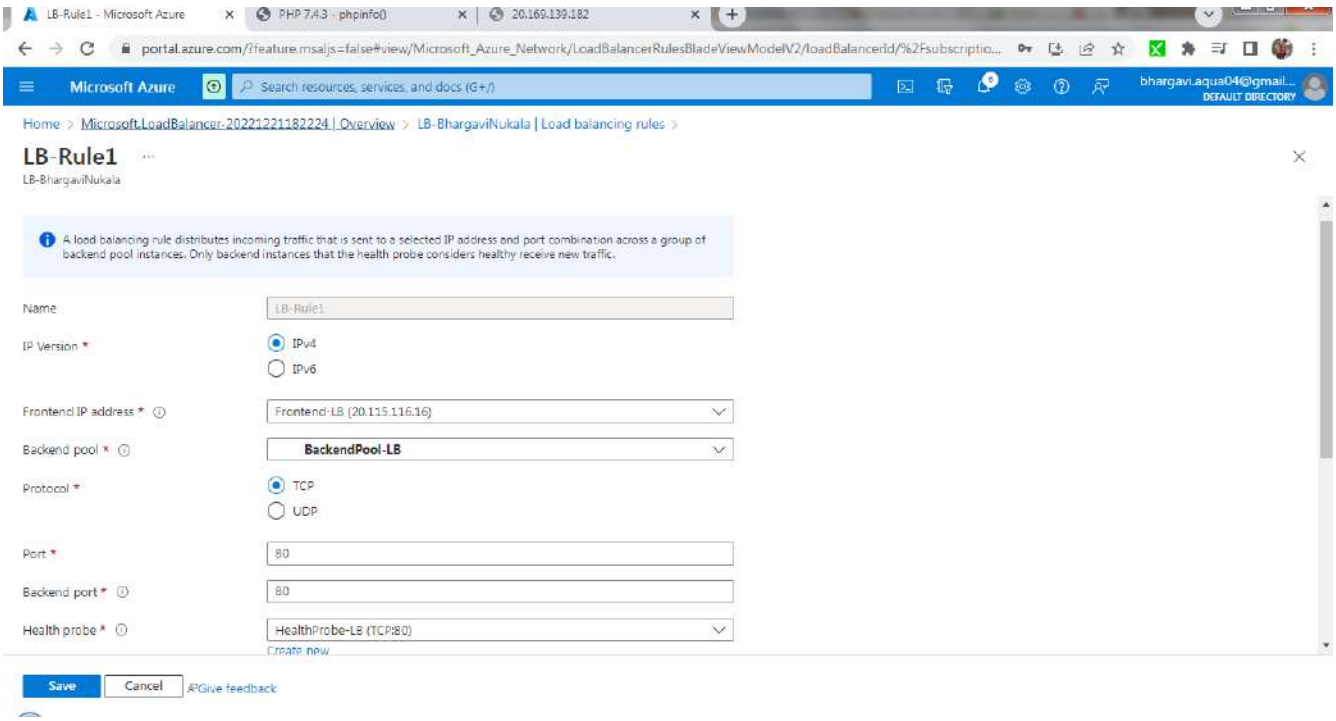
Now add Backend pools for LB. Select the created Vnet and the machines we created in 2 subnets and add them to Backend Pool of LB.



Add a health probe for LB



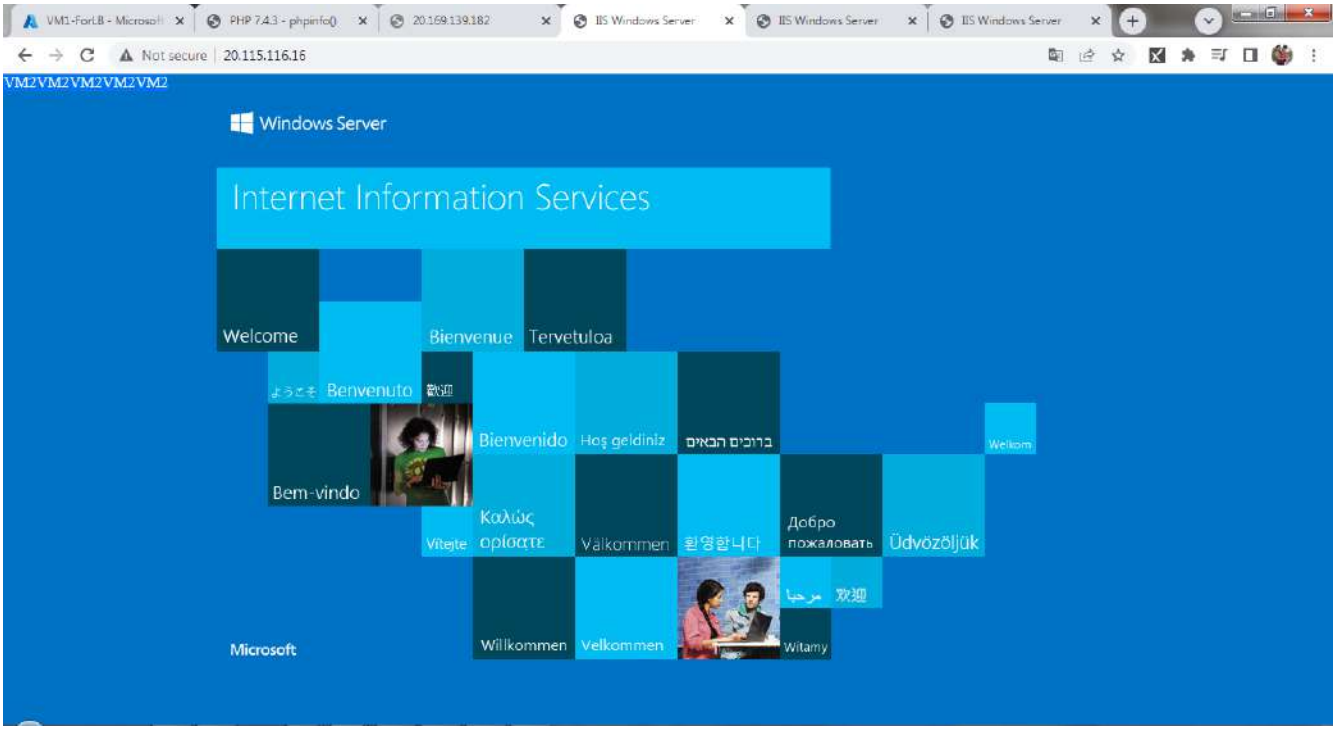
Add Load Balancing Rules by selecting the above created Frontend, Backend pool and health probe

