Azure Experiments

Name: Gaurav Kakade

Index

Sr. no.	Name of Experiment	Page no	Date of experiment
1	To attach- managed, unmanaged & shared disk to VM To take snapshot & encrypt disk	3	07/06/2021
2	To deploy VM in Availability set, Availability zone & proximity placement Group	25	14/06/2021
3	Creation of Web App & management app service log, app service network configuration	39	23/06/2021
4	Create a web app instance and a virtual machine which contains database, and try to access the database through web app instance.	51	12/07/2021
5	Creation of Virtual Machine, installation of Docker Engine, Image Management, Launching container	60	19/07/2021
6	Working with container Instance, Kubernetes cluster, network configuration	70	22/07/2021
7	Azure Networking, Address space, & attaching secondary NIC to Virtual Machine.	77	28/07/2021
8	Creation of VNET & Security rule management of Virtual Machine level and subnet level.	83	03/08/2021
9	Creation of Virtual network using ARM management.	93	06/08/2021
10	Creating load balancer in azure- Basic and Standard.	98	10/08/2021

Experiment No: 1

AIM: To attach managed disk, unmanaged disk, shared disk to virtual machine, to take snapshot of managed disk and to encrypt the data disk.

PREREQUISITES: Azure Portal, RDP.

DESCRIPTION:

Azure managed disk*:* Azure Managed Disks simplifies disk management for Azure IaaS VMs by managing the storage accounts associated with the VM disks. You only have to specify the type (Premium or Standard) and the size of disk you need, and Azure creates and manages the disk for you.

Azure unmanaged disk*:* Microsoft Azure unmanaged disk is a Microsoft-managed cloud service that provides storage that is highly available, secure, durable, scalable, and redundant. Microsoft takes care of maintenance and handles critical problems for you.

Shared disk*:* Shared disks is a feature of Azure Disk Storage that allows a single disk to be attached to multiple virtual machines. This enables you to run your most demanding enterprise applications like clustered databases, parallel file systems, persistent containers and machine learning applications in the cloud, without compromising well-known deployment patterns for fast failover and high availability.

ALGORITHM:

Steps to create a virtual machine:

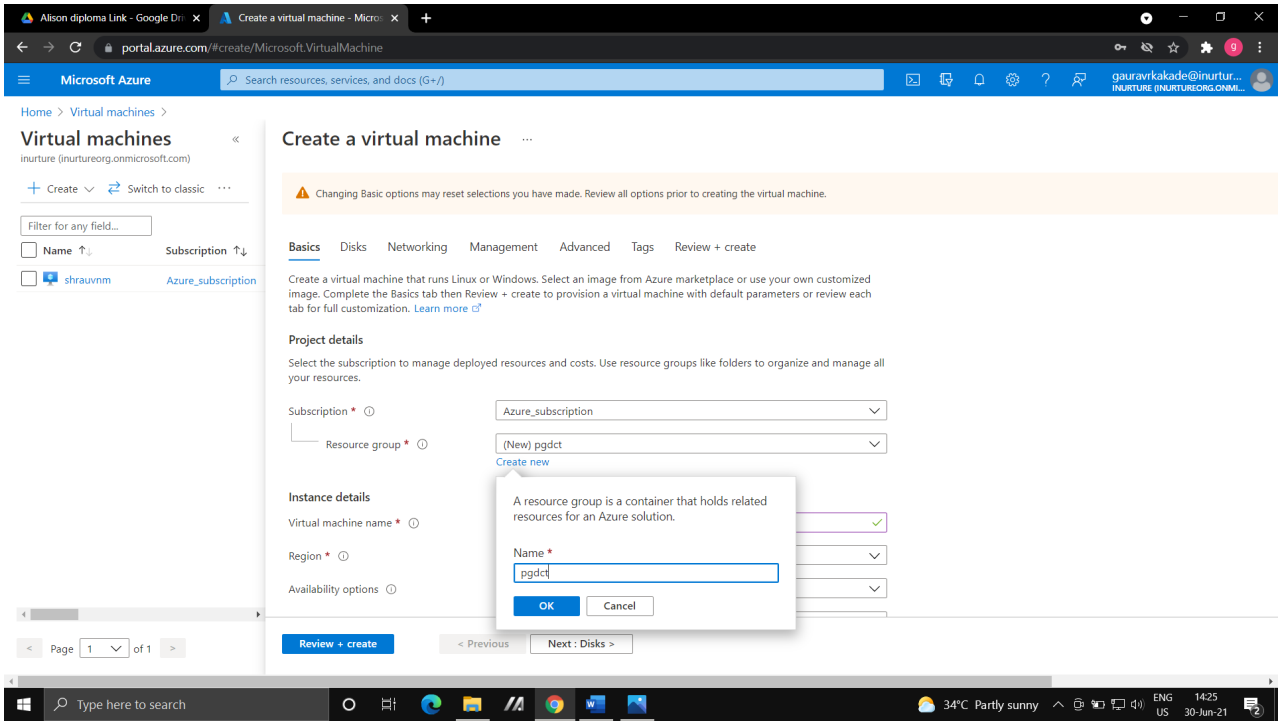
1.Click on create virtual machine.

2. a. Create new resource group.

b. Enter name of resource group.

3. a. Enter the Virtual machine name.

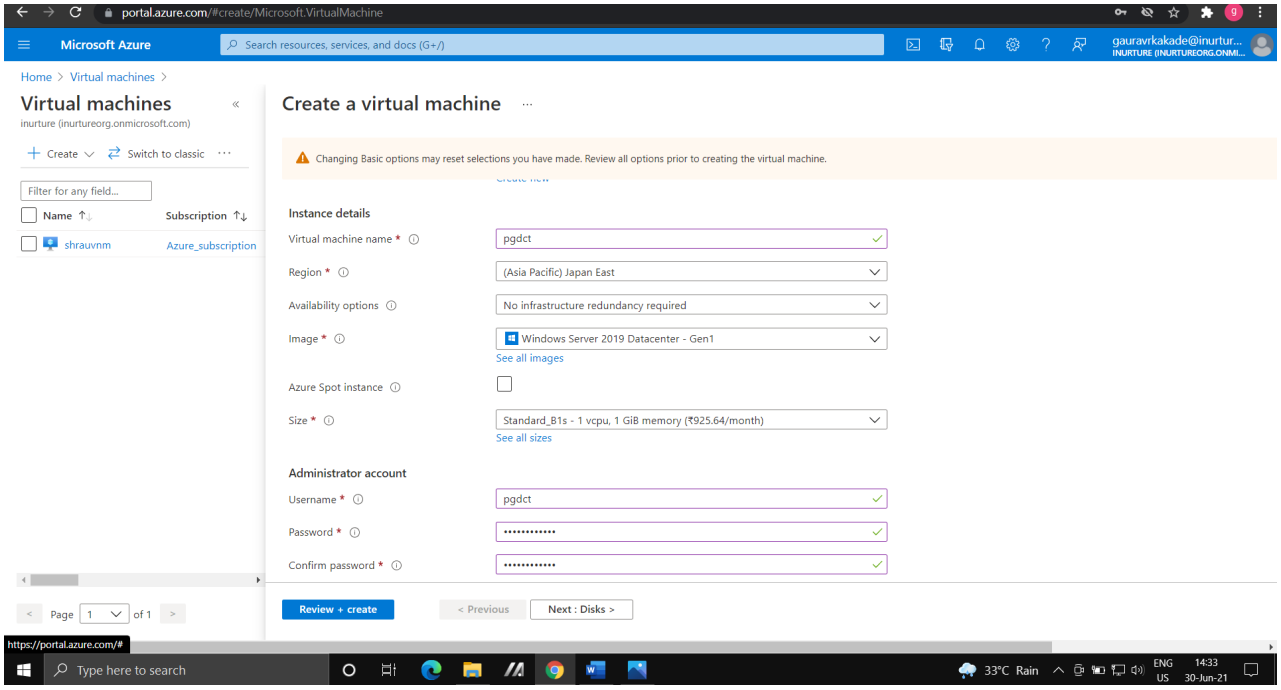
b. Select the region.



4. a. Select the image

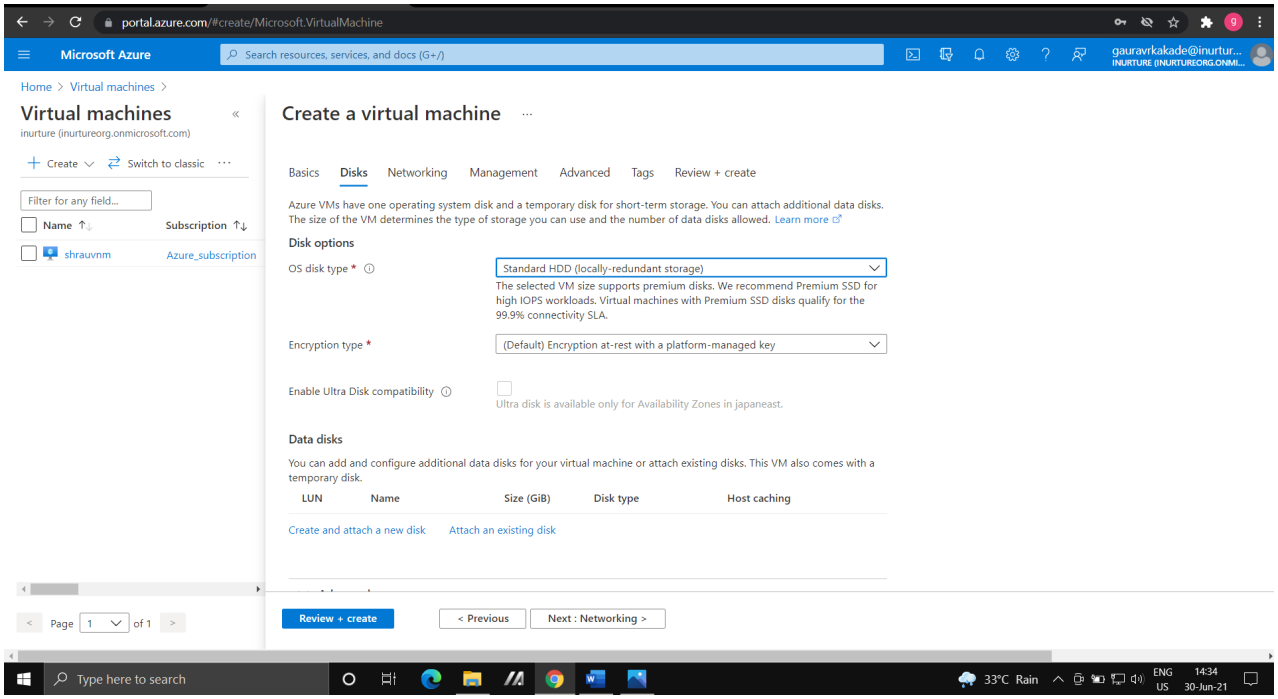
   b. Select the size of virtual machine.

5. Give the username and password for your virtual machine.



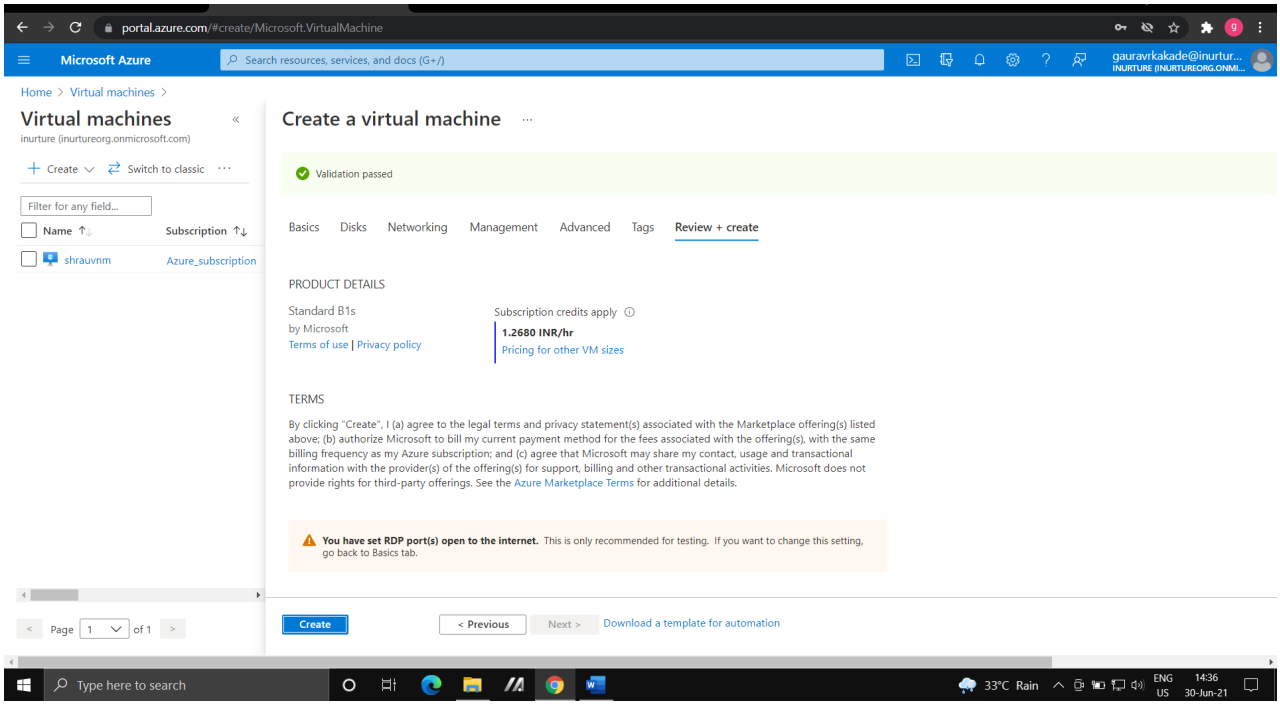
6. Select the inbound port rules.

7. Select the OS disk type as Standard HDD.



8. Go with the default settings in disks, networking, management, Advanced and tags.

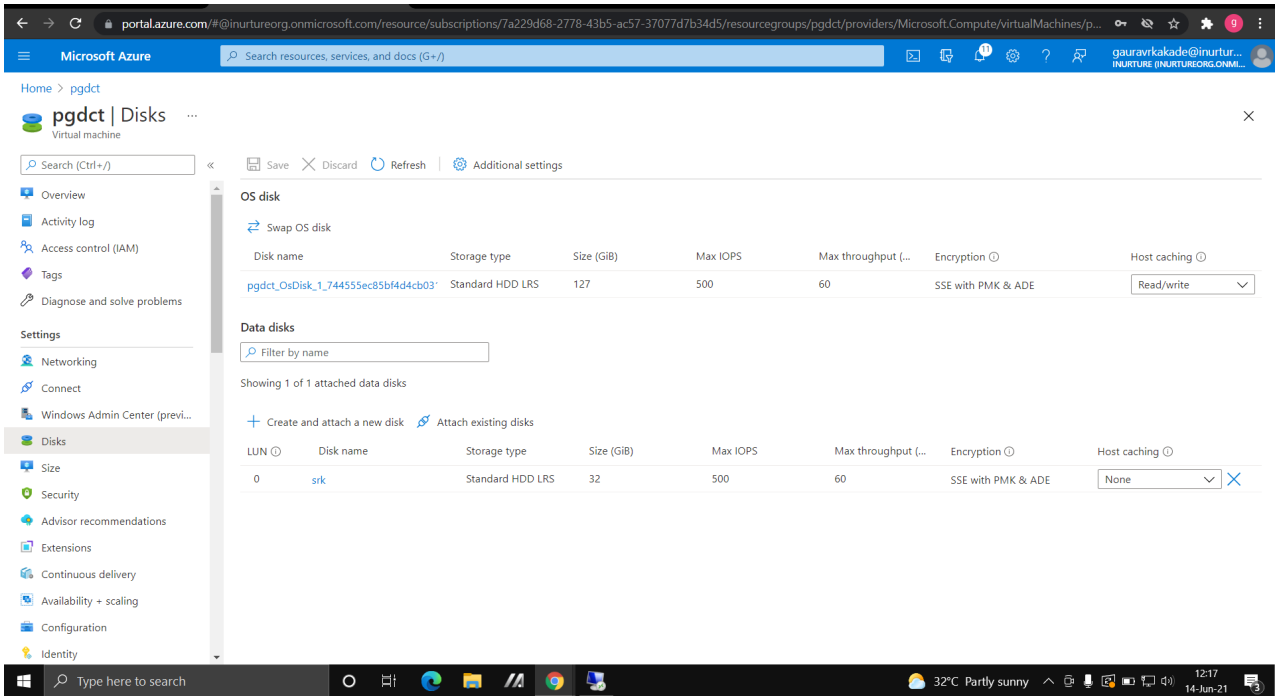
9. Review and Create.



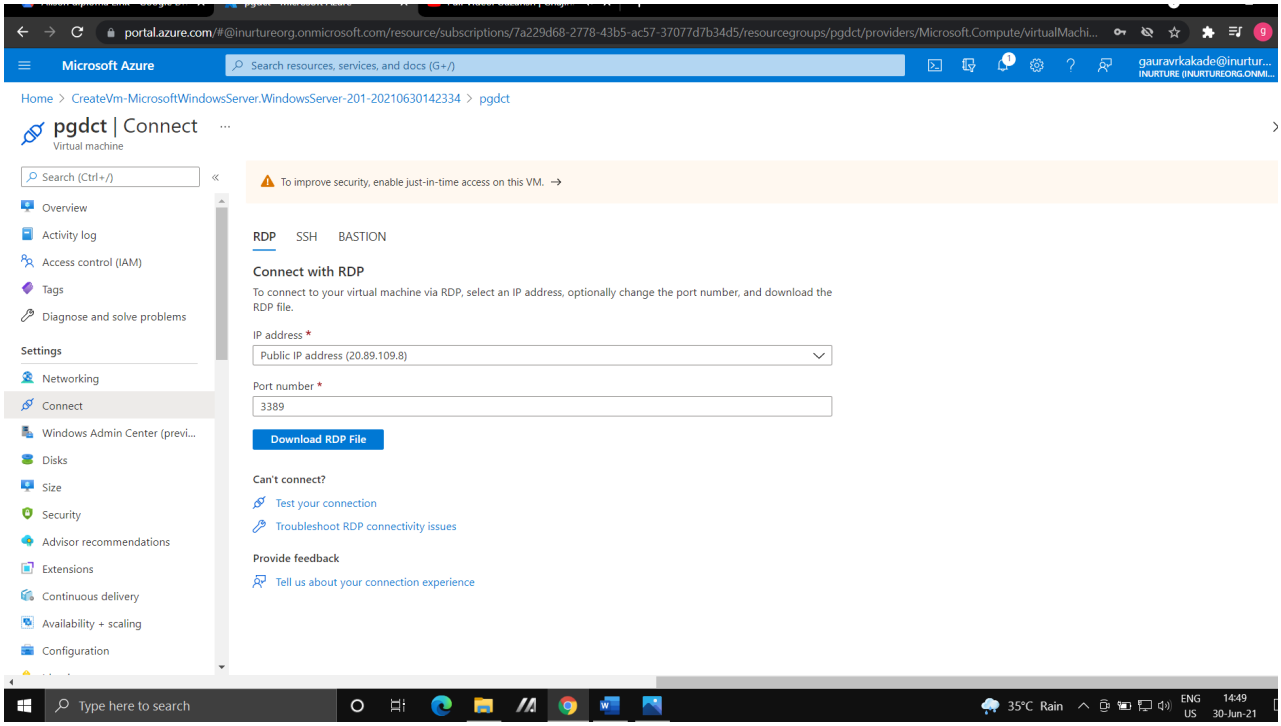
Steps to attach data/managed disk to virtual machine:

1. Go to the Virtual machine.
2. Select Disk.
3. Click on Create and attach new disk.

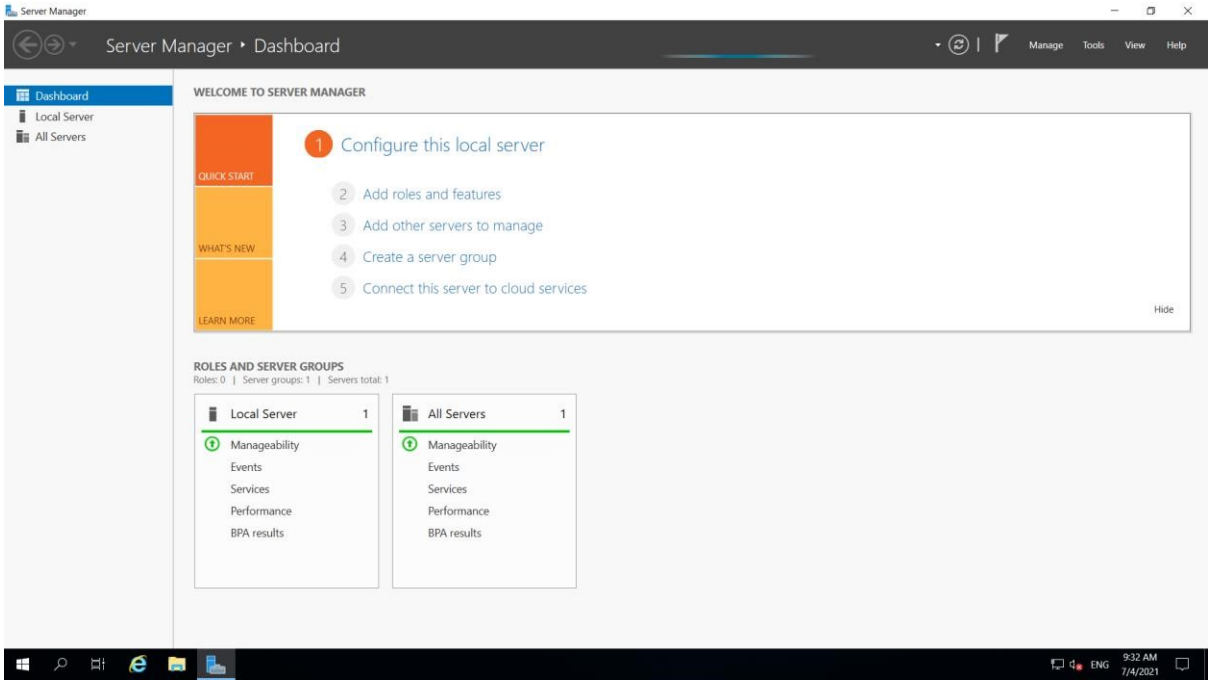
4. Give the name of disk, choose type of disk as Standard HDD and choose size then save it.



5.Now Connect your virtual machine with RDP.



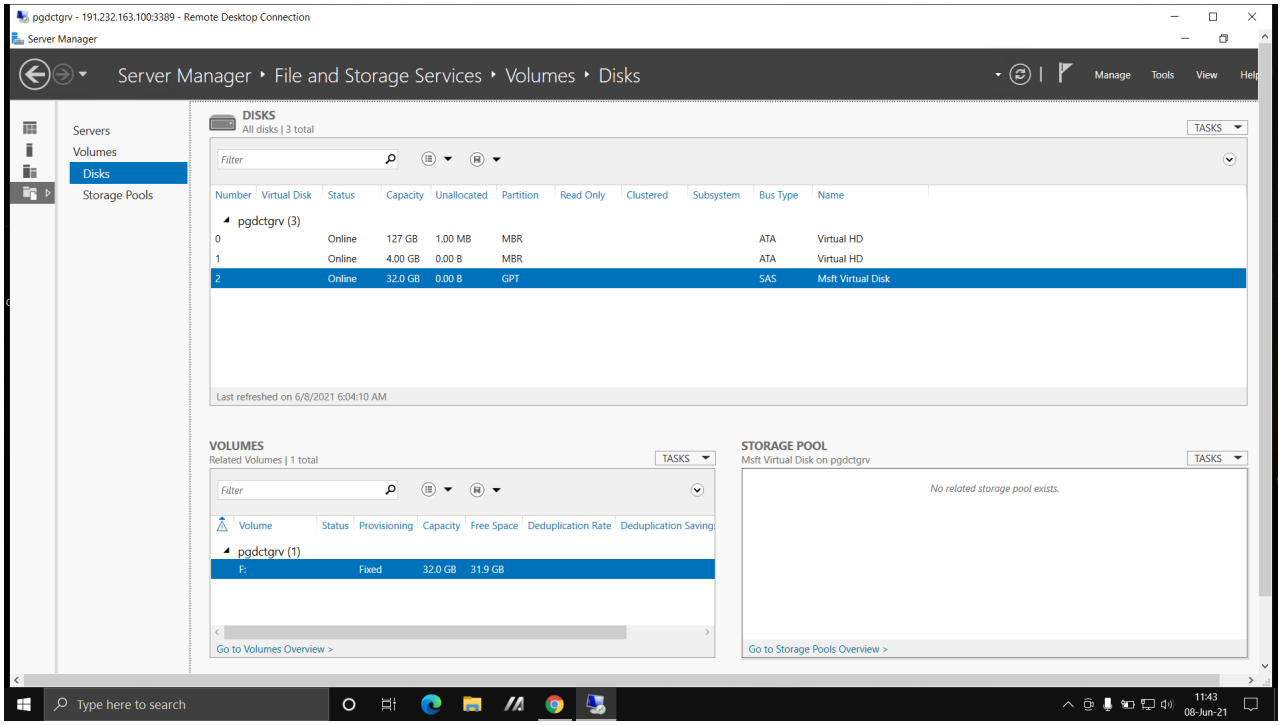
6.a. Open the Server Manager.



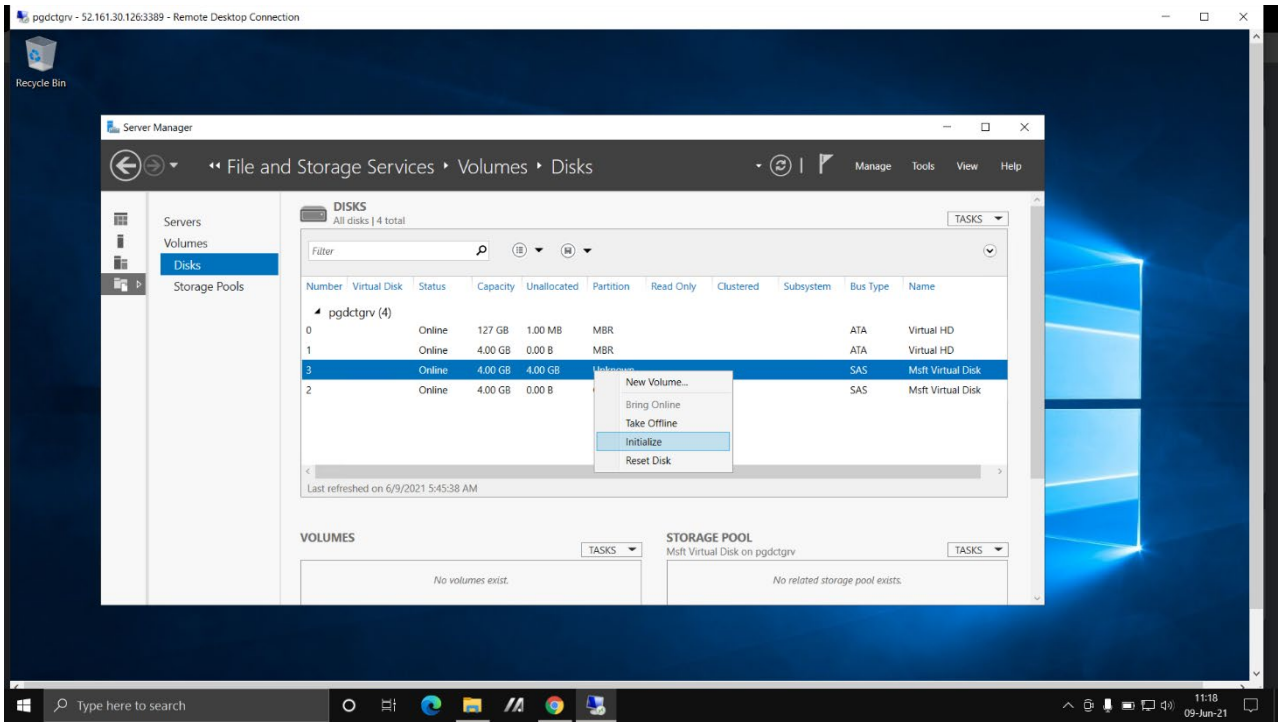
b. Go to the File and Storage block

c. Select Disk

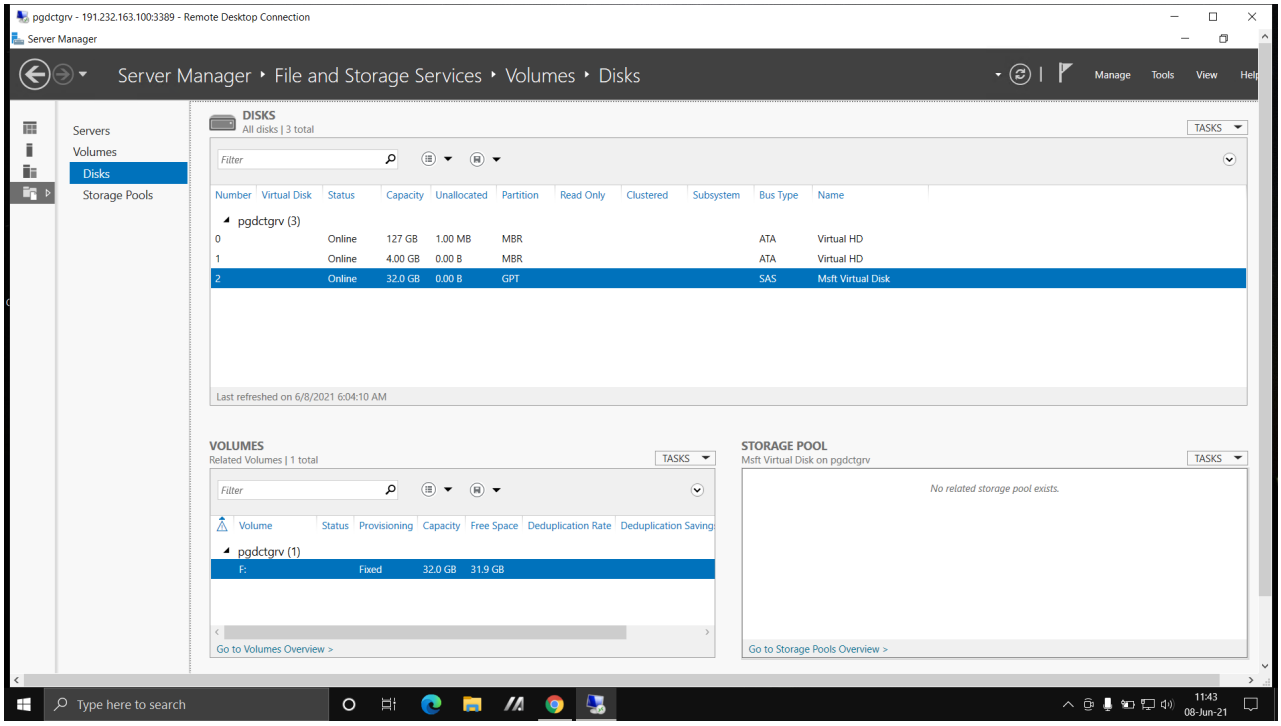
d. You will see the disk is attached.



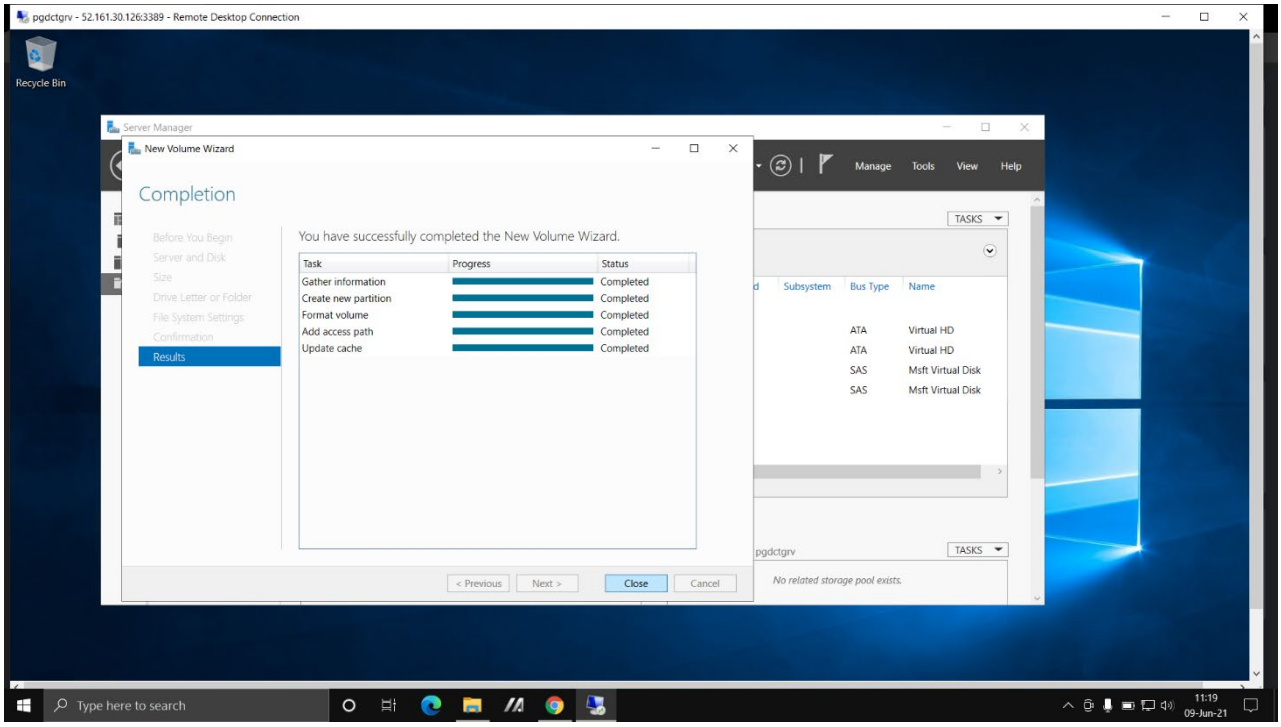
7. a. Right click on disk; you will see the initialize



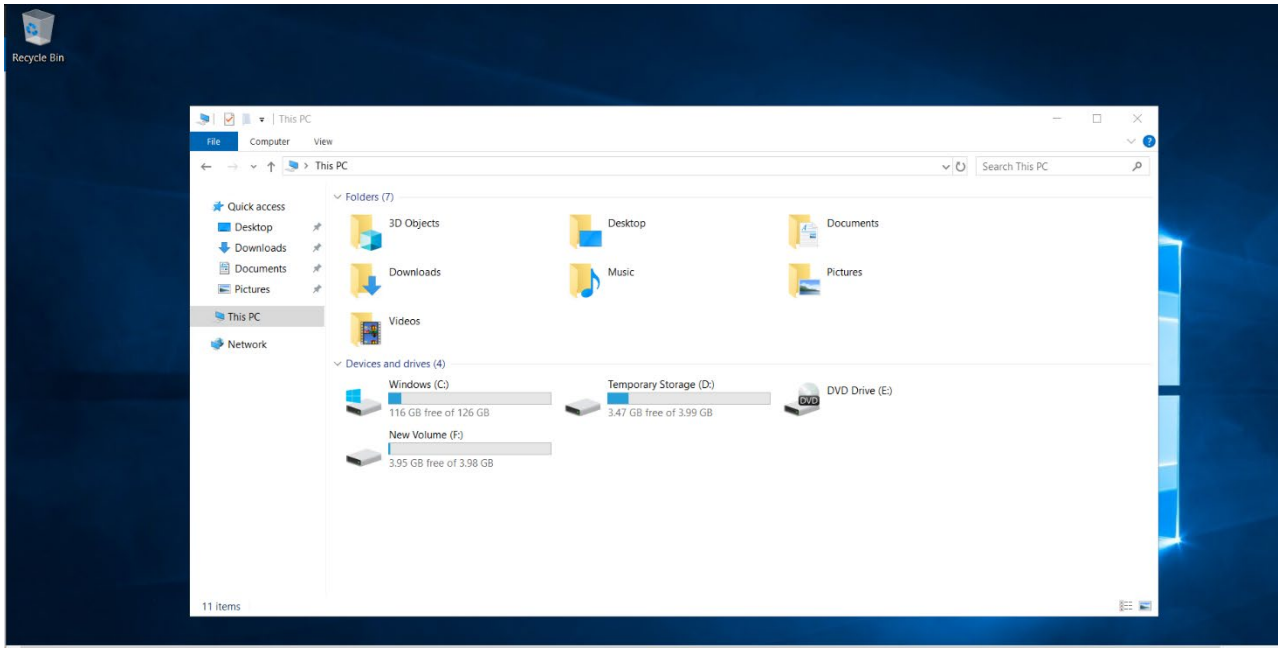
b. Again click on disk and then click on new volume.



8. Assign a letter to your disk and click create.

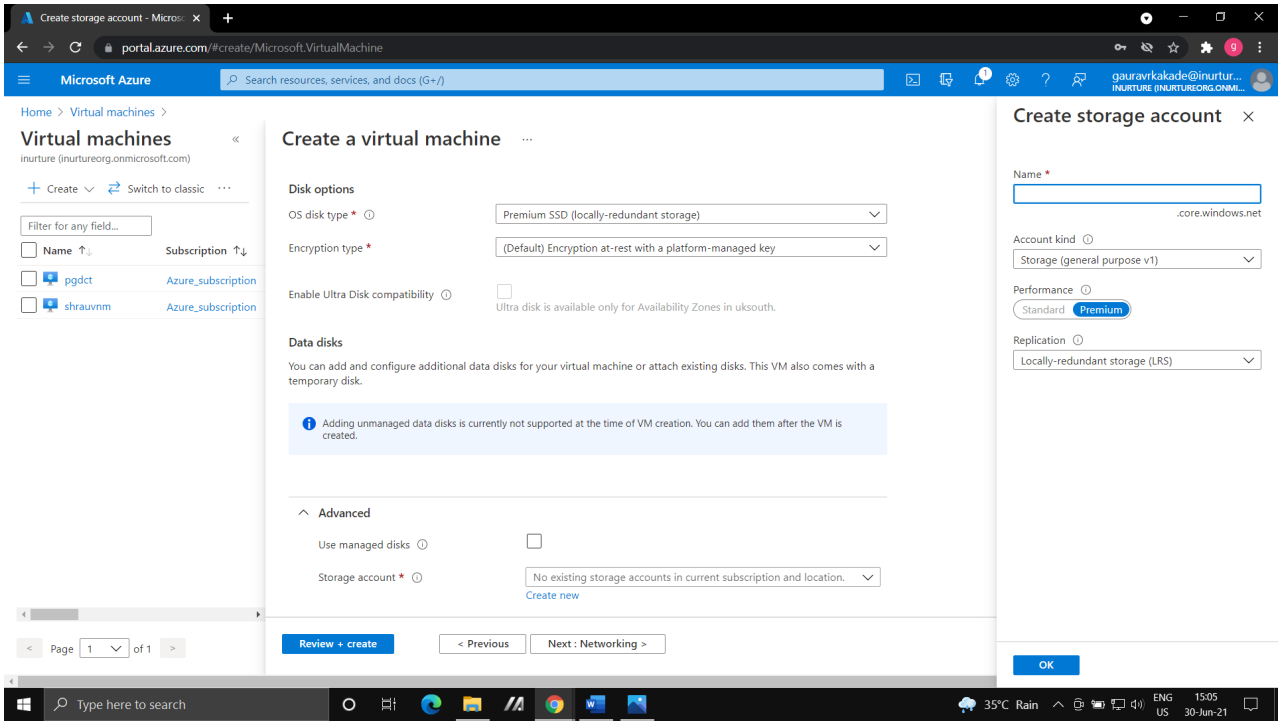


9.Go to file manager you will see your disk is attached.



Steps to attach unmanaged disk to virtual machine:

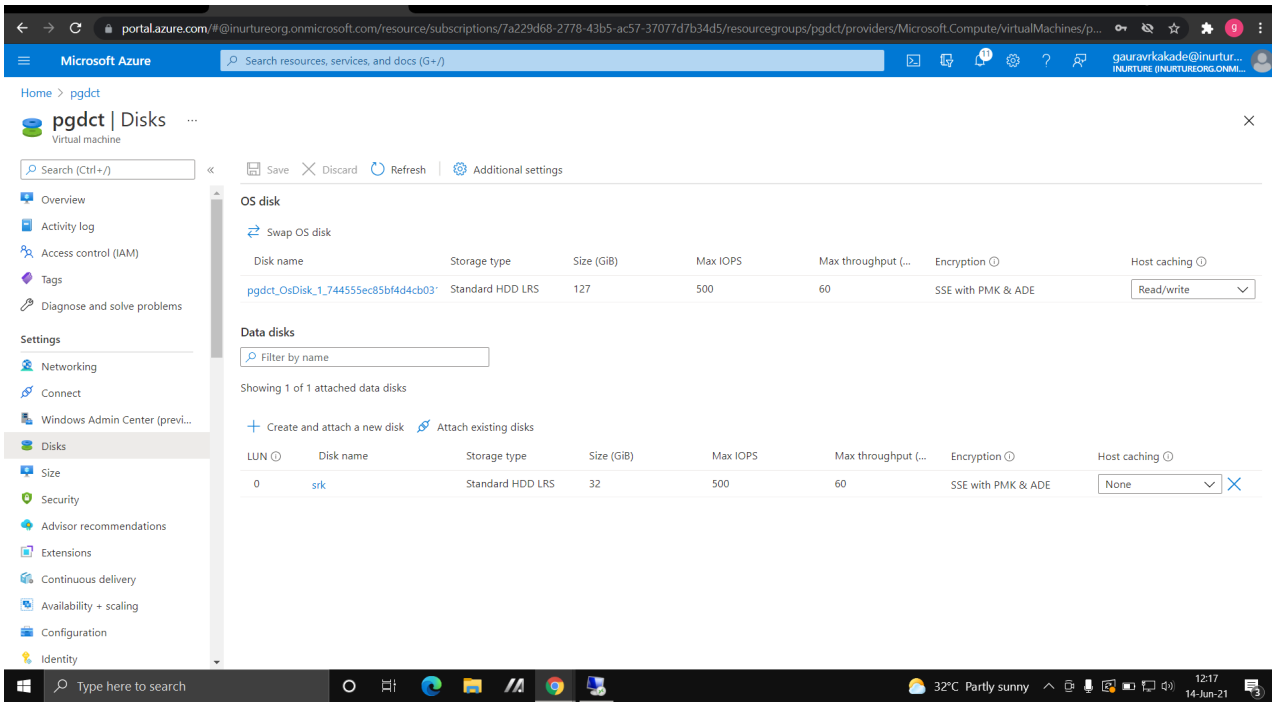
1. Create the Virtual Machine.
2. In disk go to the advanced tab and untick the managed disk option.
3. It will show you to create storage account, give name to storage account and click OK.



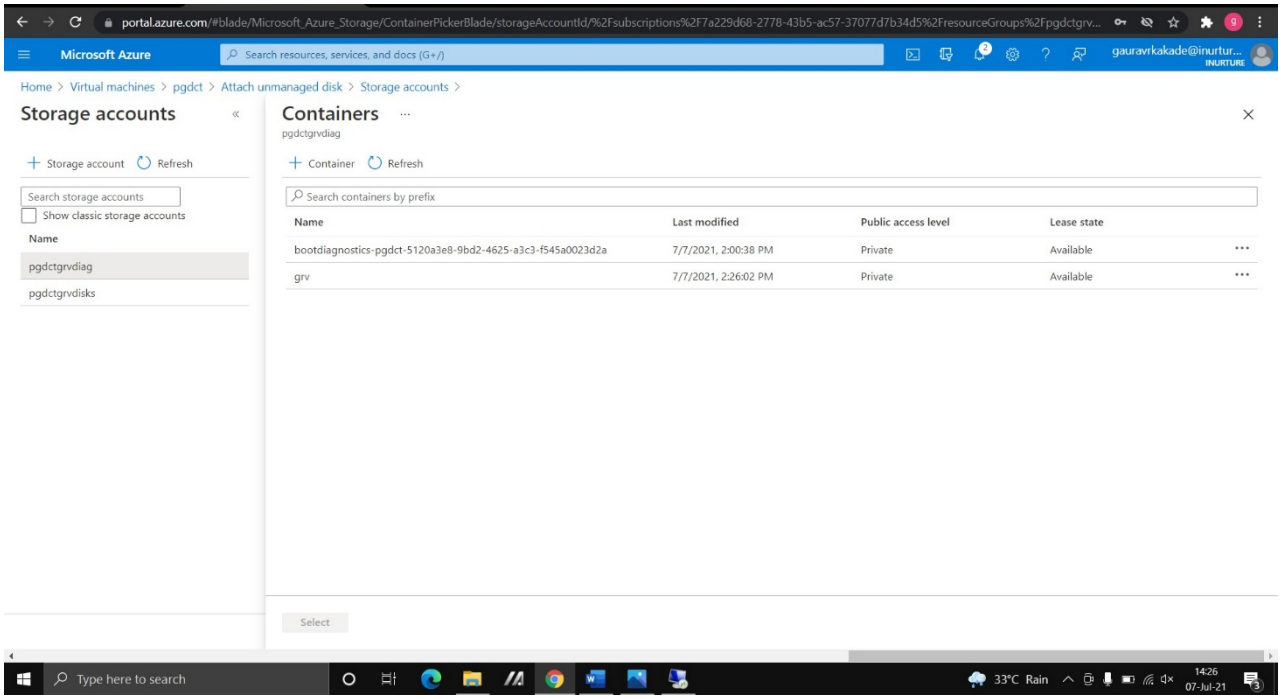
4.Go to the default settings and create virtual machine.

5.Go to disk, click on data disk and it will show you attach unmanaged disk.

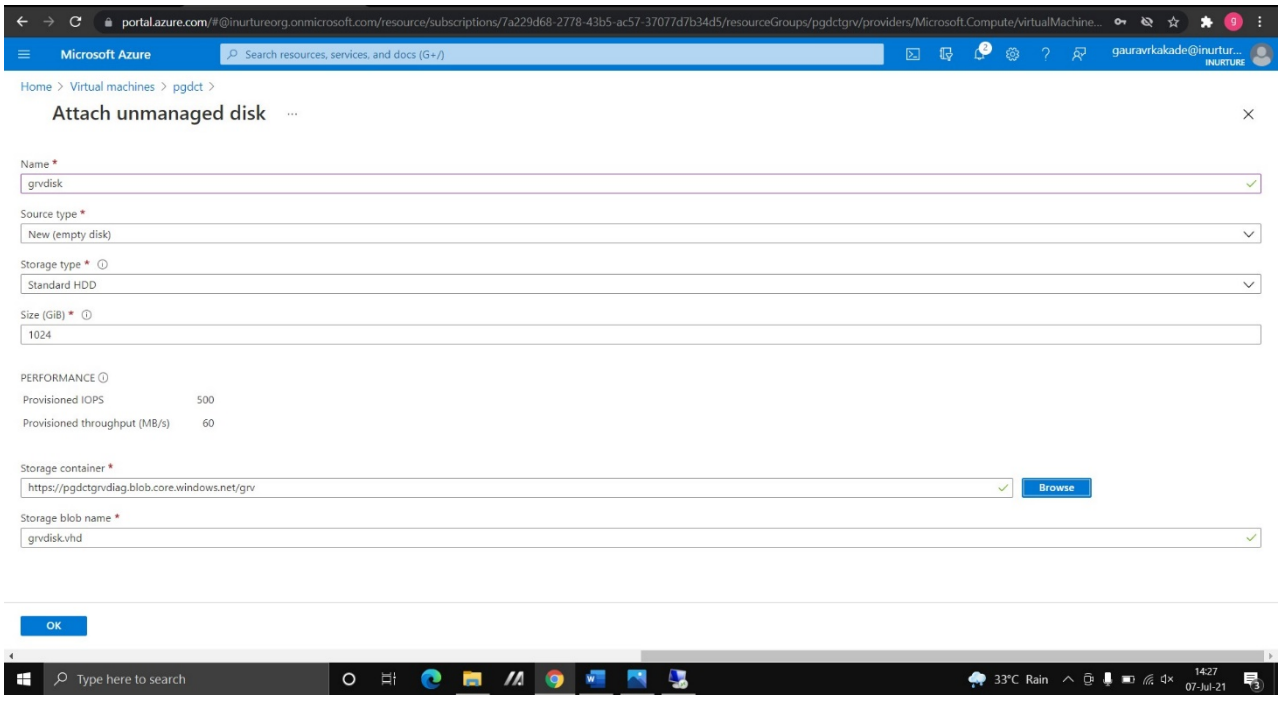
6.Give the name to disk, choose storage type as standard HDD.



7.In storage container, go to your storage account and create container.



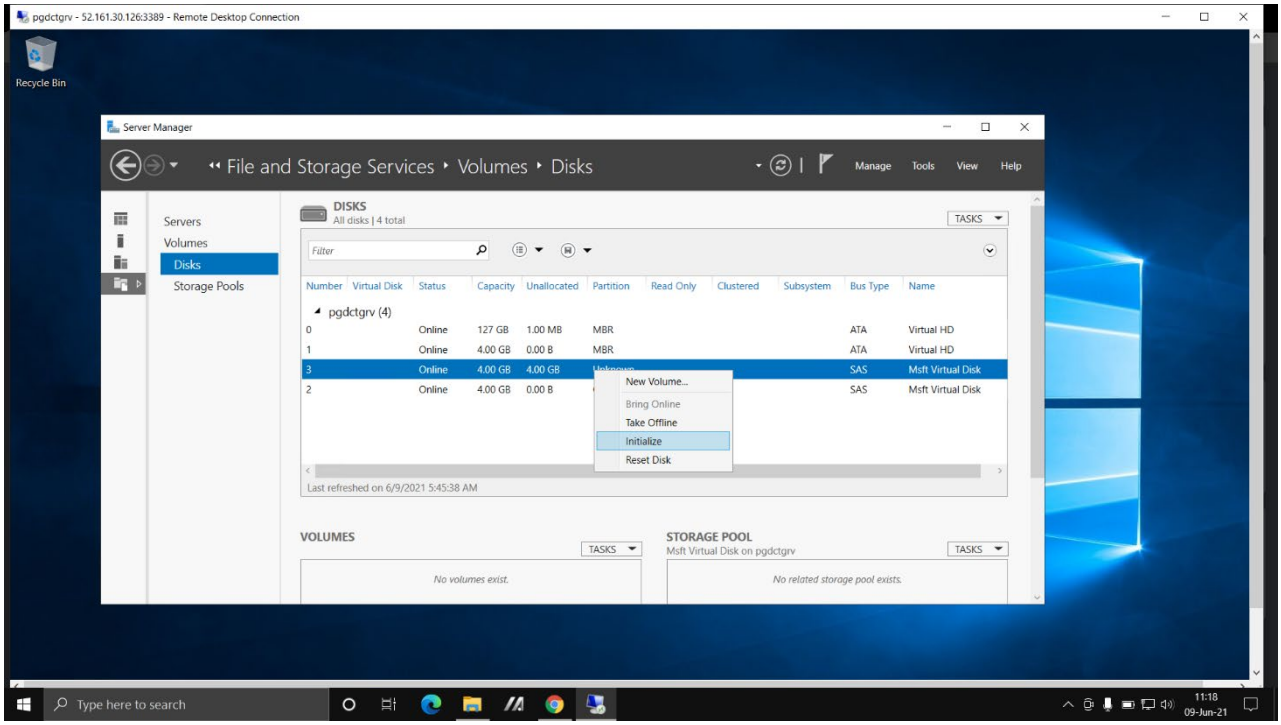
8.Go to the properties, copy the URL and paste it in storage container option.



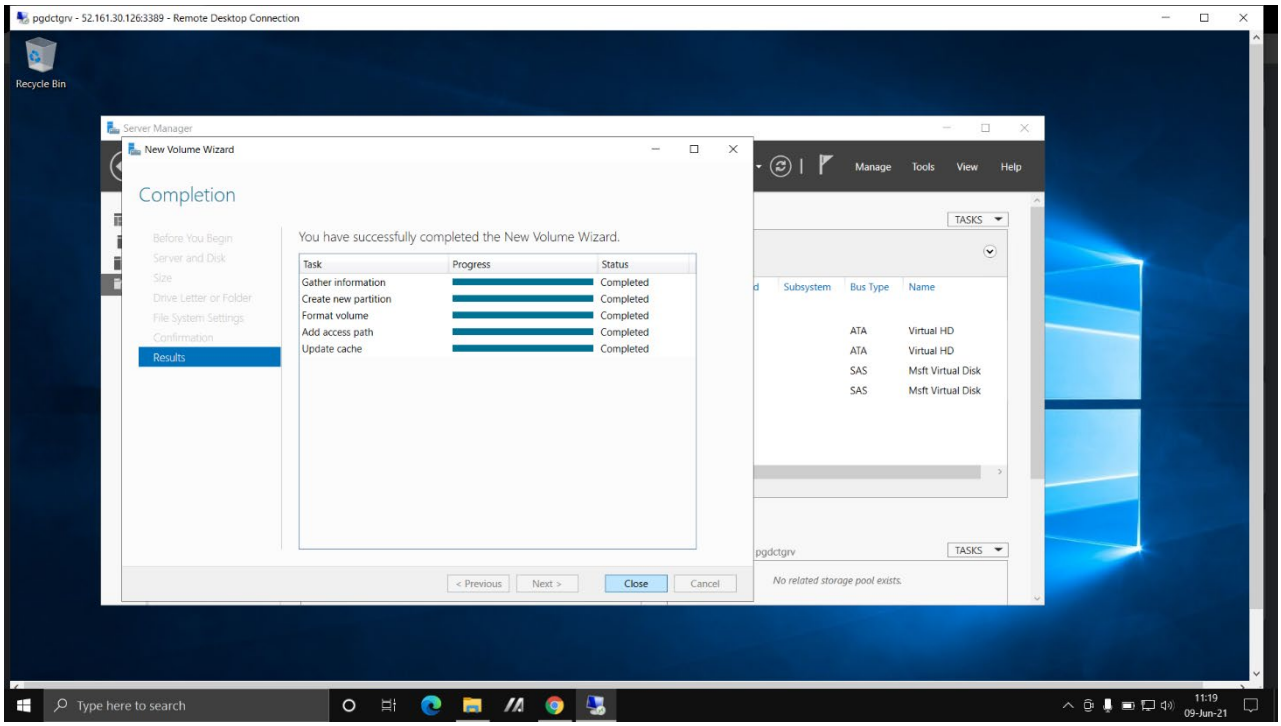
9.Now connect with RDP and go to server manager.

a. Go to the file and storage.

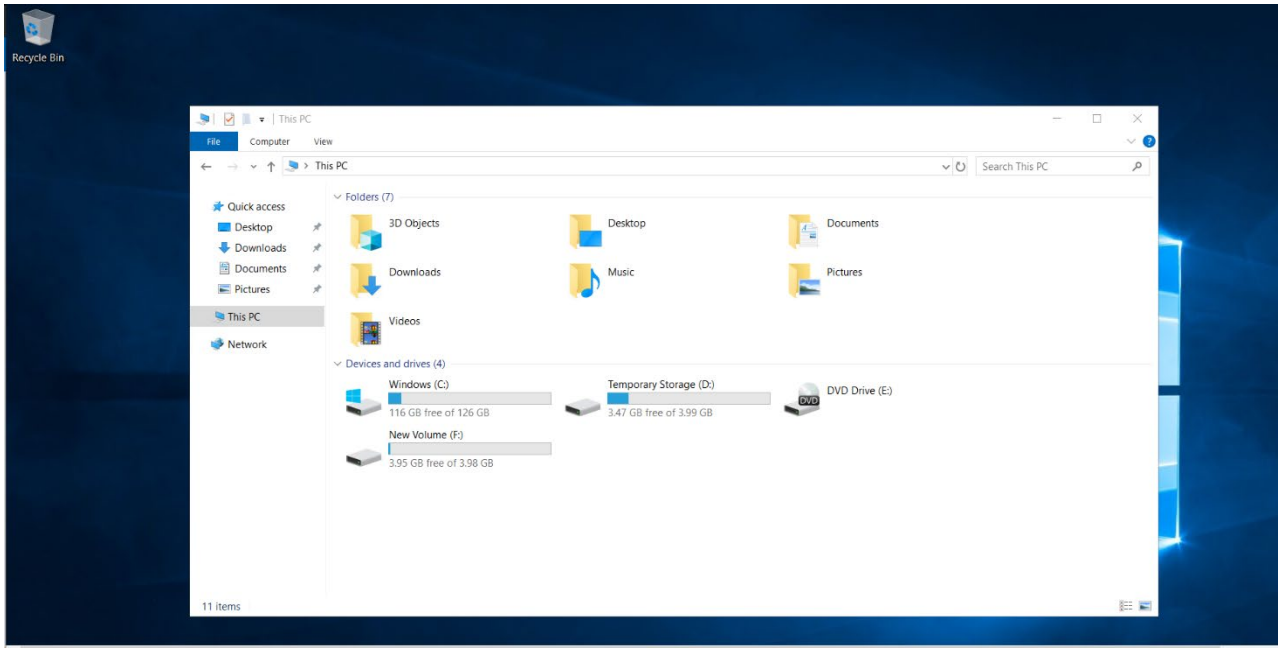
b. Go to the Disk and you will see the disk.



10.Initialize the disk and create new volume.

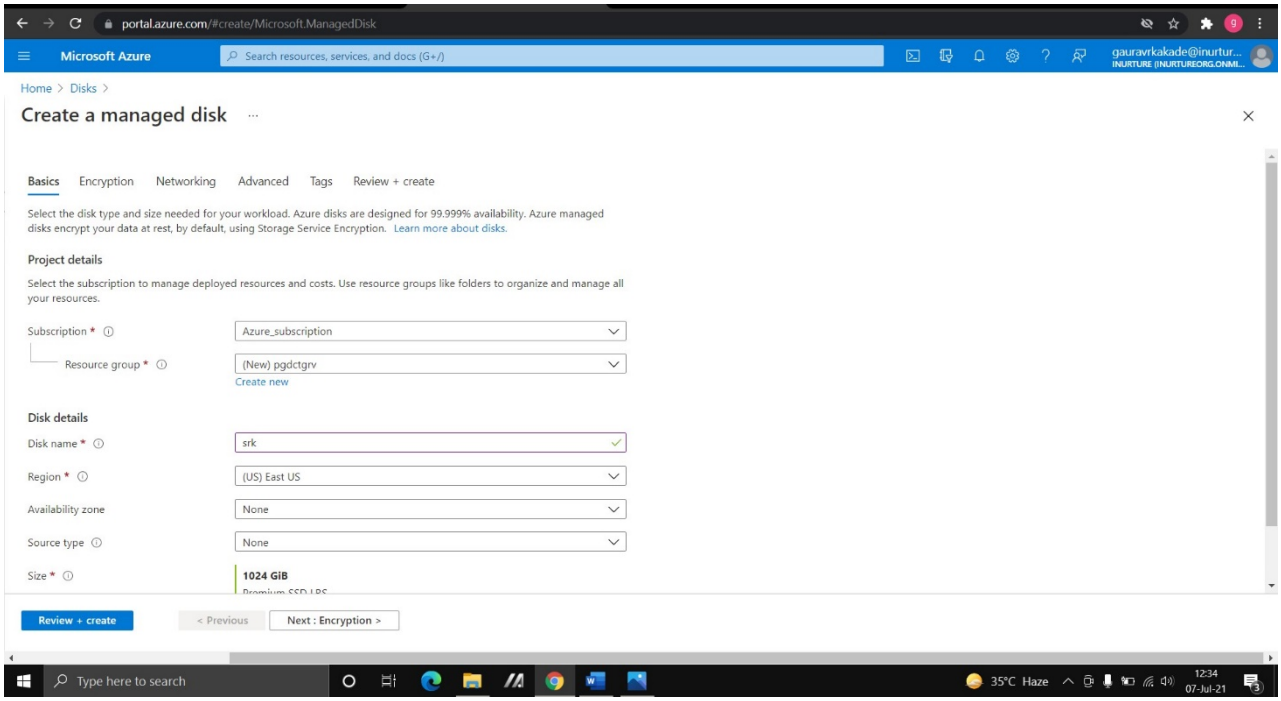


11.Go to file manager you will see the unmanaged disk has been attached.

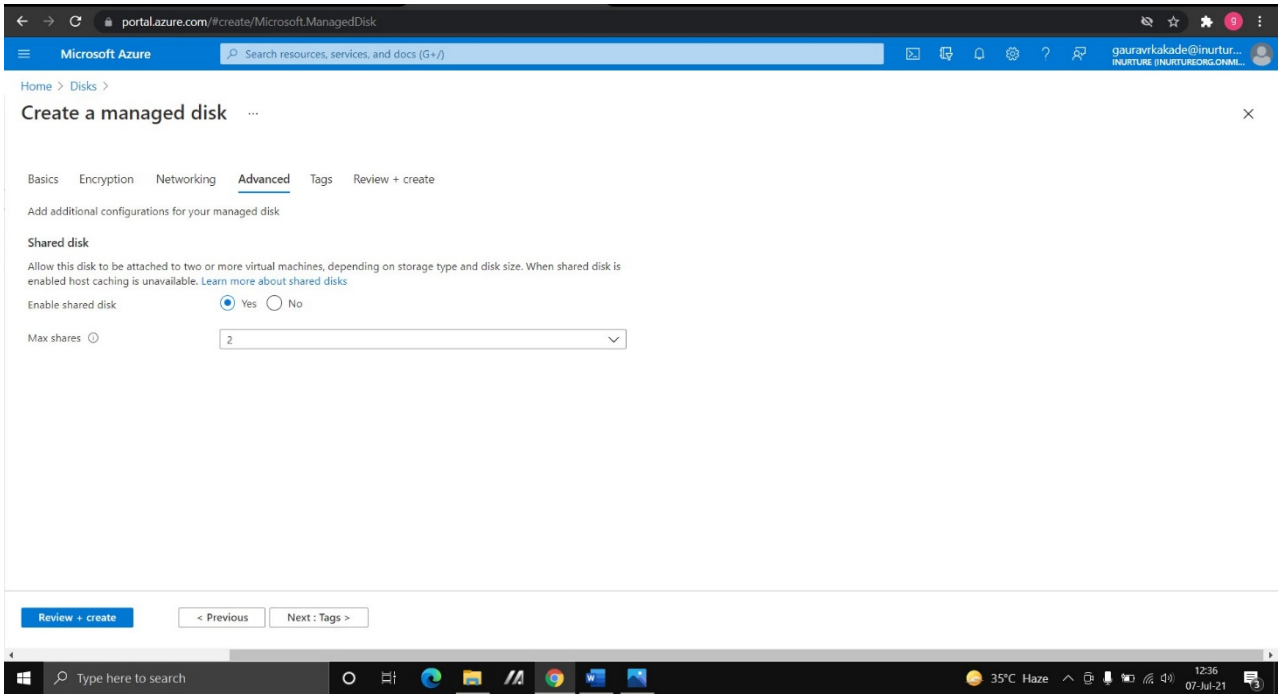


Steps to attached shared disk to virtual machine:

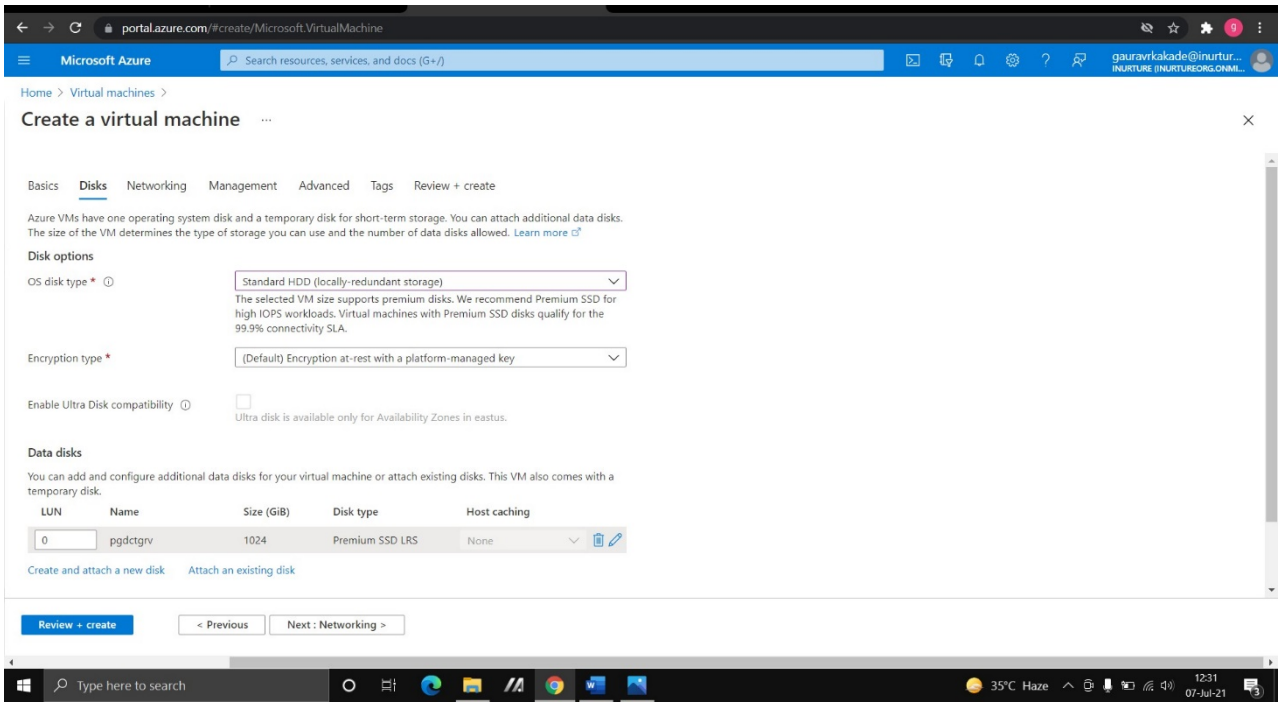
1. Go to disk.
2. Click on create data disk.
3. Choose your resource group and give the name of disk and choose region.



4. Go with default settings.
5. In advanced enable shared disk option and click on next and create disk.

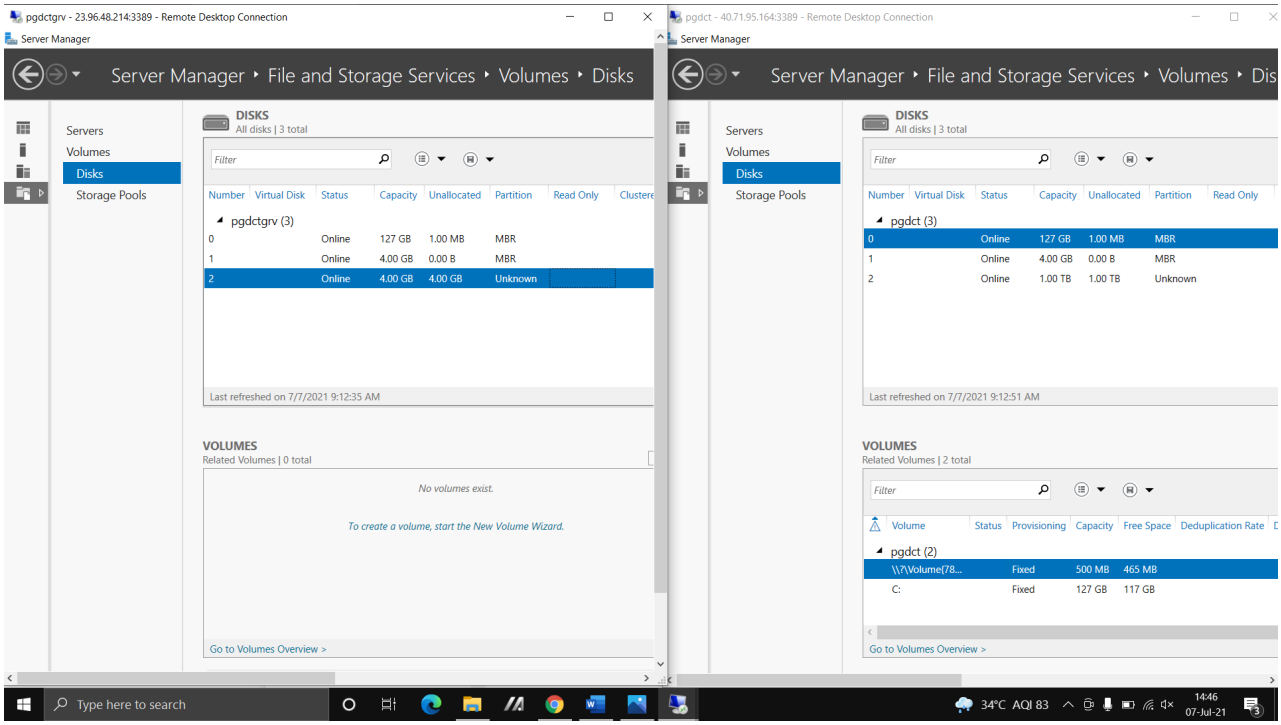


6.Create two virtual machines and in disk option click on attach existing disk.



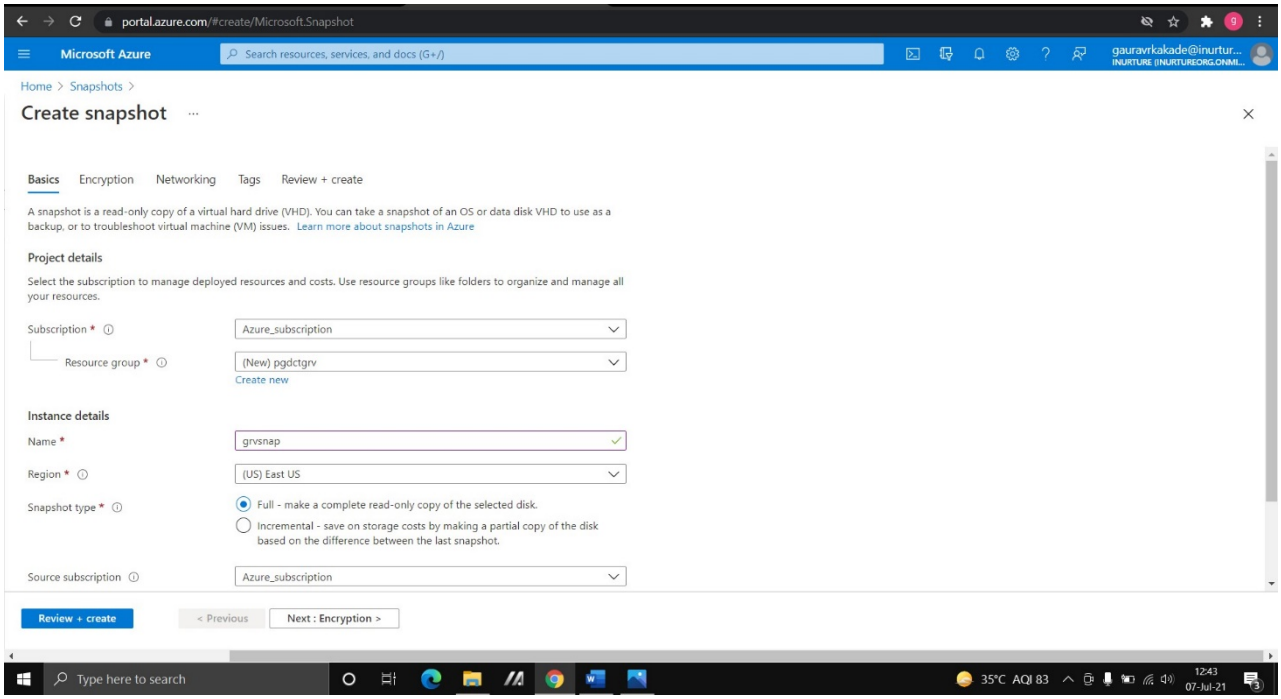
7.Now connect both the virtual machine with RDP.

8.In server manager go to file and storage, then go to disk ,your shared disk is attached to both the virtual machines.

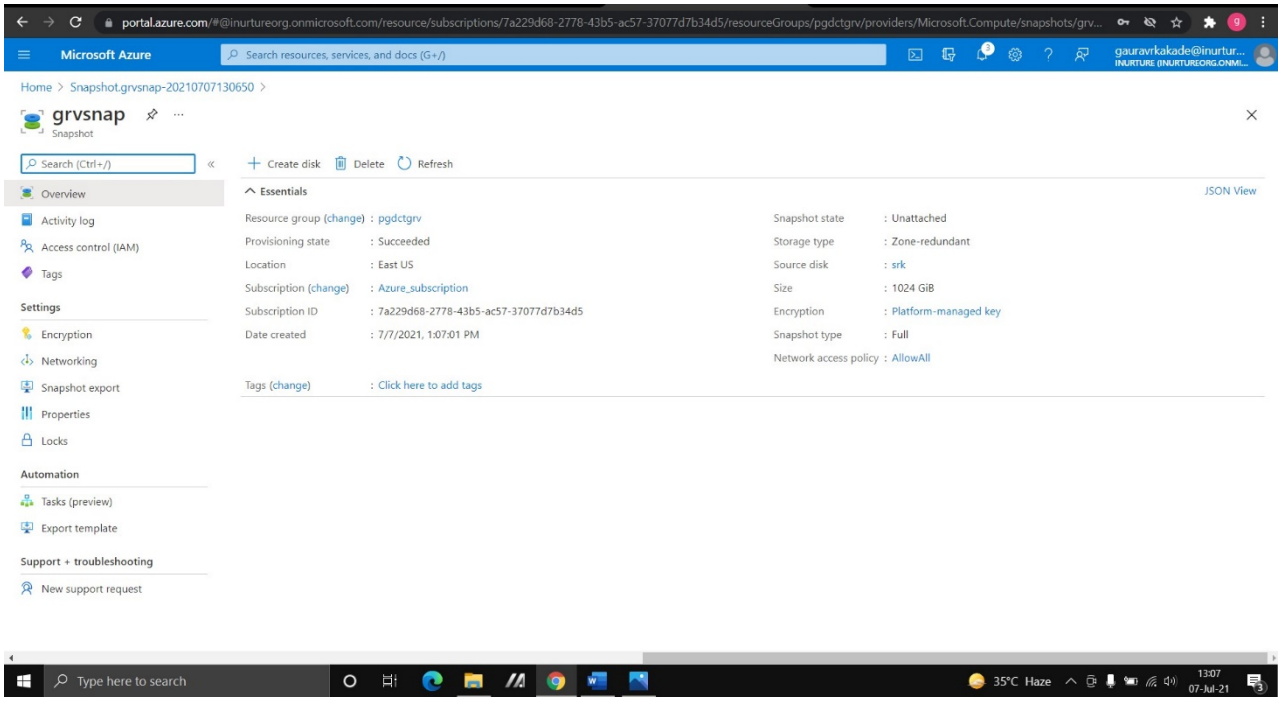


Steps to take snapshot of managed disks:

1. Create a virtual machine.
2. Create a disk.
3. Go to snapshot, enter the name of snapshot and select region & select snapshot type.
4. Select the disk whose snapshot you have to take.
5. Select the storage type as standard HDD.



6. Create snapshot.

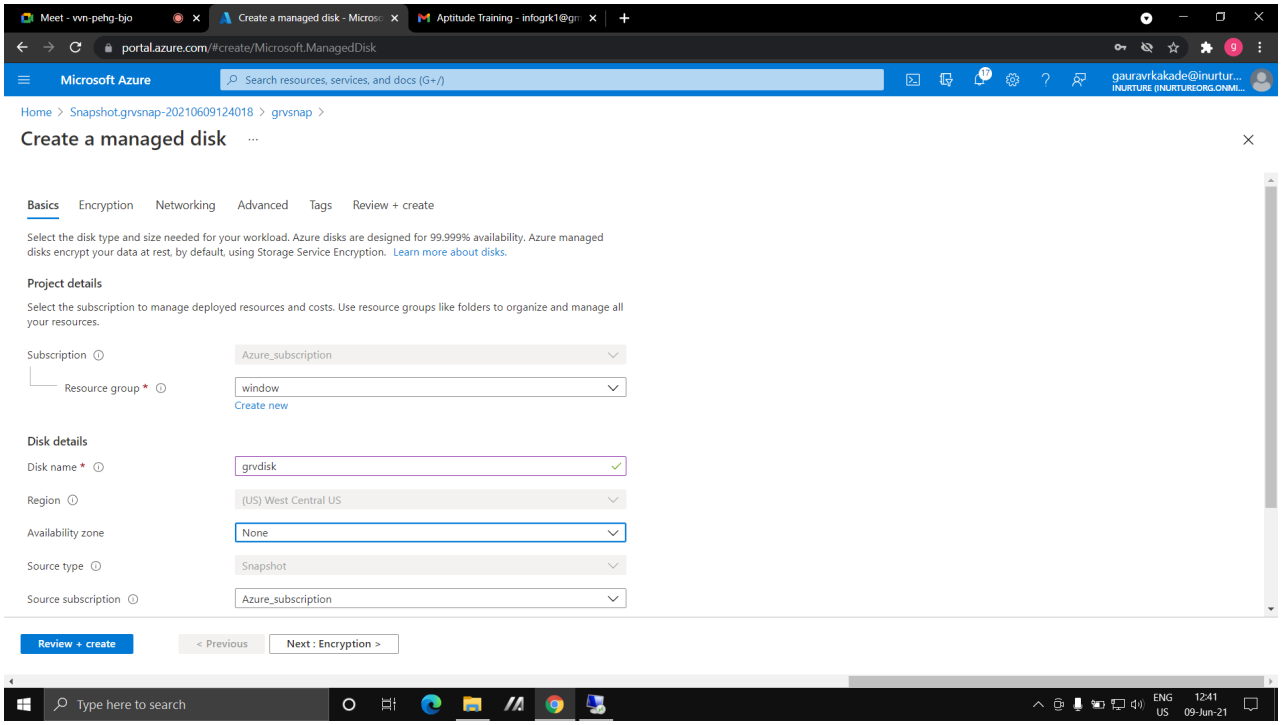


7. Click on the create disk.

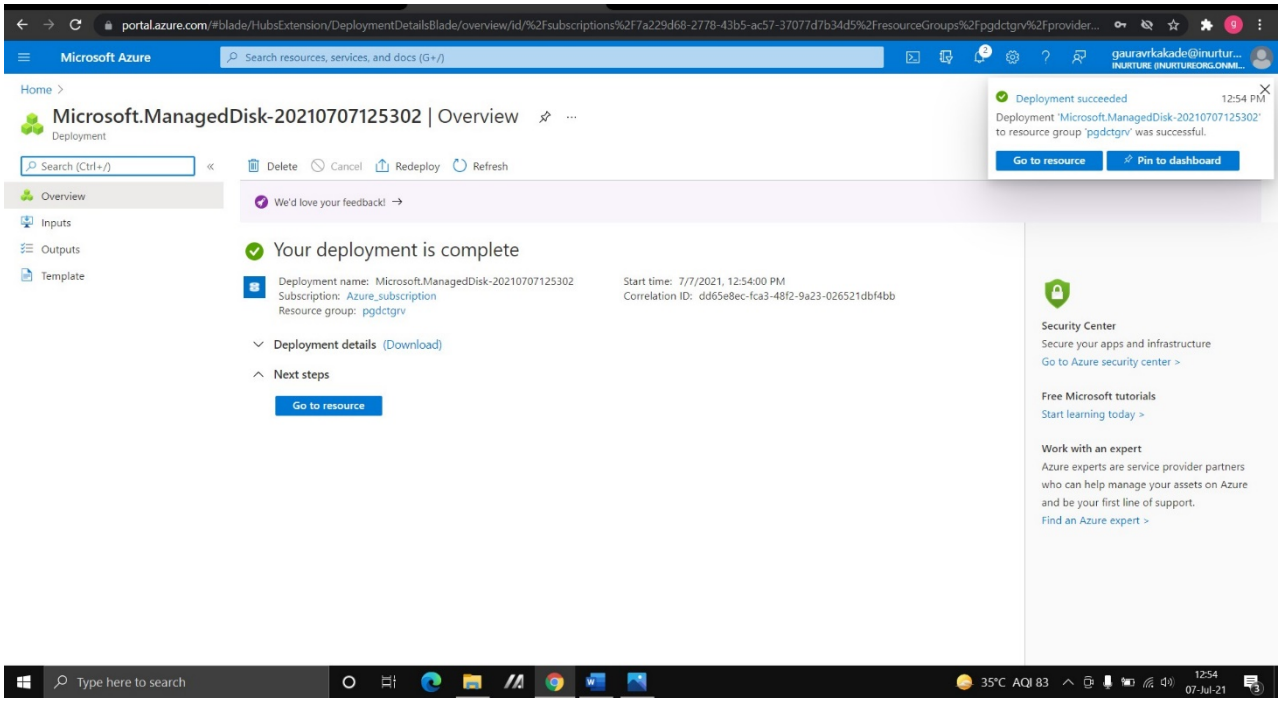
   a. Enter the disk name

   b. Size of the disk

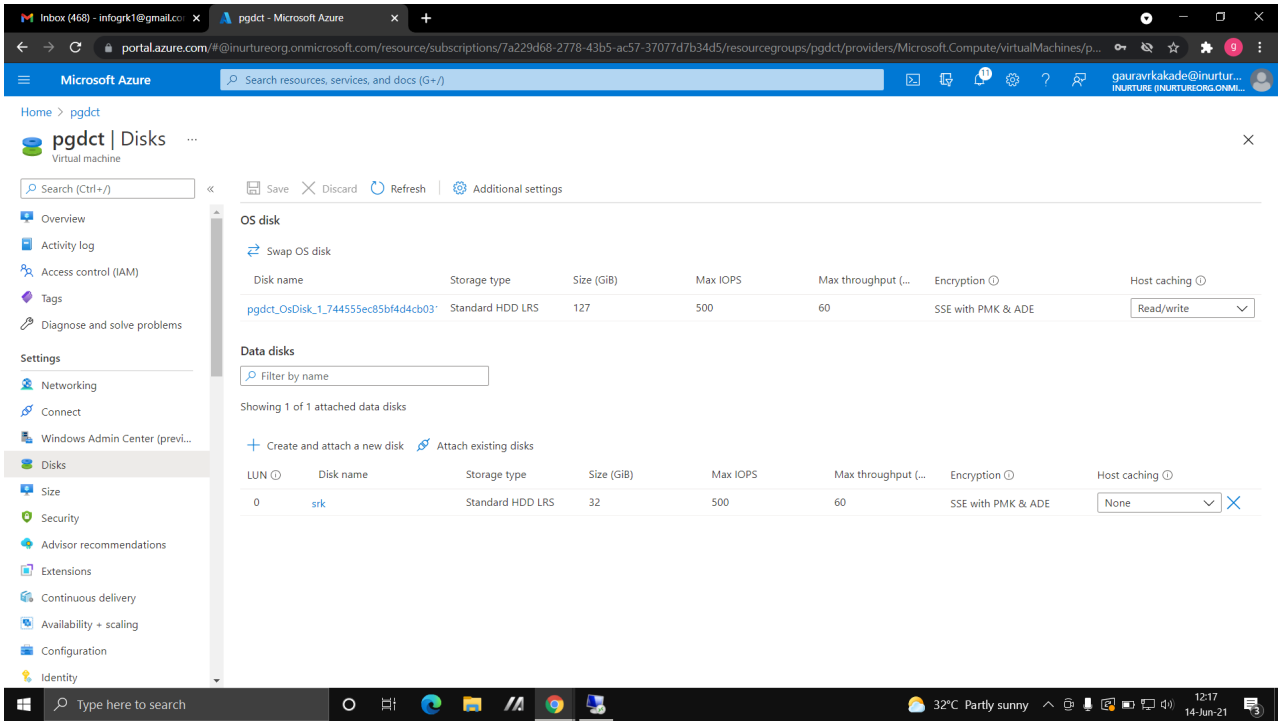
   c. Go to the default settings and create the disk.



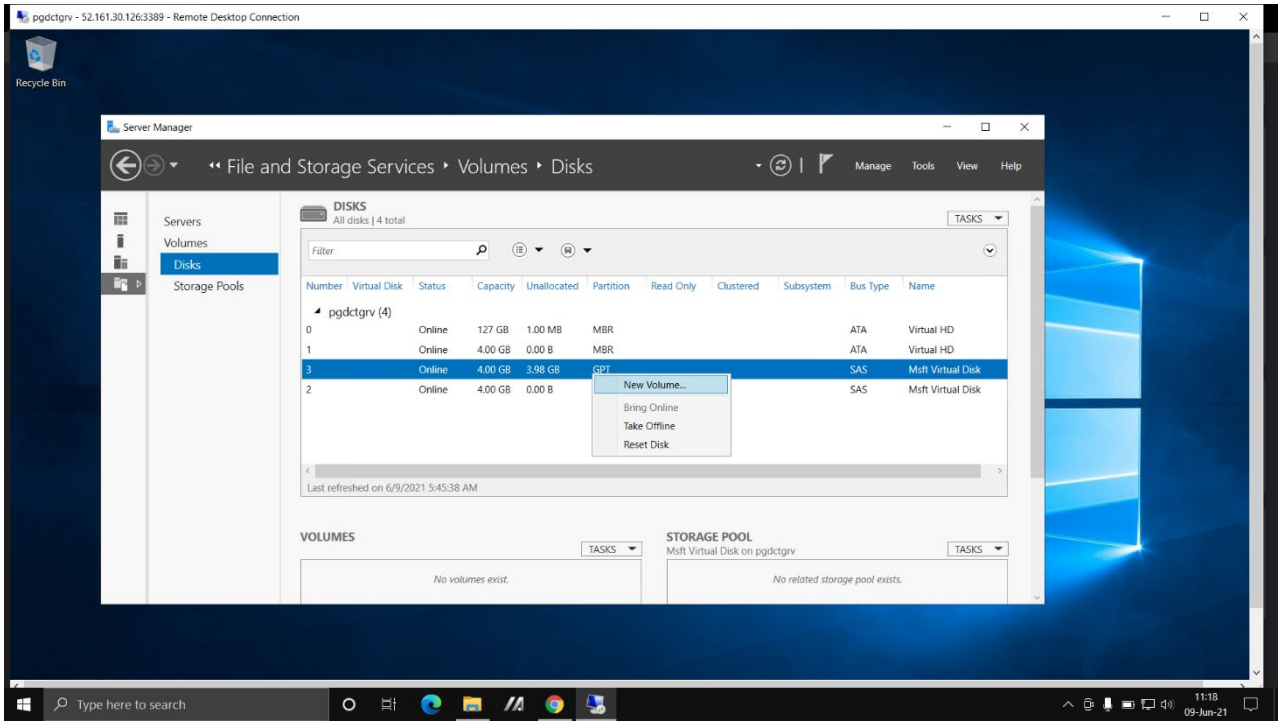
8.Go to virtual machine and create a new virtual machine.



9. Go to disk and click on attached existing disk and attach the disk whose snapshot you have been created.

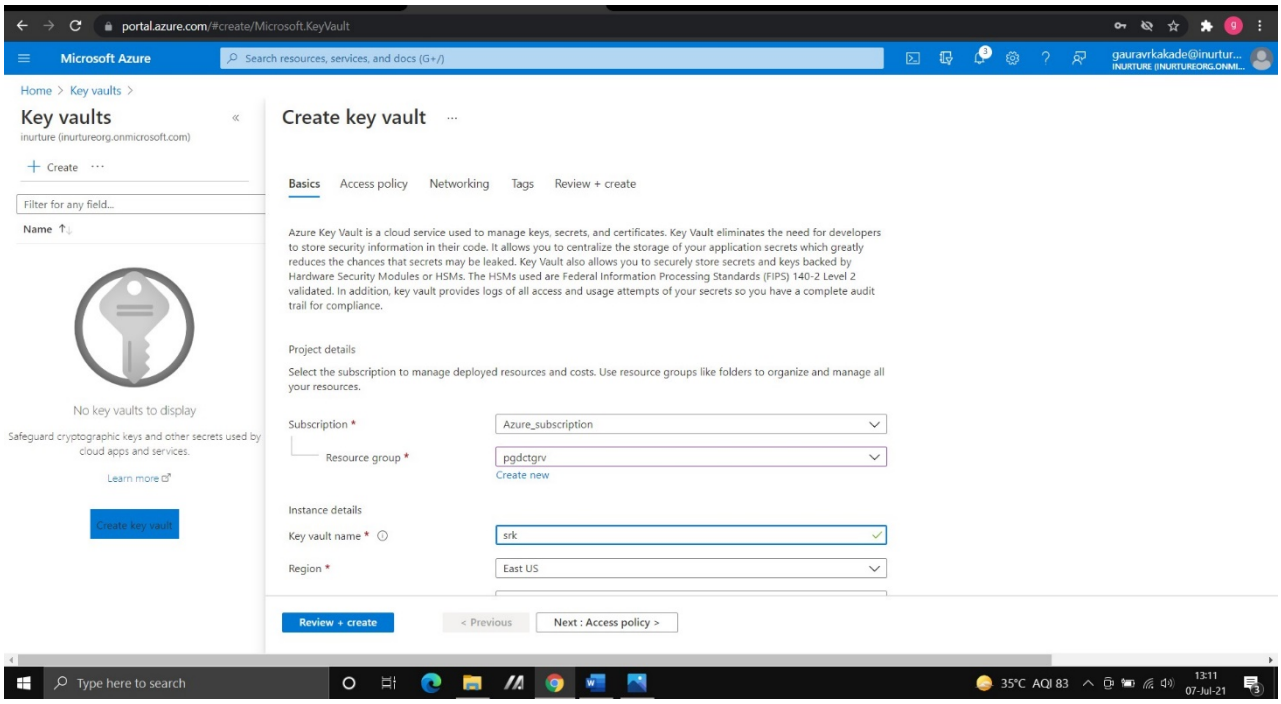


10.Open the virtual machine and in server manager you will see your disk has attached.

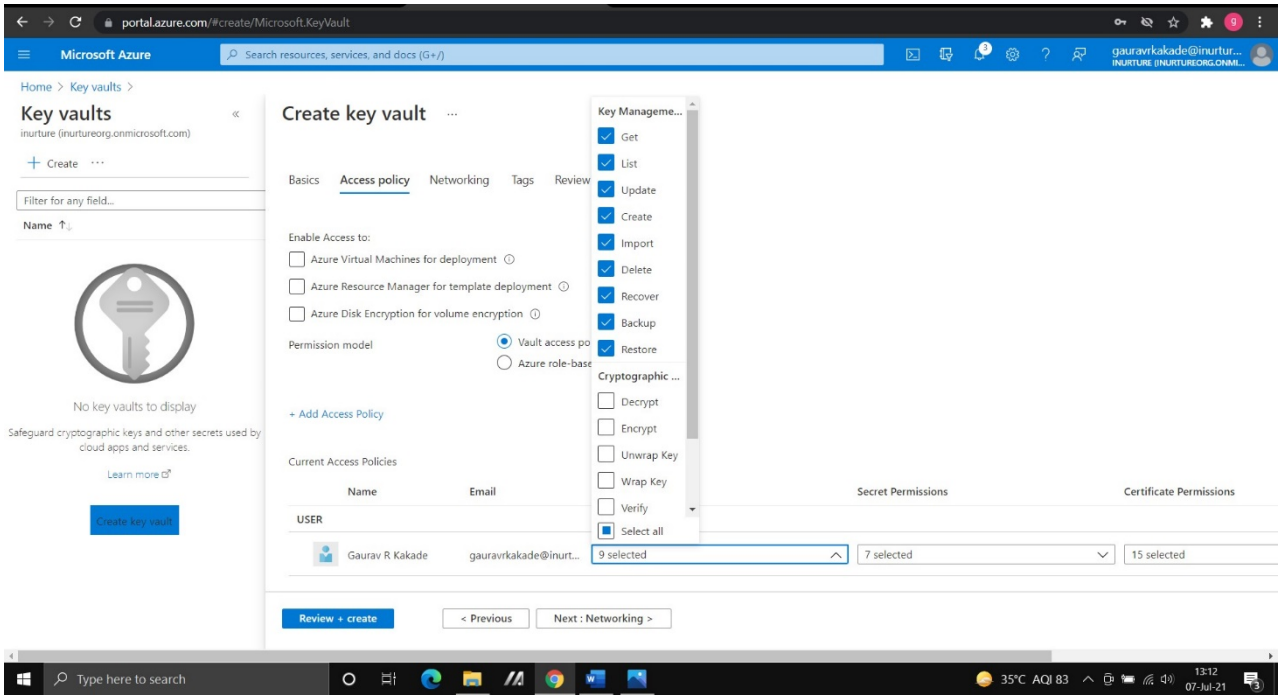


Steps to Encrypt the Disk:

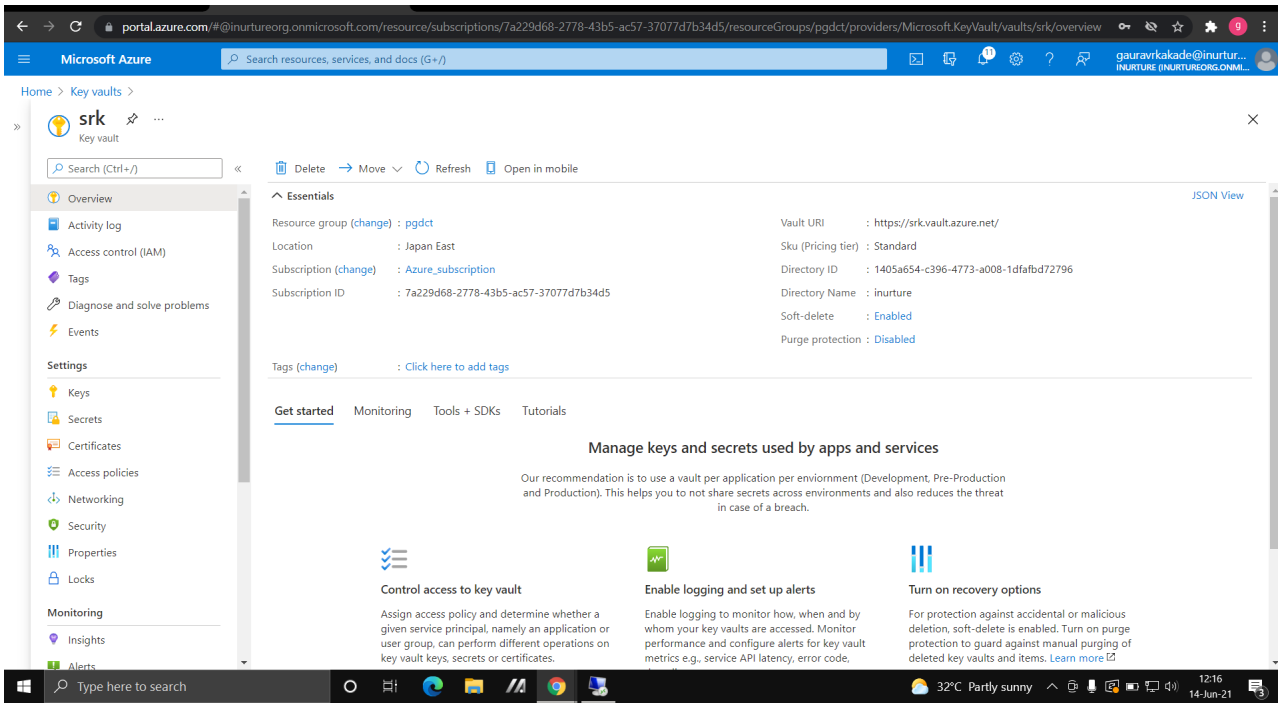
1. Create a Virtual Machine.
2. Connect to the RDP.
3. Go to the Vaults and create key vault.
4. Enter the name of key vault, region and choose pricing tier as standard.



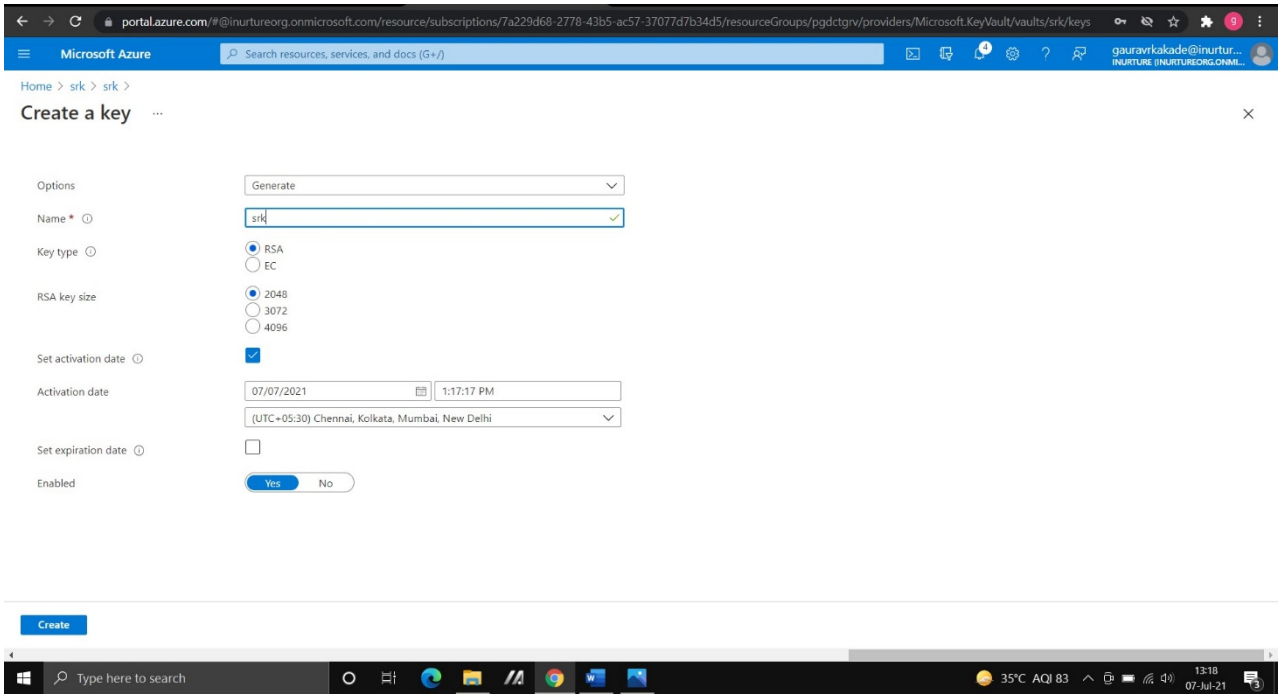
5. In enable access to select disk encryption for volume encryption.
6. In key permission select all the options.