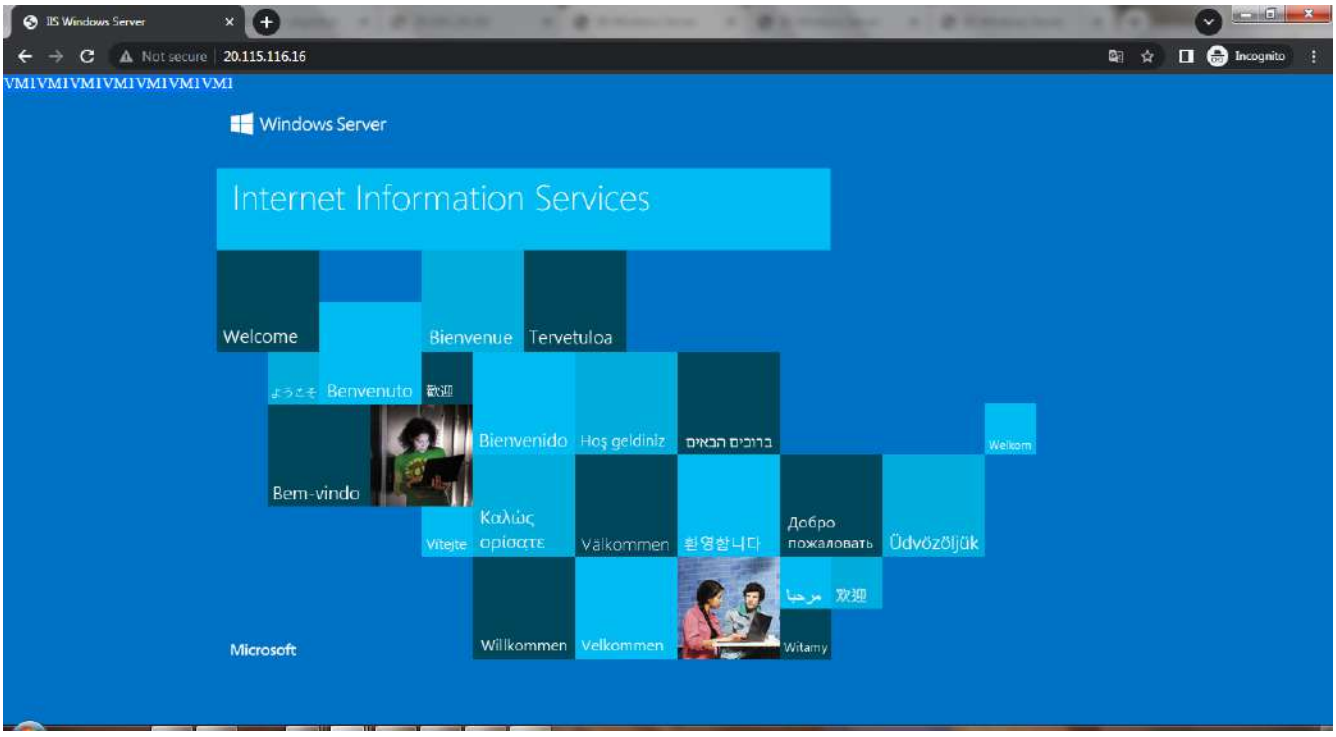


Check that IIS server is installed on two VMs. Make changes in web pages located at C:\inetpub\wwwroot to display VM1 and VM2.