7. Create key vault.
8. Go to keys and click on Generate and Import.



9. Enter the name of the key and select the activation and expiration date.

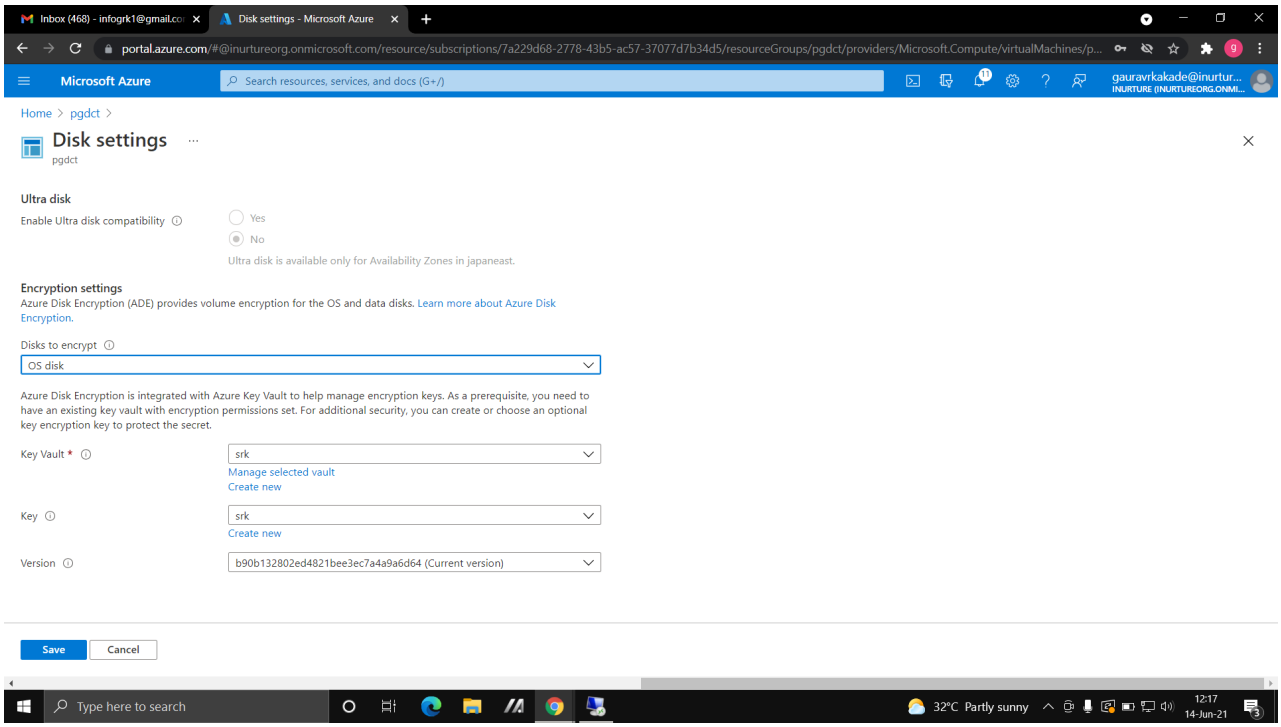


10.Click on create the key.

11.Go to virtual machine, click disk, create a data disk and save it.

12. Go to additional setting in disk.

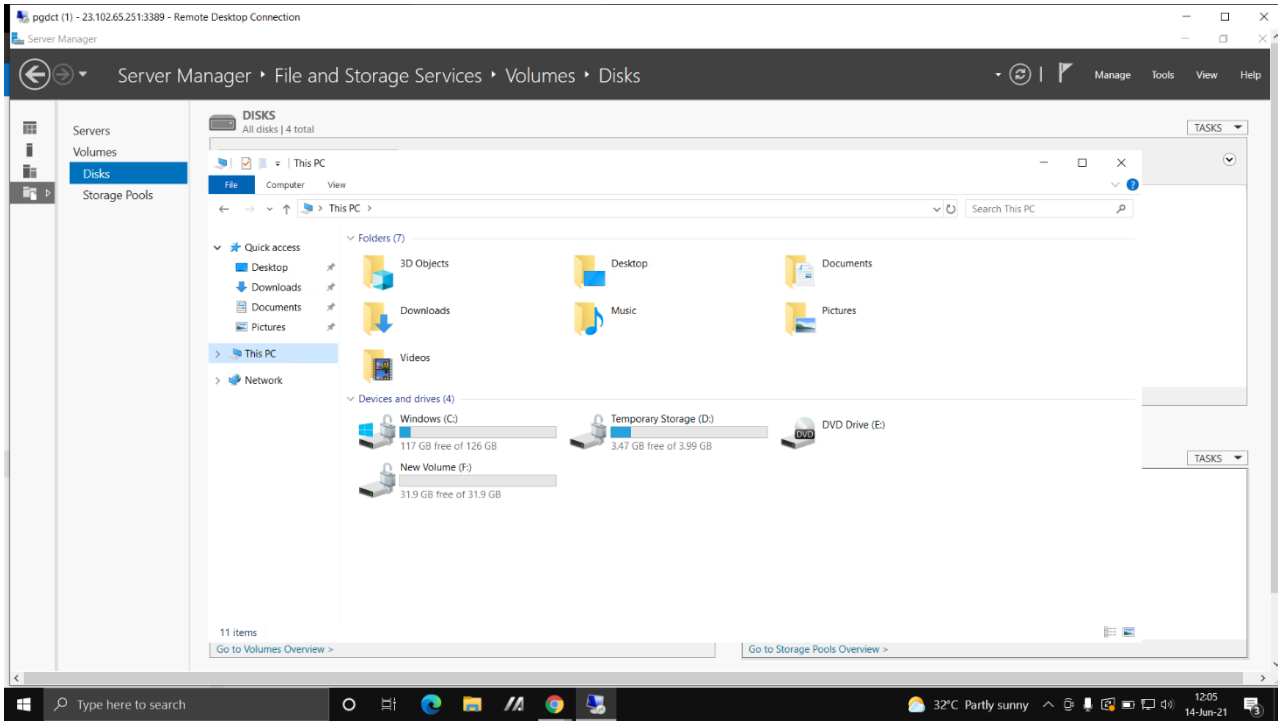
13.In disk to encrypt choose OS and data disk, select the key vault and save it.



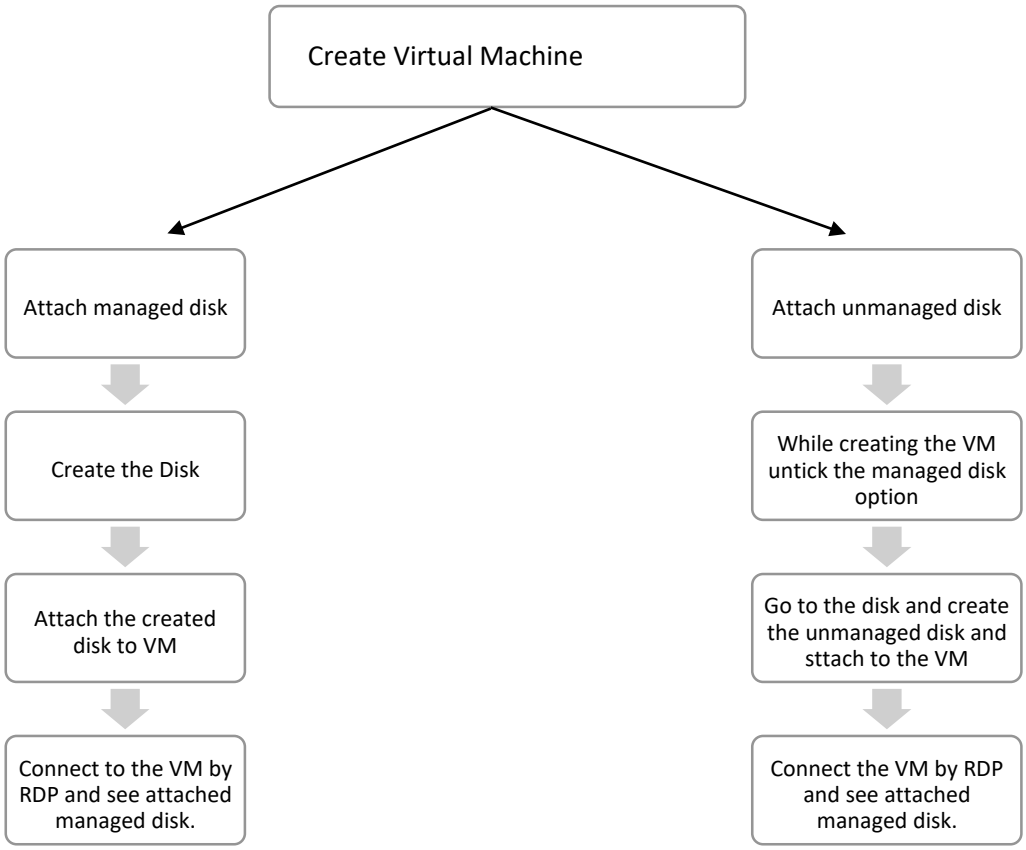
14. Now open the virtual machine and go to server manager, then to file and storage and to Disk.

15. You will see your data disk is attached.

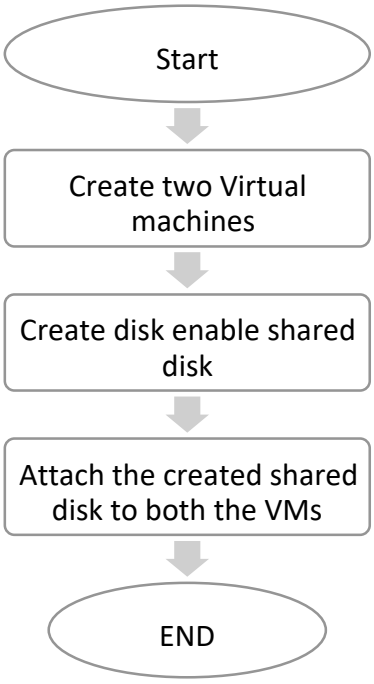
16.Go to file manager and you will see the OS disk and data disk is encrypted.


pgdct (1) - 23.102.65.251:3389 - Remote Desktop Connection
Server Manager
Server Manager ‣ File and Storage Services ‣ Volumes ‣ Disks
Manage
Tools
View
Help
Servers
Volumes
Disks
Storage Pools
DISKS
All disks | 4 total
TASKS
This PC
File
Computer
View
Search This PC
Quick access
Desktop
Downloads
Documents
Pictures
This PC
Network
Folders (7)
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Devices and drives (4)
Windows (C:)
117 GB free of 126 GB
Temporary Storage (D:)
3.47 GB free of 3.99 GB
DVD Drive (E:)
New Volume (F:)
31.9 GB free of 31.9 GB
11 items
Go to Volumes Overview >
Go to Storage Pools Overview >
Type here to search
32°C Partly sunny
12:05
14-Jun-21

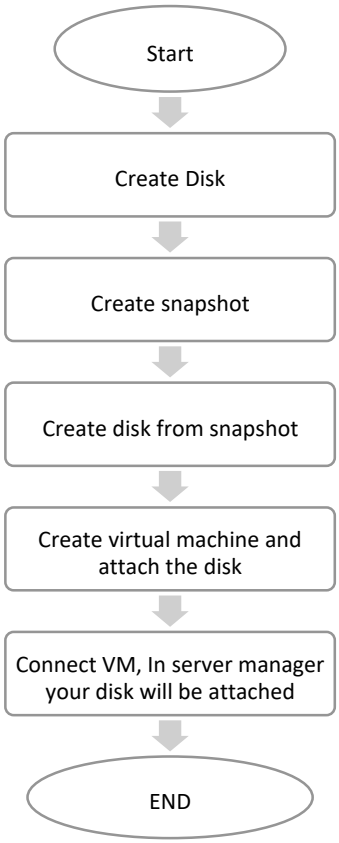
FLOWCHART:



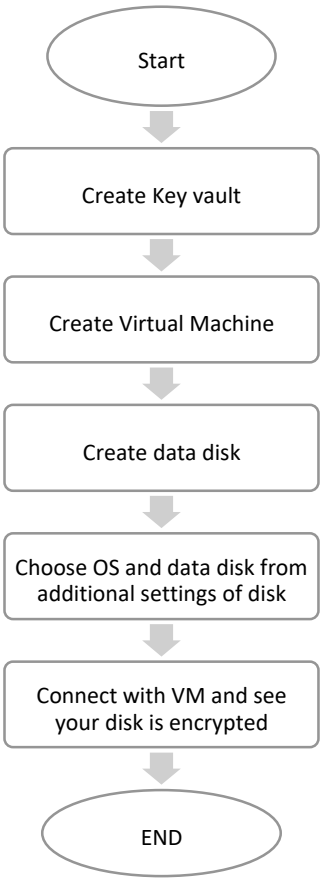
Shared Disk:



Taking Snapshots:



To Encrypt the Disk:



RESULT:

1. We can attach the data managed, unmanaged and shared disk to virtual machine and see the corresponding output by connecting virtual machine through RDP. Then in server manager we can see our disk is attach and we can also initialize the disk and can attach the disk by creating new volume.
2. We can also take snapshot of disk and then can attach the disk to virtual machine.
3. We can also encrypt the disk.

CONCLUSION:

We have successfully attached the data, shared and unmanaged disk to virtual machine and also have successfully take the snapshot of disk and encrypt the disk.

Experiment No. 2

Name of Experiment:
1.Deploying virtual machine in availability sets.

2.Deploying virtual machine in availability zone.

3.Deploying virtual machine in proximity placement group.

Prerequisites: Azure Portal, RDP.

Description:

Availability set: An availability set is a logical grouping of VMs that allows Azure to understand how your application is built to provide for redundancy and availability. Each virtual machine in your availability set is assigned an update domain and a fault domain by the underlying Azure platform. Each availability set can be configured with up to three fault domains and twenty update domains.

Availability zone: An Availability Zone is a high-availability offering that protects your applications and data from datacenter failures. Availability Zones are unique physical locations within an Azure region. Each zone is made up of one or more datacenters equipped with independent power, cooling, and networking.

Proximity placement group: To get VMs as close as possible, achieving the lowest possible latency, you should deploy them within a proximity placement group. A proximity placement group is a logical grouping used to make sure that Azure compute resources are physically located close to each other. Proximity placement groups are useful for workloads where low latency is a requirement.

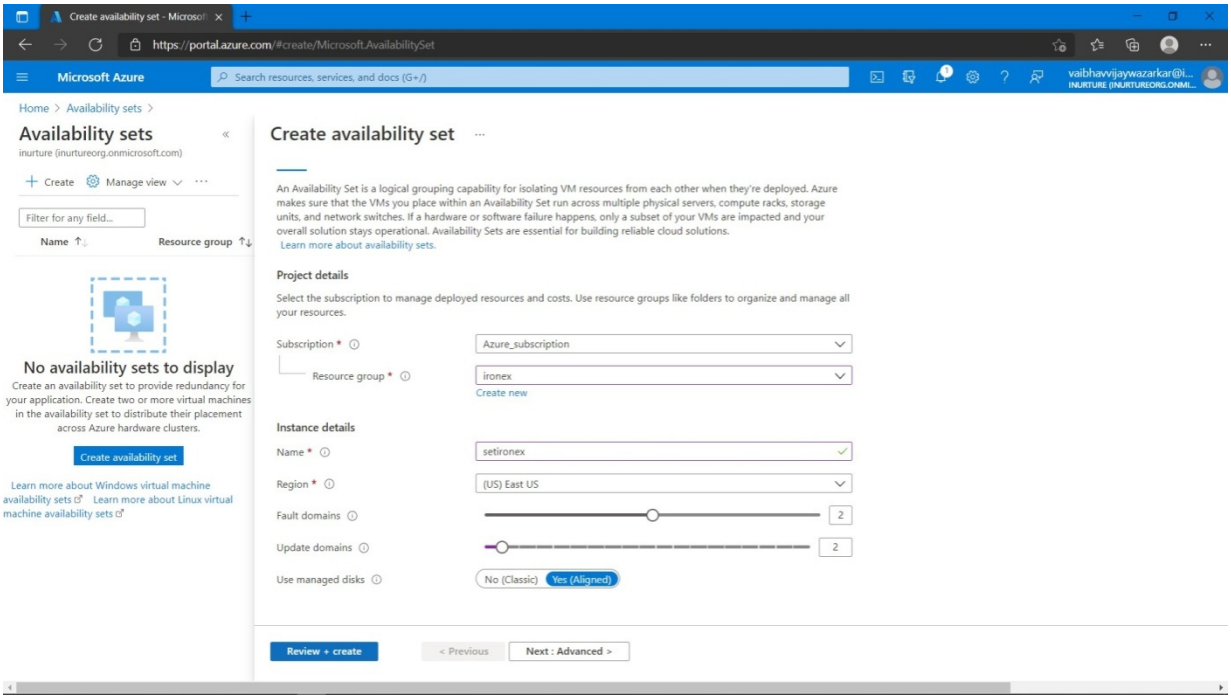
Algorithm:

Steps to deploy virtual machine in availability sets:

1.Go to resource group and create a resource group.

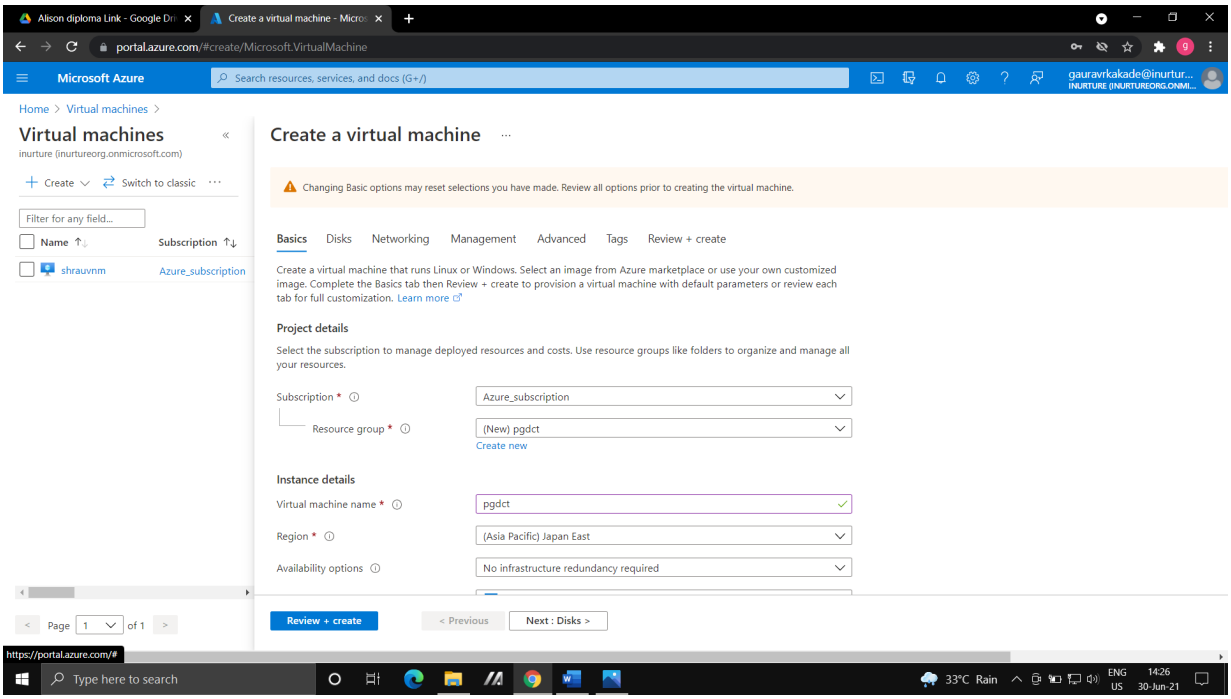
2.Go to availability sets>>create availability sets>>enter the name of availability sets.

3.Select 2 fault domain and 2 update domain.

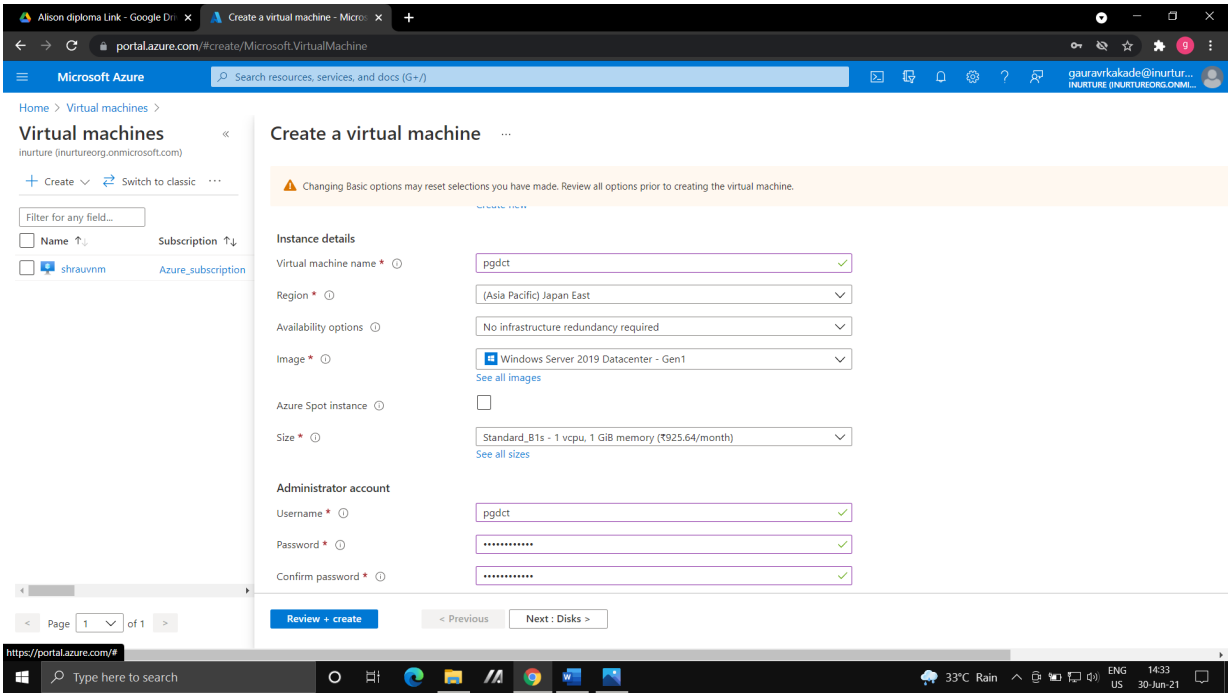


4.Create availability set.

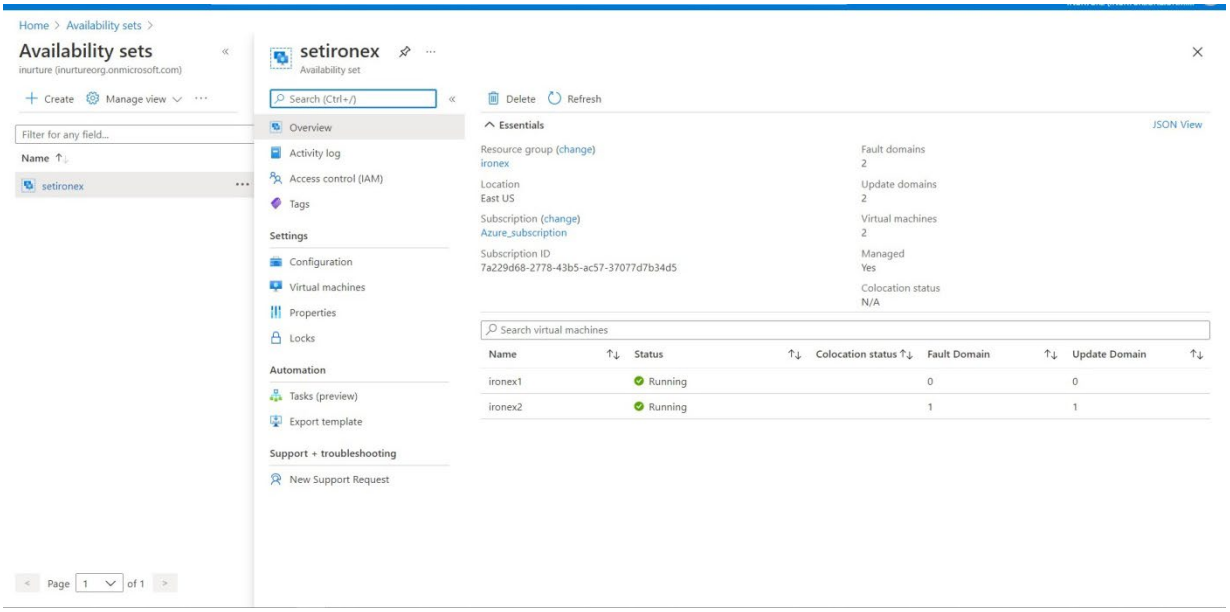
5.Go to virtual machine>>select availability set you have created>>go with the default setting.



6.Create another virtual machine and choose availability set that you have created >>create virtual machine.



7.Go to availability set>>select your availability set>>you will see 2 virtual machine that you have created.

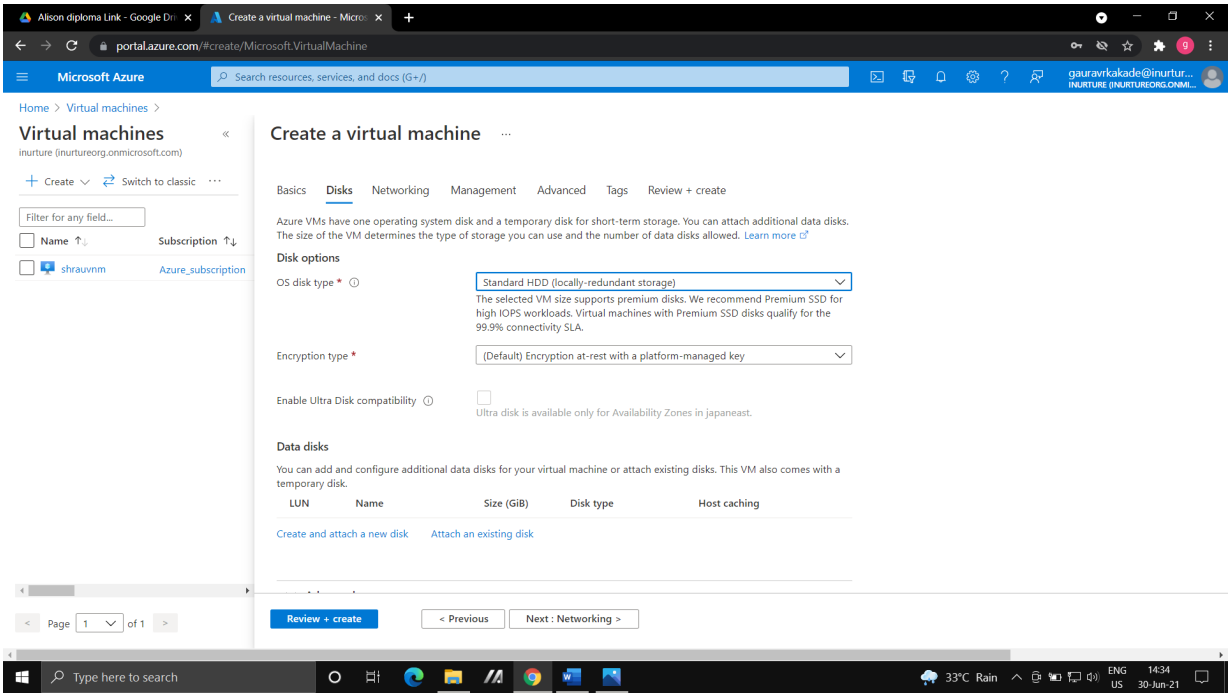


Steps to deploy virtual machine in availability zone:

1.go to virtual machine>>create new virtual machine.

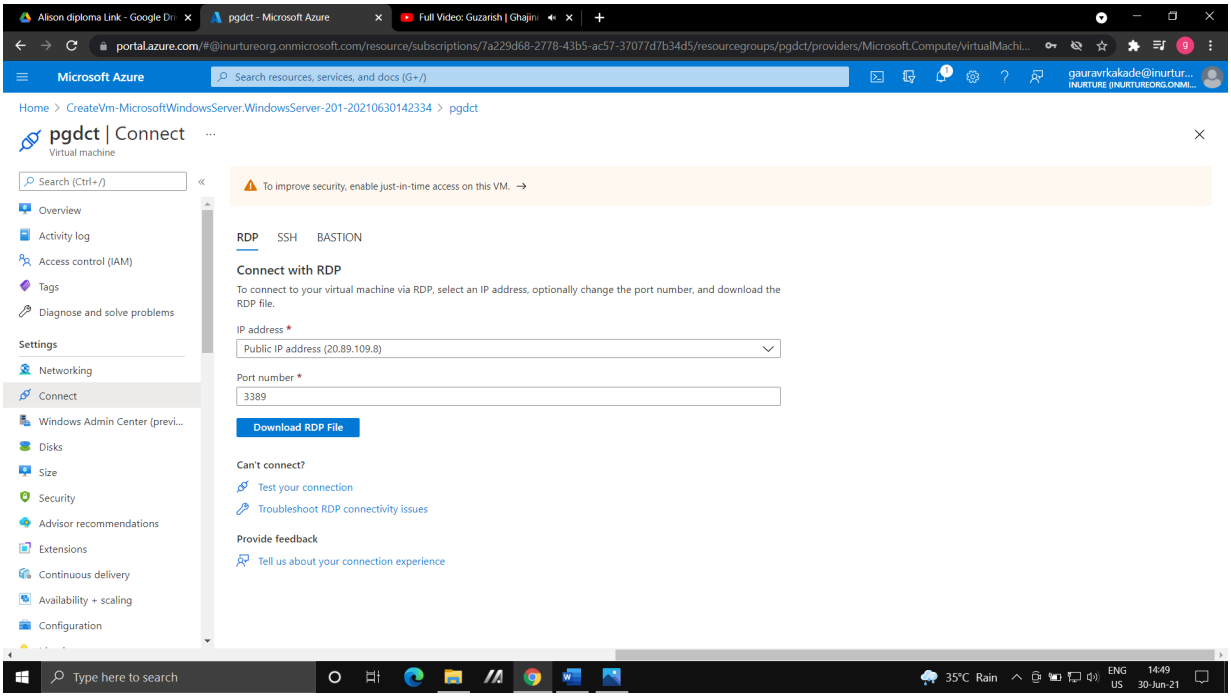
2.Select the region where availability zone is available.

3.In availability options>select availability zone.

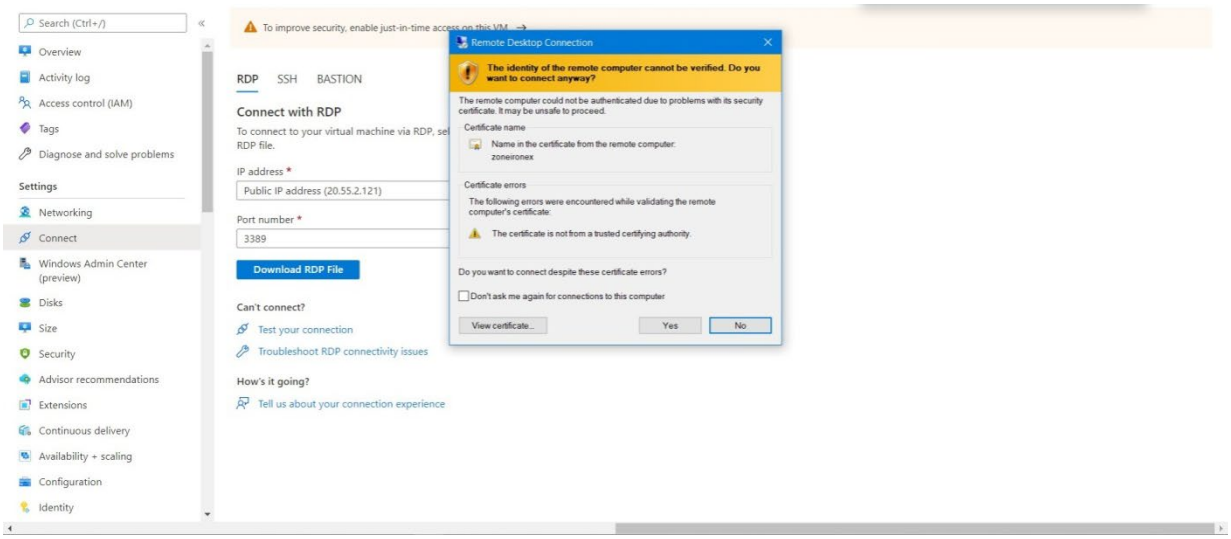


4.Select in which availability zone you want your virtual machine.

5.Keep everything as default and create virtual machine.



6.Try to connect with RDP.

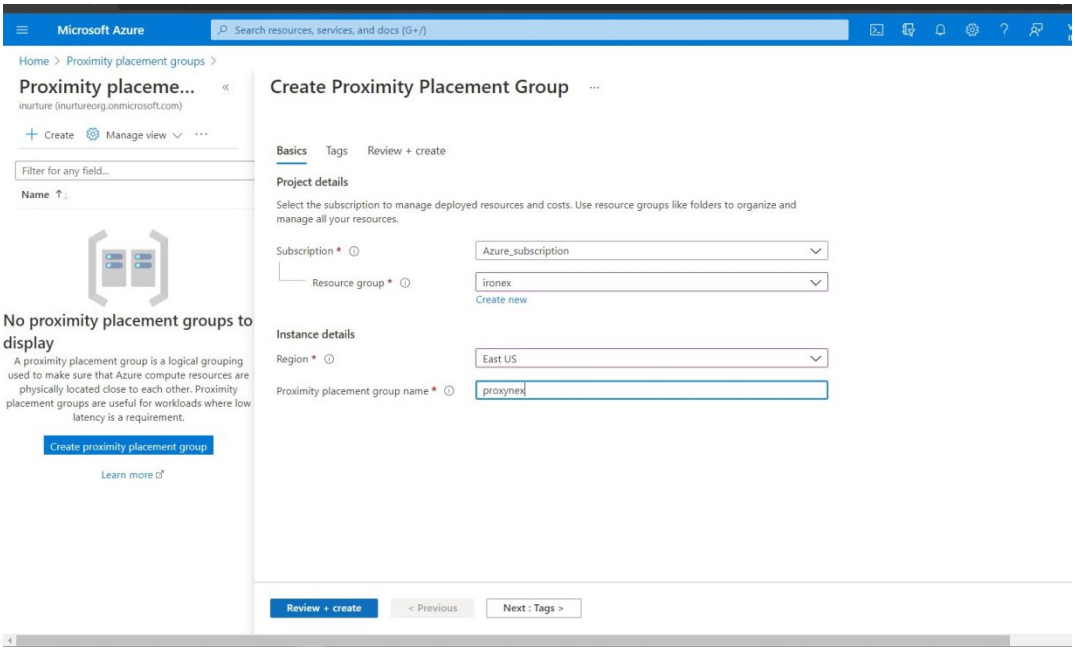


Steps to deploy virtual machine in proximity placement group:

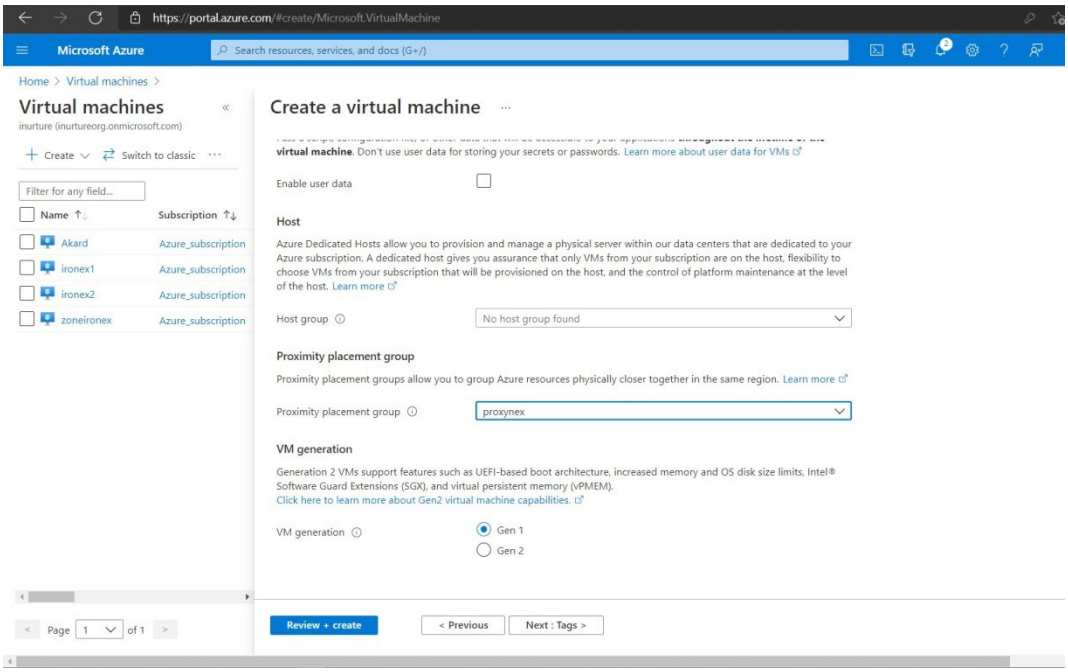
First use case: without availability set deploy virtual machine in proximity placement group.

1.Create resource group.

2.Create proximity group.



3.Go to virtual machine>>In advance select the proximity placement group that you have created.

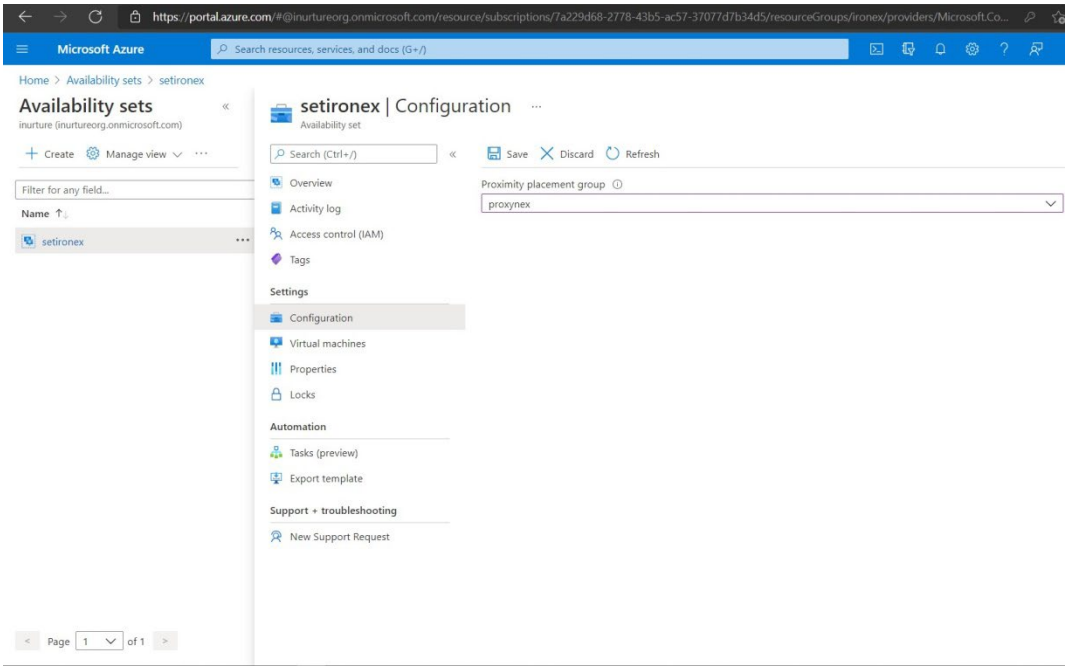


4.Now try to connect the virtual machine with RDP.

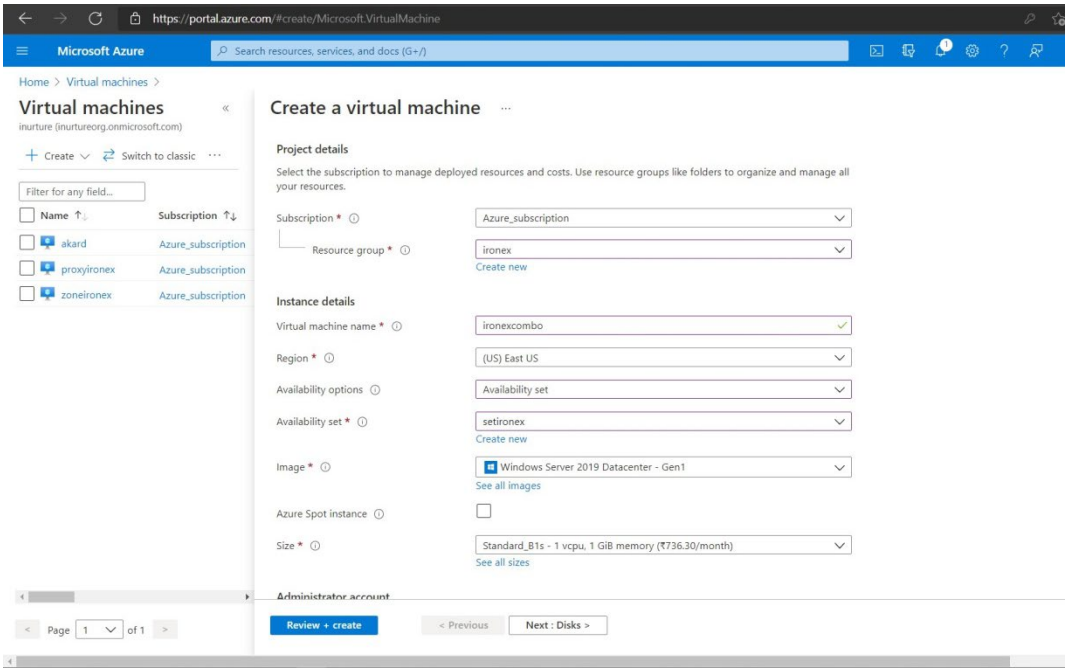
Second Use Case: to check the combination of availability set and proximity placement group.

1.Create availability set.

2.Select your proximity placement group>>create availability set.



3.Go to virtual machine>>choose the availability set that you have created.



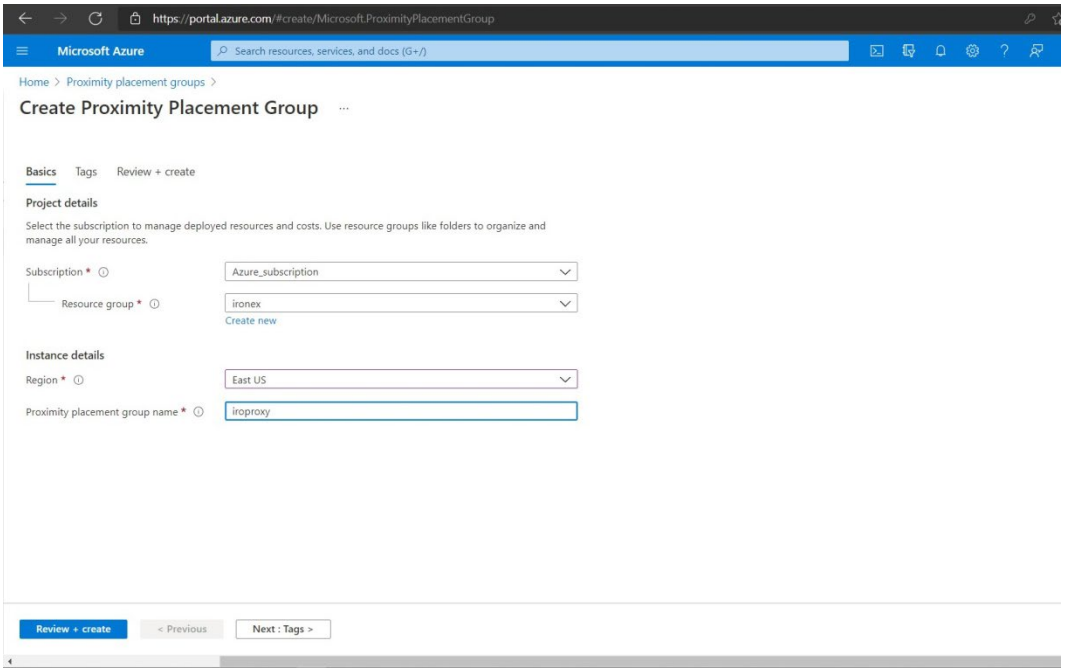
4.Go with default option>>create virtual machine.

5.Try to connect with RDP.

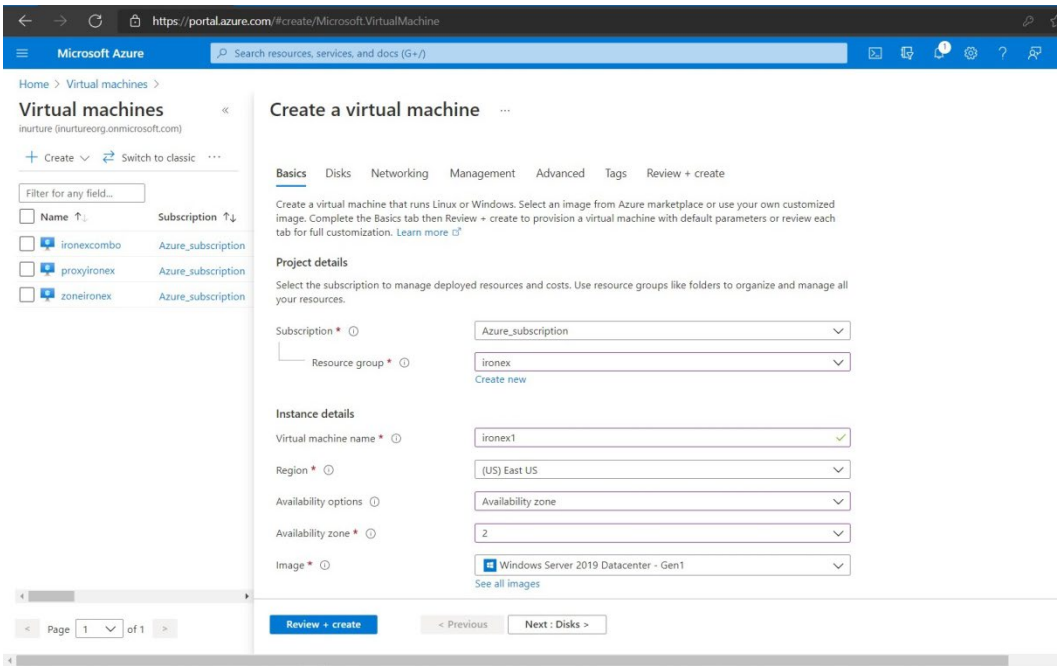
Third Use Case: we will create 2 virtual machine with same availability zone and attach proximity group to 2 virtual machine.

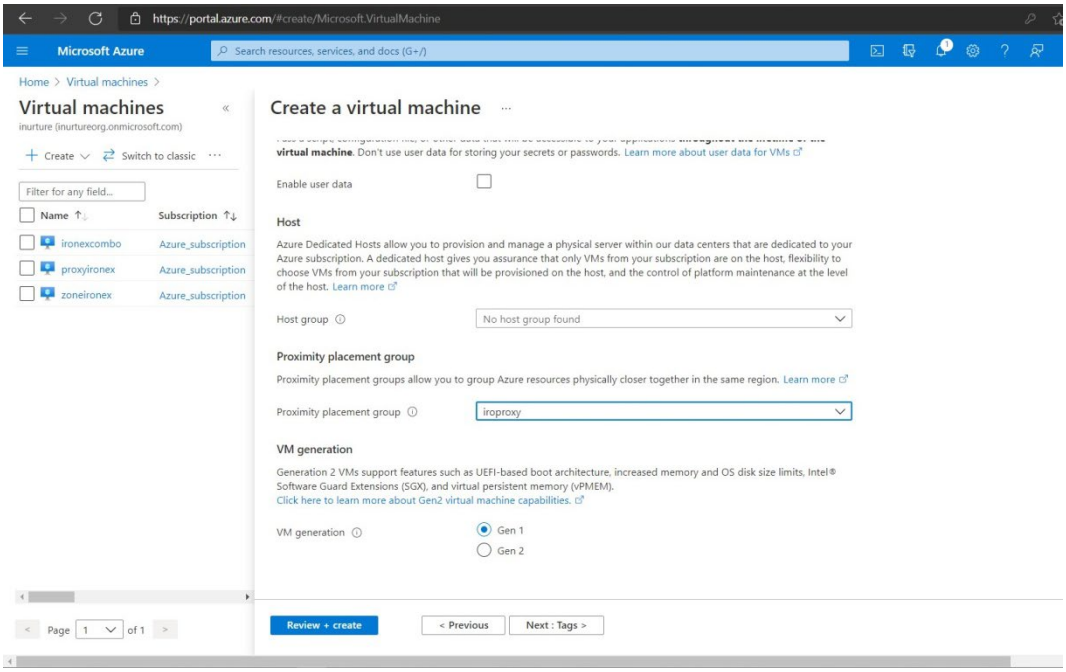
1.Create resource group.

2.Create proximity placement group.

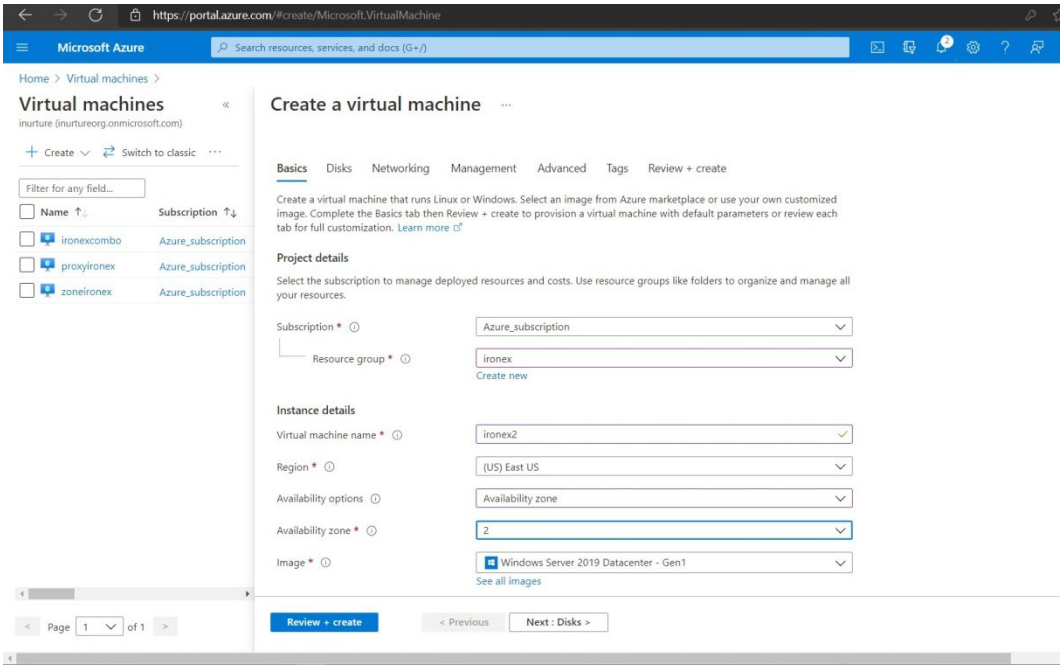


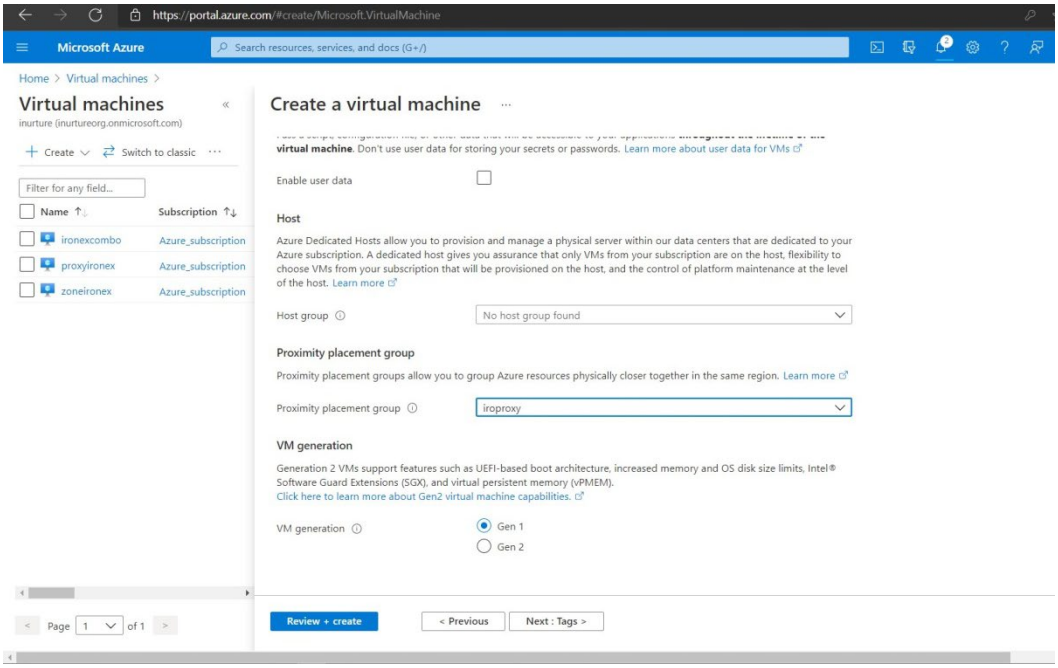
3.Create 2 virtual machine with same availability zone.





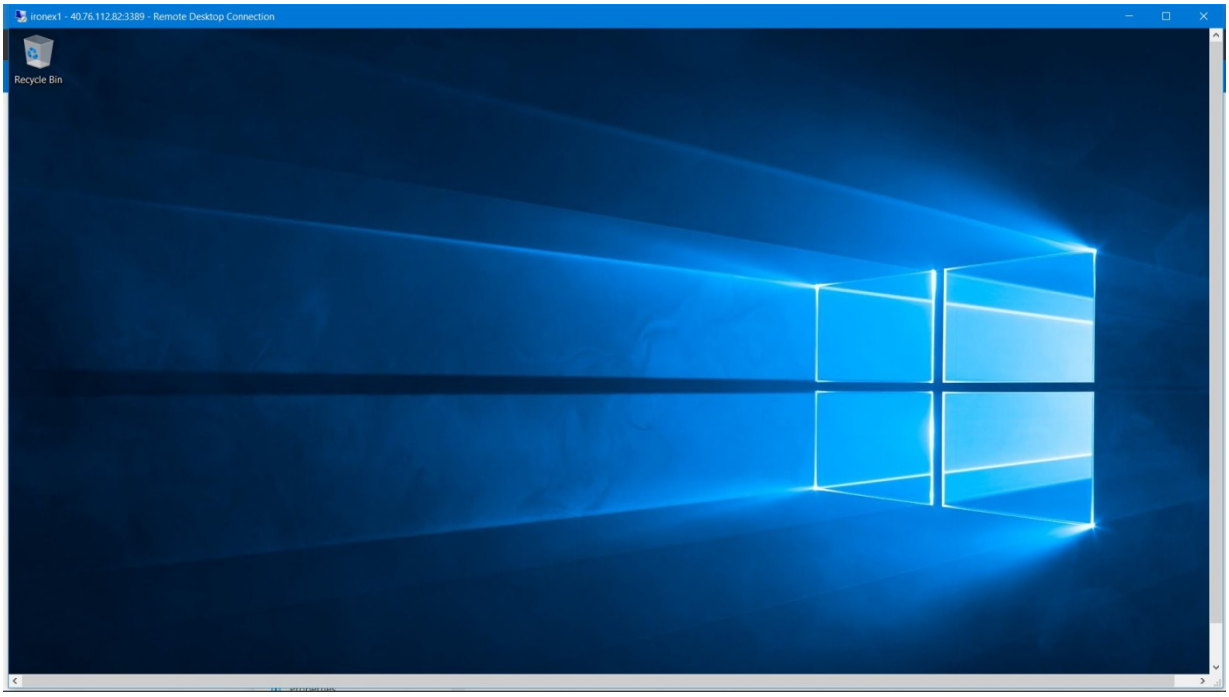
4.In advanced, select the proximity placement group.

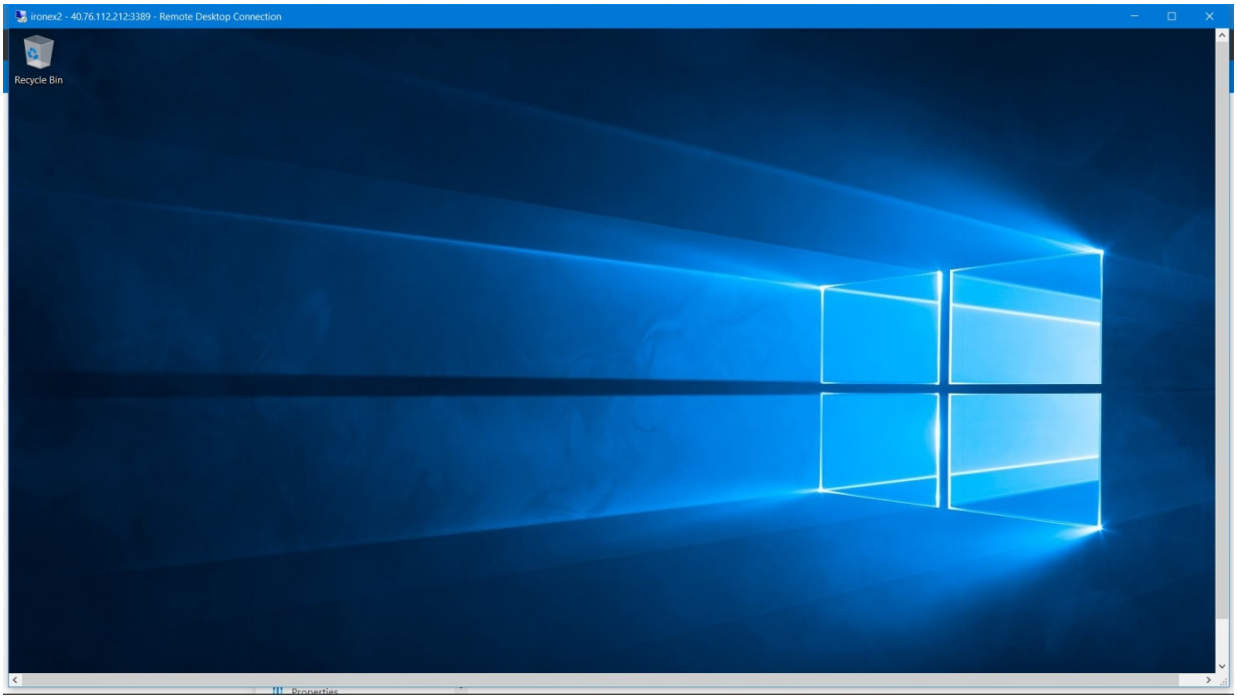




5.Create virtual machines.

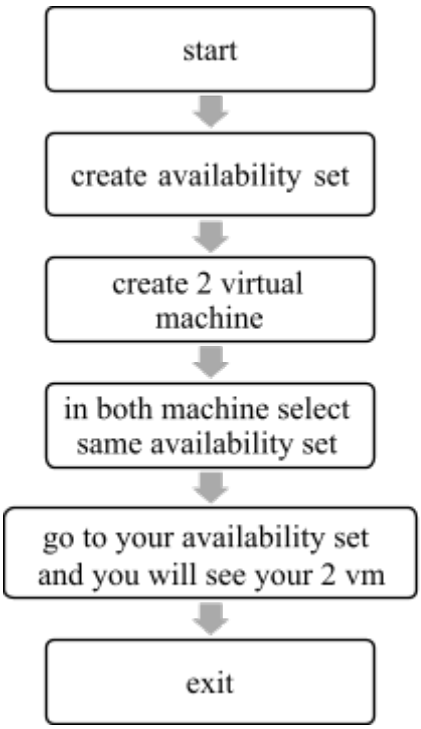
6.Try to connect both virtual machine with RDP.



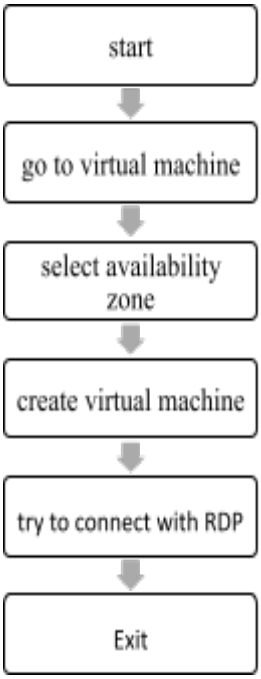


Flowchart:

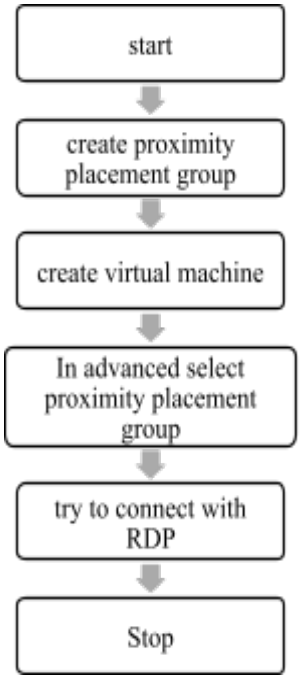
For availability set:

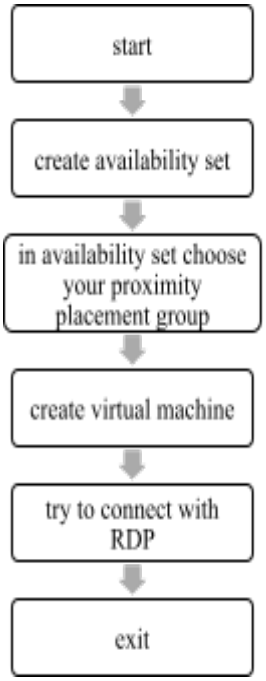


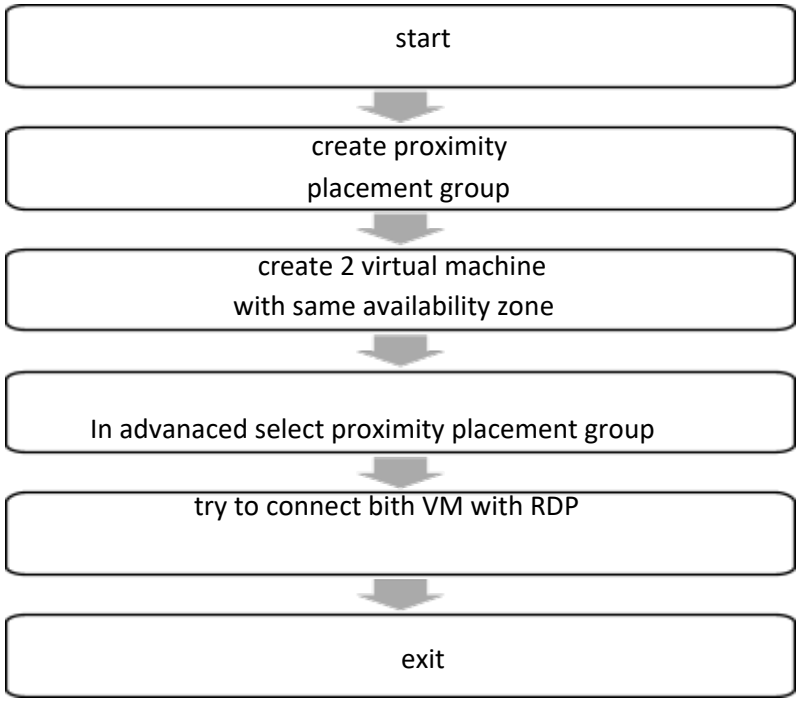
For availability zone:



For proximity placement group:







Result: Different virtual machine has been deployed in availability set, availability zone and proximity placement group. We have also seen the combination of proximity placement group with availability set and availability zone.

Conclusion: we have successfully deployed virtual machines in availability set, availability zone and proximity placement group.

EXPERIMENT NO: 3

AIM: creation of web app and to see app service logs.

PREREQUISITES: Azure portal, visual studio.

DESCRIPTION:

Azure web app:

- Azure Web Apps provides a platform to build an App in Azure without having to deploy, configure and maintain your own Azure VM's.
- You can build Web App using the ASP.NET, PHP, Node.js and Python.
- They also integrate common development environments which could be Visual Studio and GitHub.
- Azure Web App provides a host service that developers can use it to develop mobile or web app. Apart from this the developer can use to build API apps or Logic apps, which provides integration with SaaS.

ALGORITHM:

Steps to create web app:

1.Go to app service plan.

2.Give the name of plan.

3.Select the pricing tier as free f1>>create app service plan.

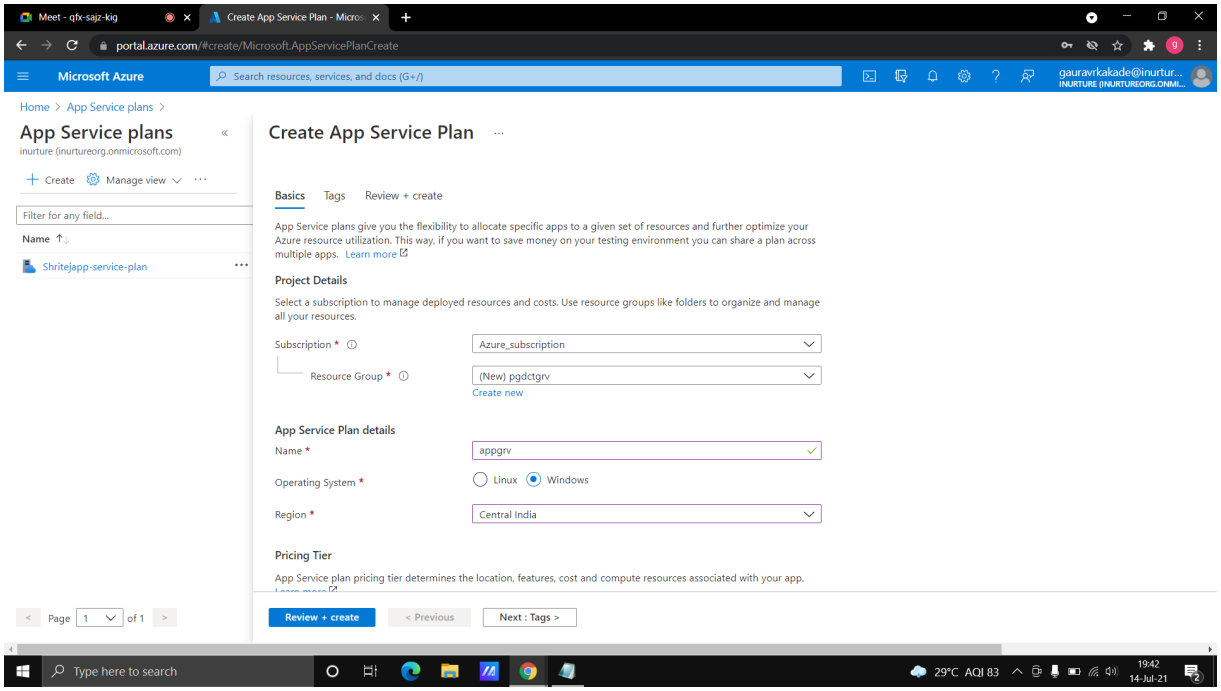


Fig: create app service plan

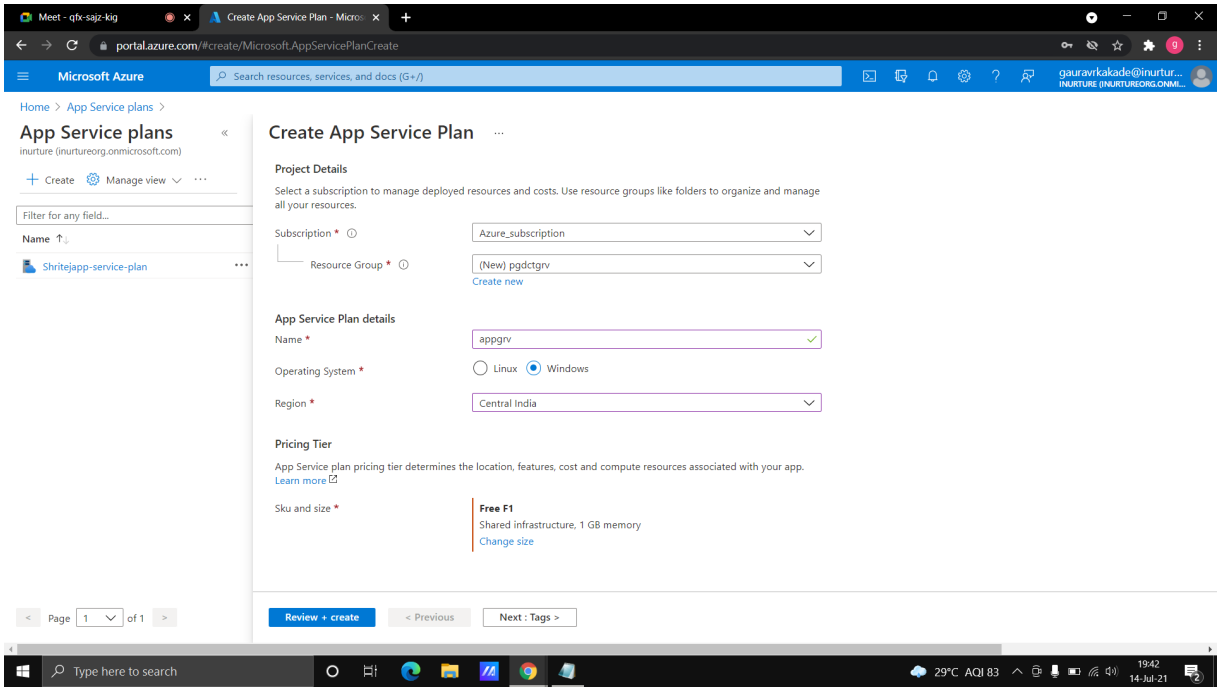


Fig: select size

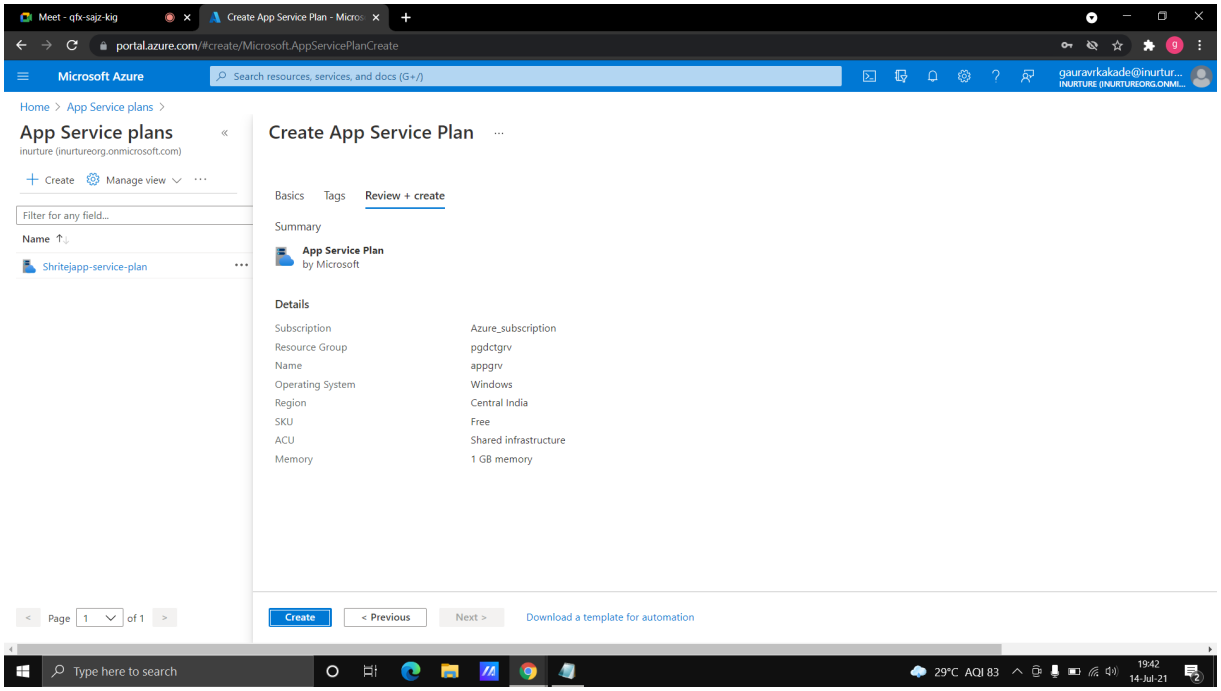


Fig: create app service plan

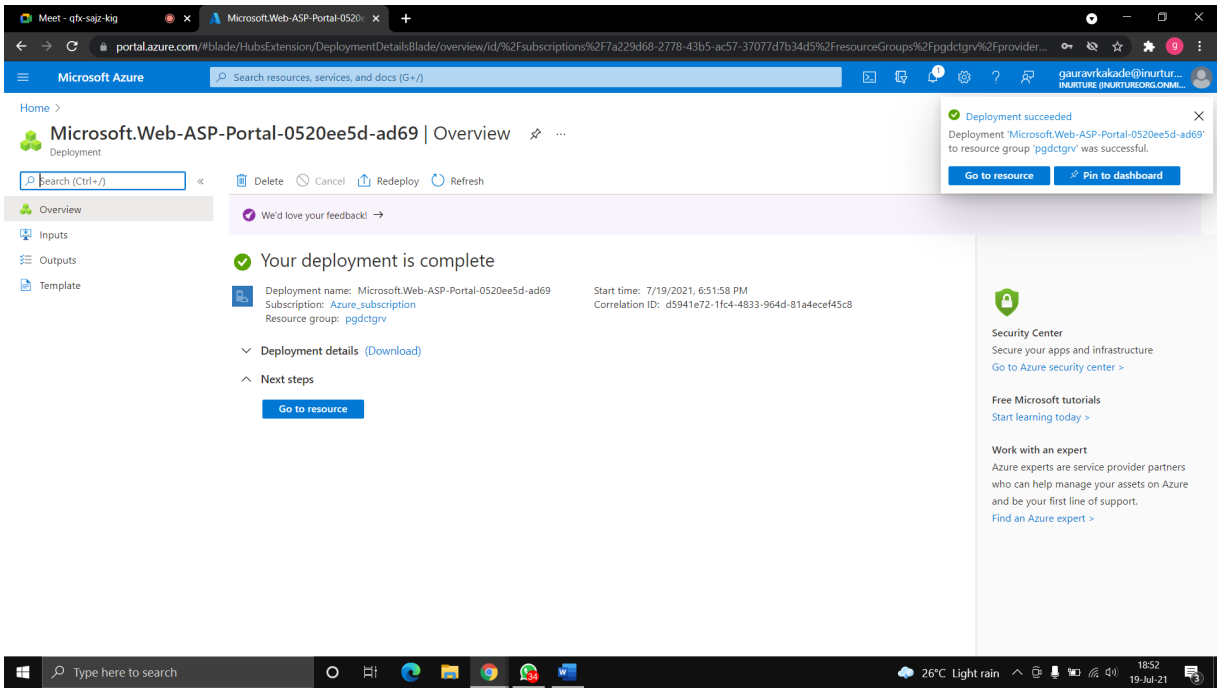


Fig: created app service plan

4.Go to web app.

5.Give the name of web app>>select the runtime stack as .NET core 3.1(LTS).

6.In windows plan select your windows plan.

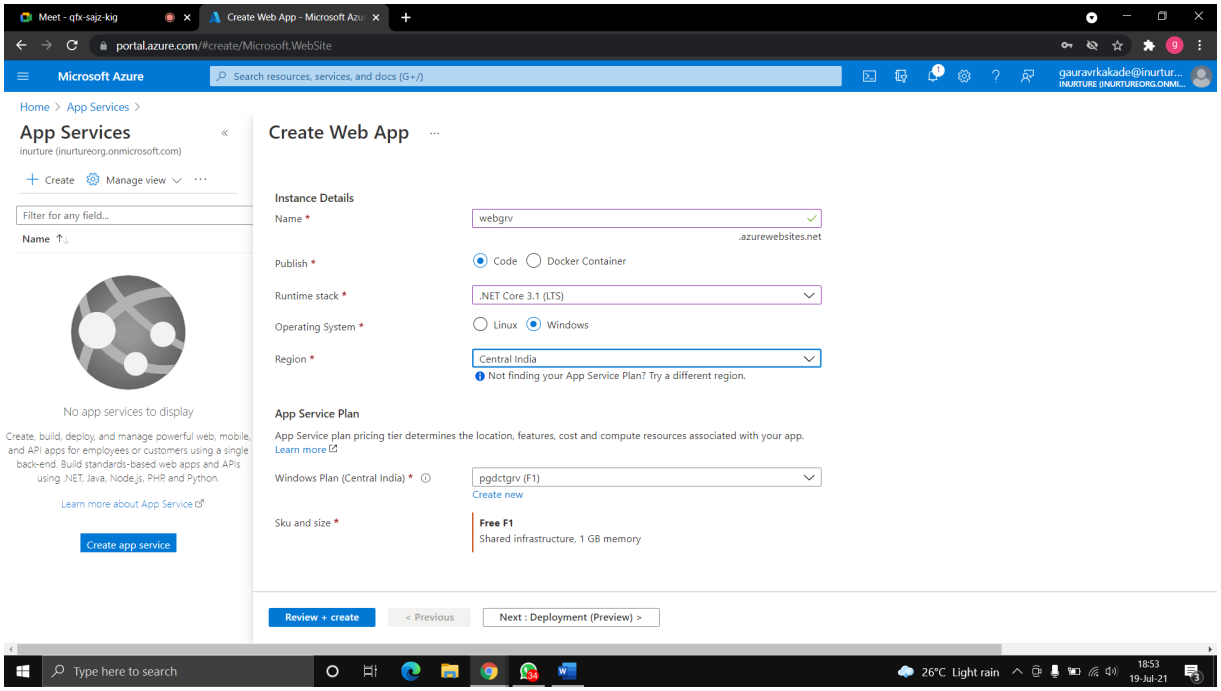


Fig: create web app

7.Keep the deployment setting as disable.

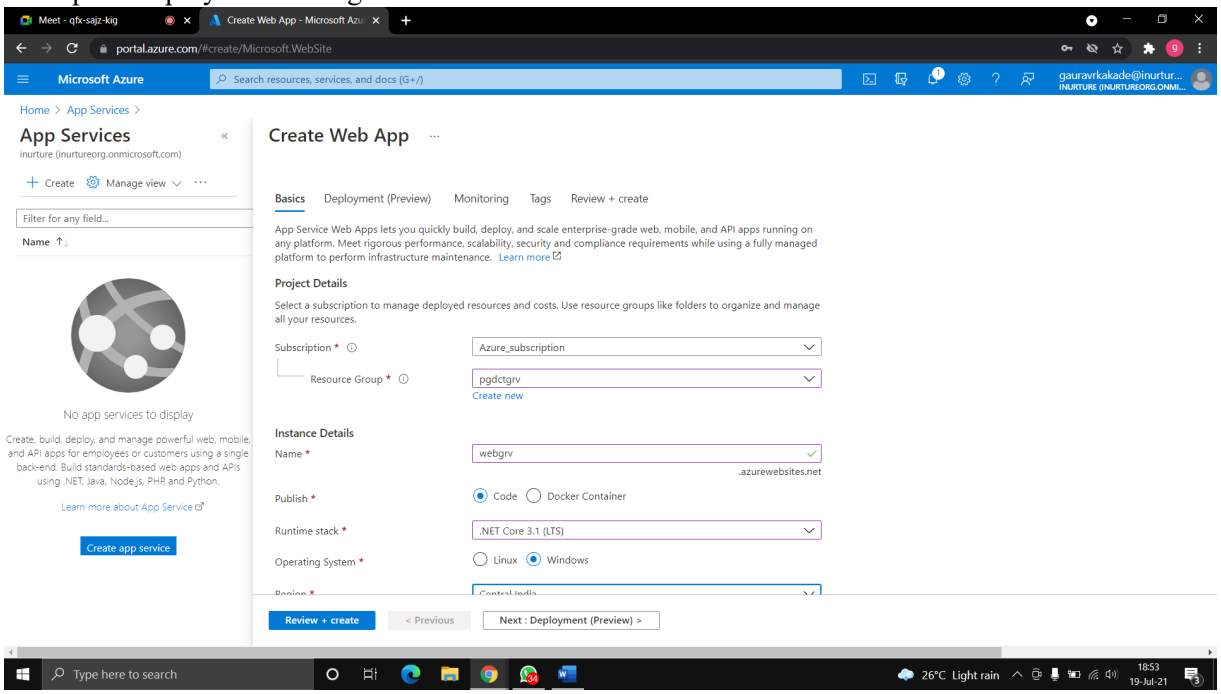


Fig: deployment settings

8.In monitoring keep application insights as no.

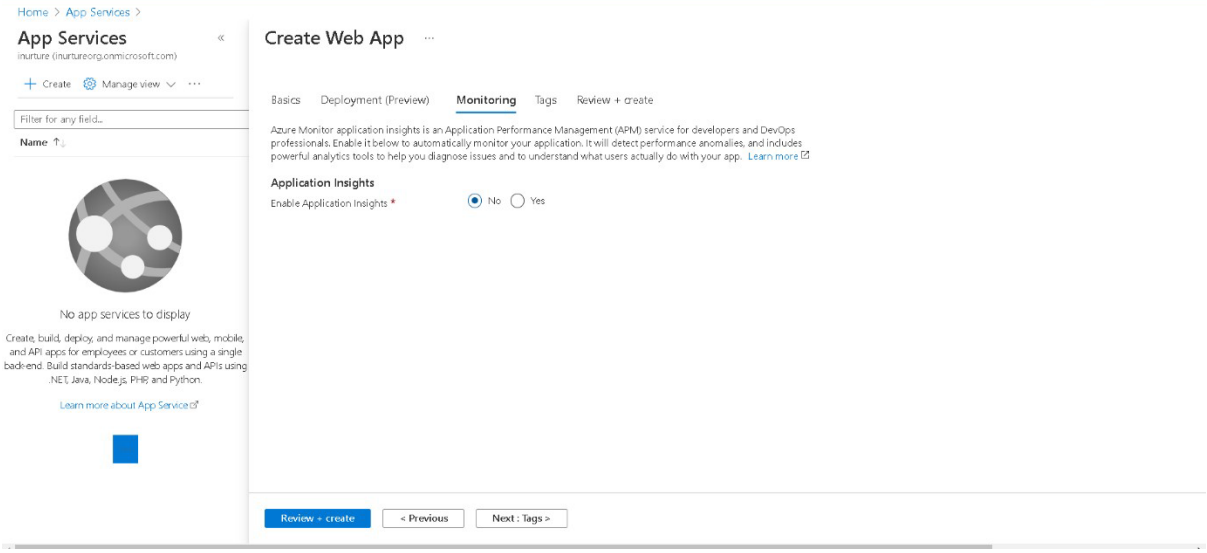


Fig: configure monitoring

9.Create the web app.

~ Steps to install visual studio and sink with azure:

1.Go to browser and download visual studio.

2.While installing you can select which features you want in your visual studio.

3.Open the visual studio>>create project.

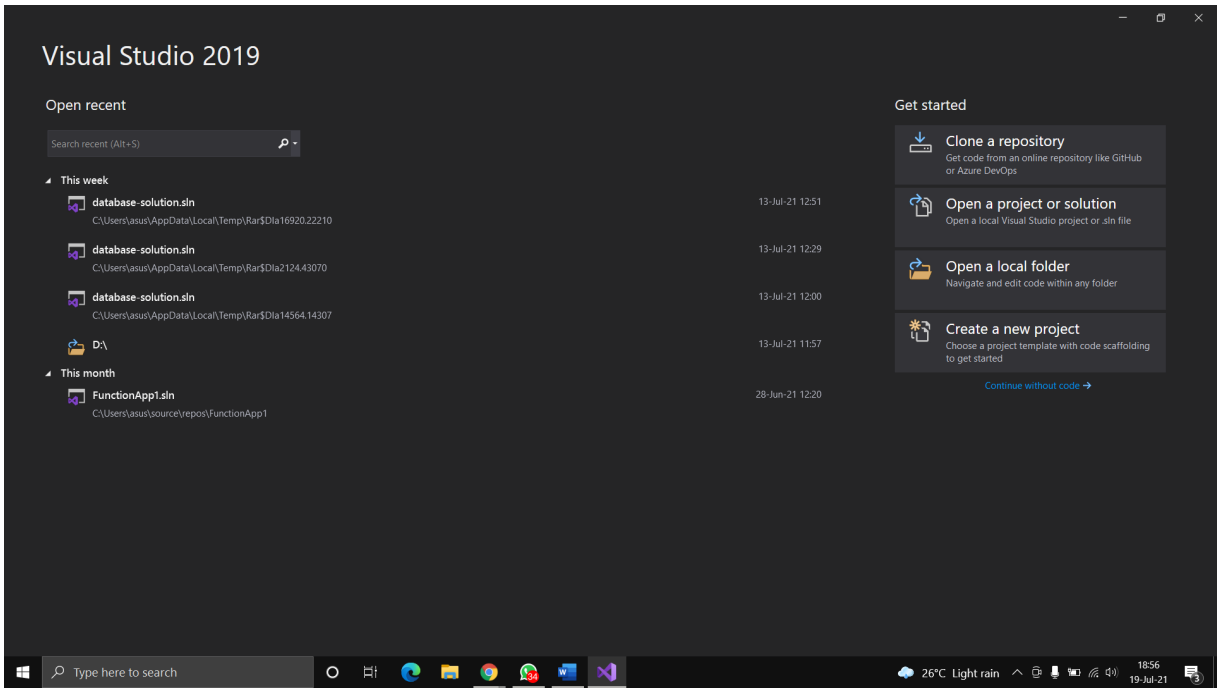


Fig: install VS and create new project

4.Select ASP.NET core web app.

5.Select target framework as .NET 3.1 and create the project.

6.Go to publish.

7.In target select azure and in specific target select azure app service(windows).

8.In app service>>sign in with your azure account.

9.After sign in>>select your resource group>>click on create.

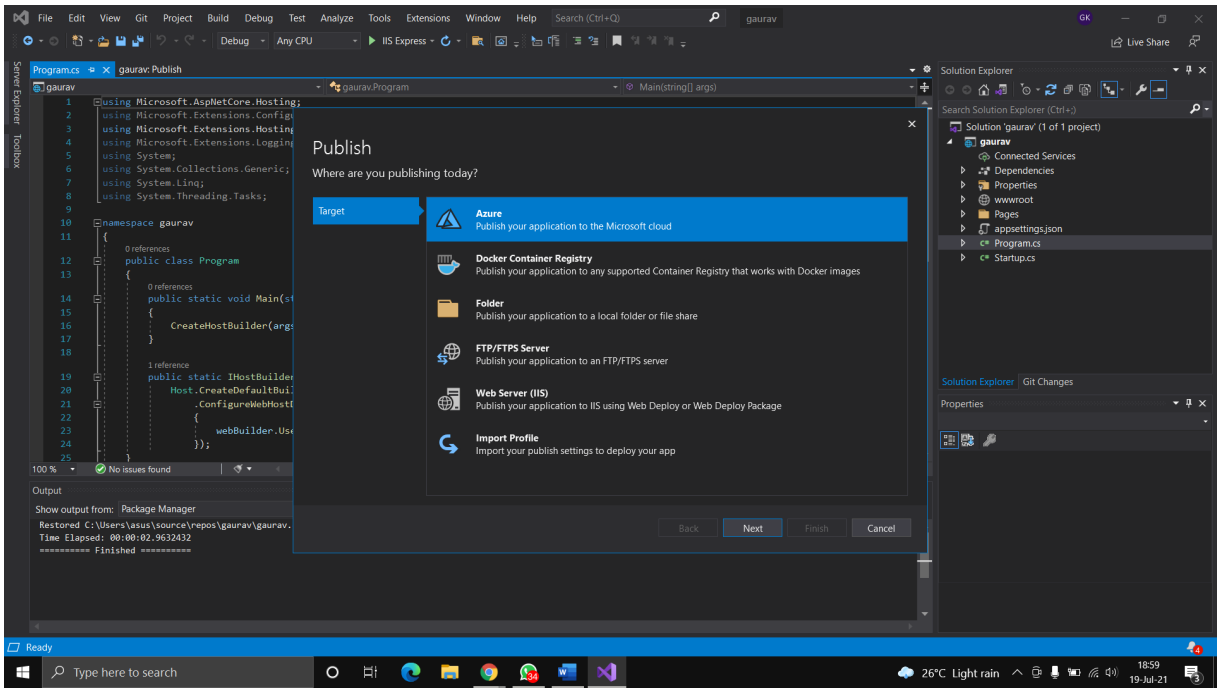


Fig: select your app service instance

~ Steps to see app service logs:

1.Go to azure portal>>app service logs.

2.Go to web server logging>>storage.

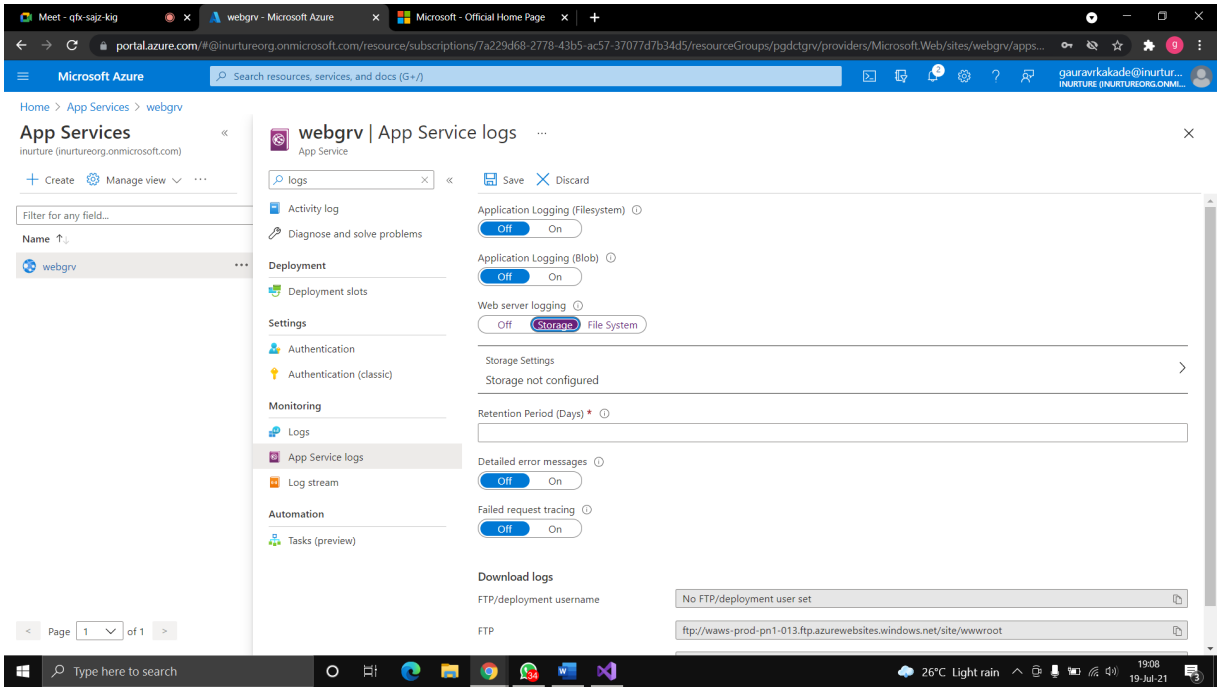


Fig: create app service logs

3.Create storage account>>create container.

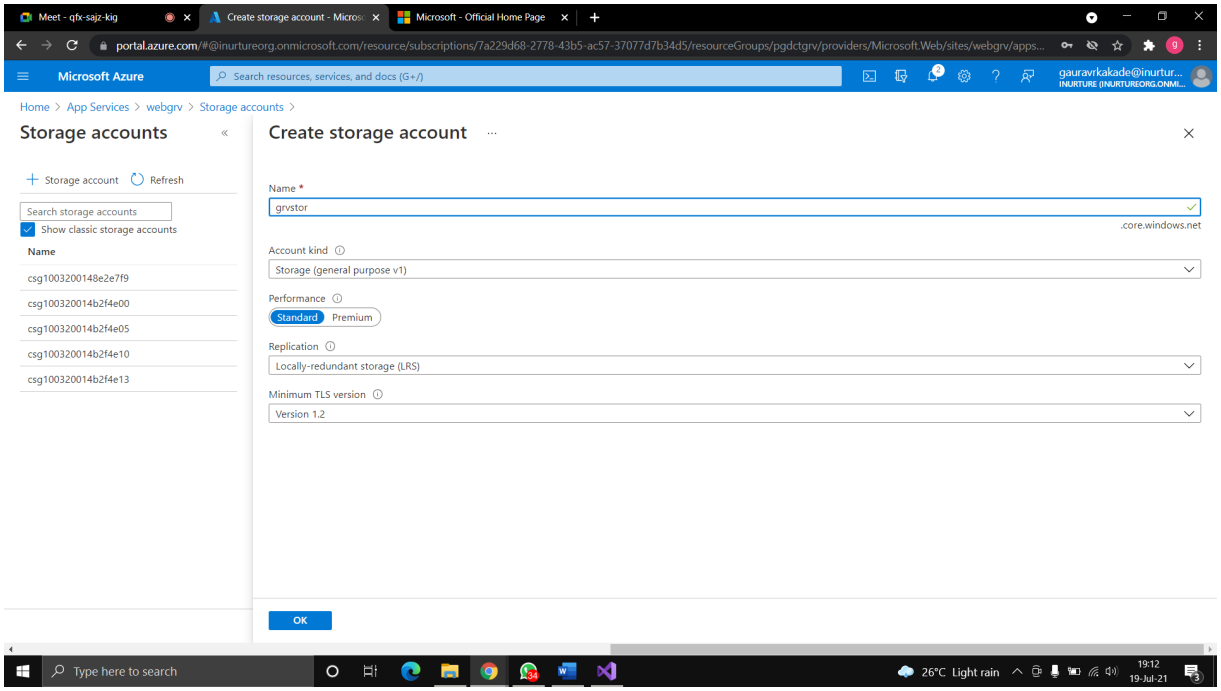


Fig: create storage account

4.Select the container and save.

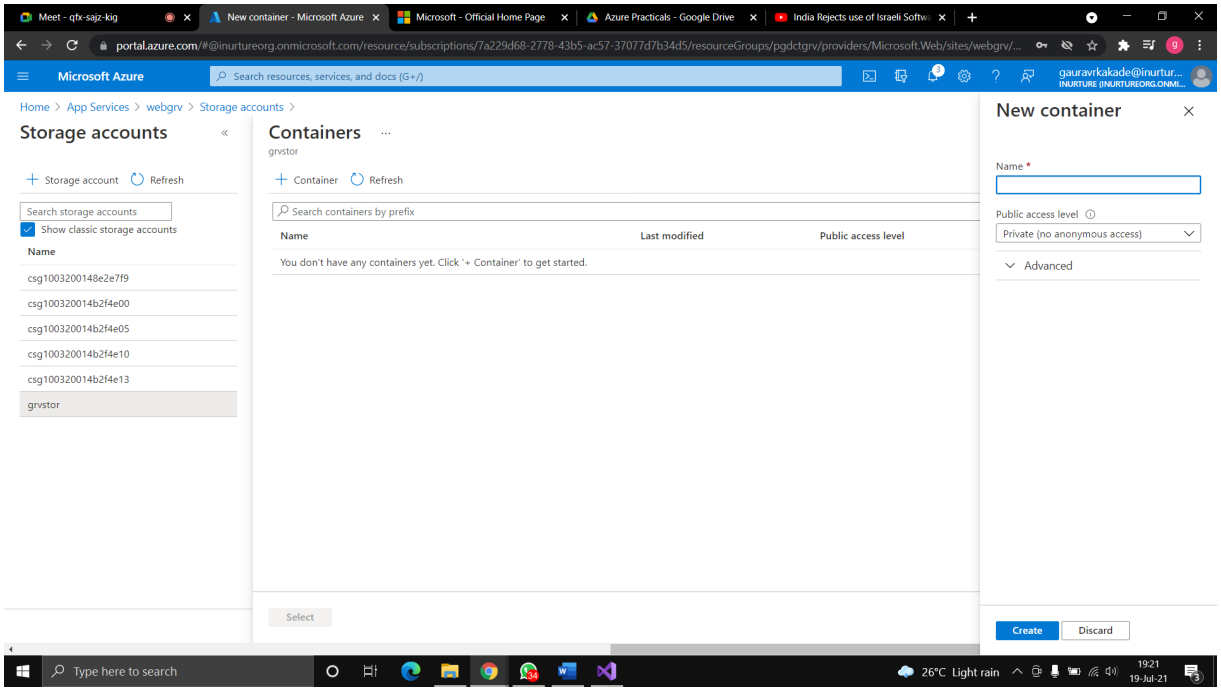


Fig: create new container

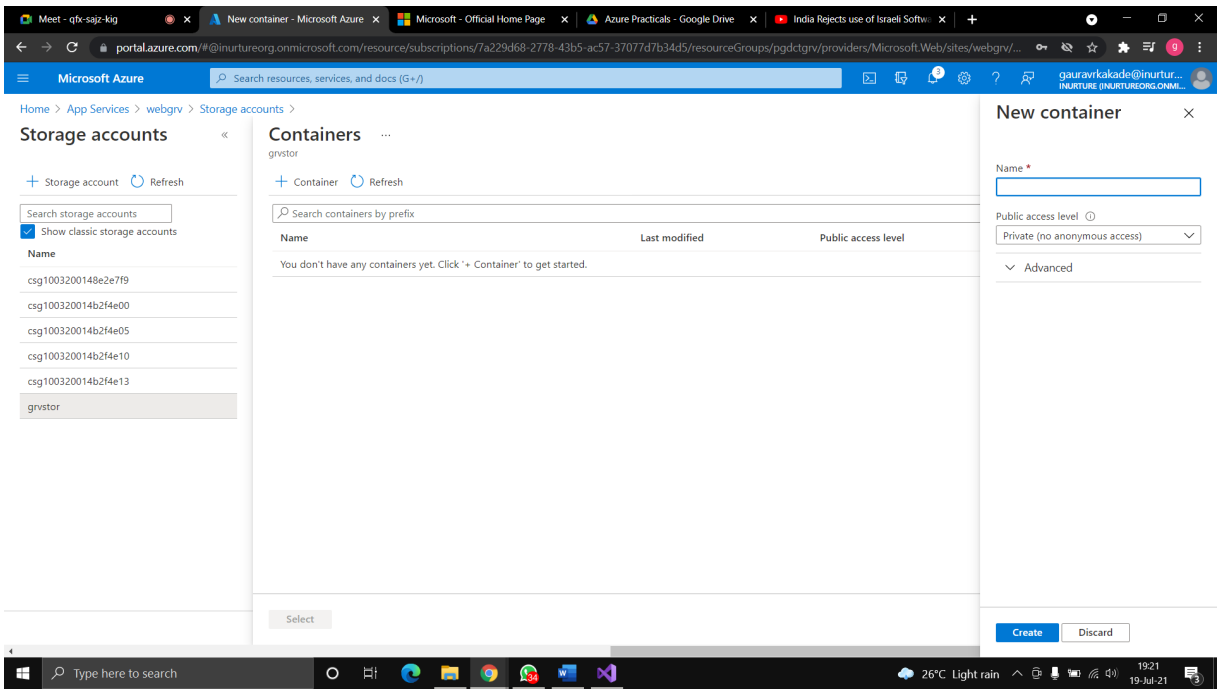


Fig: created container

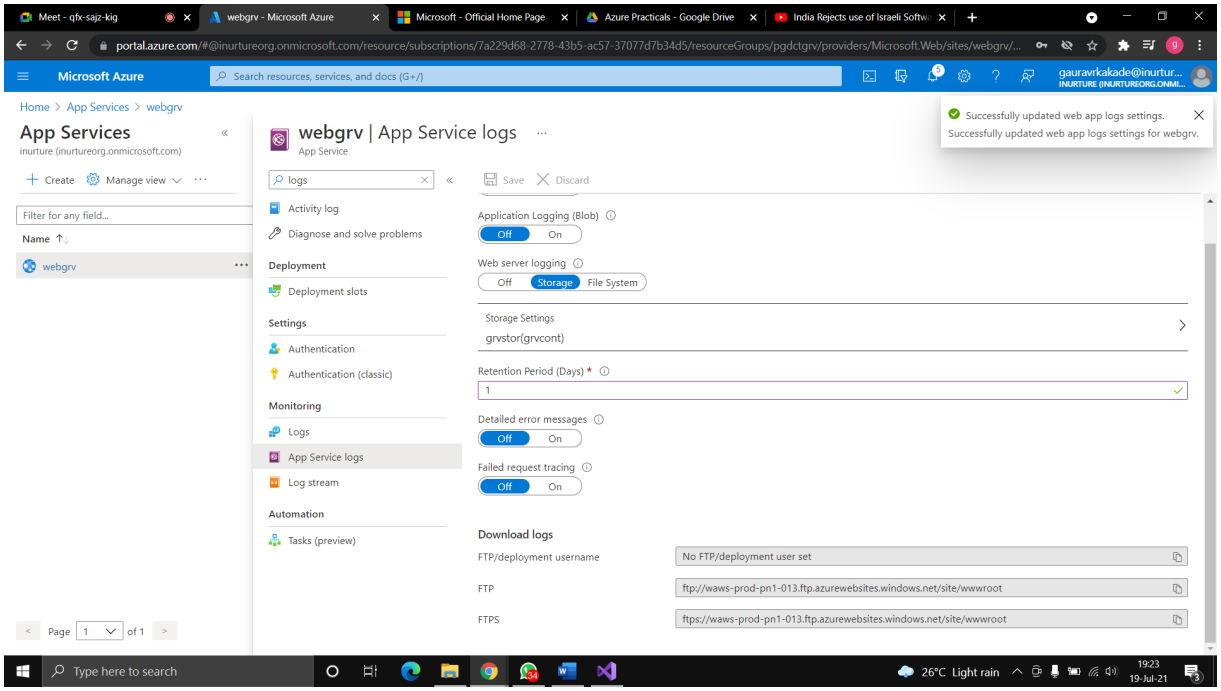


Fig: created container

5.Now go to storage>>container.