Access webpage from browser using public IP of VM.

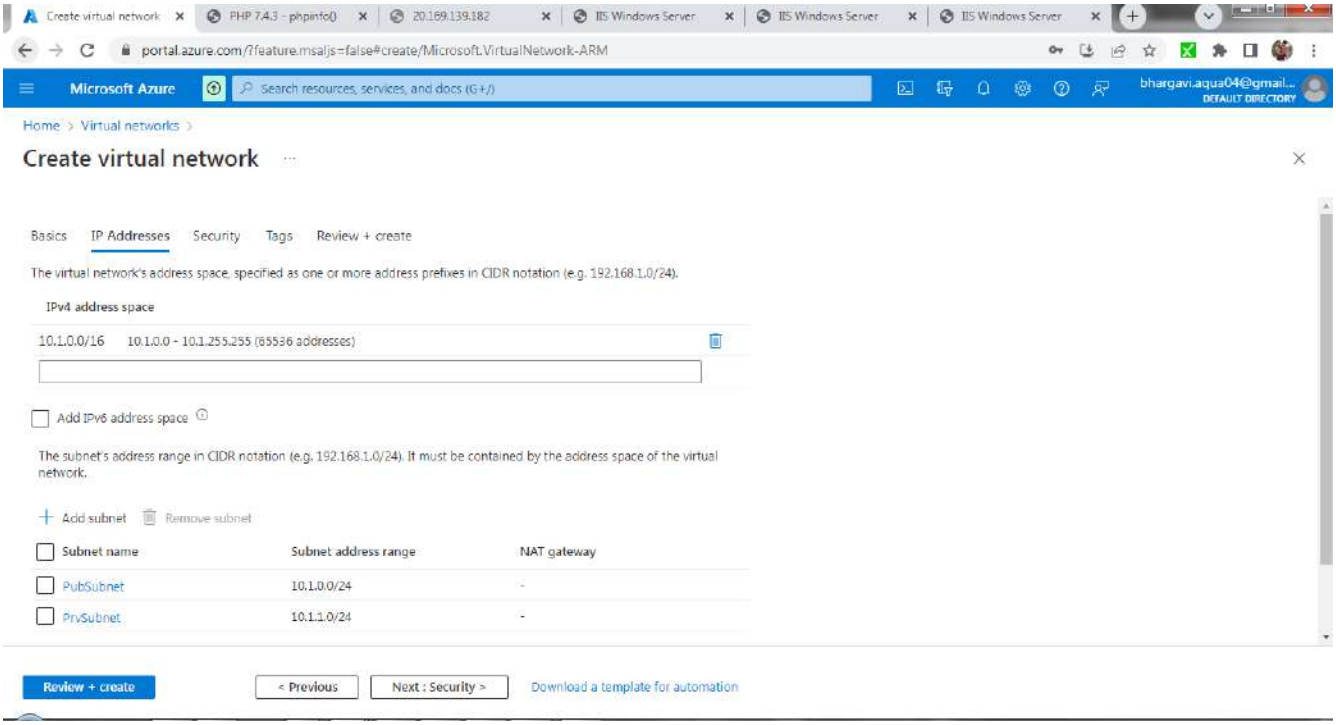
As seen, once VM1 is called and once VM2 is called by Load Balancer.