6.You will see one folder is created.

7.Open that folder>>you will see your logs.

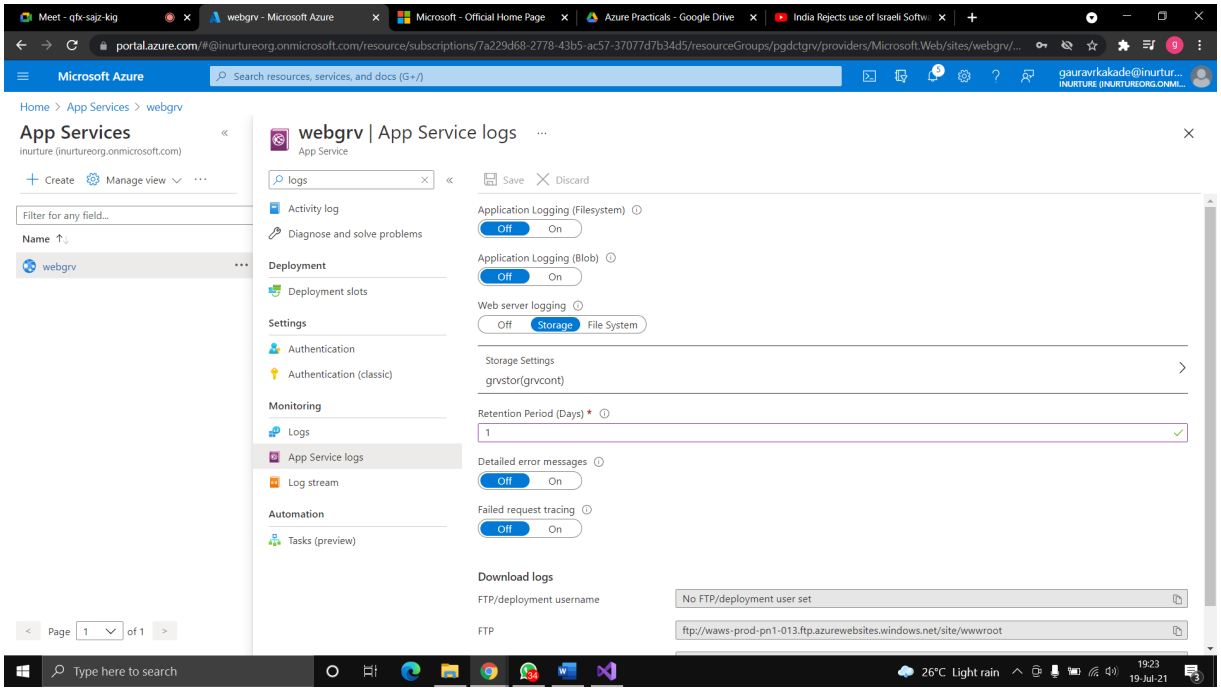


Fig: overview window

8.Go to overview>>copy the url and paste in browser.

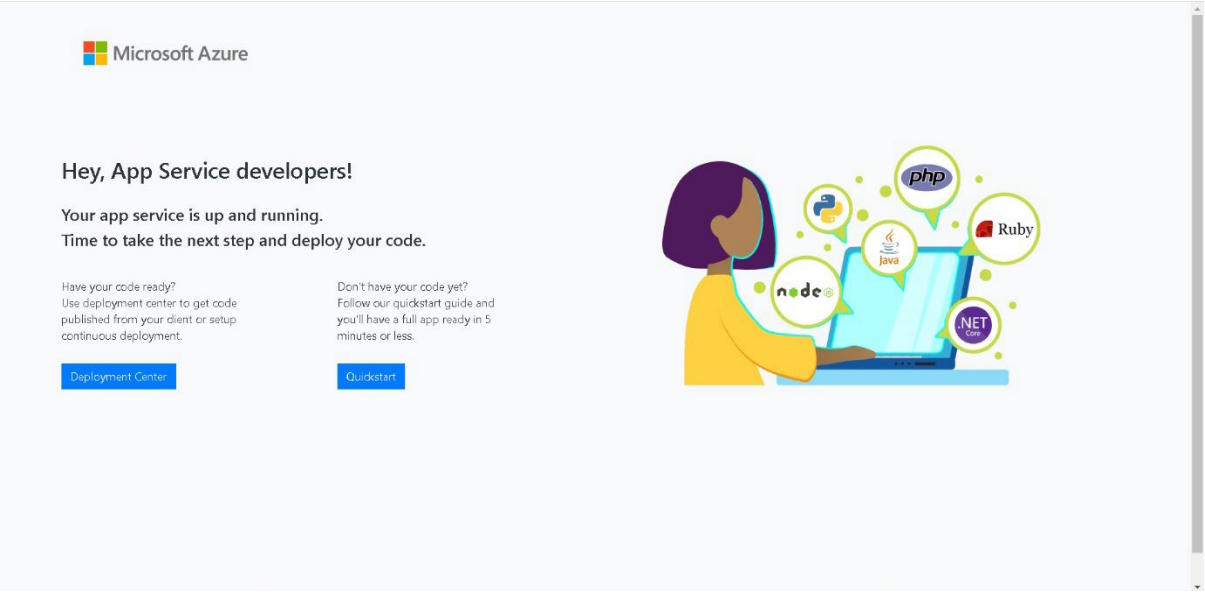
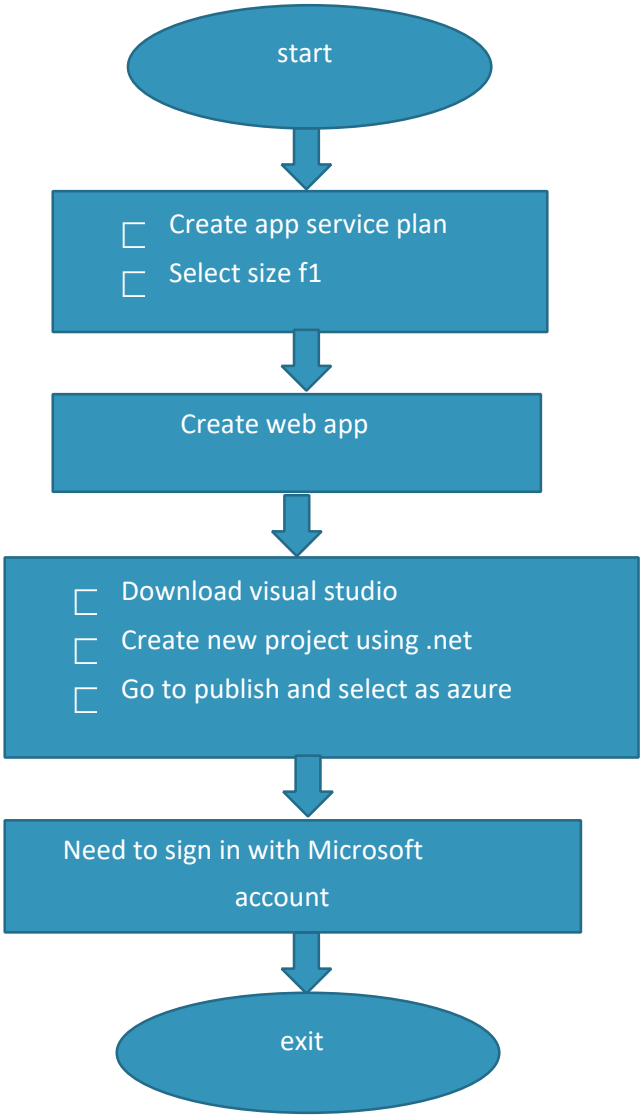
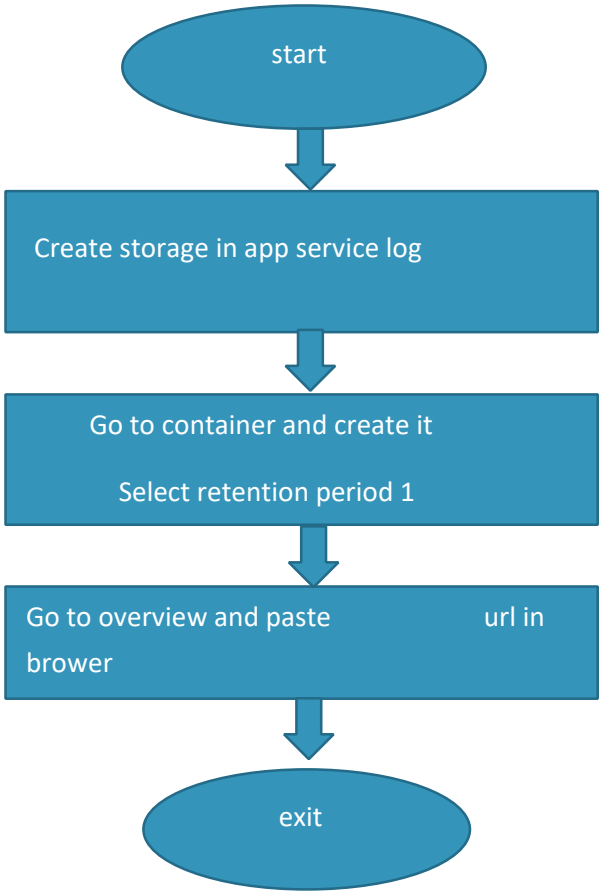


Fig: Page of AppService web

Flowchart:



Flowchart to see the logs:



Result: The web app has been created and the azure account has been sink with visual studio, where we can see our web app. The app service logs was created and can been seen in container.

Conclusion: The web app was successfully created and azure account was successfully sink with visual studio.

EXPERIMENT NO: 4

AIM: Create a web app instance and a virtual machine which contains database, and try to access the database through web app instance.

PREREQUISITES: Azure portal, visual studio, SQL server management studio.

DESCRIPTION:

Azure app Service plan: An App Service plan defines a set of compute resources for a web app to run. Azure Web Apps provides a platform to build an App in Azure without having to deploy, configure and maintain your own Azure VM's.You can build Web App using the ASP.NET, PHP, Node.js and Python. They also integrate common development environments which could be Visual Studio and GitHub.

Visual studio: Microsoft Visual Studio is an integrated development environment from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps.

SQL server management studio:SQL Server Management Studio is a software application first launched with Microsoft SQL Server 2005 that is used for configuring, managing, and administering all components within Microsoft SQL Server.

ALGORITHM:

1.Create a virtual network.

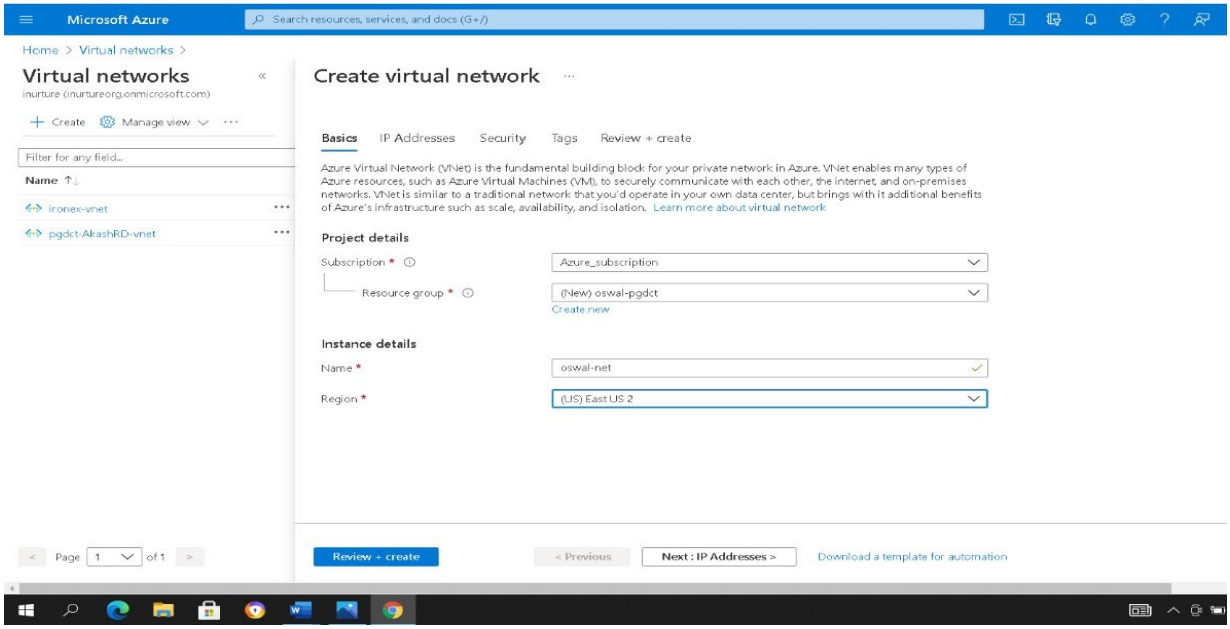


Fig: create virtual network 2. Go to app

service plan>>create a app service plan.

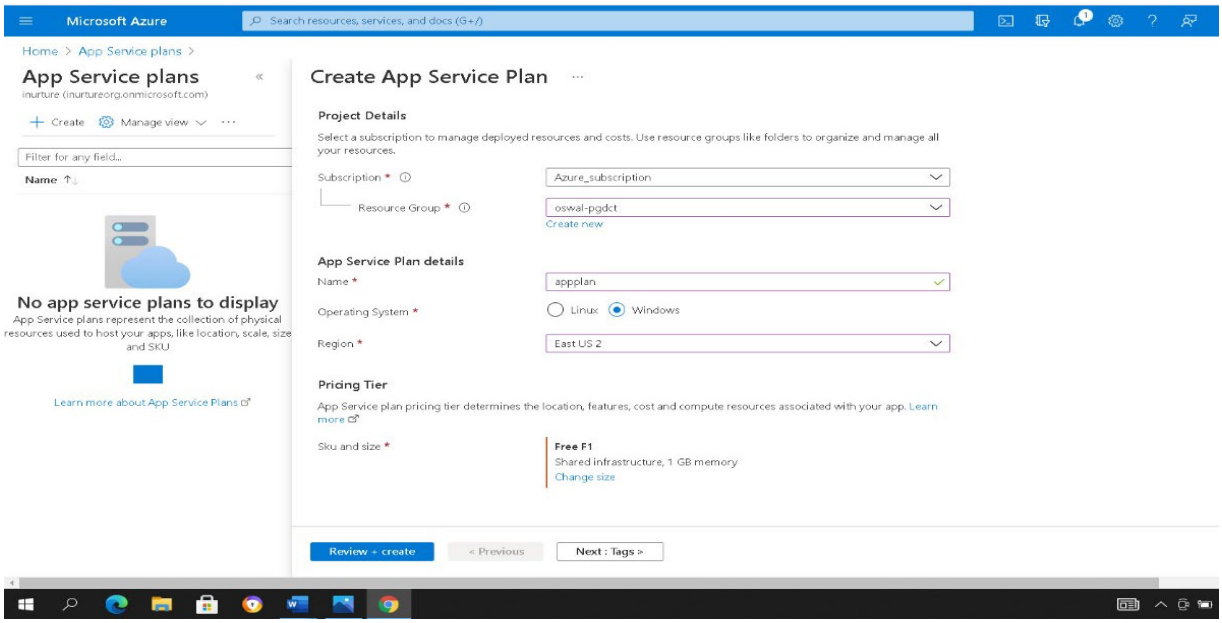


Fig: create app service plan

3.Go to resource of app service plan>>Networking>>go to VNet integration.

4.Choose the plan **S1** and apply.

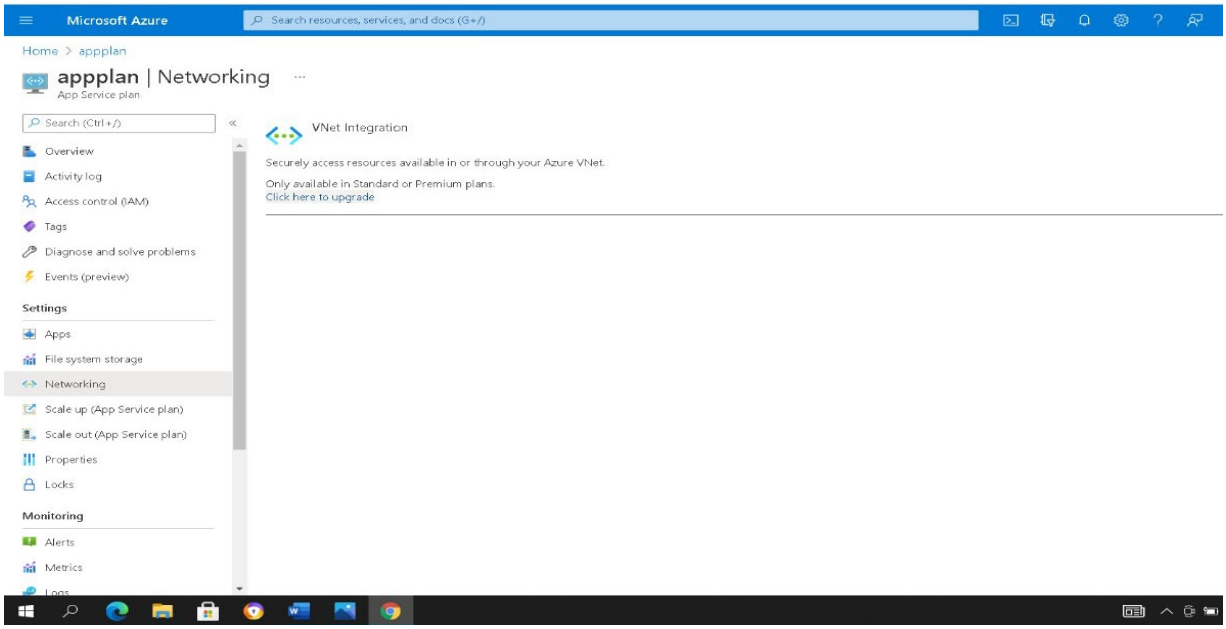


Fig: upgrade vnet integration

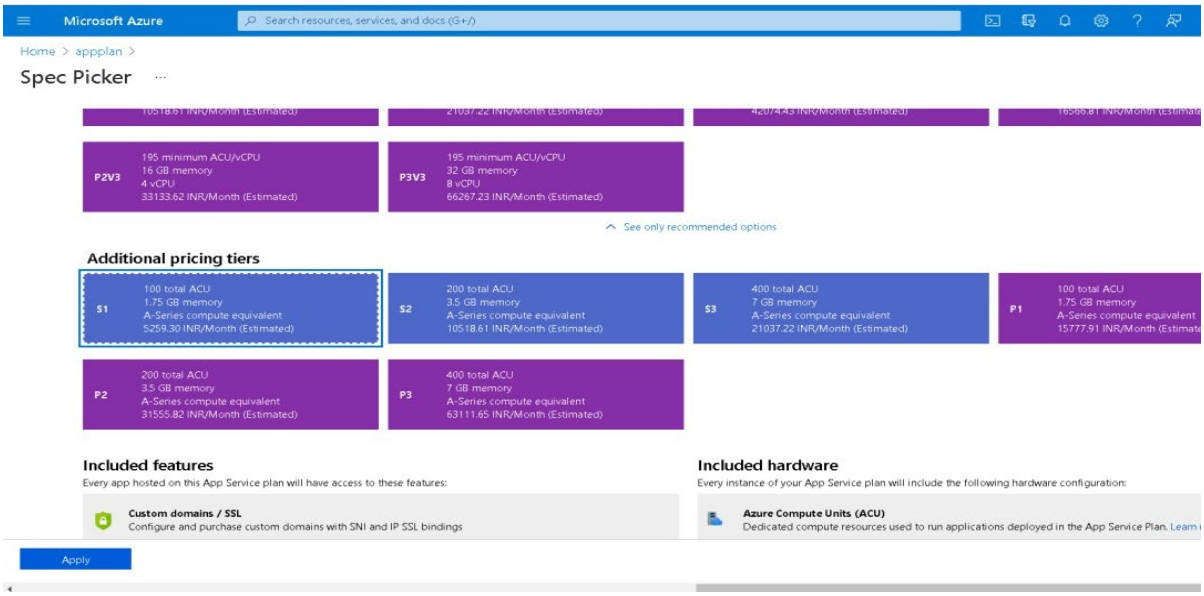


Fig: select S1 pricing tiers

5.Go to app service>>click on create a web app.

6.Give the name of instance>>choose runtime as **ASP.NET V4.8**.

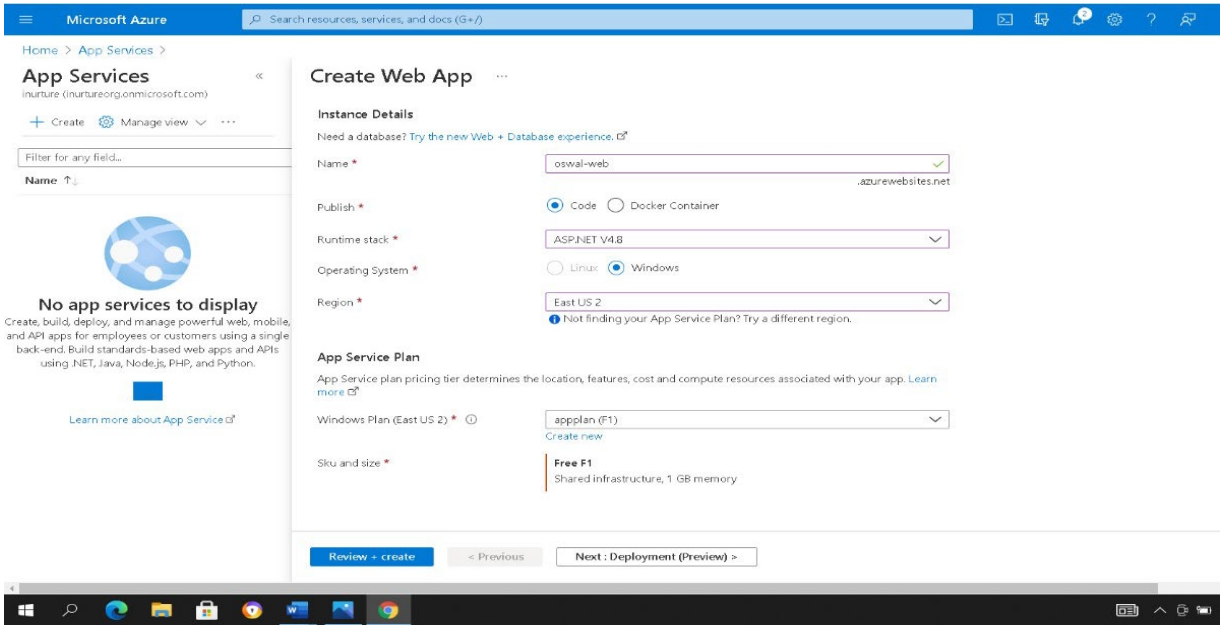


Fig: create web app

7.Keep enable application insights as no.

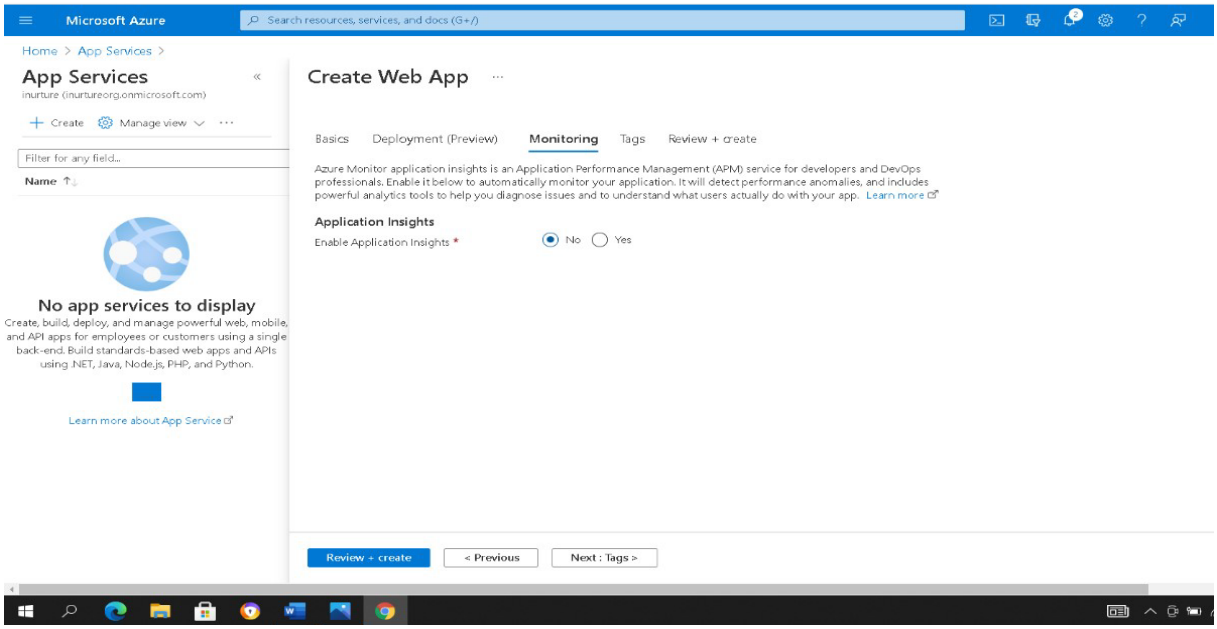


Fig: configure monitoring

8.Create web app.

9.Go to resource>>Networking>>go to VNet integration.

10.Click on add VNet>>select your virtual network and click on ok.

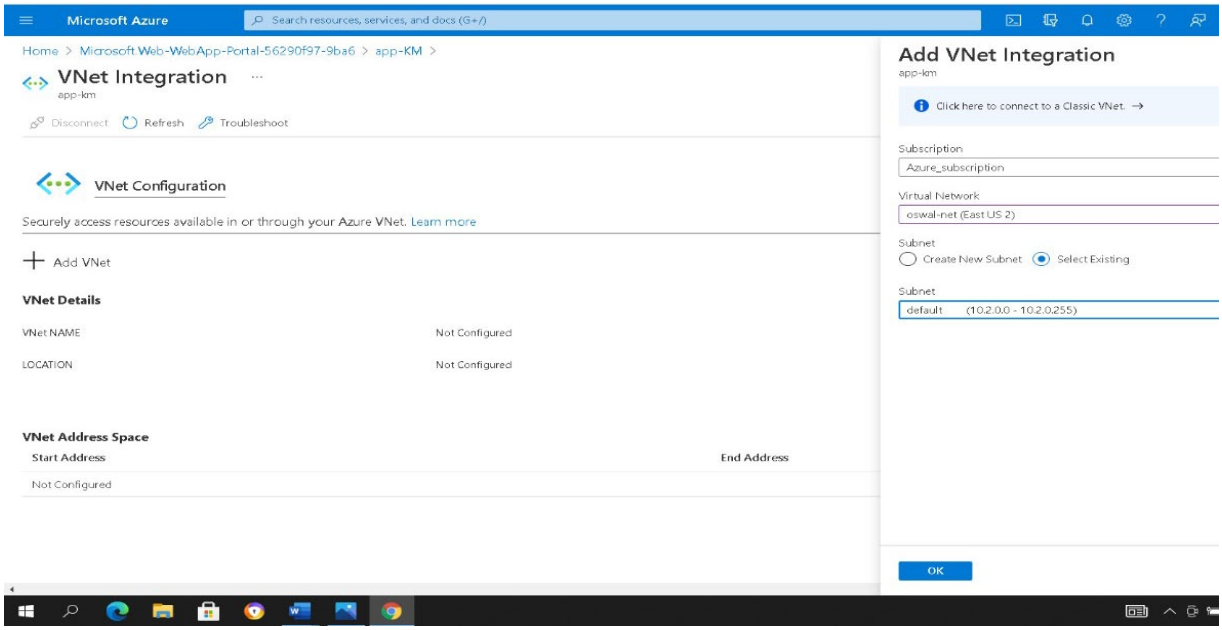


Fig: Add vnet integration

11.Now create a virtual machine with the image of **SQL server 2019 web on windows server 2019-Gen1.**

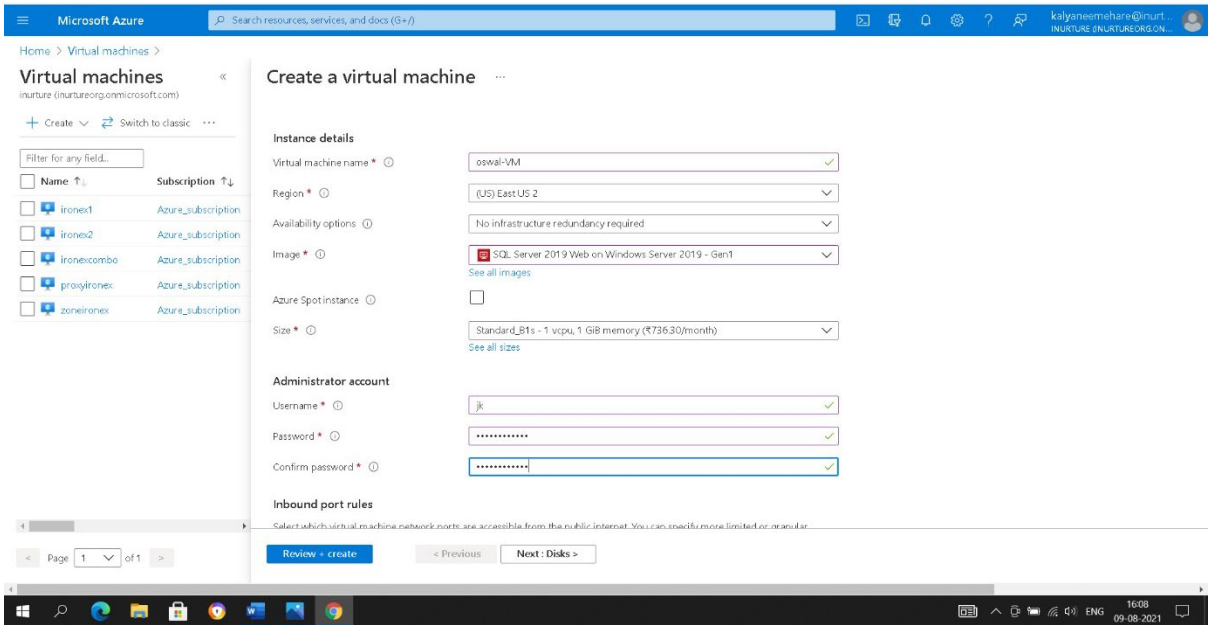


Fig: create virtual machine

12.In networking>>go to subnet by selecting manage configuration and create a new subnet.

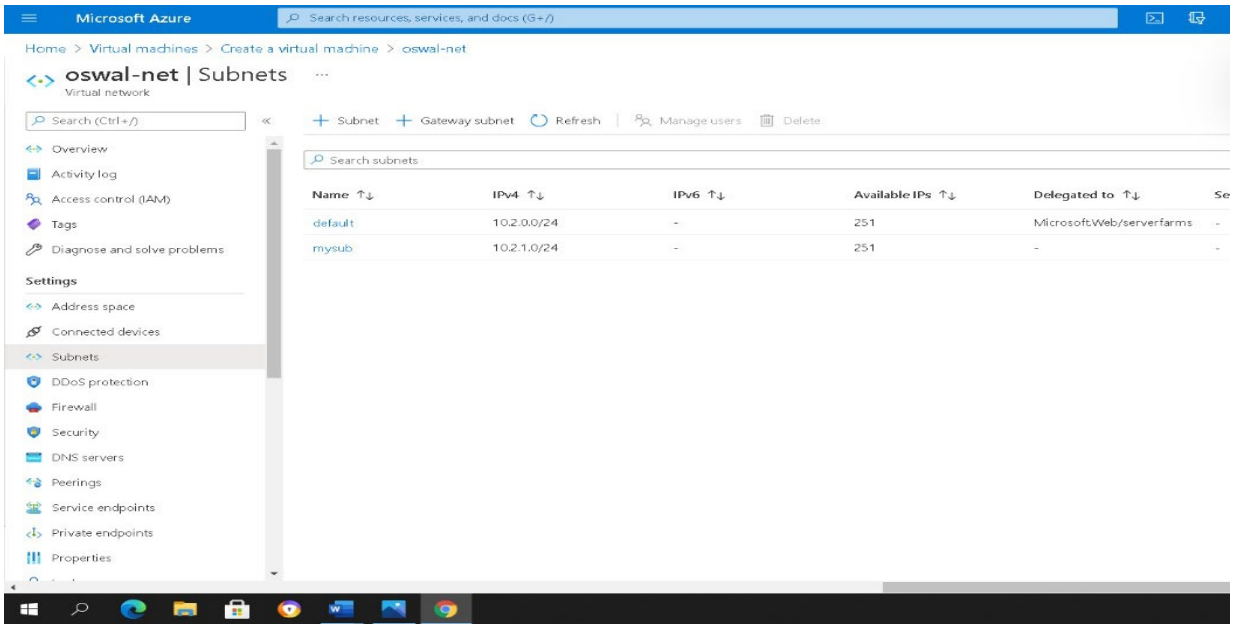


Fig: create subnet

13.In SQL server settings>>keep the SQL connectivity as public and enable SQL authentication, enter the password and username.

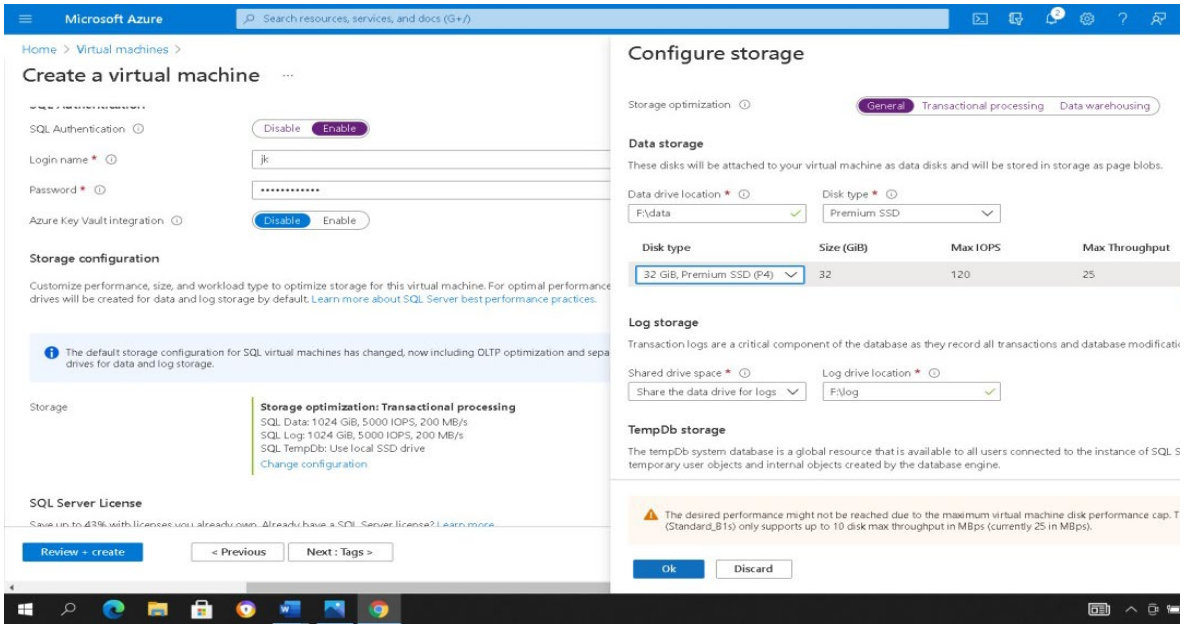


Fig: configure storage

14.Create the virtual machine.

15.Open the SQL server management studio, In server name enter your virtual machine IP address and choose SQL authentication, enter username and password and click on connect.

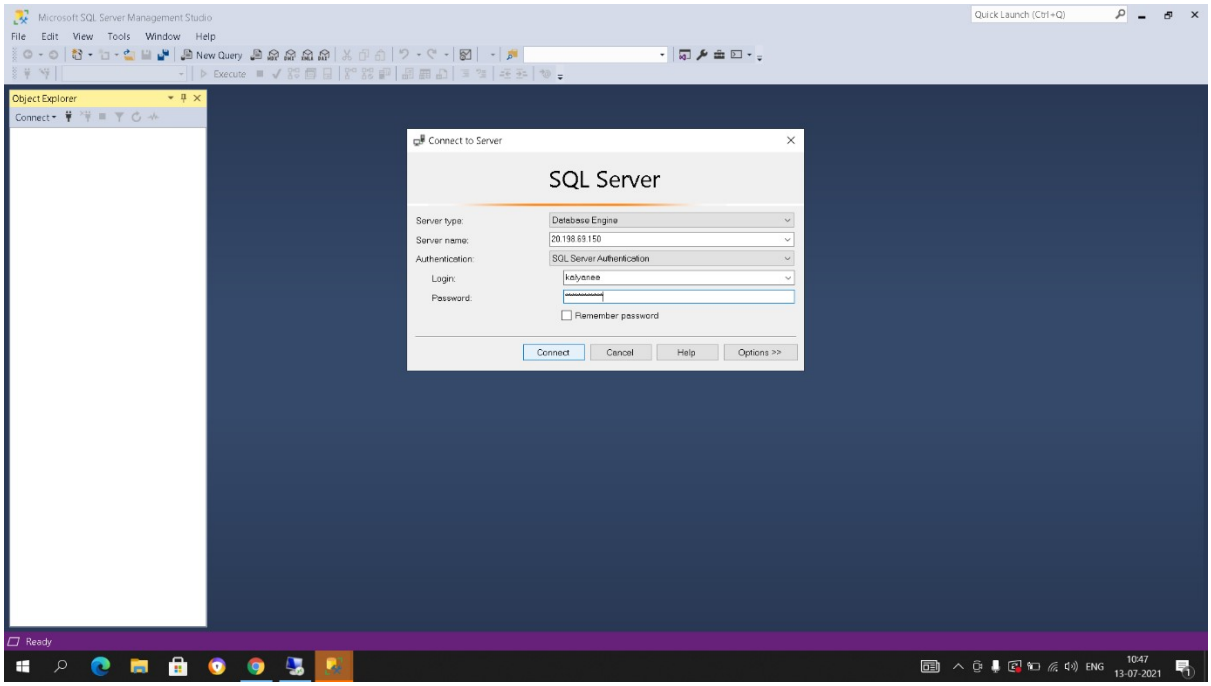


Fig: give credentials

16.Create new database and create a table.

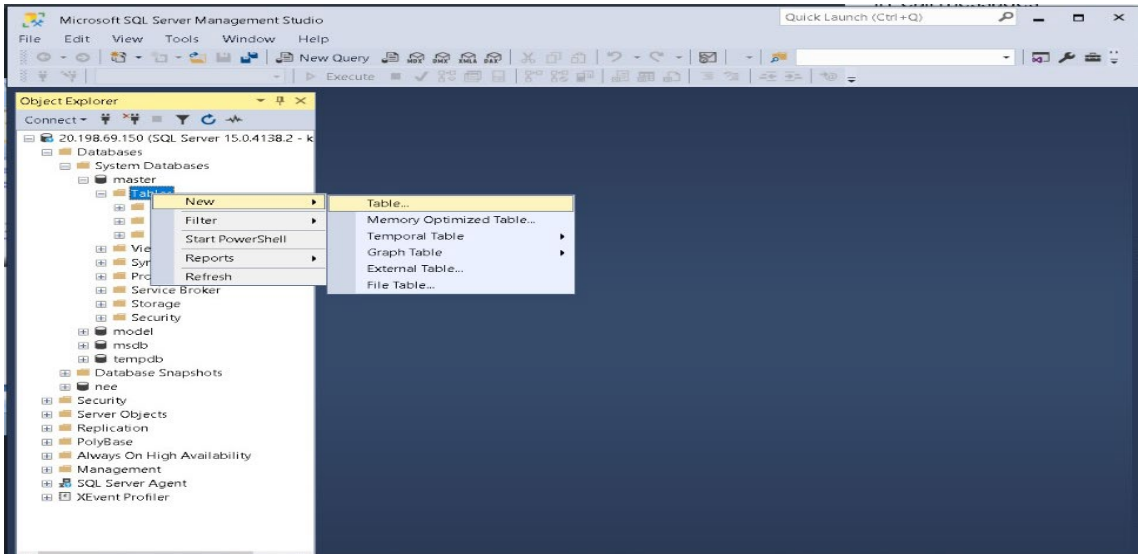


Fig: Add new tables

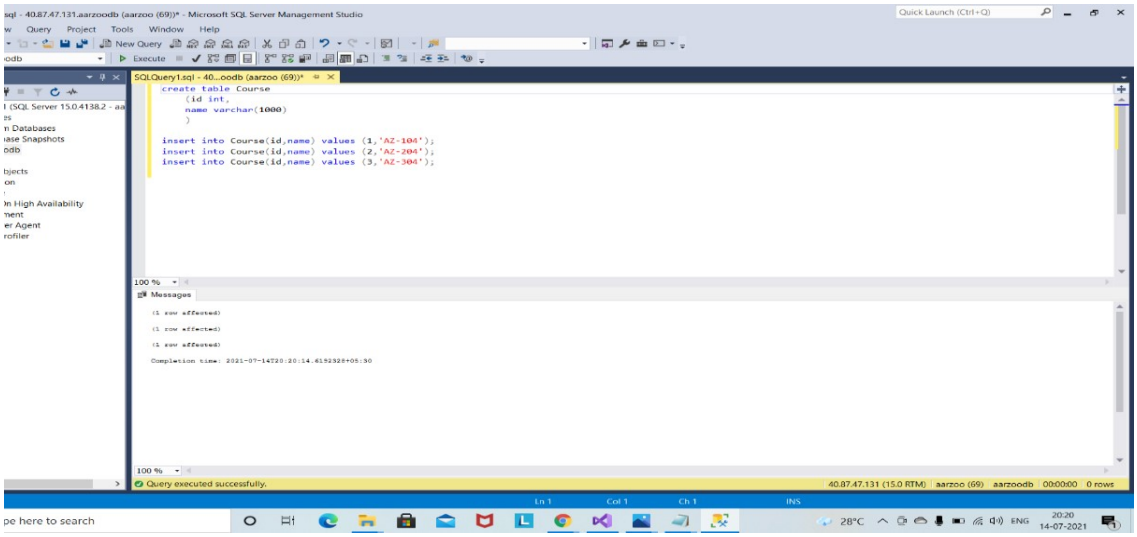


Fig: created table

17.Go to virtual machine>>networking>>go to public IP address and disassociate the public IP.

18.Open the database project in visual studio, go to webconfig file edit the code and enter the proper credentials of your virtual machine in that code.

19.Now publish and sink with azure account, select your resource group and click on finish.

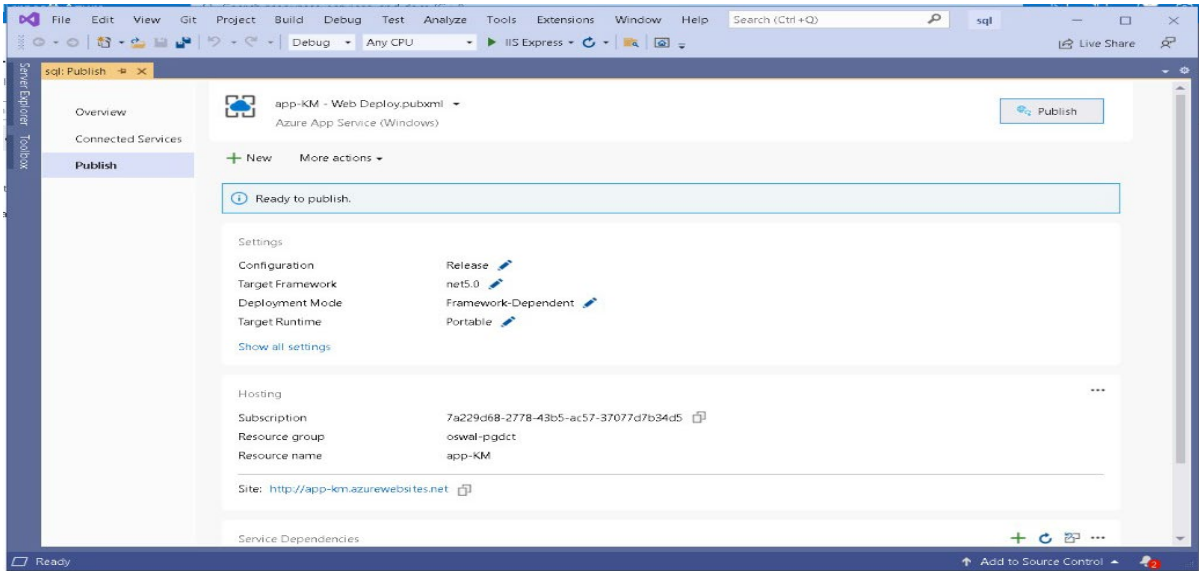


Fig: published

20.Now copy the URL and browse it, you will see your data.

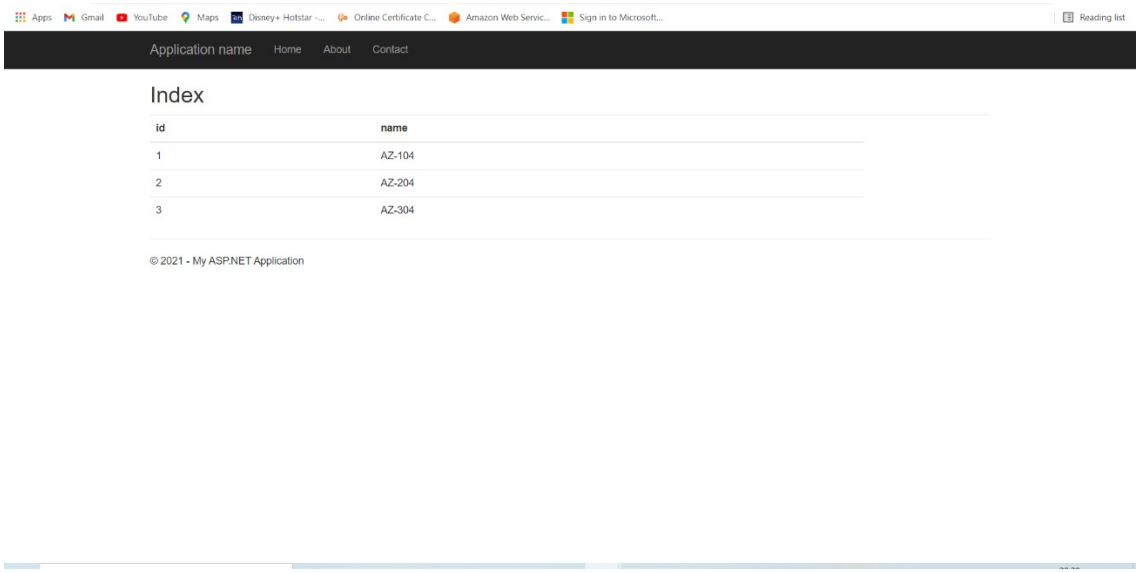
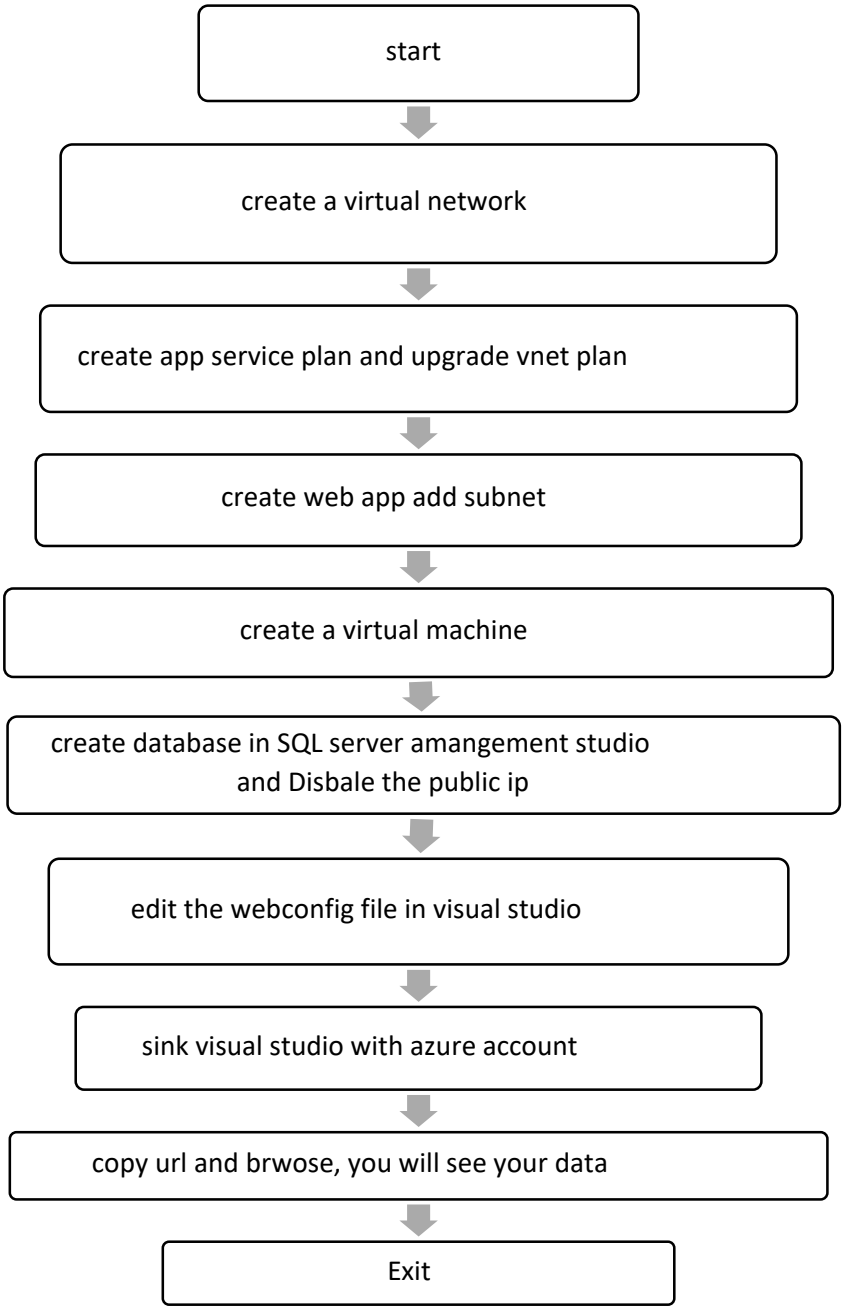


Fig: output of created table

Flowchart :



Result: We have created the web app instance and a virtual machine in which we have our database, we have seen that how web app has accessed our database.