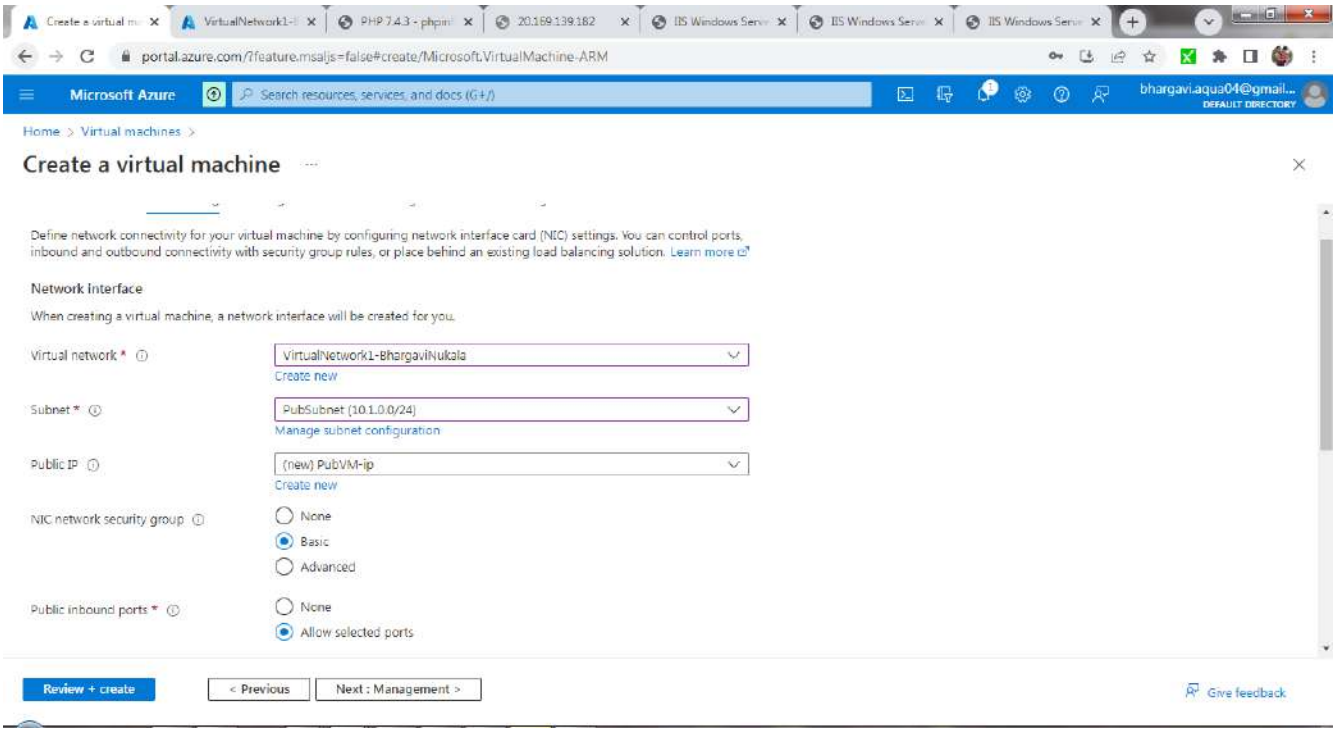


VPC:

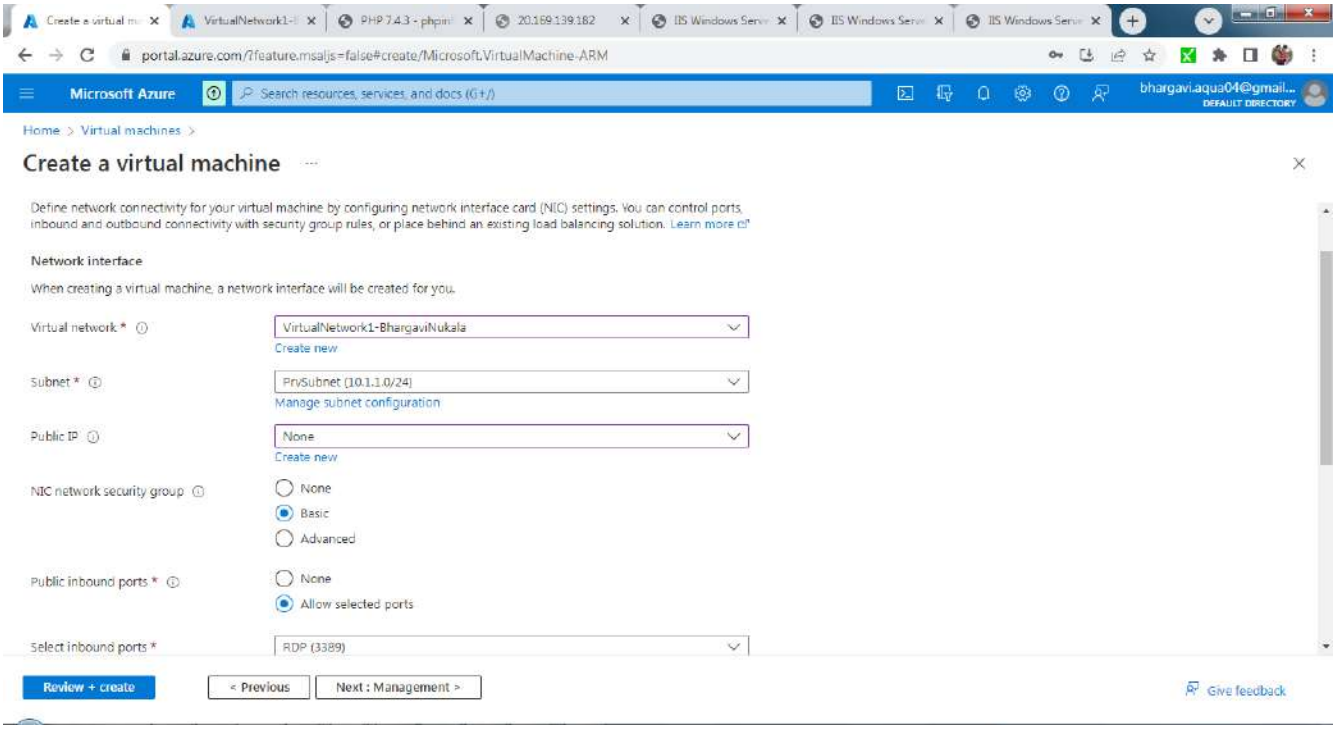
Create a Virtual Network. Add 2 subnets, one public subnet and one private subnet.



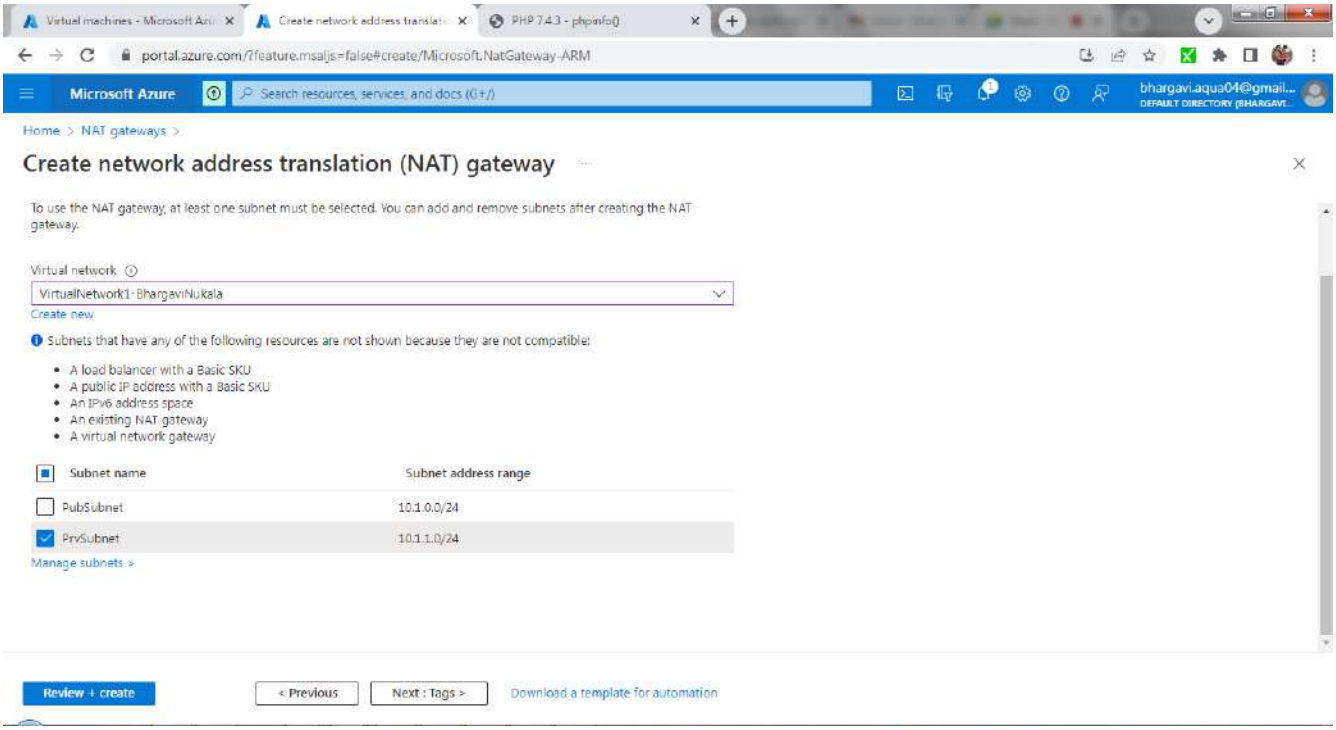
Create 2 Vms, one in Public subnet and one VM in private subnet.
Public VM should be created by selecting the public IP.



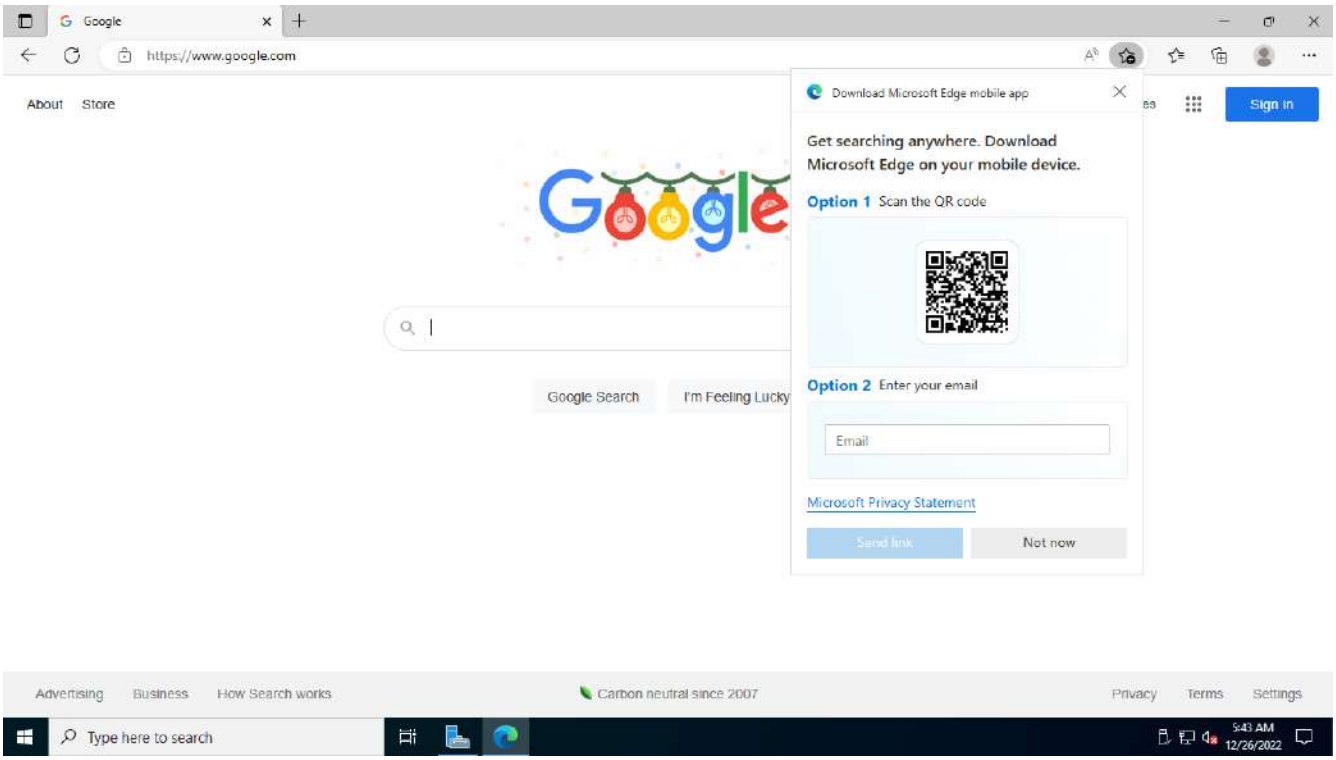
Private VM should be created by not selecting the public IP.