Conclusion: we have successfully accessed the database through web app.

EXPERIMENT NO: 5

AIM: Creation of Virtual Machine, installation of Docker Engine, Image Management, Launching container

PREREQUISITES: Azure portal **DESCRIPTION:**

Docker:

Docker is a set of platforms as a service (PaaS) product that use OS-level virtualization to deliver software in packages called containers. Containers are isolated from one another and bundle their own software, libraries and configuration files; they can communicate with each other through well-defined channels. With Docker, you can manage your infrastructure in the same ways you manage your applications.

Container instance:

Azure Container Instances is a solution for any scenario that can operate in isolated containers, without orchestration. Run event-driven applications, quickly deploy from your container development pipelines, and run data processing and build jobs

ALGORITHM:

1. First, we need to create virtual machine.
2. Go to create virtual machine
3. create resource group>>give it a name and select region.
4. Select image ubuntu server 18.0-LTS -gen1>> password as authentication type
5. Give username and password>>open http and rdp ports

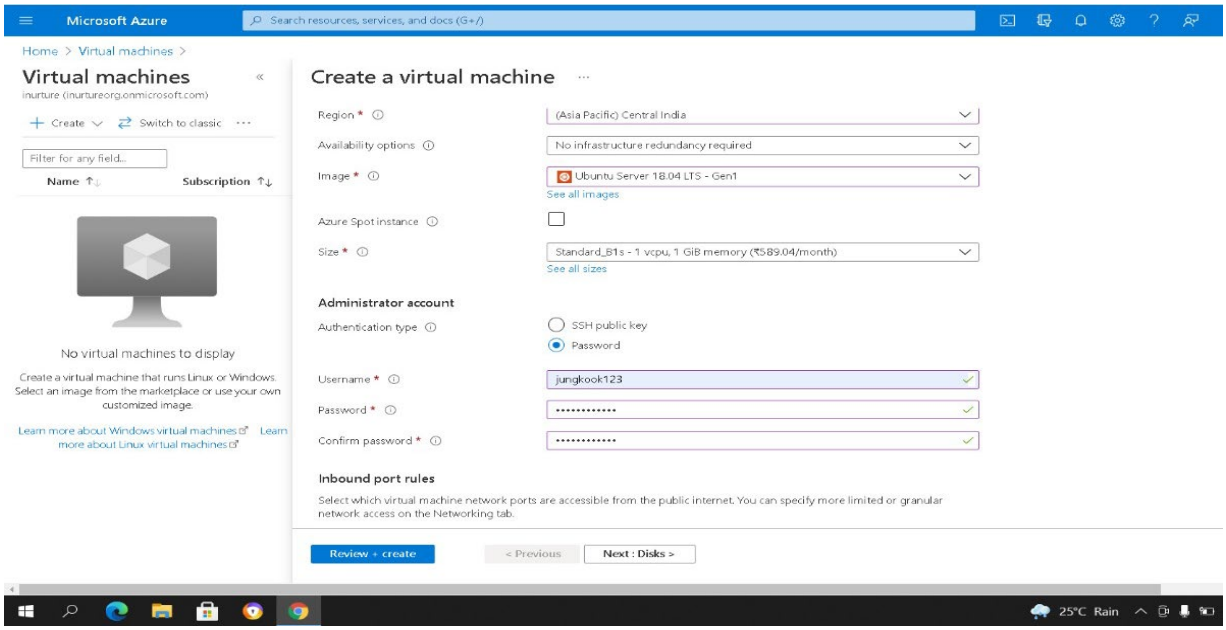


Fig: create virtual machine

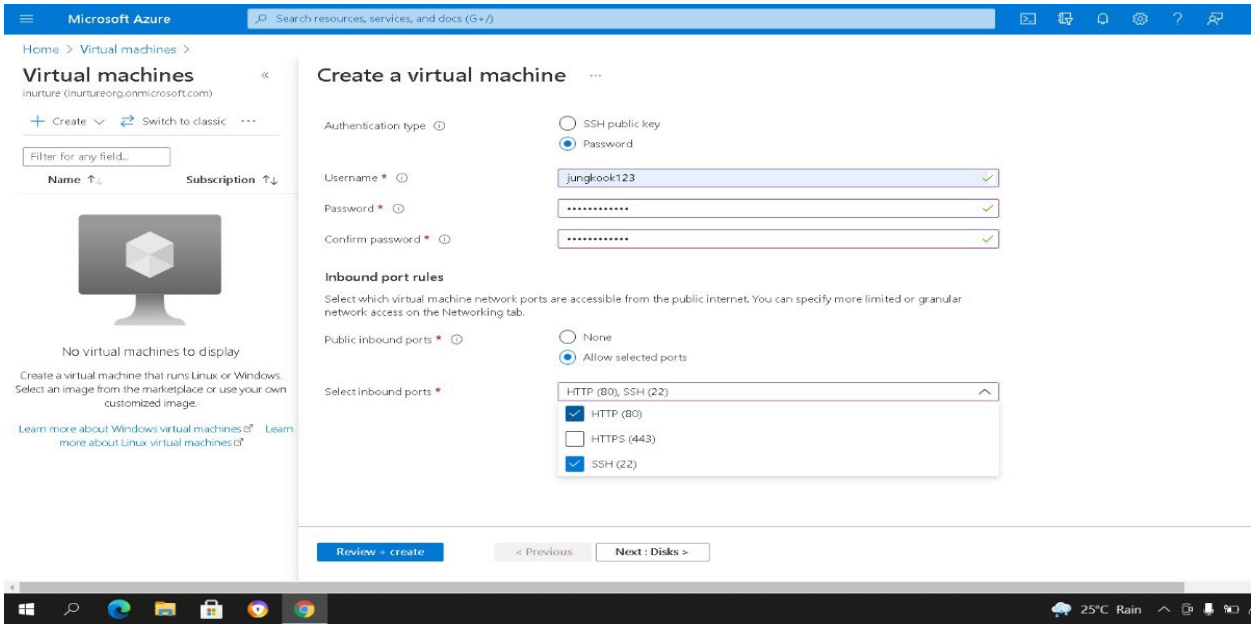


Fig: open port

6.Select os disk type standard HDD and everything set as a default and create virtual machine.

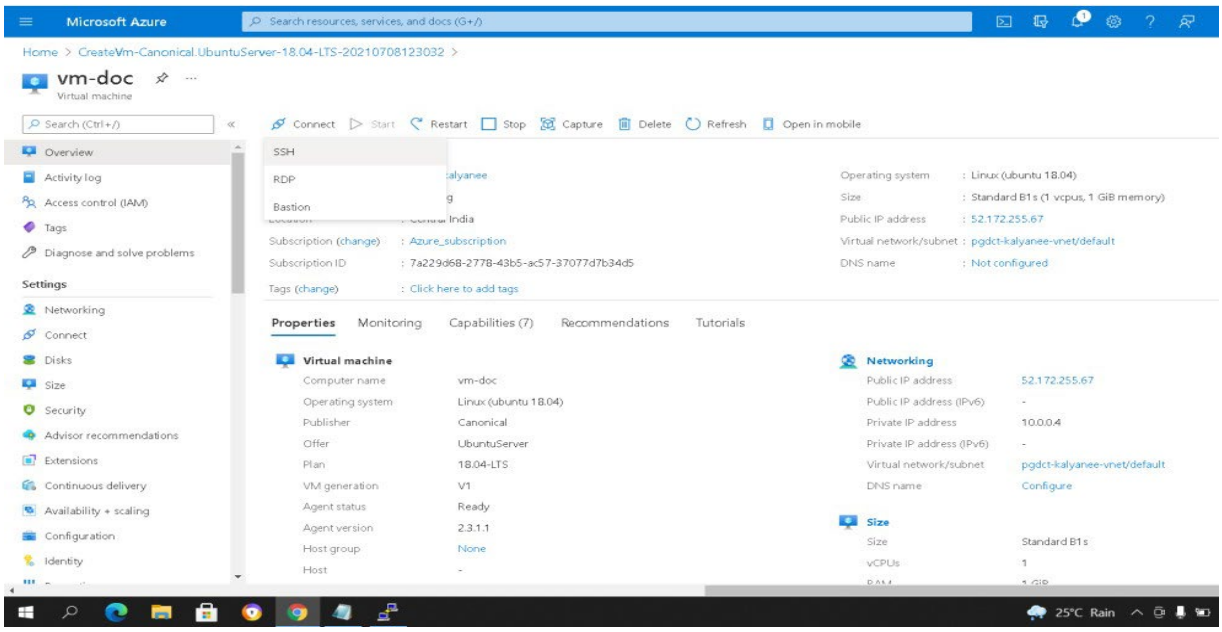


Fig: connect with ssh

7.After creation just connect vm to ssh

8.Open putty and add the username and password in it.

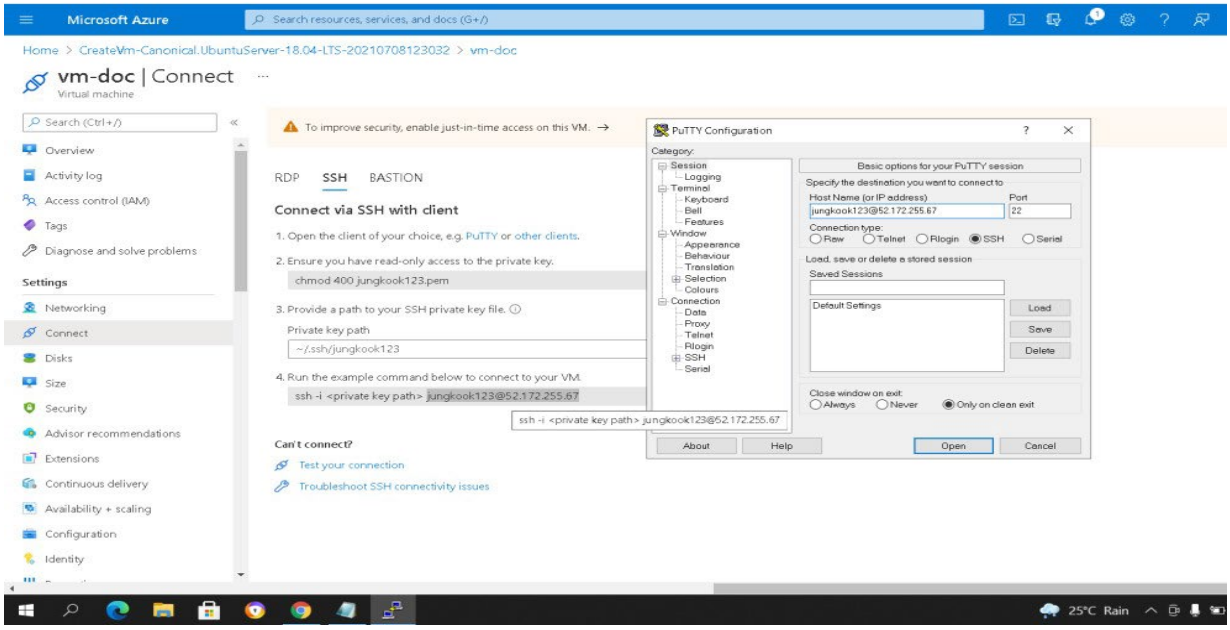


Fig: enter username and ip

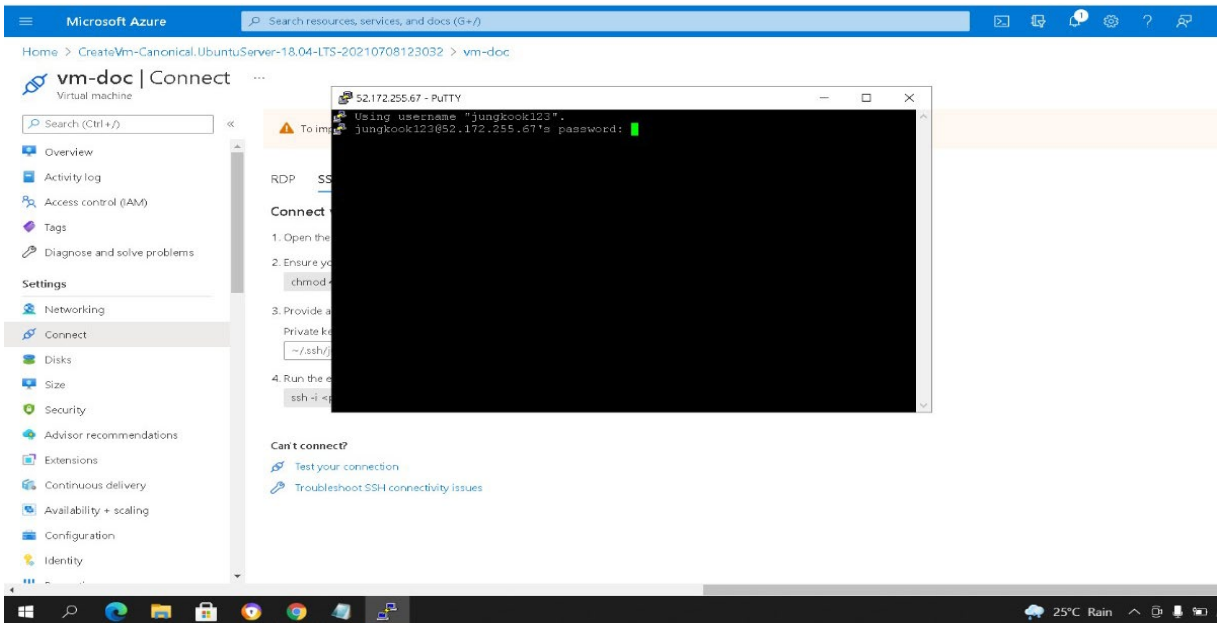


Fig: connected putty

9.after connecting with putty just enter your password and enter following commands.

10.sudo apt-get update

11.it will update packages

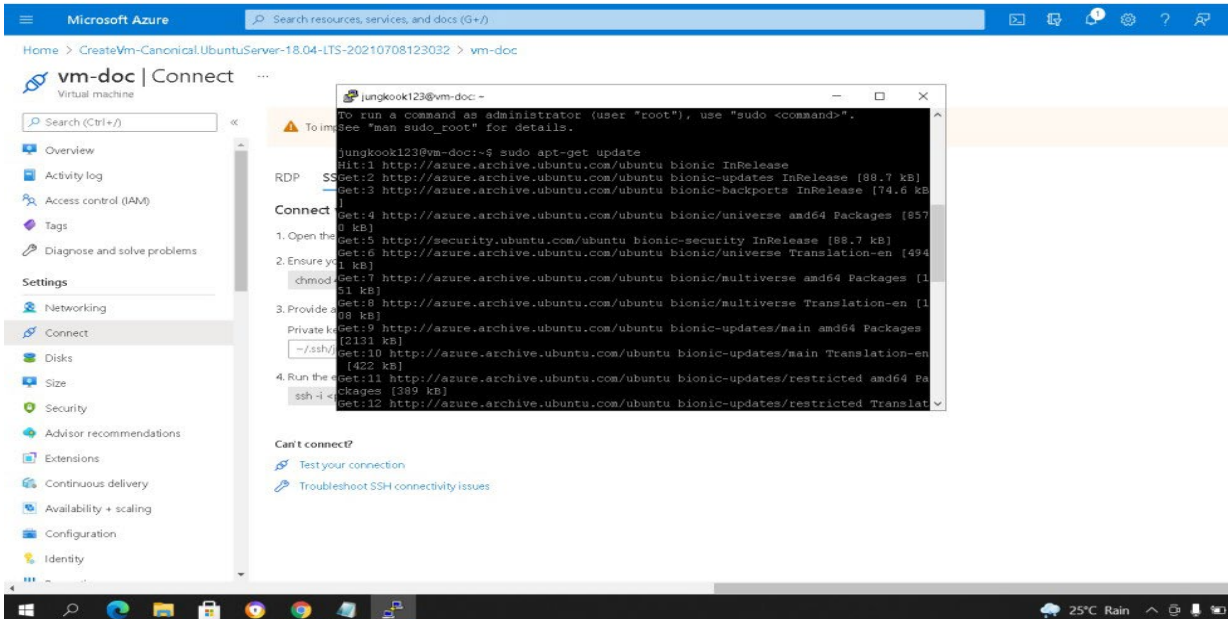


Fig: used commands

12.Install different packages for docker

13.sudo apt install apt-transport-https ca-certificates curl software-propertiescommon

~curl –fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add

~sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable"

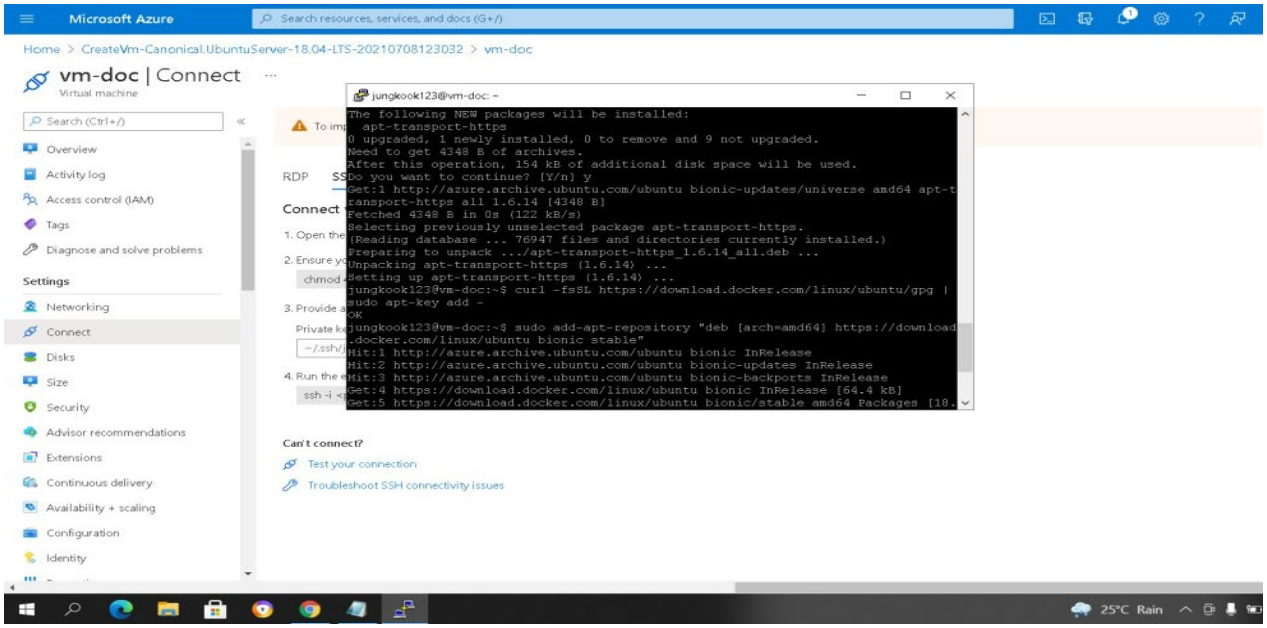


Fig: used commands

~sudo apt-get update

~Command for installing docker

~sudo apt-get install docker-ce

~command pull image for docker

~sudo docker pull nginx:1.17.0

~this command will create container with name sampleapp

~sudo docker run --name sampleapp -p 80:80 -d nginx:1.17.0

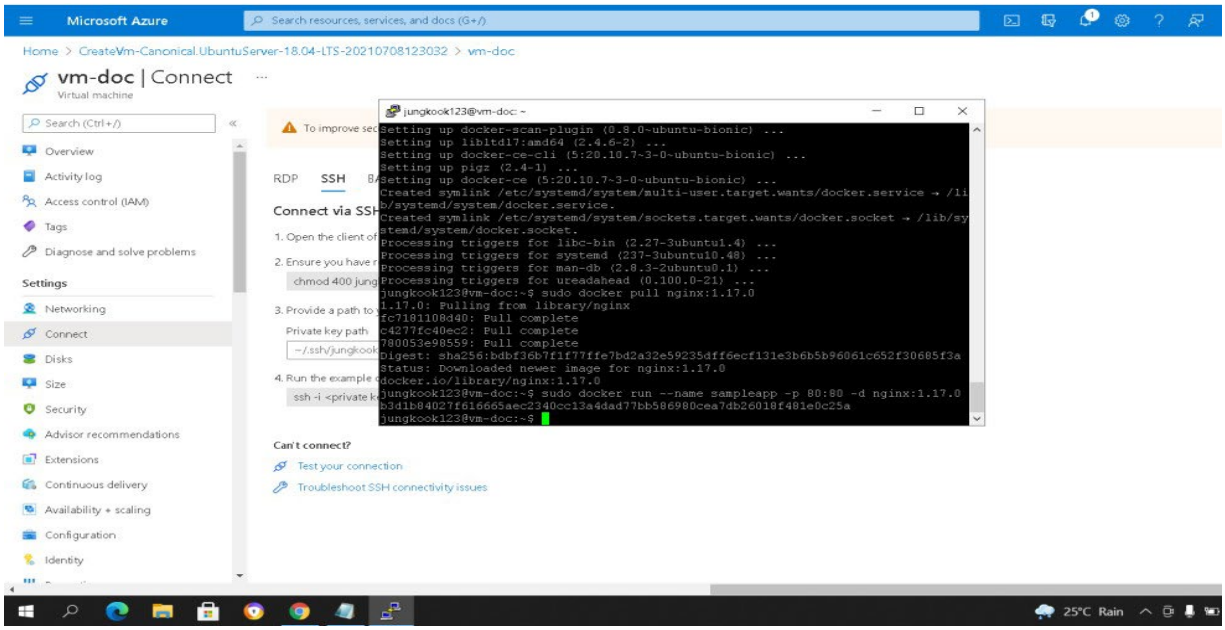


Fig: used commands

14.Then just grab public ip and browse it .

15.You will see the page of nginx server

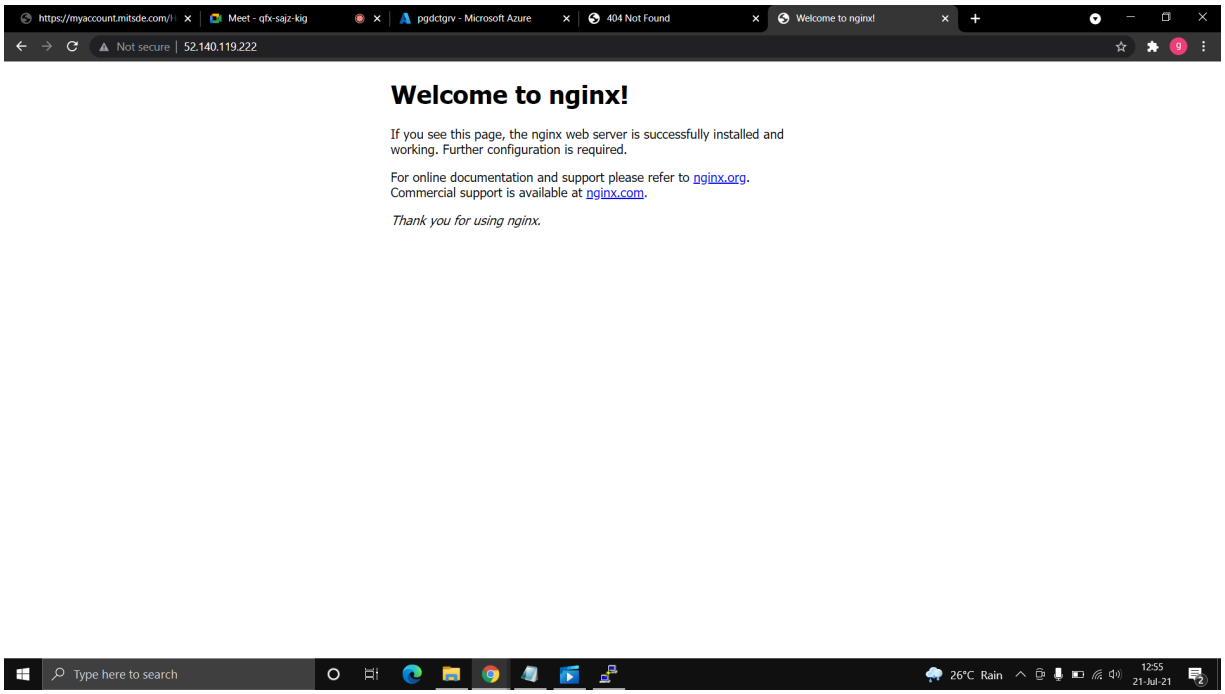


Fig: webpage of nginx server

16.Create Container instances

17.Search container instances in search bar

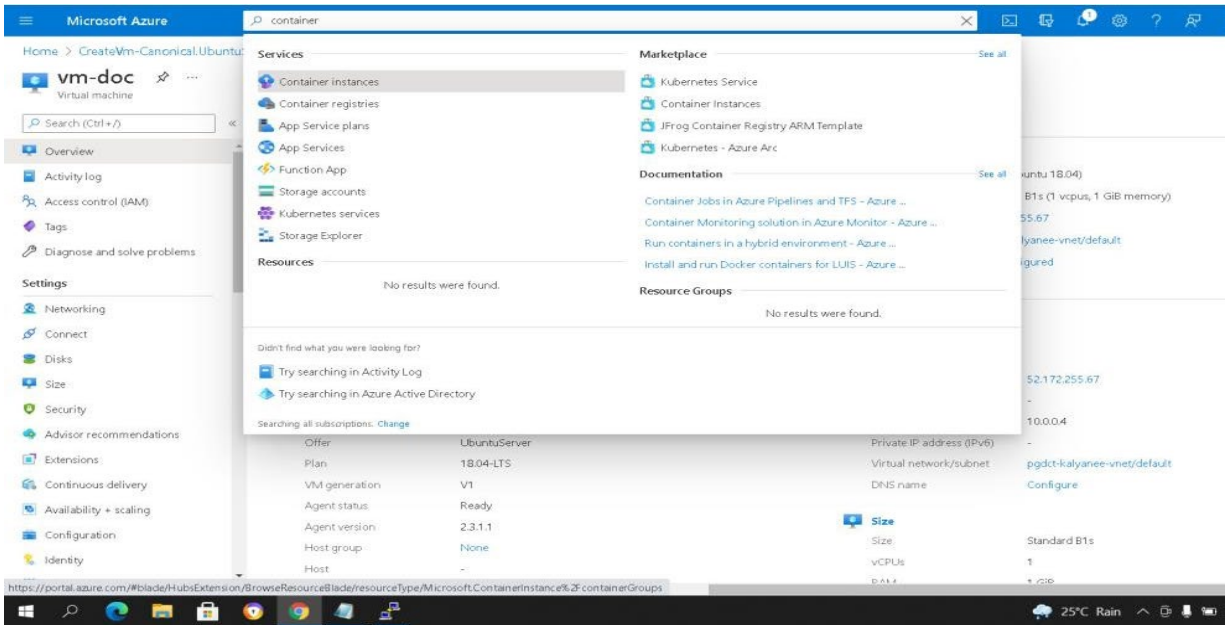


Fig: search container instance

18.Add resource group>>container name>>region

19.Select image source as docker hub

20.Image type public>>image nginx>>os type-Linux

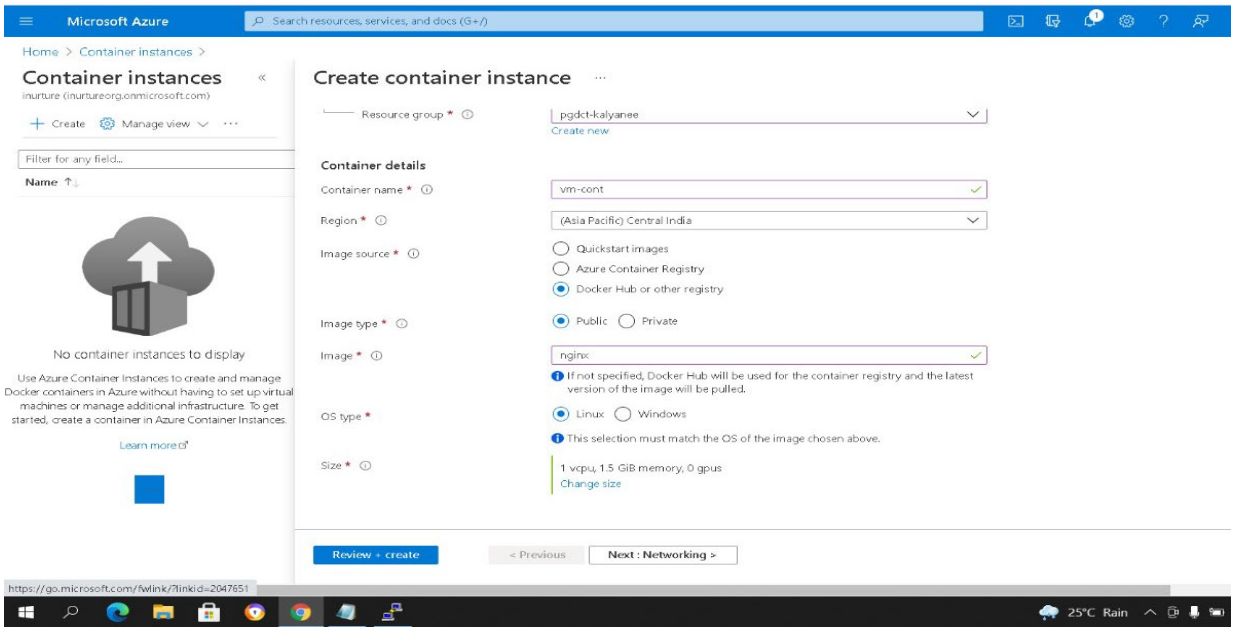


Fig: configure container instance

21.And set everything as default.

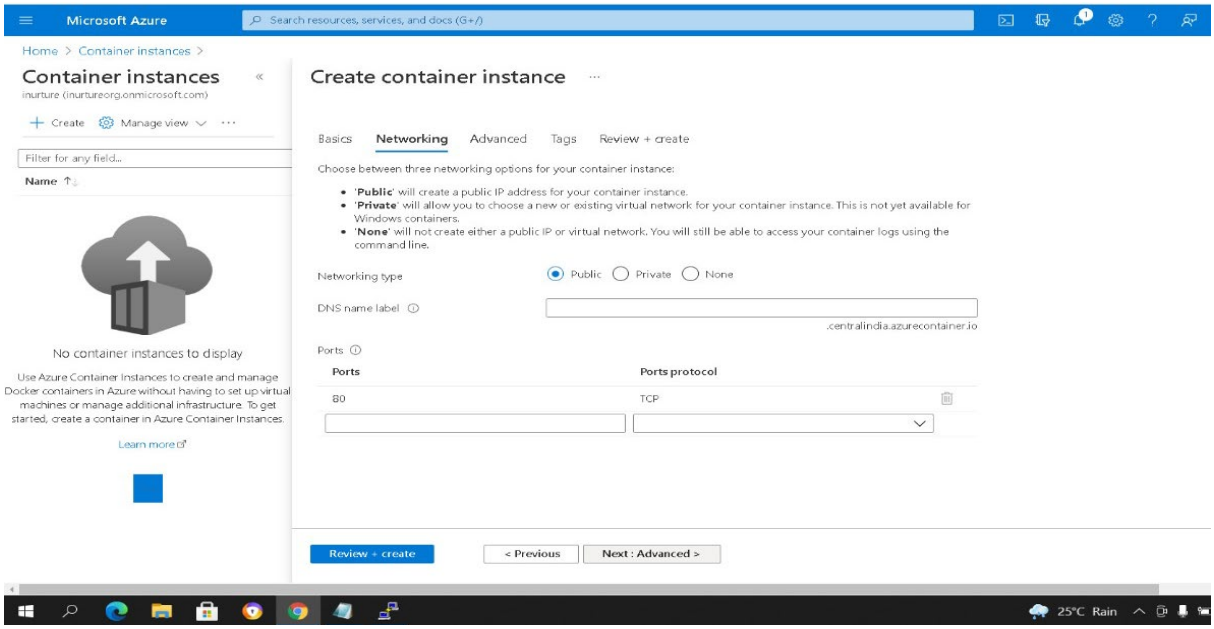


Fig: choose networking type

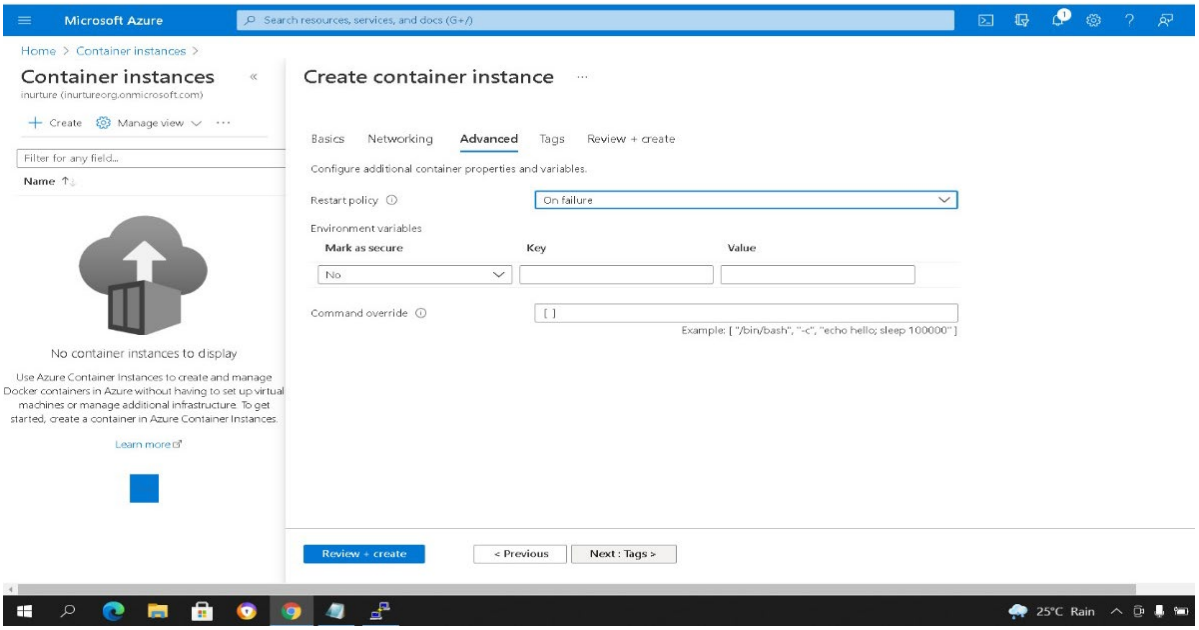


Fig: default advance

22.Here we are created container instance successfully.

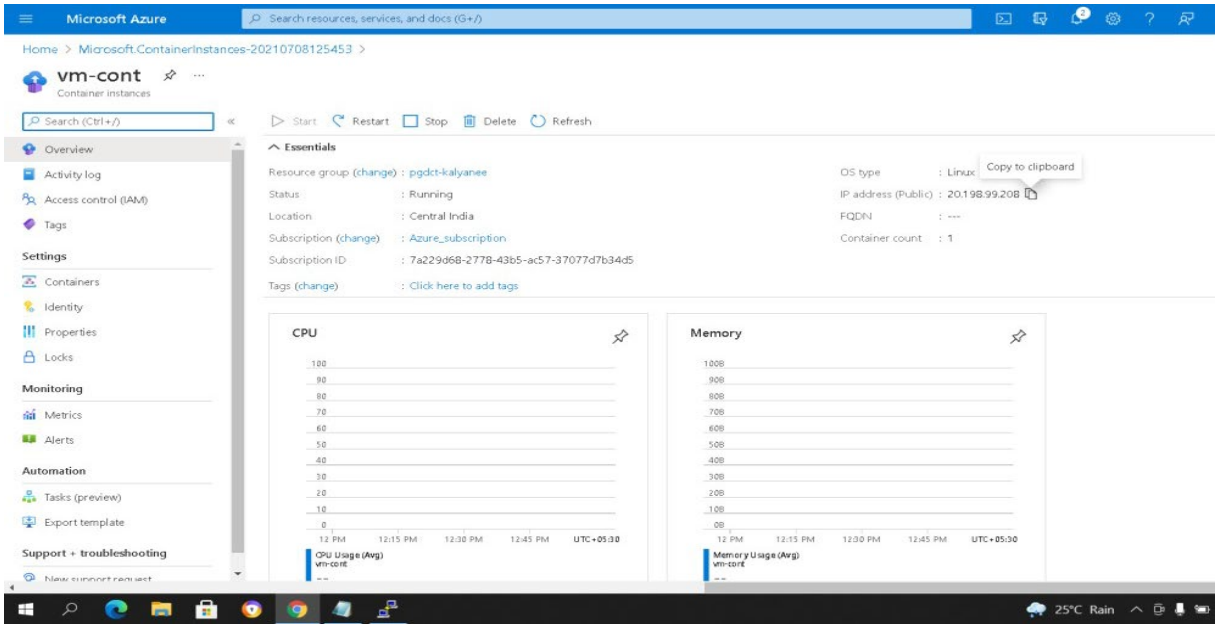


Fig: grab public ip

23.Just grab public ip and browse it you will see nginx web page.

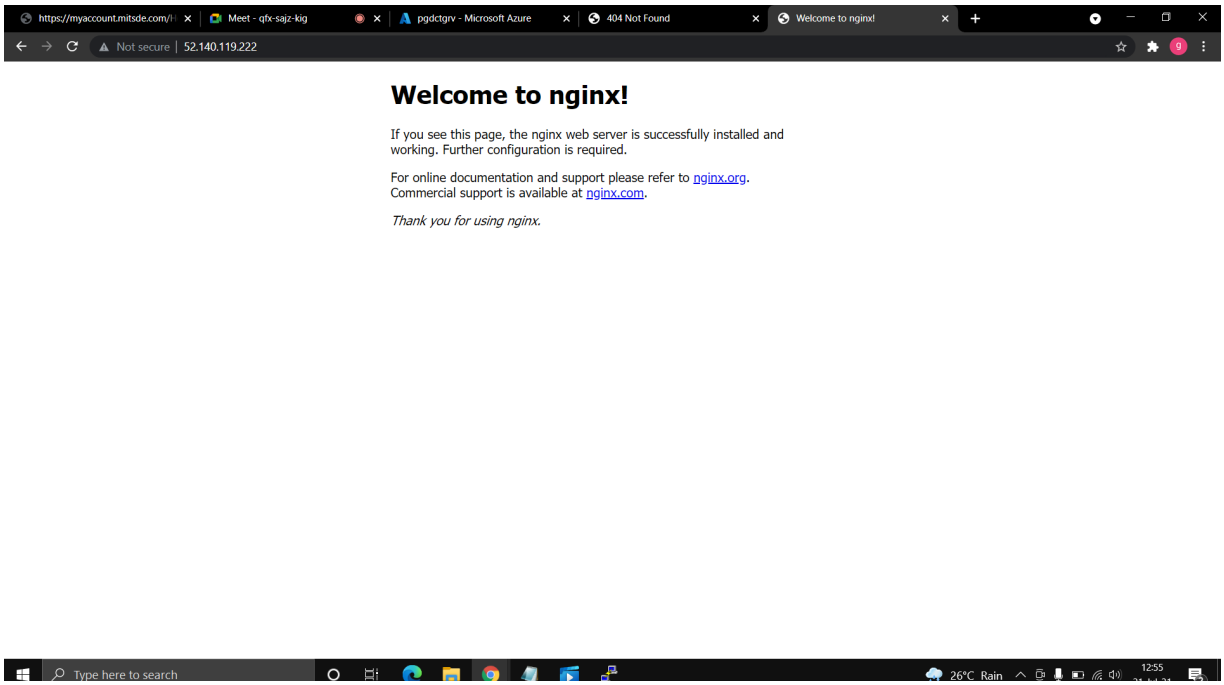
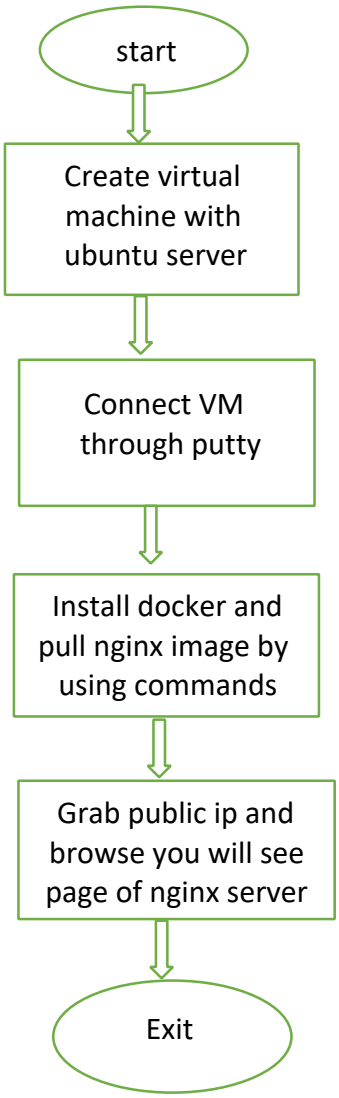
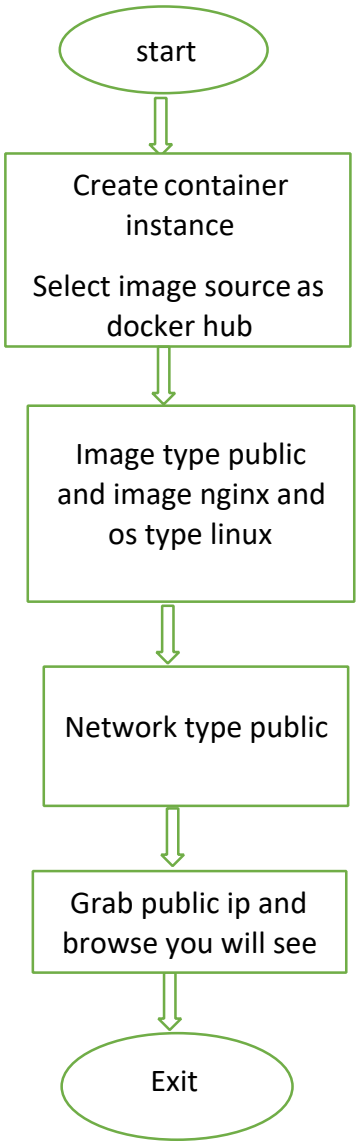


Fig: webpage of nginx server

Flowchart: To install docker on Vm and pull image using docker



To create container instance using docker



Result: We have created virtual machine then connect using putty and install docker engine using commands. We have created container instance using docker hub.

Conclusion: We have successfully Created of Virtual Machine and installed Docker Engine and done with Image Management. We have successfully Launched container instance.

Experiment No.: 6

Name of Experiment: Working with container Instance, Kubernetes cluster, network configuration

Prerequisites: Azure Portal, putty.

Description: **Azure Kubernetes Service:** Azure Kubernetes Service (AKS) simplifies deploying a managed Kubernetes cluster in Azure by offloading the operational overhead to Azure. As a hosted

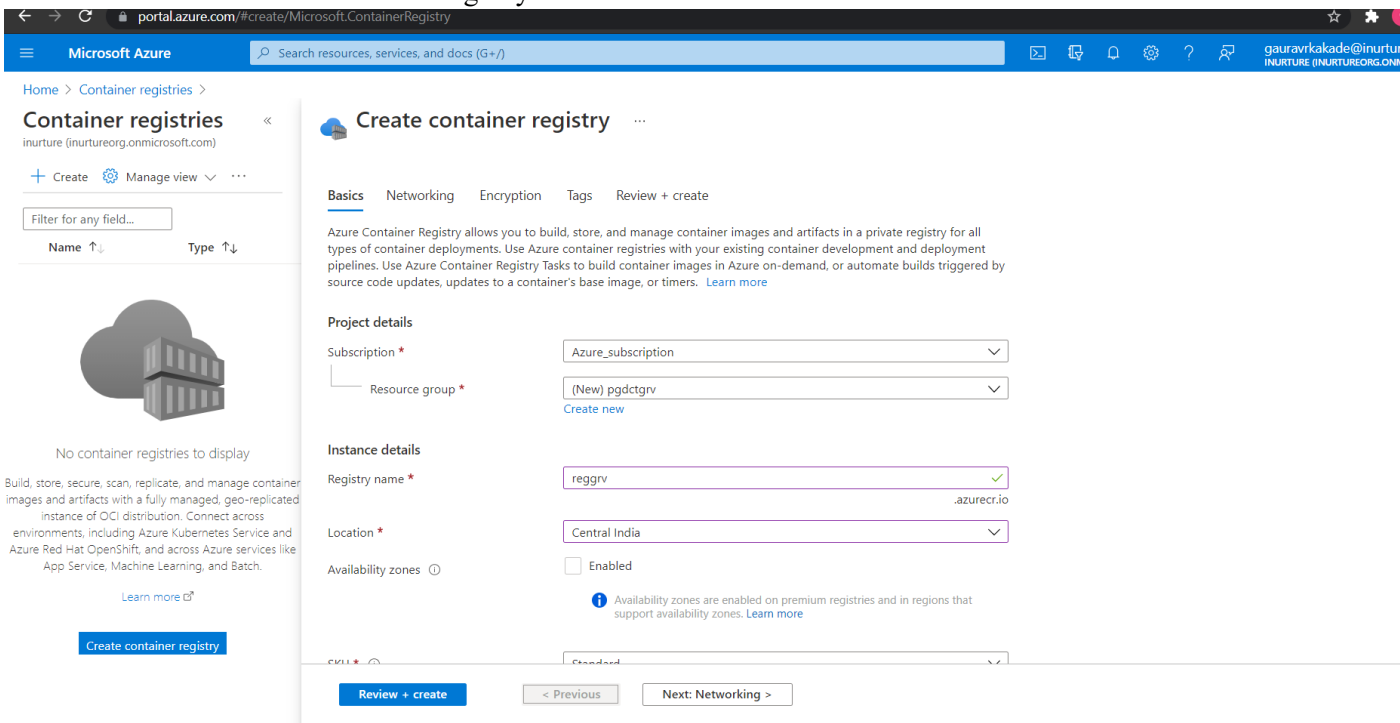
Kubernetes service, Azure handles critical tasks, like health monitoring and maintenance. Since Kubernetes masters are managed by Azure, you only manage and maintain the agent nodes. Thus, AKS is free; you only pay for the agent nodes within your clusters, not for the masters.

A Kubernetes cluster is a **set of node machines for running containerized applications**. If you're running Kubernetes, you're running a cluster. At a minimum, a cluster contains a control plane and one or more compute machines, or nodes.

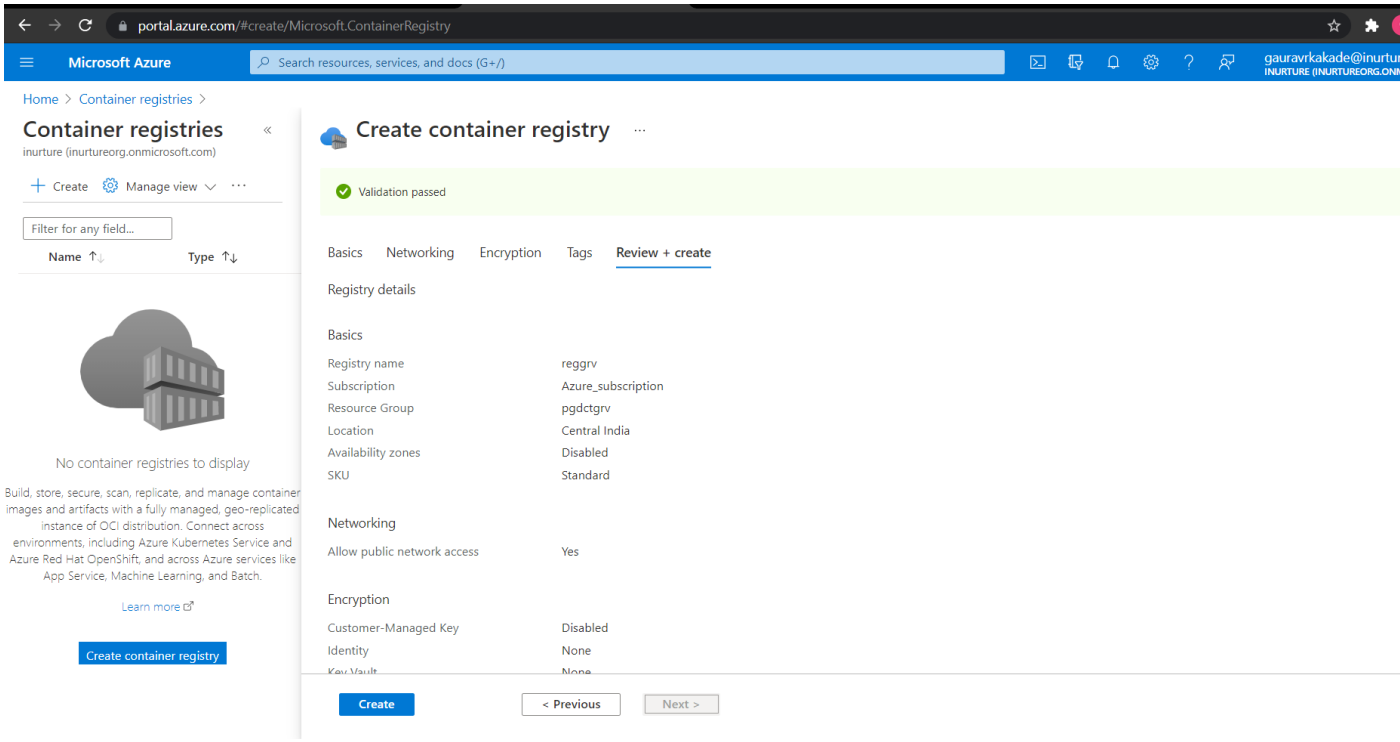
Azure Container Registry is a private registry service for building, storing, and managing container images and related artifacts. Use Docker commands to push a container image into the registry, and finally pull and run the image from your registry.