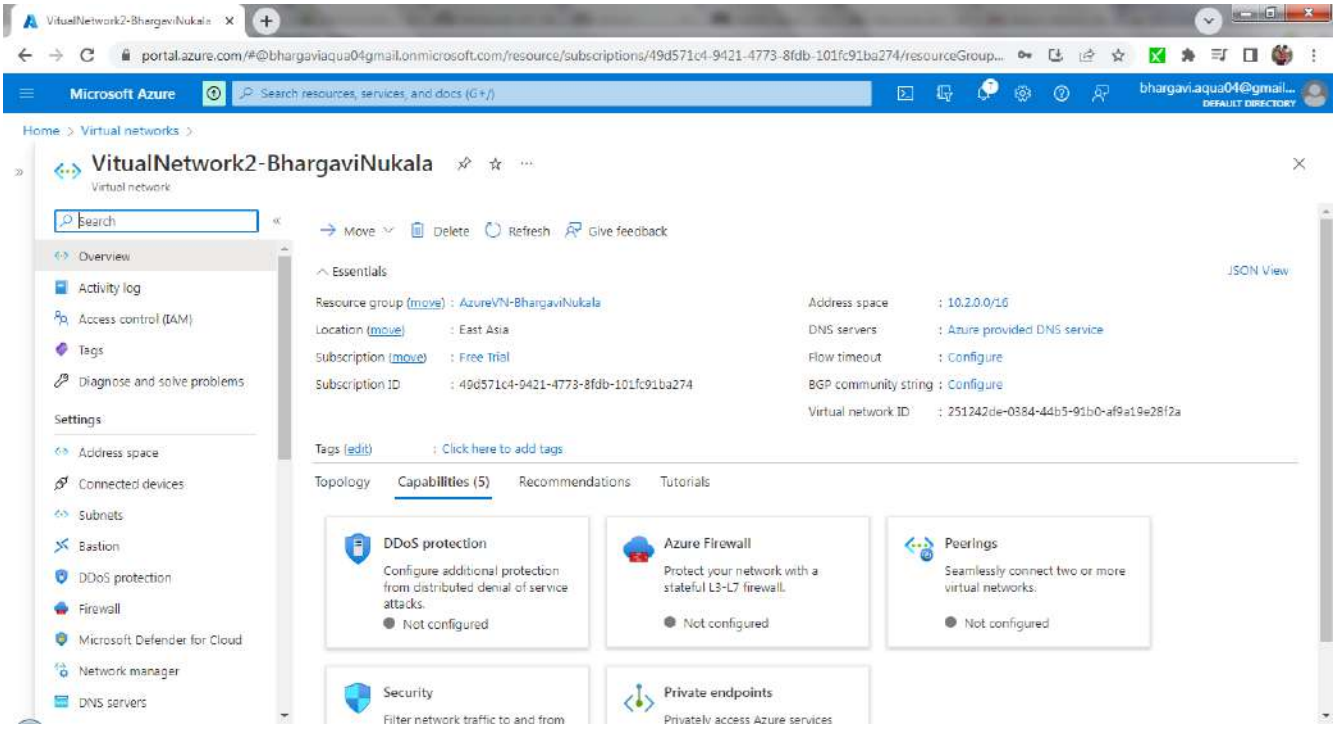
Create a NAT Gateway to get Internet to Private Subnet. Select the Virtual Network and the private subnet where you want Internet.