Algorithm:

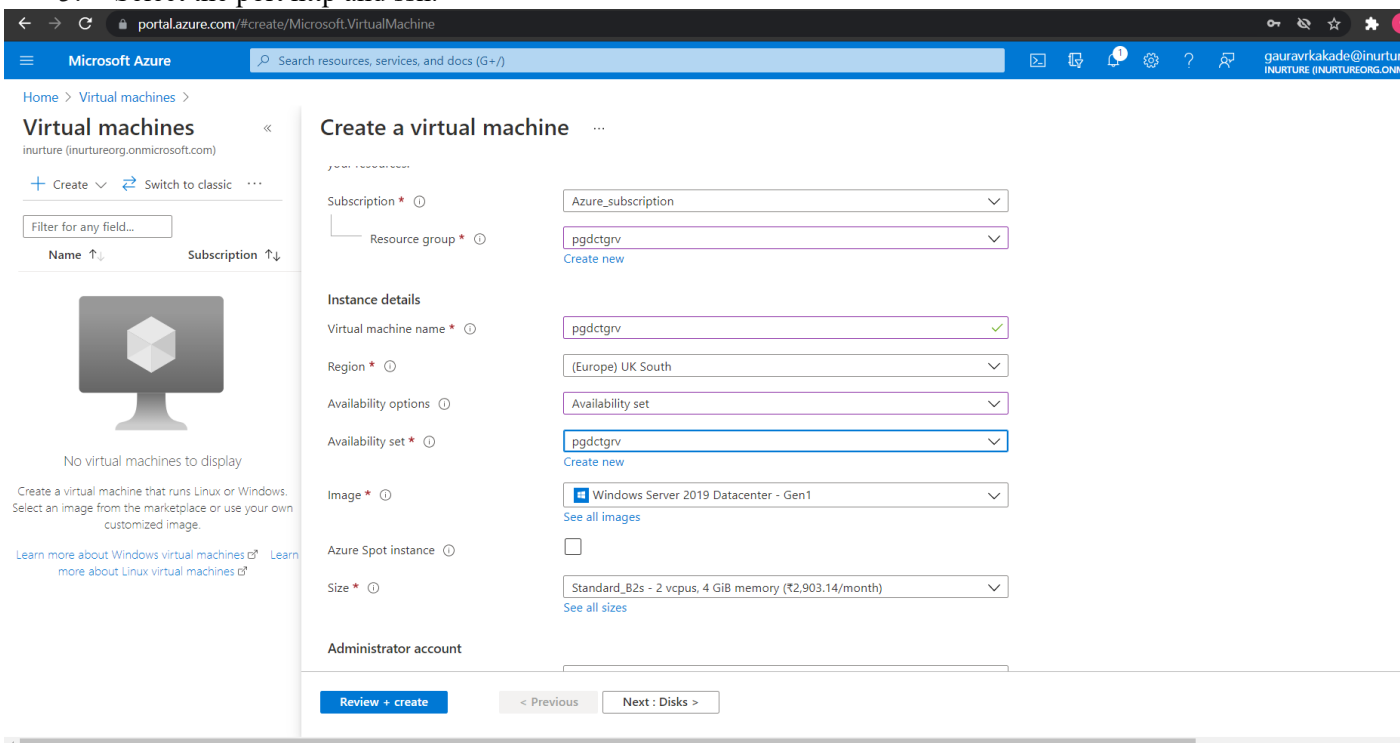
1. Go to container registry.
2. Give the name of container registry and select SKU as basic.



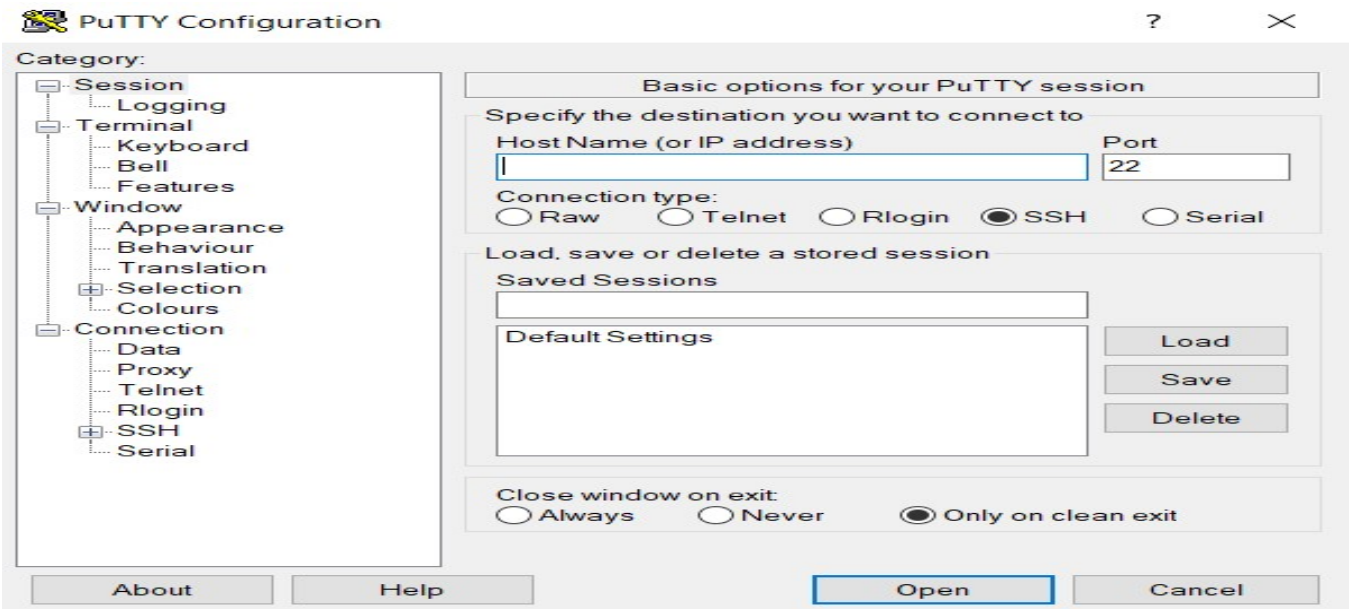
3. Go with default setting and create container registry.



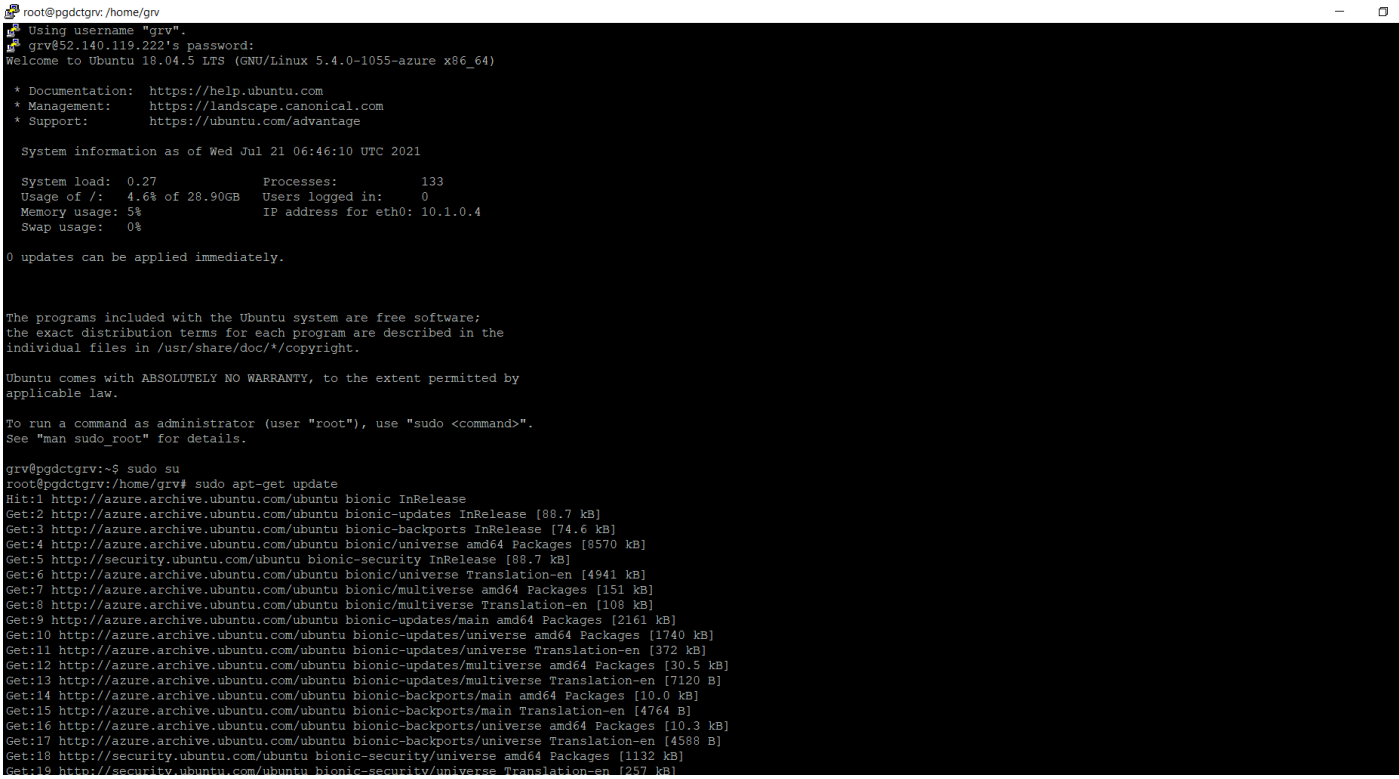
4. Now create the virtual machine with ubuntu server.
5. Select the port http and ssh.



6. Now connect the virtual machine through ssh using putty.



7. Install docker using the commands.



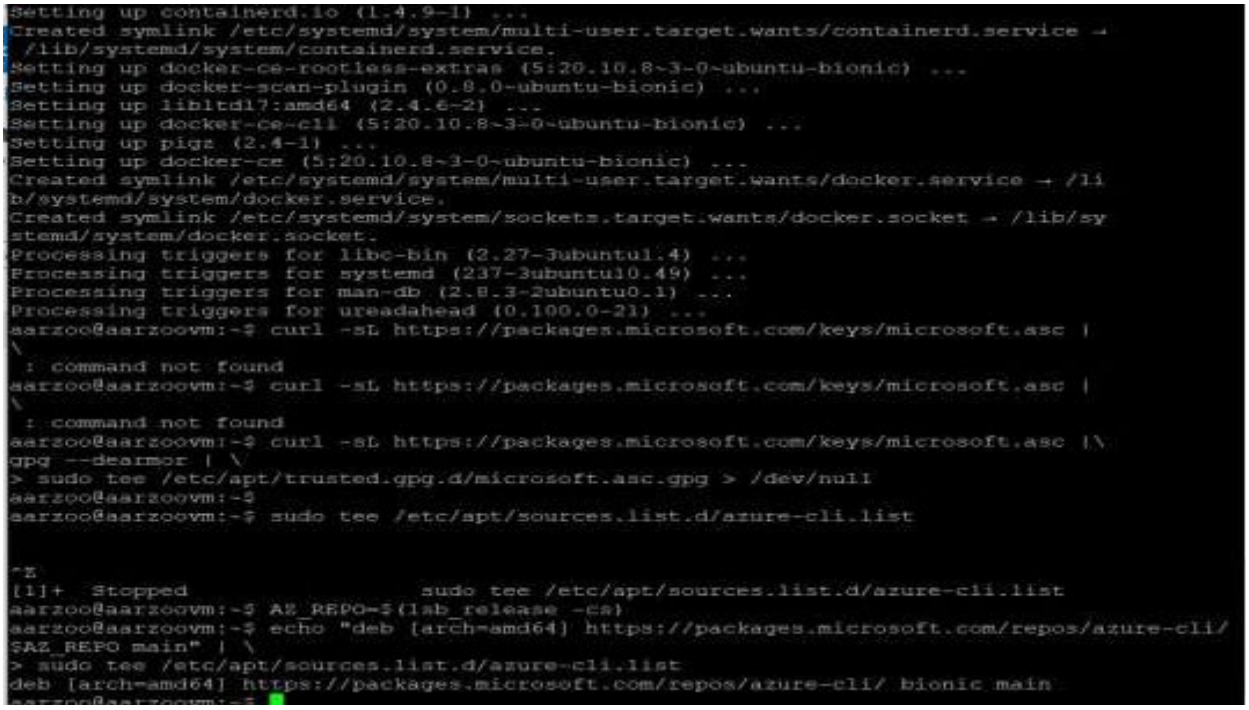
8. Now enter the following command:

curl -sL
https://packages.microsoft.com/keys/microsoft.asc
| \ gpg --dearmor | \

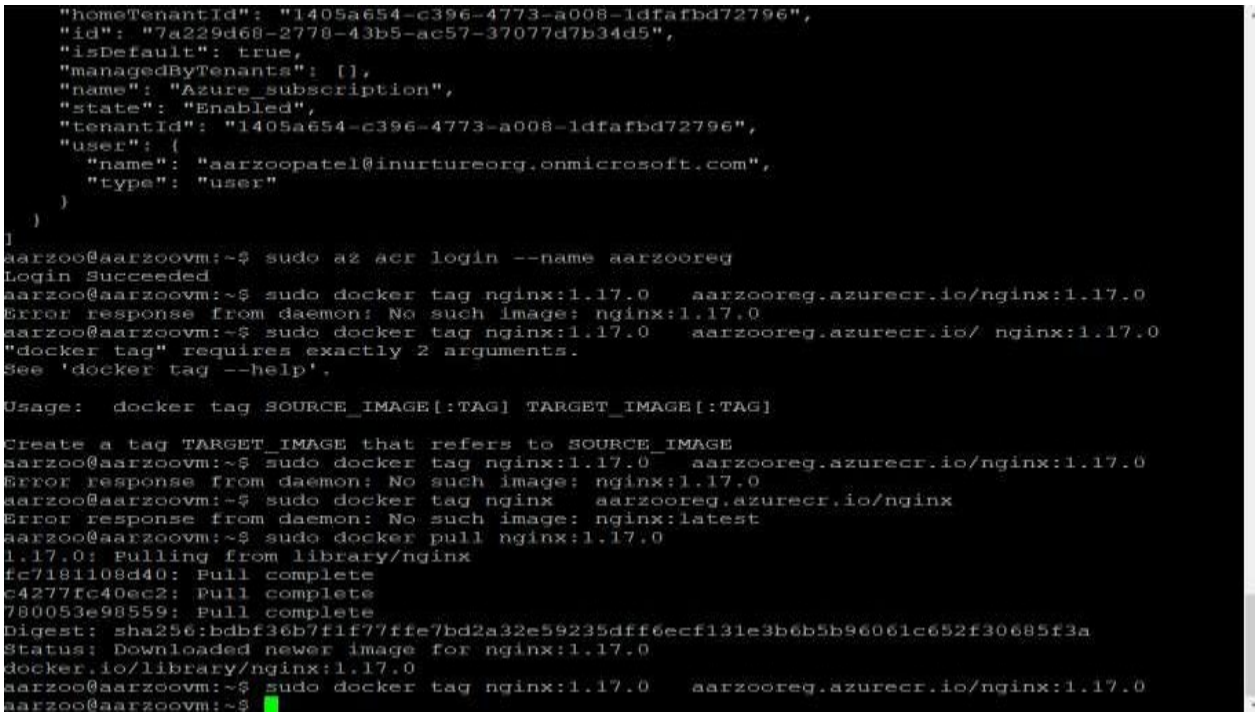
sudo tee /etc/apt/trusted.gpg.d/microsoft.asc.gpg> /dev/null

AZ_REPO=$(lsb_release -cs)

echo "deb [arch=amd64] https://packages.microsoft.com/repos/azure-cli/ $AZ_REPO
main" | \ sudo tee /etc/apt/sources.list.d/azure-cli.list



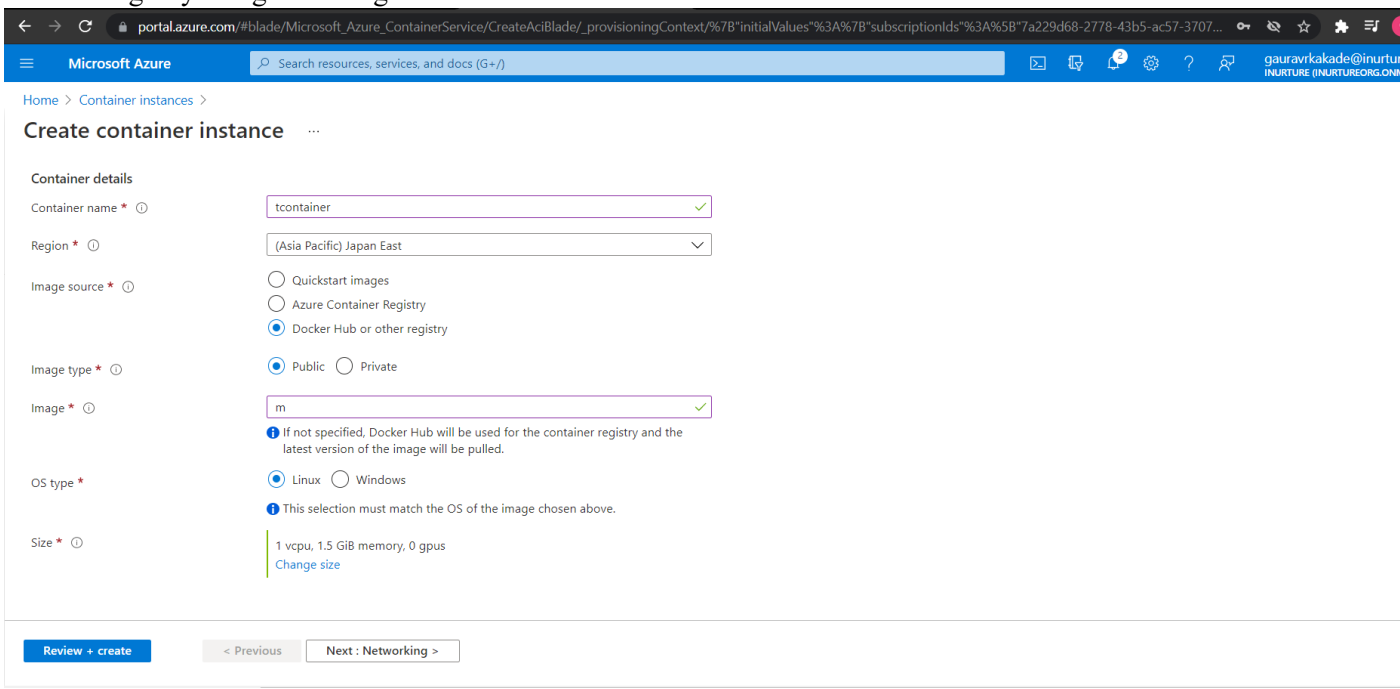
sudo apt-get update sudo apt-get install azure-cli sudoaz login sudoazacr login --name grvregsudo docker pull nginx:1.17.0



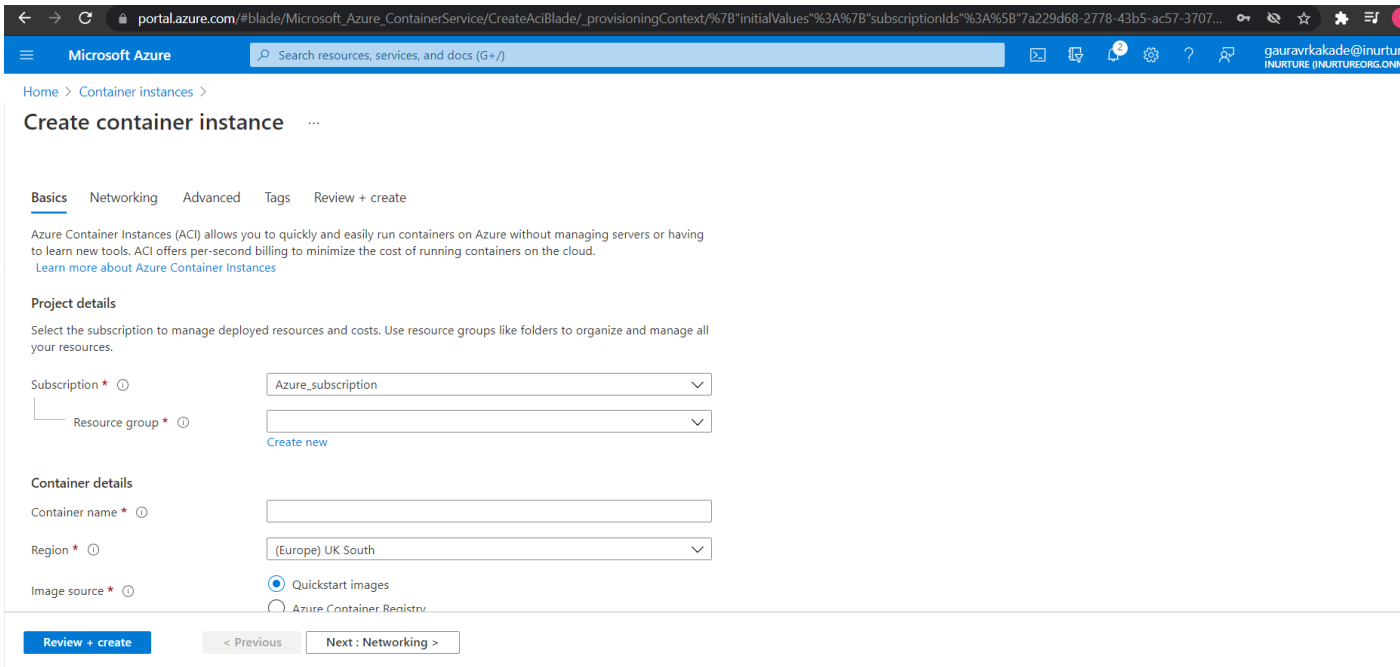
sudo docker tag nginx:1.17.0 grvreg.azurecr.io/nginx:1.17.0 sudo docker push grvreg.azurecr.io/nginx:1.17.0

```
Digest: sha256:bdbf36b7f1f77ffe7bd2a32e59235dff6ecf131e3b6b5b96061c652f30685f3a
Status: Downloaded newer image for nginx:1.17.0
docker.io/library/nginx:1.17.0
aarzoo@aarzoovm:~$ sudo docker tag nginx:1.17.0   aarzooreg.azurecr.io/nginx:1.17.0
aarzoo@aarzoovm:~$ sudo docker push aarzooreg.azurecr.io/nginx:1.17.0
The push refers to repository [aarzooreg.azurecr.io/nginx]
d7acf794921f: Pushed
d9569ca04881: Pushed
cf5b3c6798f7: Pushed
1.17.0: digest: sha256:079aa93463d2566b7a81cbdf856afc6d4d2a6f9100ca3bcbecf24ade92c9a7fe si
ze: 948
```

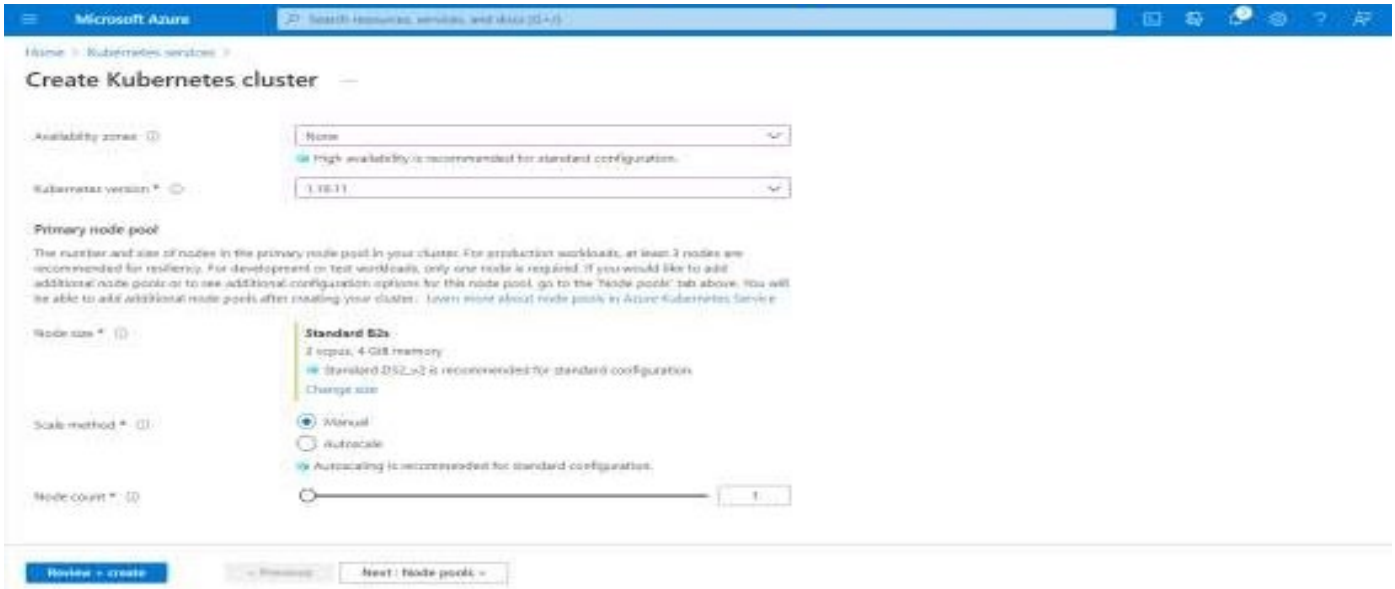
9. Now go to container instance.
10. Select image source as container registry>>now open power shell and enable container registry using following command:



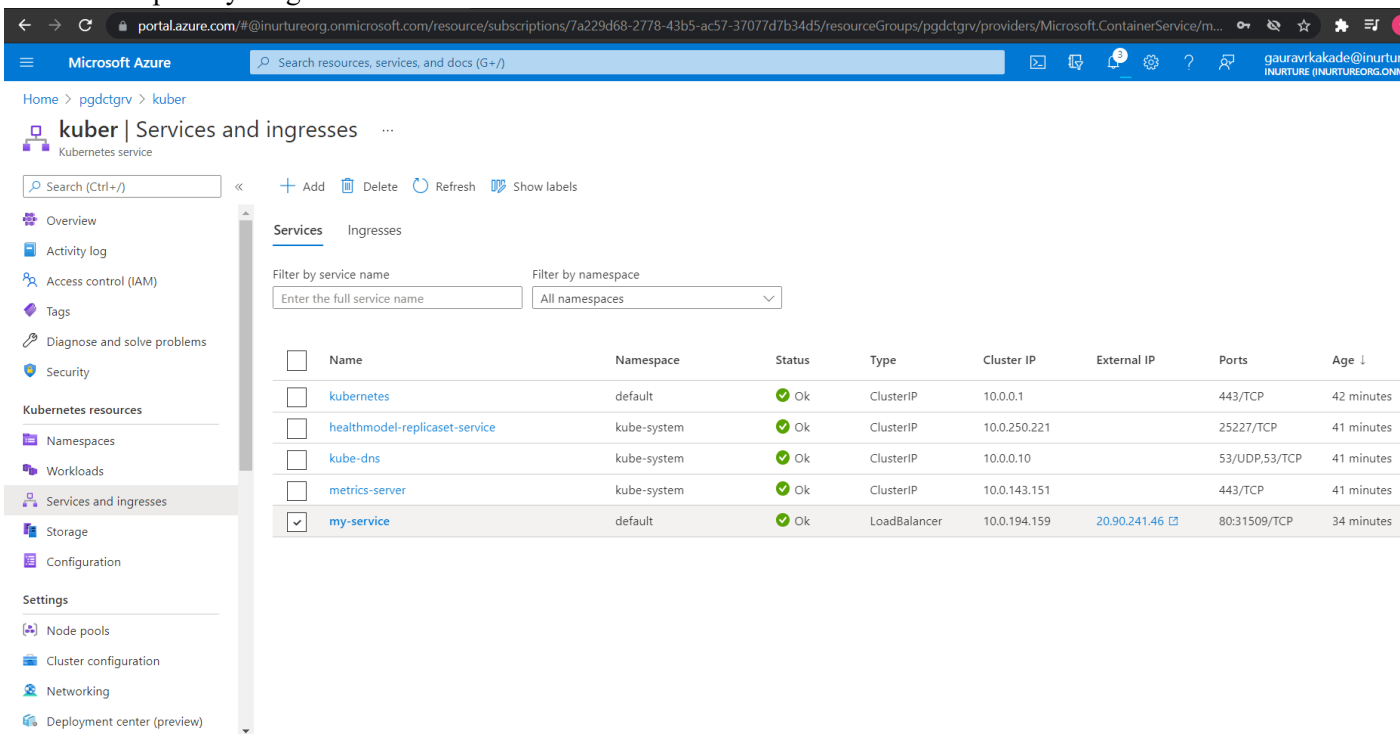
11. Create the container instance.



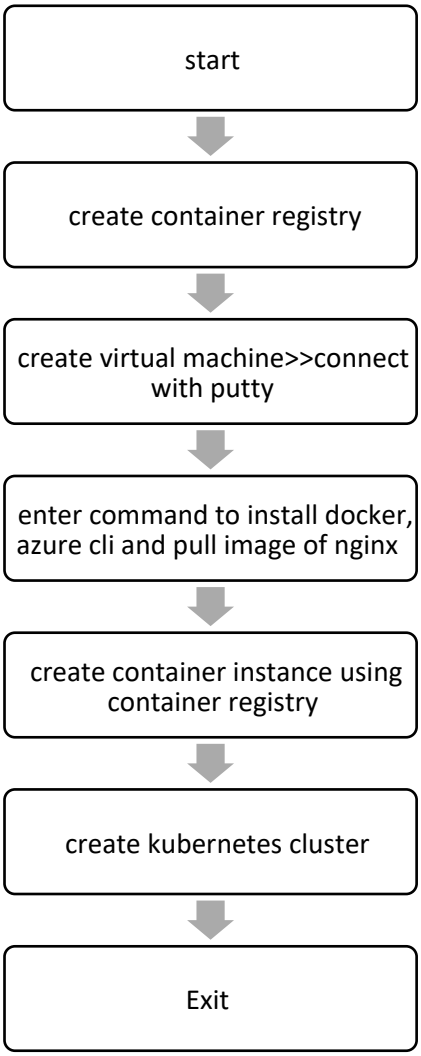
12. Go to Kubernetes services.
13. Select availability zone as none and Kubernetes version as 1.19.11.
14. Select method as manual and node count as 1.



15. In authentication felid tick AKS-managed Azure Active Directory.
16. Keep everything default and create the Kubernetes cluster.



Flowchart:



Result: It has been seen that using container registry we can create container instance and can also push the image in container registry using azure cli and have created Kubernetes cluster.

Conclusion: we have successfully created the container instance using container registry and Kubernetes cluster.

Experiment no :7

AIM: Azure Networking, Address space, & attaching secondary NIC to Virtual Machine.

Prerequisites: Azure portal.

Description:

Azure Networking: The networking services in Azure provide a variety of networking capabilities that can be used together or separately. Some of the services are mentioned below:

- **Connectivity services:** Connect Azure resources and onpremises resources using any or a combination of these networking services in Azure - Virtual Network (VNet), Virtual WAN, ExpressRoute, VPN Gateway, Virtual network NAT Gateway, Azure DNS, Peering service, and Azure Bastion.

- **Application protection services:** Protect your applications using any or a combination of these networking services in Azure - Load Balancer, Private Link, DDoS protection, Firewall, Network Security Groups, Web Application Firewall, and Virtual Network Endpoints.

- **Application delivery services:** Deliver applications in the Azure network using any or a combination of these networking services in Azure - Content Delivery Network (CDN), Azure Front Door Service, Traffic Manager, Application Gateway, Internet Analyzer, and Load Balancer.

- **Network monitoring:** Monitor your network resources using any or a combination of these networking services in Azure - Network Watcher, ExpressRoute Monitor, Azure Monitor, or VNet Terminal Access Point (TAP).

Public IP address: Public IP addresses allow Internet resources to communicate inbound to Azure resources. Public IP addresses enable Azure resources to communicate to Internet and public-facing Azure services. The address is dedicated to the resource, until it's unassigned by you.

Address Space: Address space is the amount of memory allocated for all possible addresses for a computational entity, such as a device, a file, a server, or a networked computer. Address space may refer to a range of either physical or virtual addresses accessible to a processor or reserved for a process.

Network Interface: A Network Interface (NIC) is an interconnection between a Virtual Machine and the underlying software network. An Azure Virtual Machine (VM) has one or more network

interfaces (NIC) attached to it. Any NIC can have one or more static or dynamic public and private IP addresses assigned to it.

~ Configuring the network interface:

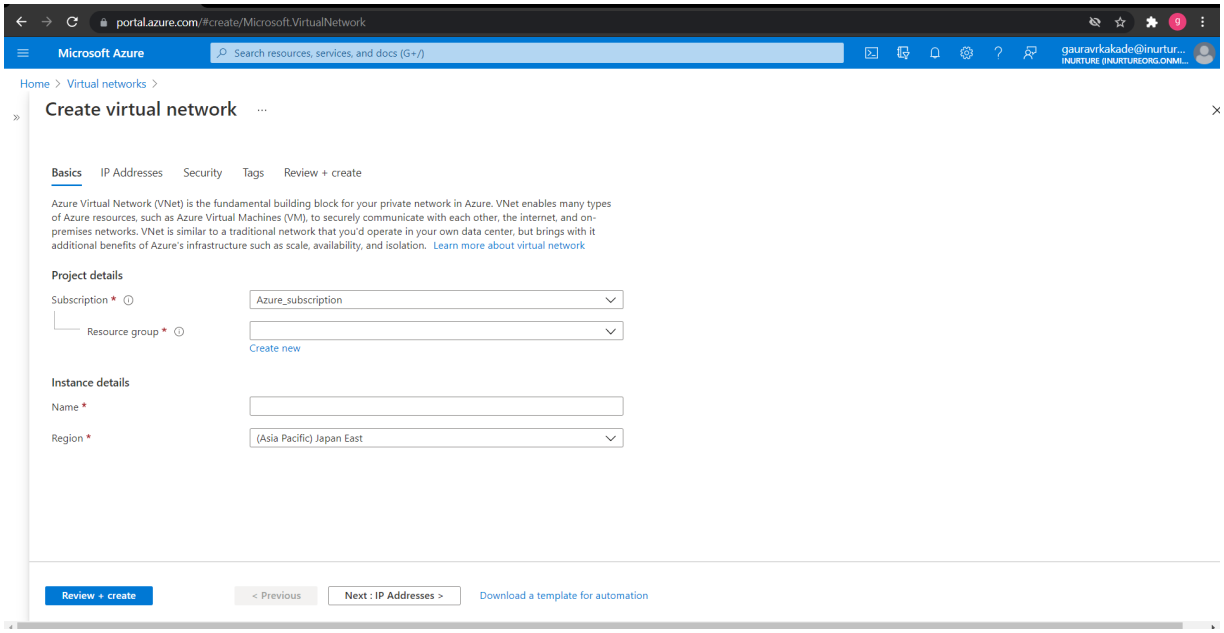
Virtual network & subnets: we can attach a network interface to a VNet and Subnet, and once we deployed a NIC into a VNet, we can't change it.

IP configuration: Public and private IP addresses will be assigned at the NIC level. Primary & secondary IP configurations.

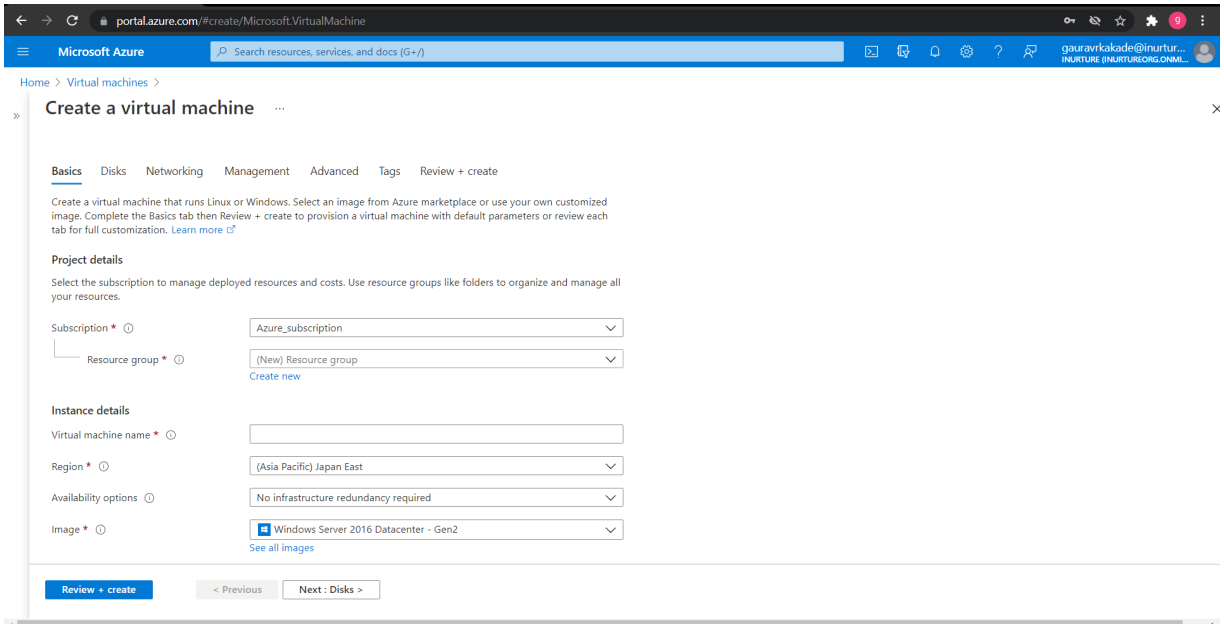
ALGORITHM: We will see how to attach as secondary NIC to a

virtual machine.

1.Create a virtual Network.



2.Create a Virtual Machine.

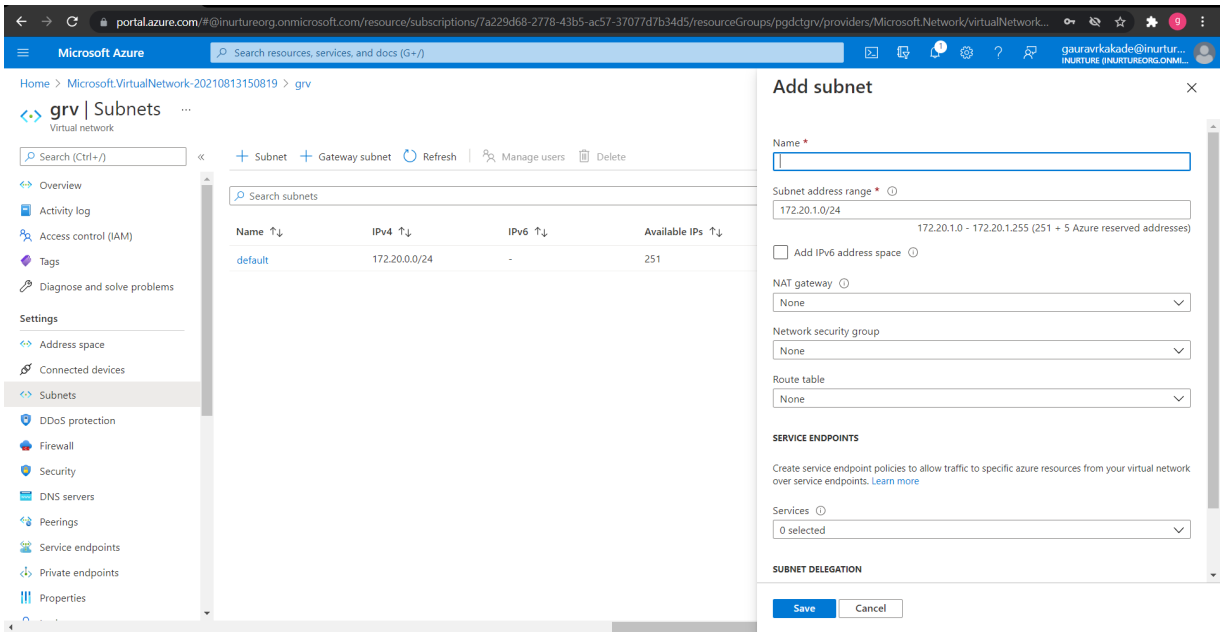


3. To see the default NIC go to networking and then go to network interface.

4. Now to attach the secondary NIC to virtual machine, we have to stop the virtual machine.

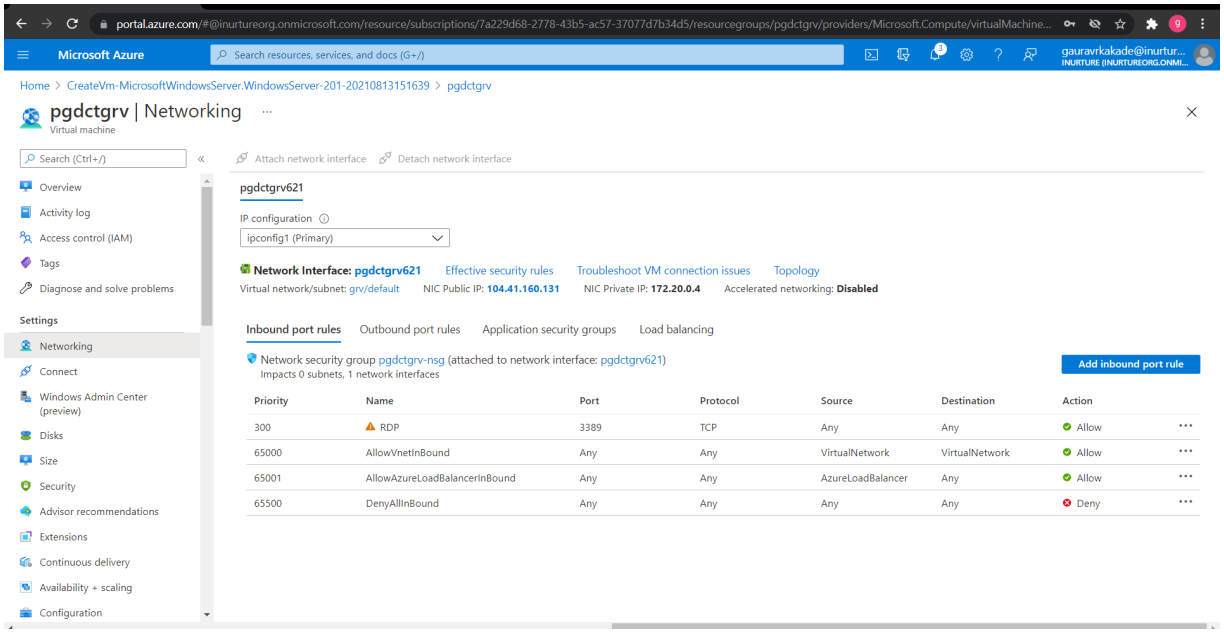
5. Go to overview of virtual machine and stop the virtual machine.

6. Create a new subnet.

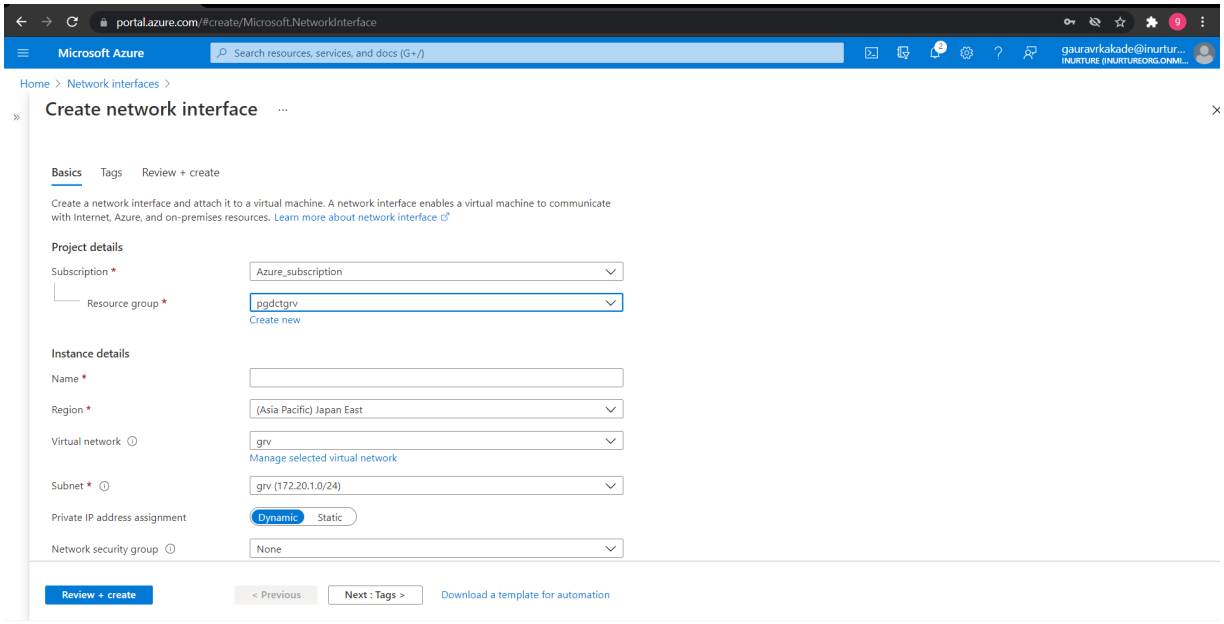


7. Now go to the VM and then to Networking.

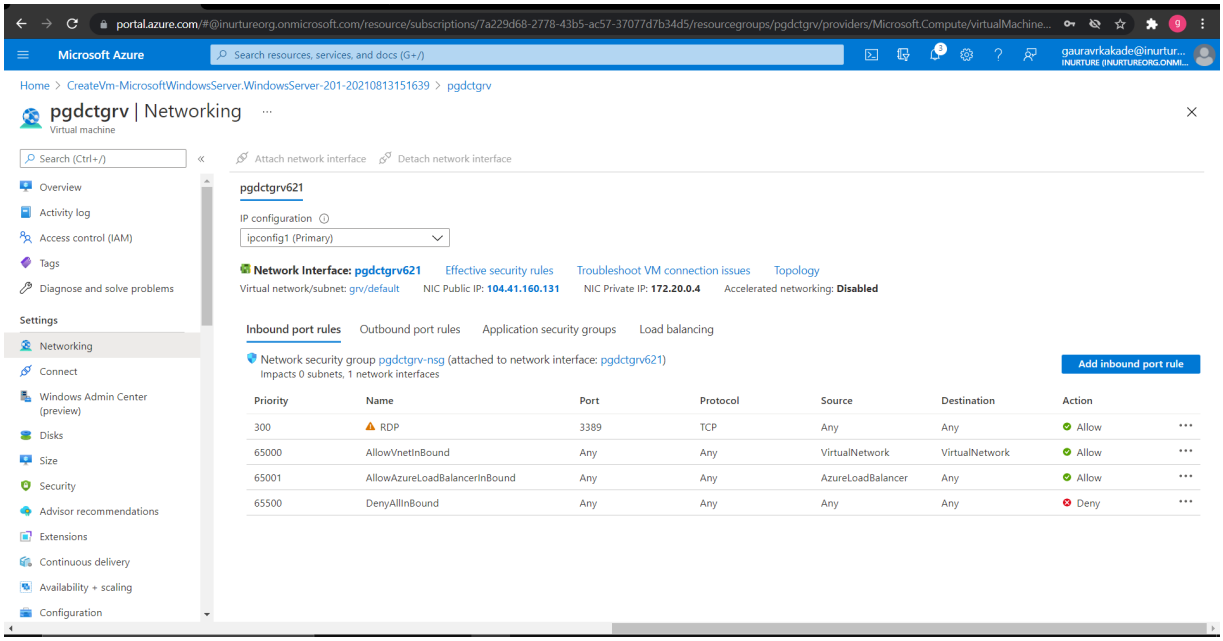
8. Click on attach network interface and create new interface.



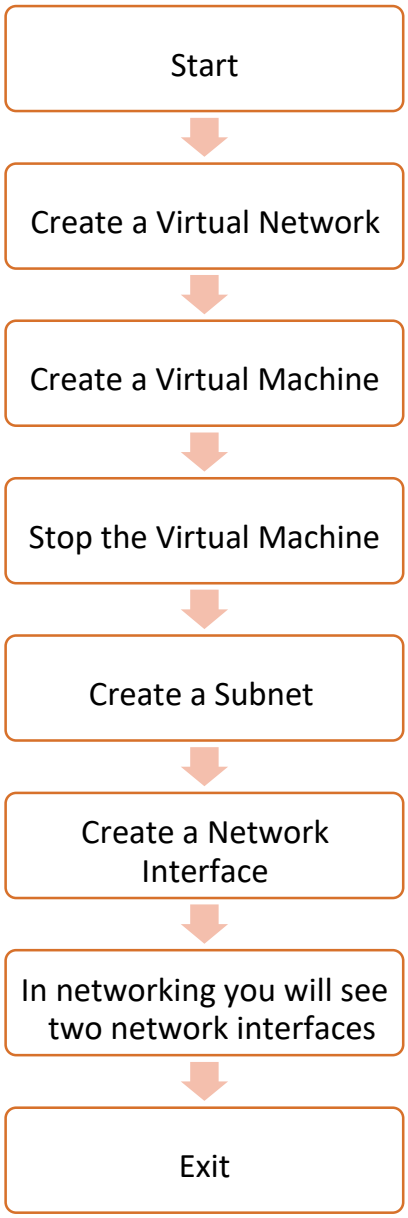
9. Enter the name of networking interface and keep everything default and then add the subnet that you have created and click on network interface.