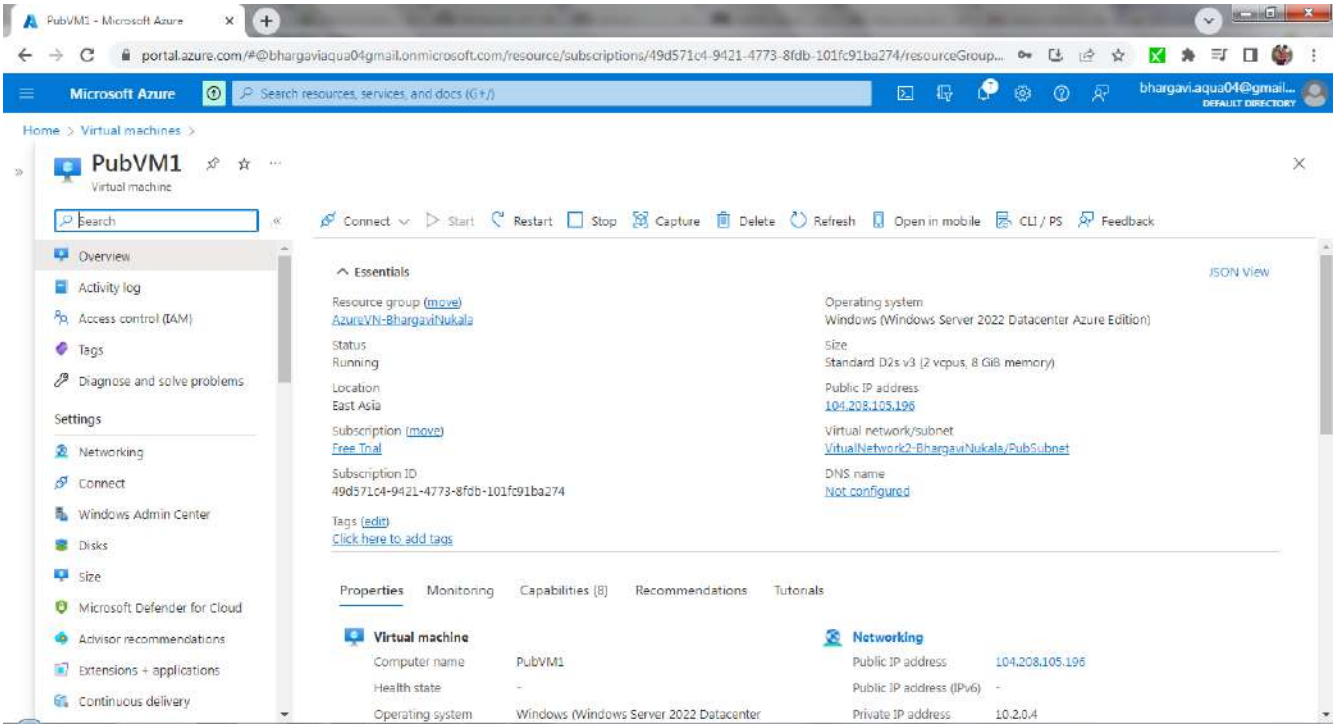
Now connect to private machine from public machine using remote desktop connection of public VM. Access google.com on private VM and the page is displayed.



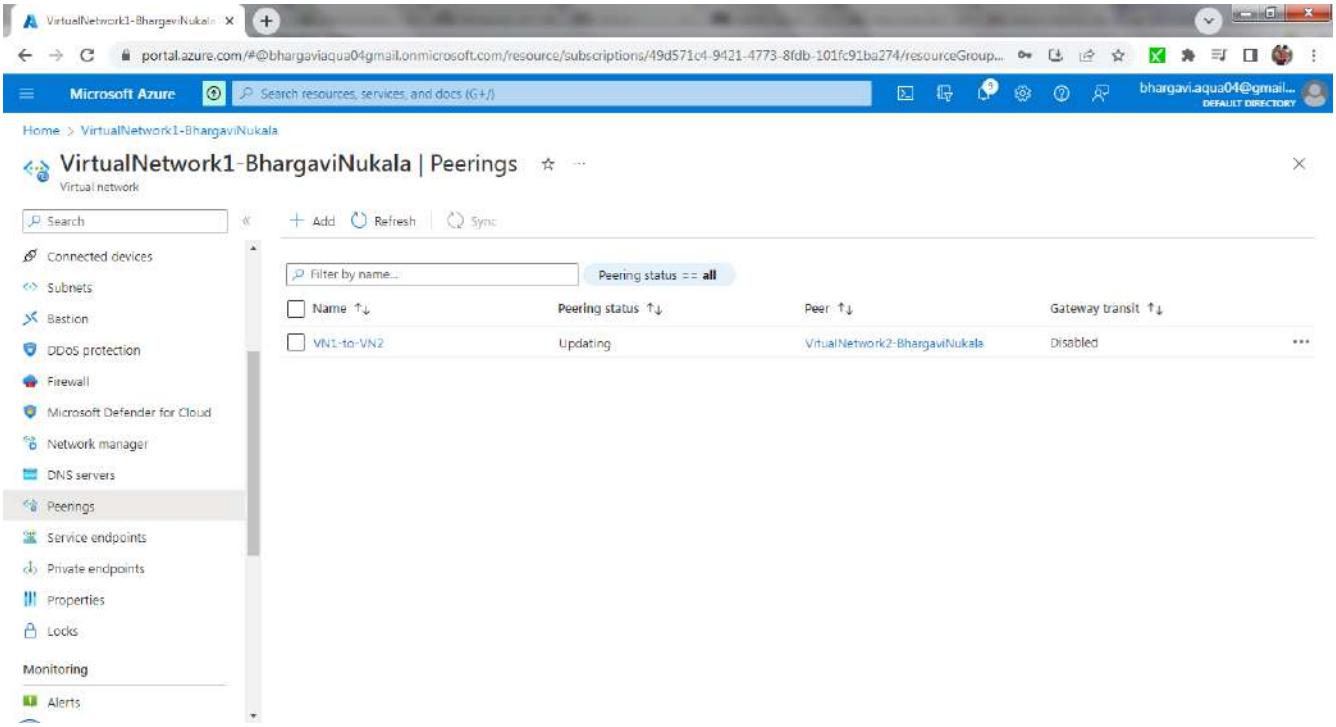
VPC Peering:
Create another Virtual Network in another region.



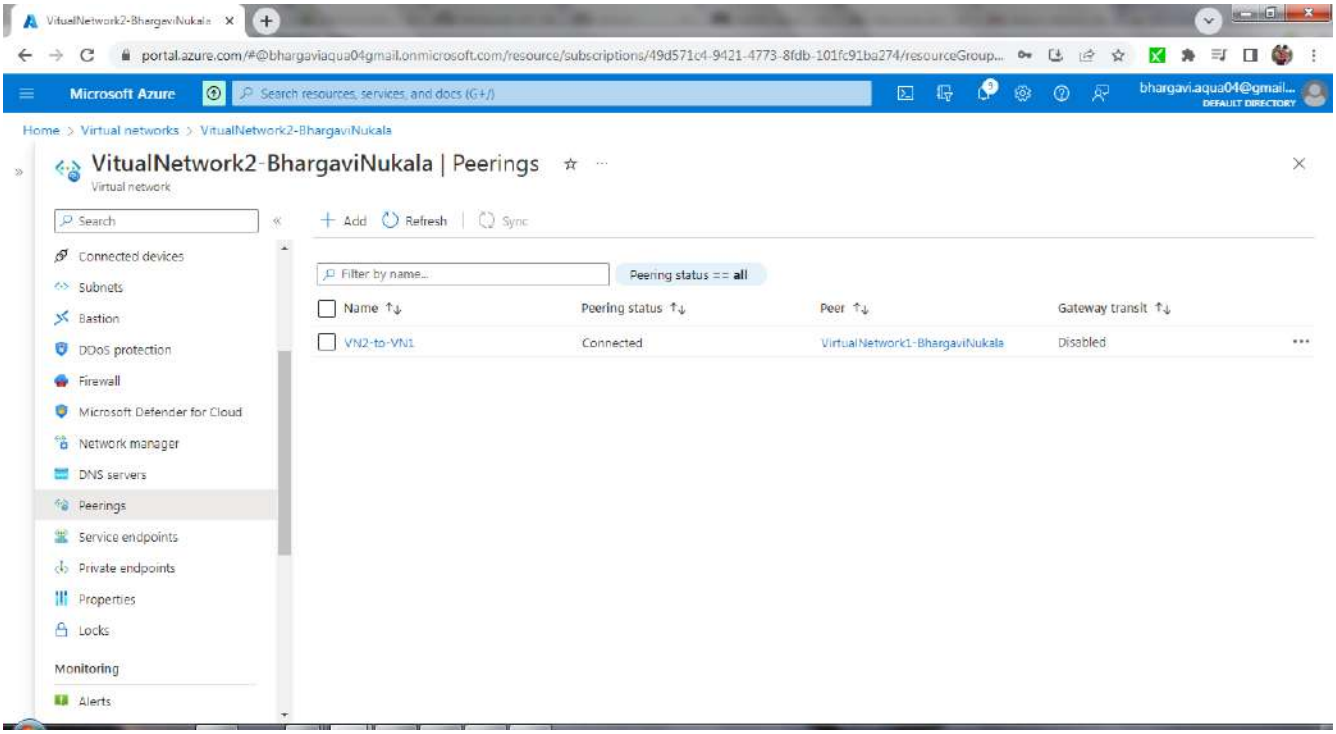
Now create a VM in Virtual Network2.



Add Peering in Virtual Network1. Goto VirtualNetwork1 and click on Peerings. Add peering between both Vnets. As seen,Two peerings are created between 2 Vnets.
VN1->VN2



VN2->VN1



Now a peering is established and when you use private IP of VM of VN2 to connect from VM of VN1, you should be able to connect.