10. Now in networking you can see the two-network interface.


portal.azure.com/#@inurtureorg.onmicrosoft.com/resource/subscriptions/7a229d68-2778-43b5-ac57-37077d7b34d5/resourcegroups/pgdctgrv/providers/Microsoft.Compute/virtualMachine...
Microsoft Azure
Search resources, services, and docs (G+/)
gauravrkakade@inurtur...
INURTURE (INURTUREORG.ONMI...
Home > CreateVm-MicrosoftWindowsServer.WindowsServer-201-20210813151639 > pgdctgrv
pgdctgrv | Networking
Virtual machine
Search (Ctrl+/)
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Settings
Networking
Connect
Windows Admin Center (preview)
Disks
Size
Security
Advisor recommendations
Extensions
Continuous delivery
Availability + scaling
Configuration
Attach network interface
Detach network interface
pgdctgrv621
IP configuration
ipconfig1 (Primary)
Network Interface: pgdctgrv621
Effective security rules
Troubleshoot VM connection issues
Topology
Virtual network/subnet: grv/default
NIC Public IP: 104.41.160.131
NIC Private IP: 172.20.0.4
Accelerated networking: Disabled
Inbound port rules
Outbound port rules
Application security groups
Load balancing
Network security group pgdctgrv-nsg (attached to network interface: pgdctgrv621)
Impacts 0 subnets, 1 network interfaces
Add inbound port rule
Priority
Name
Port
Protocol
Source
Destination
Action
300
RDP
3389
TCP
Any
Any
Allow
65000
AllowVnetInBound
Any
Any
VirtualNetwork
VirtualNetwork
Allow
65001
AllowAzureLoadBalancerInBound
Any
Any
AzureLoadBalancer
Any
Allow
65500
DenyAllInBound
Any
Any
Any
Any
Deny

FLOWCHART:



Result: we have seen the concept of azure networking, address space. By default, a network interface card is always associated with a virtual machine but if we want to attach a secondary NIC we can also do that.

Conclusion: we have successfully attached the network interface card.

Experiment No.: 8

Name of Experiment: Creation of VNET & Security rule management of Virtual Machine level and subnet level.

Prerequisites: Azure portal, RDP.

Description:

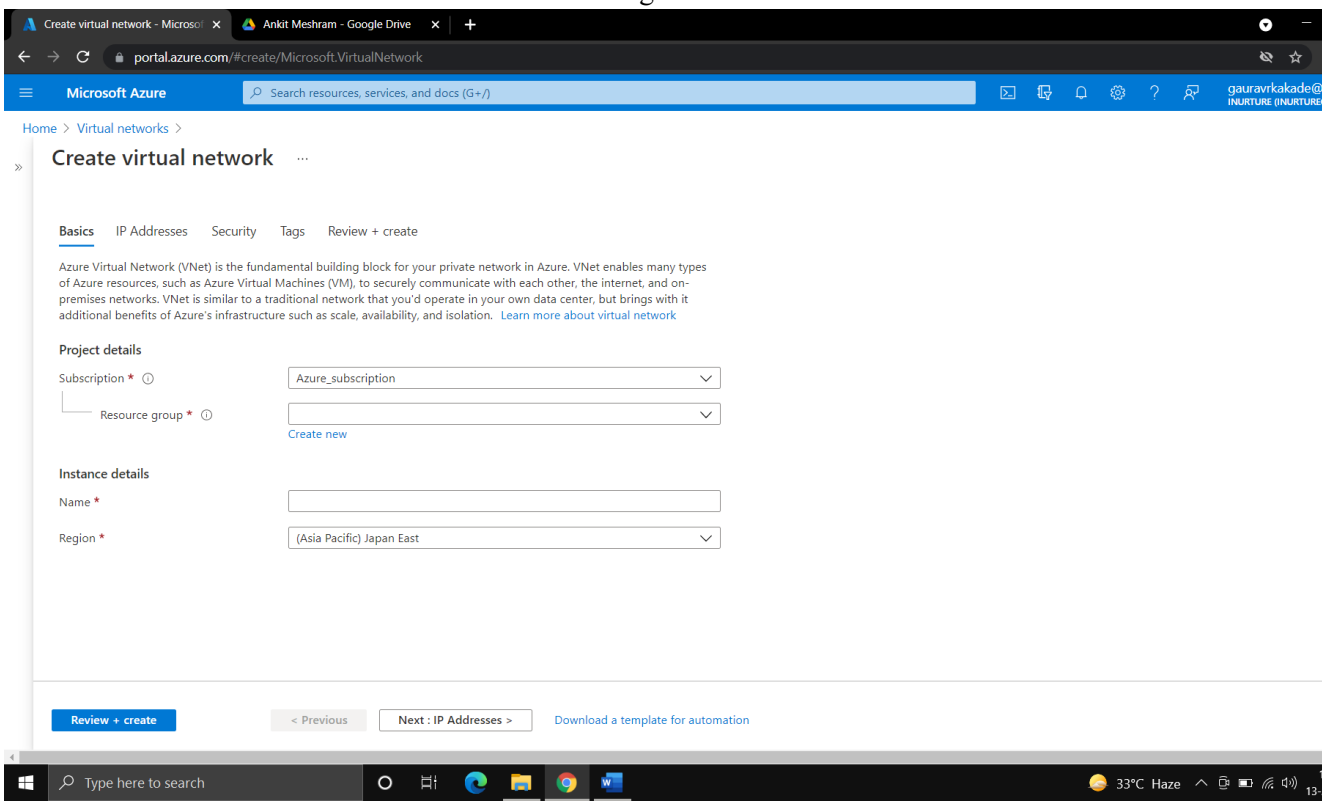
Virtual Network: Azure Virtual Network (VNET) is the fundamental building block for your private network in Azure. VNET enables many types of Azure resources, such as Azure Virtual Machines (VM), to securely communicate with each other, the internet, and on-premises networks. VNET is similar to a traditional network that you'd operate in your own data centre, but brings with it additional benefits of Azure's infrastructure such as scale, availability, and isolation.

Security Group: You can use an Azure network security group to filter network traffic to and from Azure resources in an Azure virtual network. A network security group contains security rules that allow or deny inbound network traffic to, or outbound network traffic from, several types of Azure resources. For each rule, you can specify source and destination, port, and protocol.

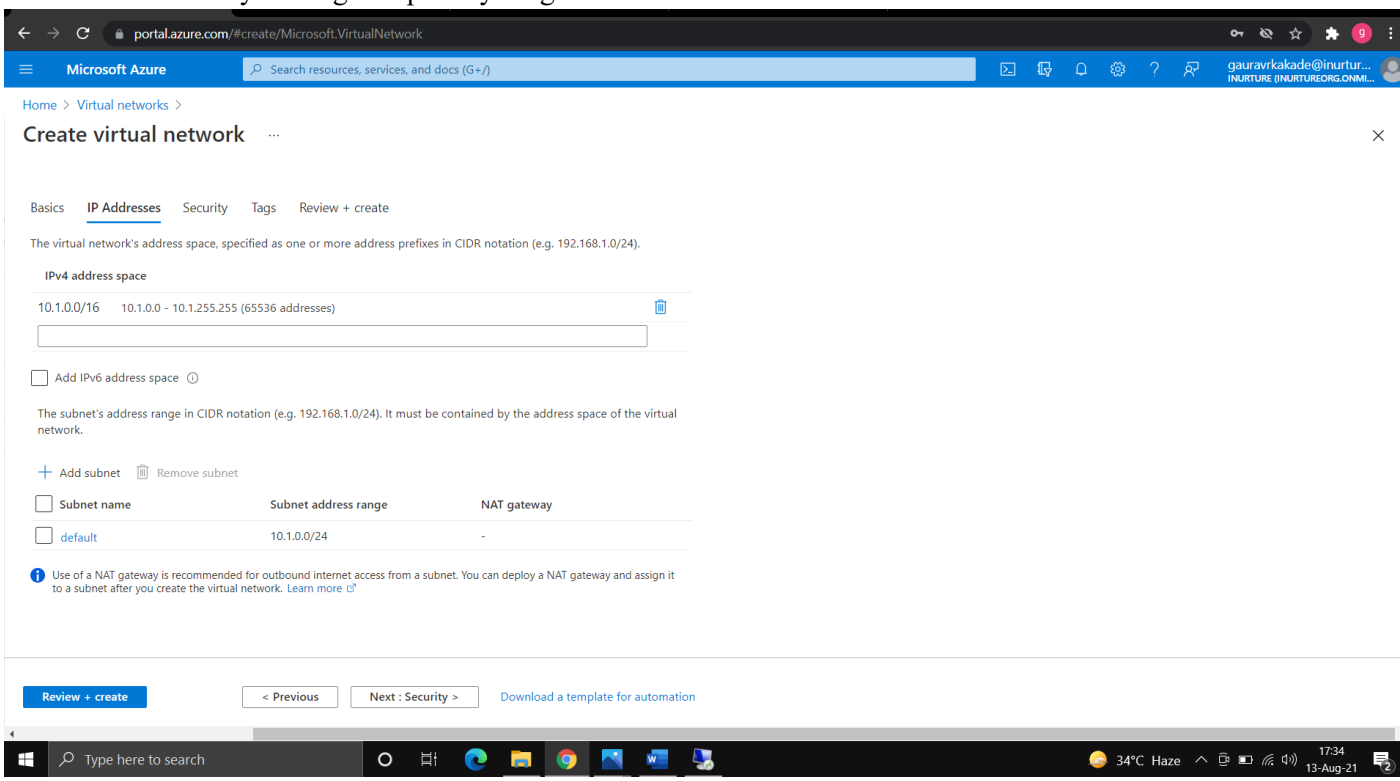
Algorithm:

Steps to create virtual network (VNET):

1. Go to virtual network.
2. Create new resource group.
3. Give the name of virtual network and select the region.



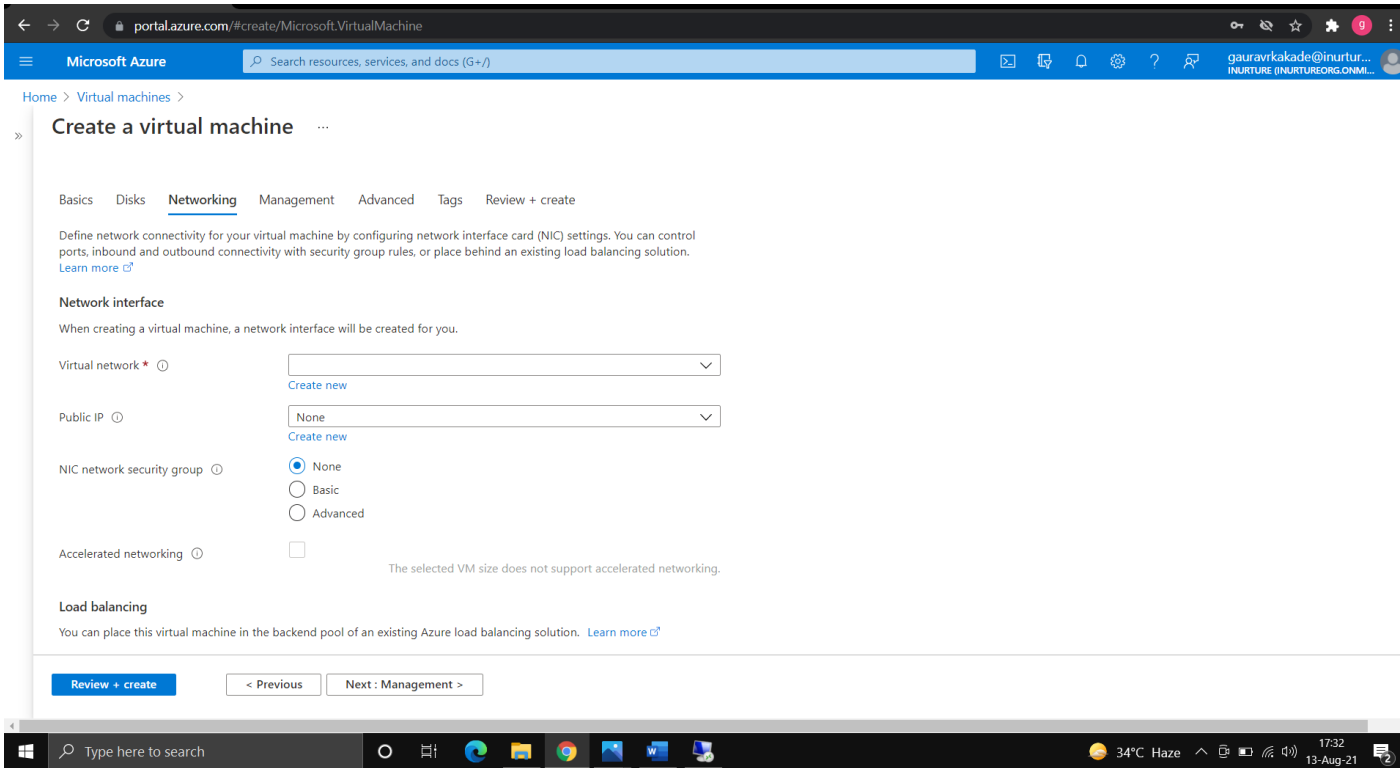
4. Go with the default IP address and subnet.
5. In security and tags keep everything default.



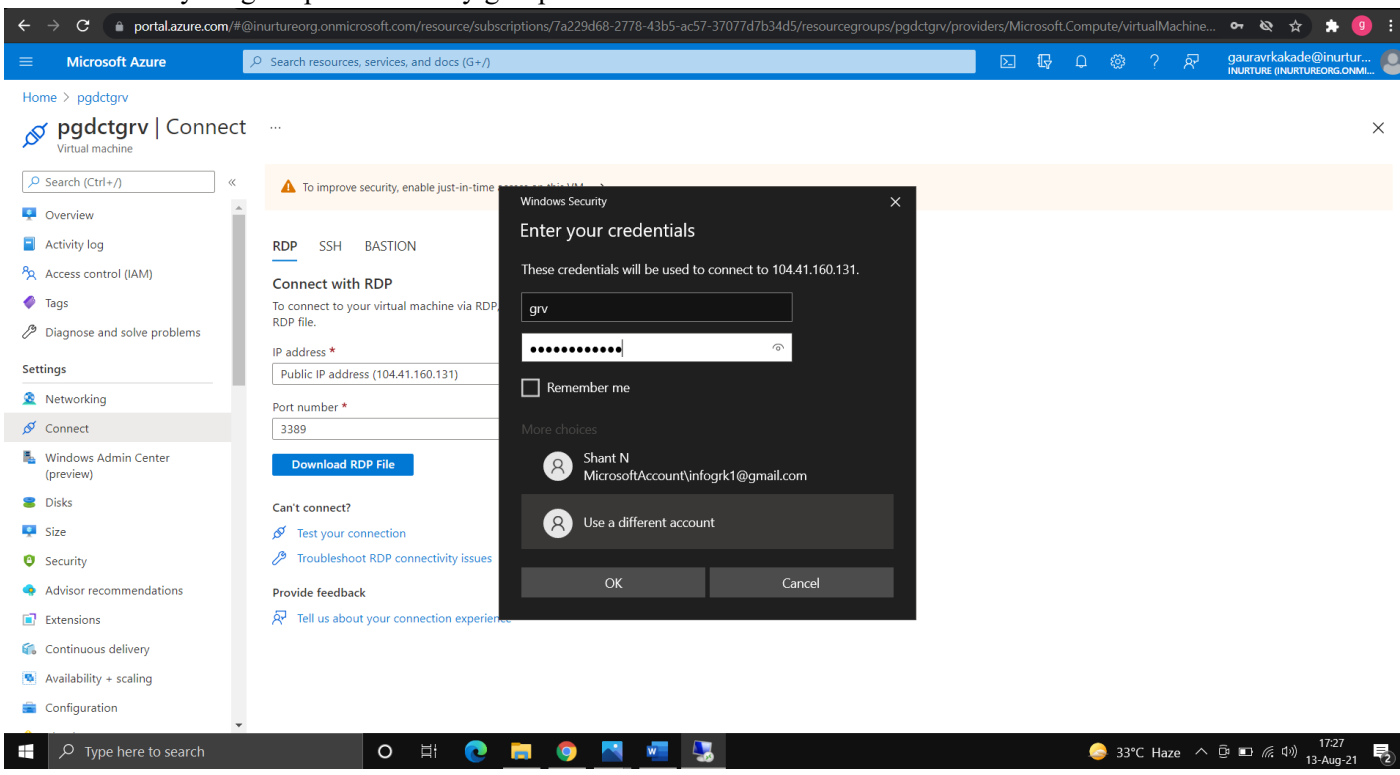
6. Create the virtual network.

Steps to create security group and attach it to virtual machine:

1. Create a virtual machine.
2. In networking select network security group as none.

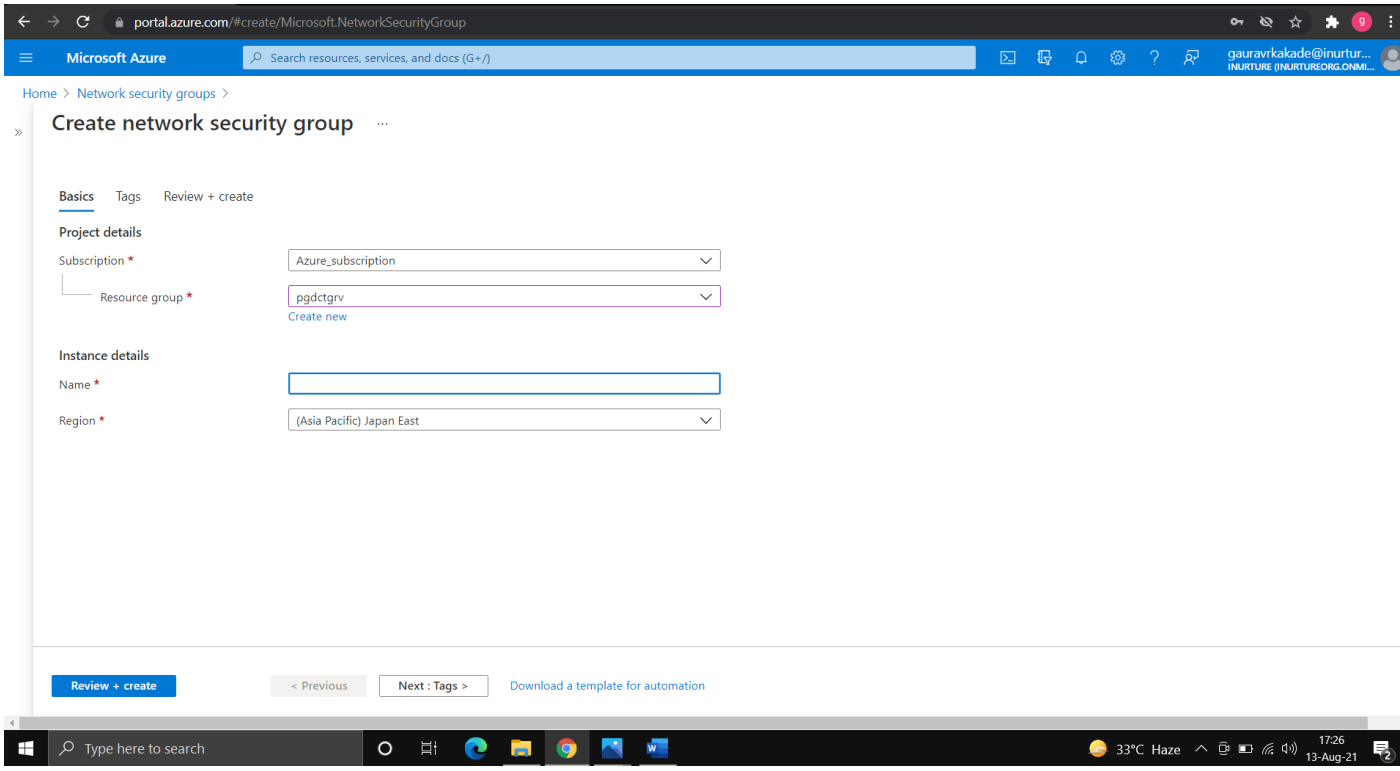


3. Keep everything default and create the virtual machine.
4. Now try to connect with the virtual machine through RDP, it will connect because in azure everything is open in security group.

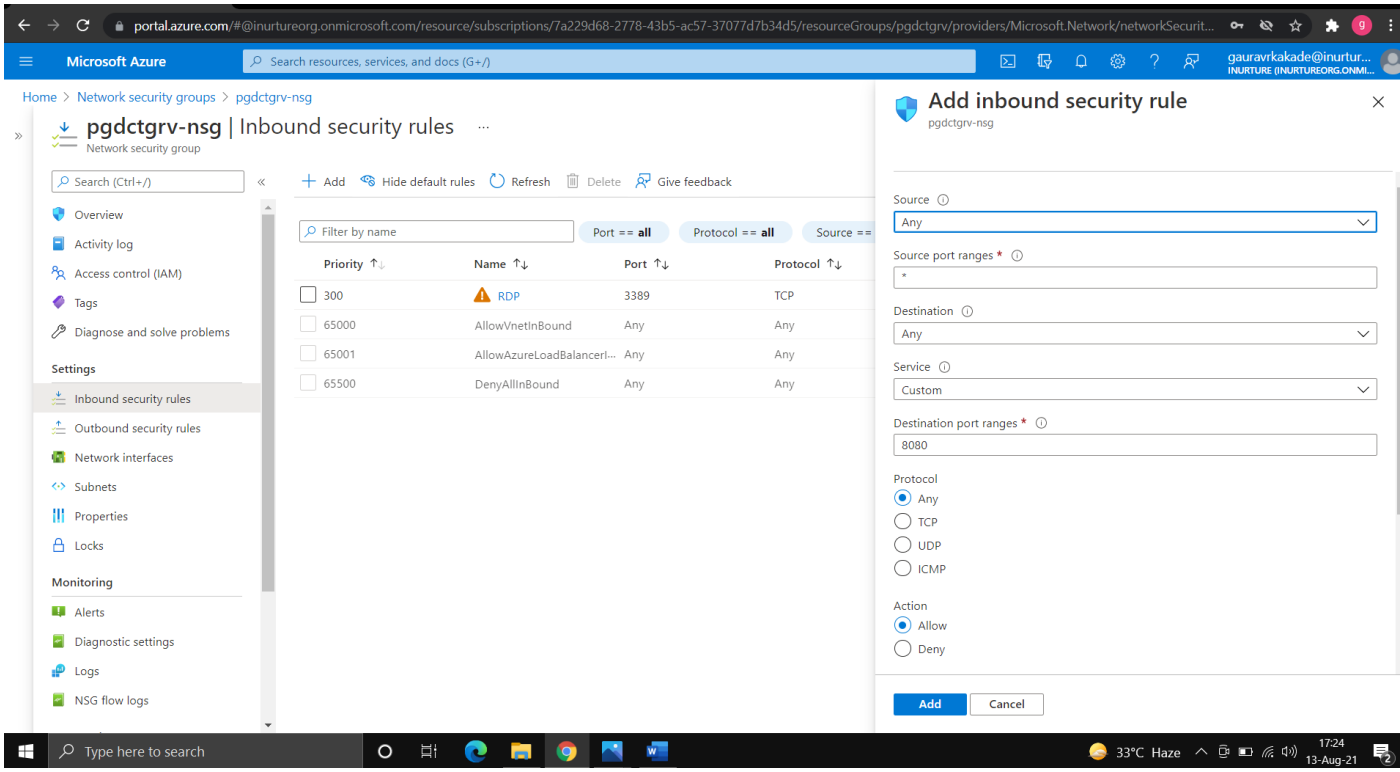


5. Now install IIS web server in virtual machine and browse the IP address of virtual machine and you will see the webpage of IIS webserver; this is because everything is open in security group by default.

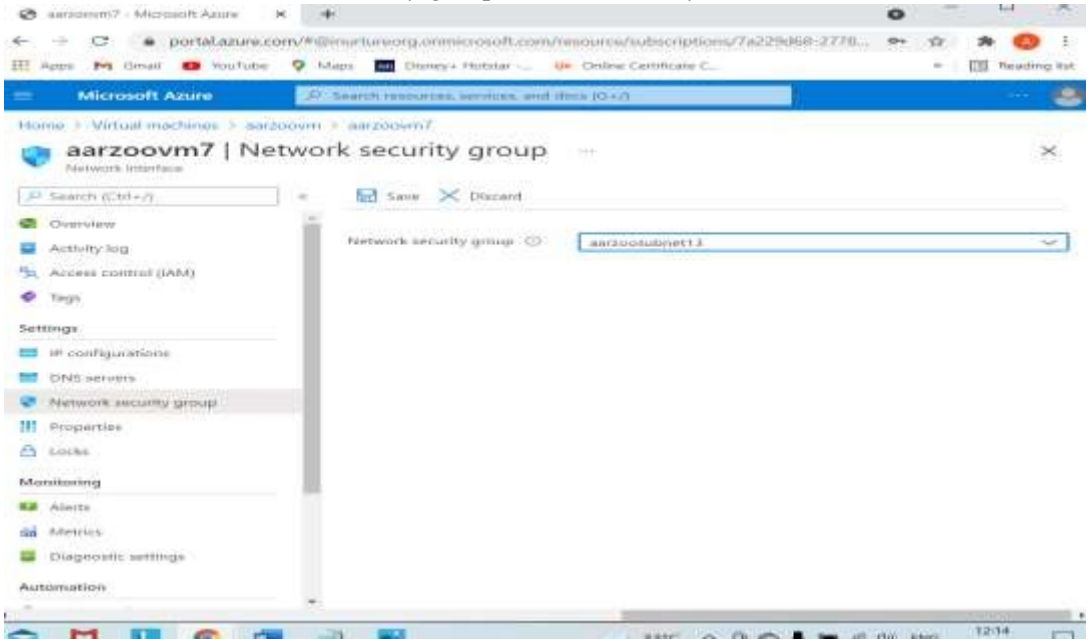
6. Now to add security group, go to virtual machine.
7. Network security group>> select the resource group>>enter the name of security group and create.



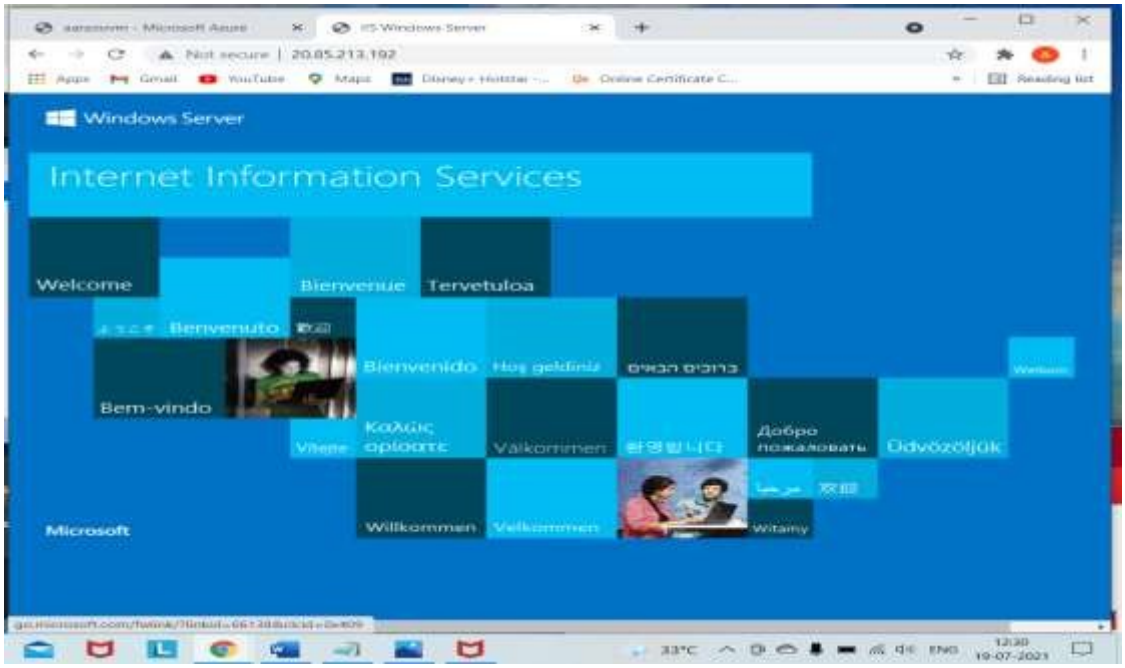
8. Now in inbound security rule>>click on add>>choose the service as http.
9. Keep action allow and enter priority as 100>> add the security rule.



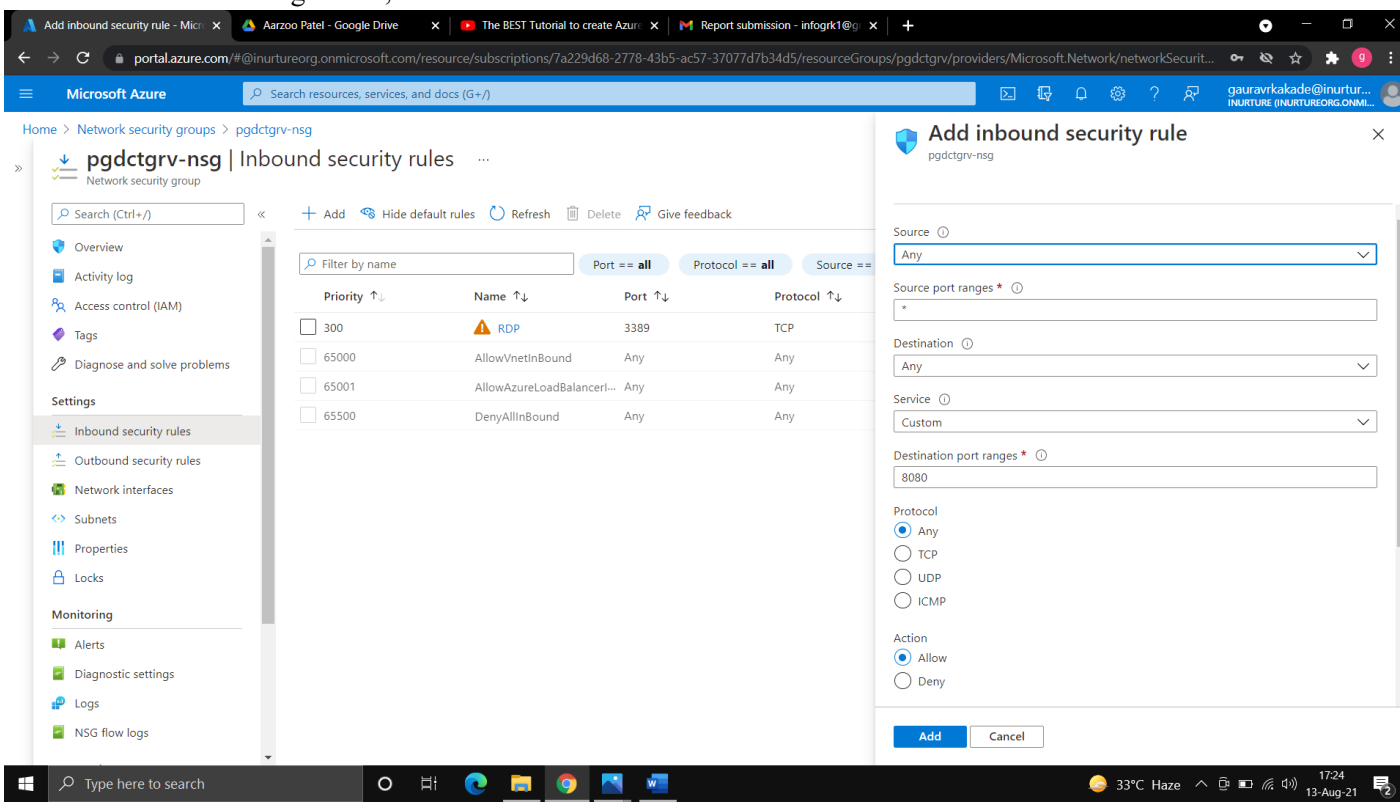
10. Now go to virtual machine>>stop the virtual machine.
11. Go to network interface>>network security group>>select your network security group and save.
12. You will see the security group has attached to your network interface.

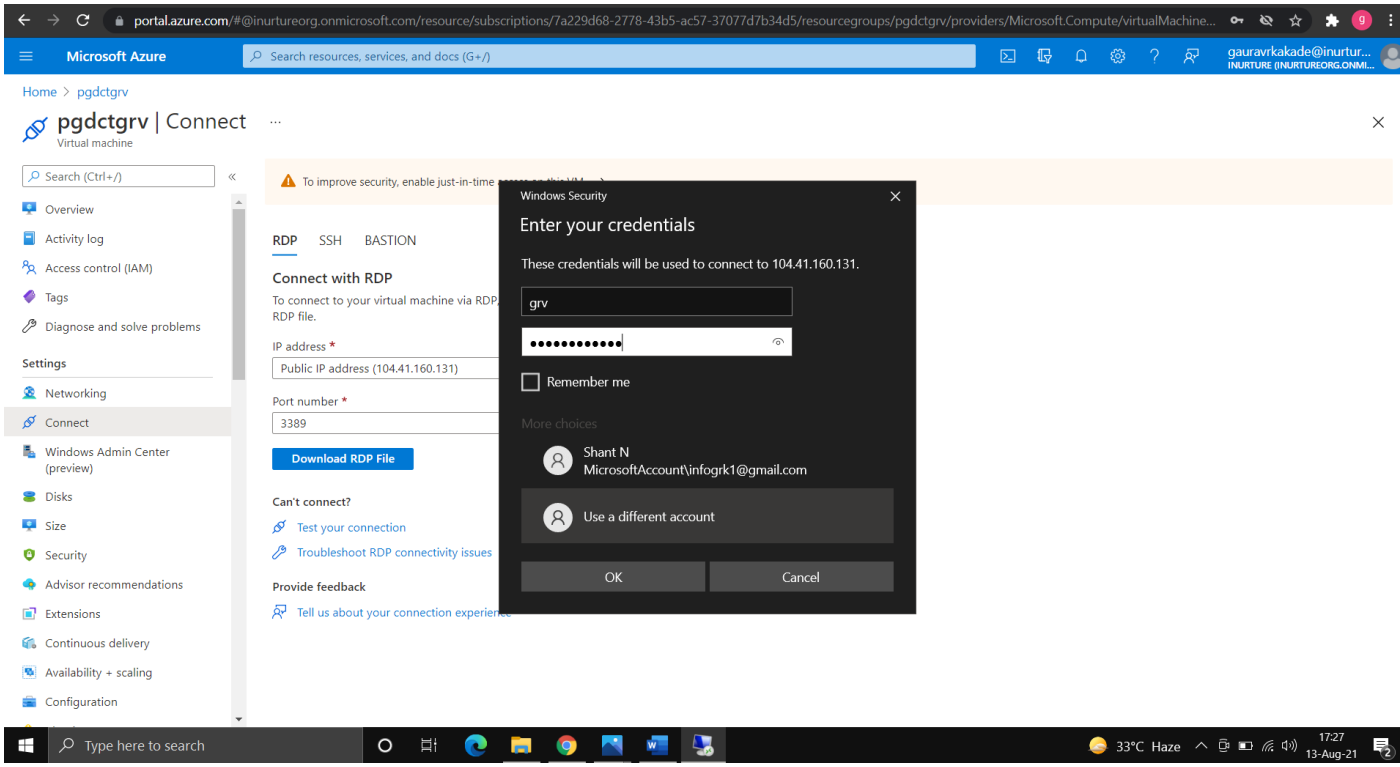


13. Now copy the public IP address of virtual machine and browse it you will see your web server.



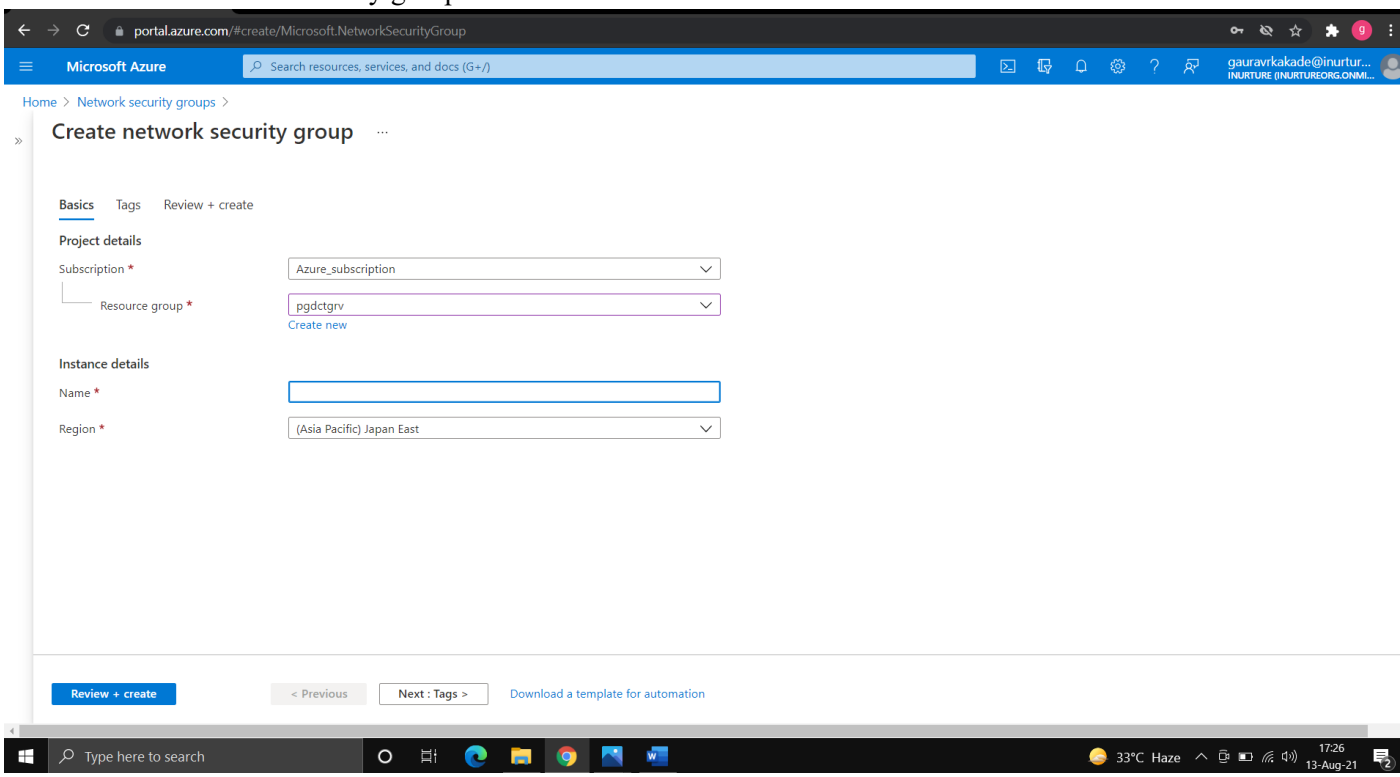
14. Also try to connect with RDP, it will not connect as we have not allowed the RDP security rule.
15. To connect through RDP>>create a inbound security rule with service RDP>>now try to connect through RDP, it will connect.



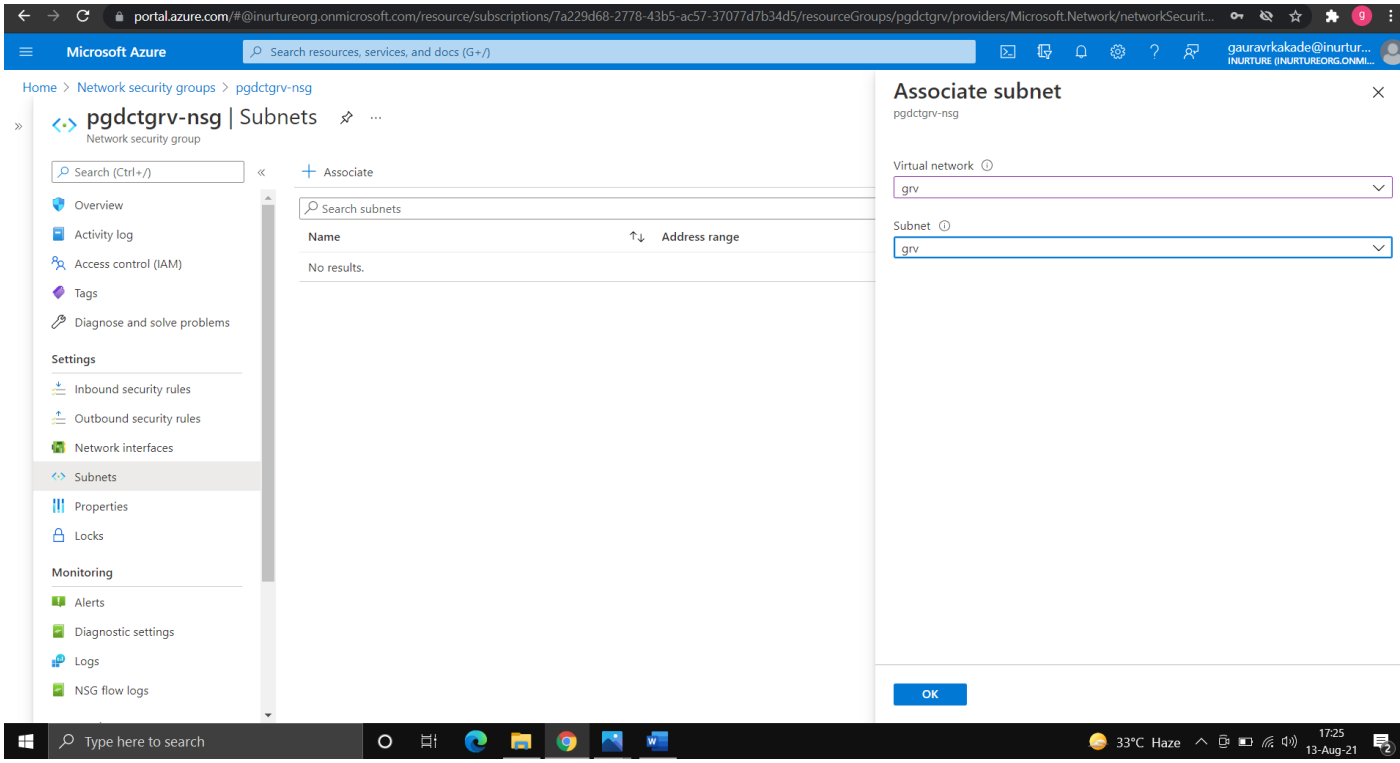


Steps to create security group and attach to a subnet:

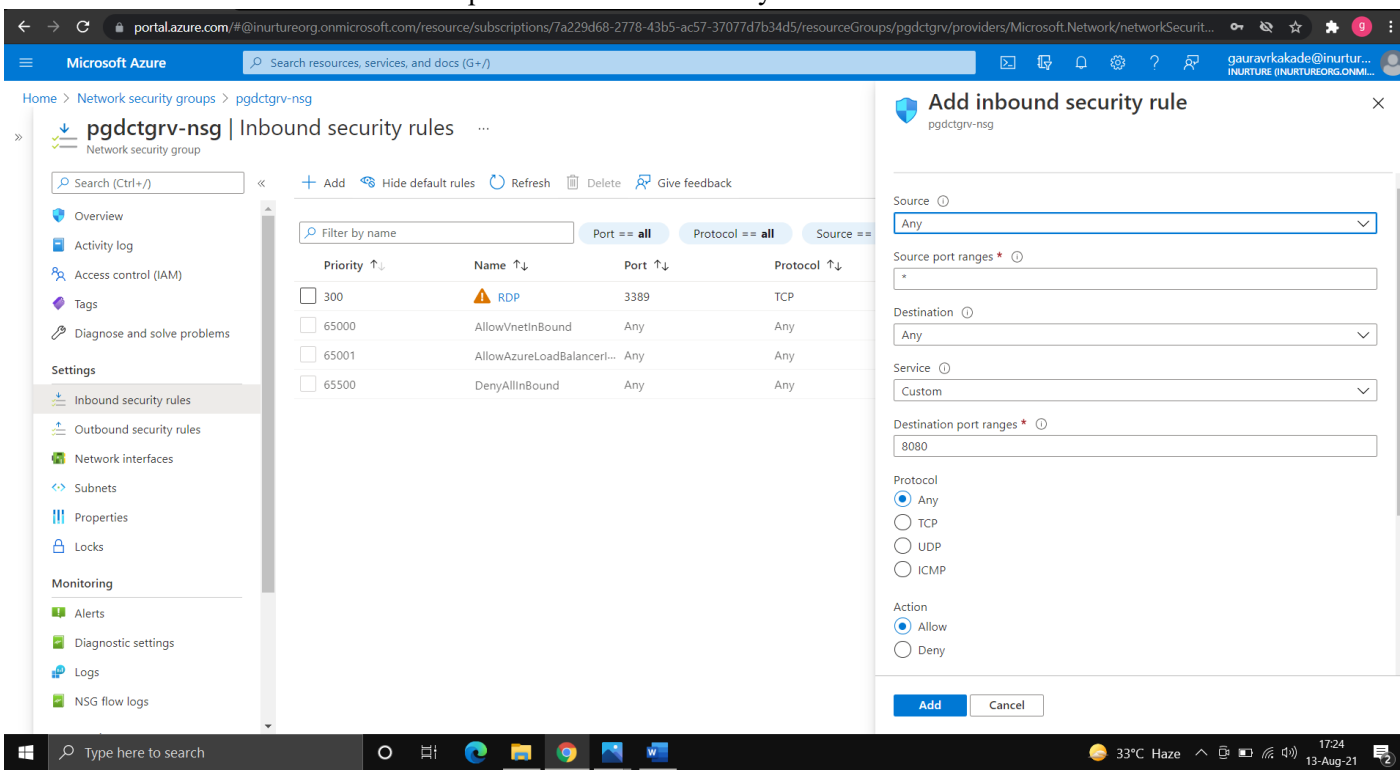
1. Create a new security group.



2. Go to virtual network>>subnets>>add

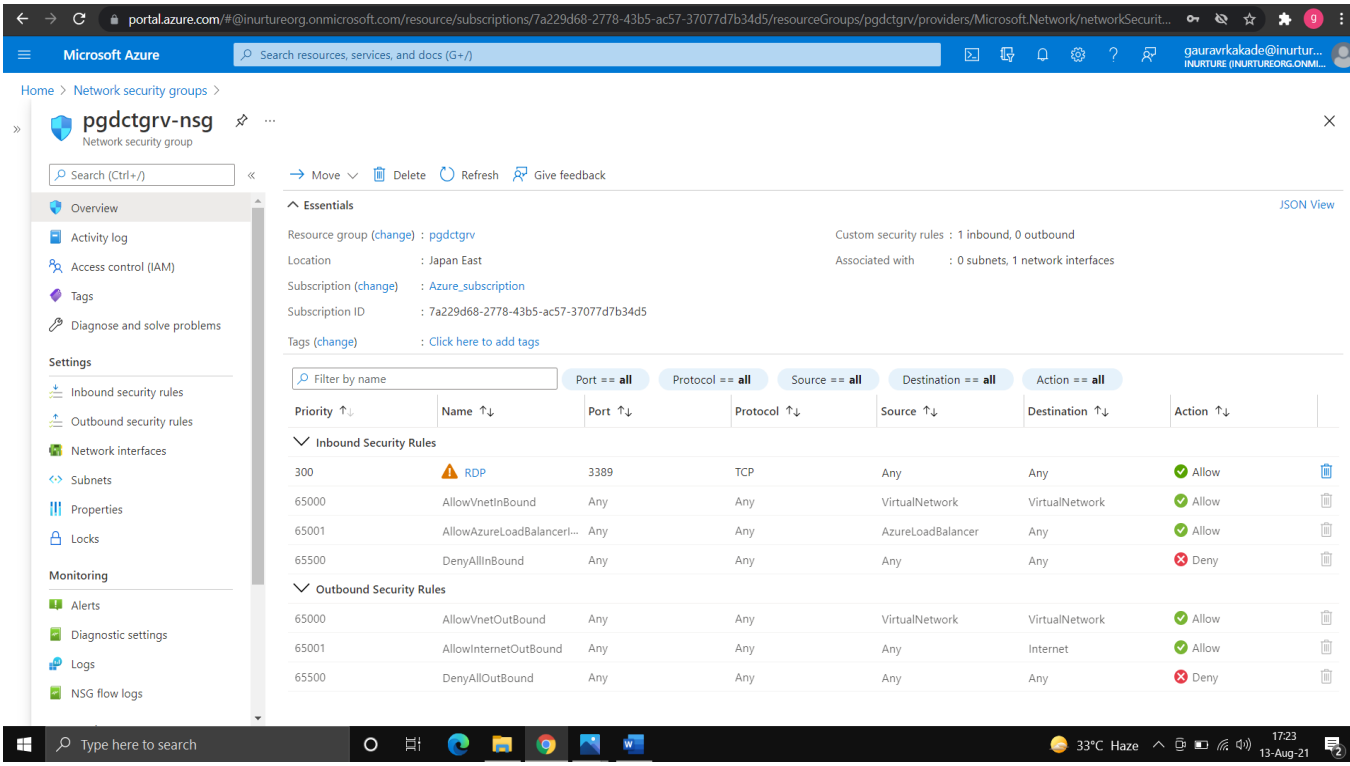


3. Go to inbound security rule.
4. Create a rule with service http>>choose action as deny and add.

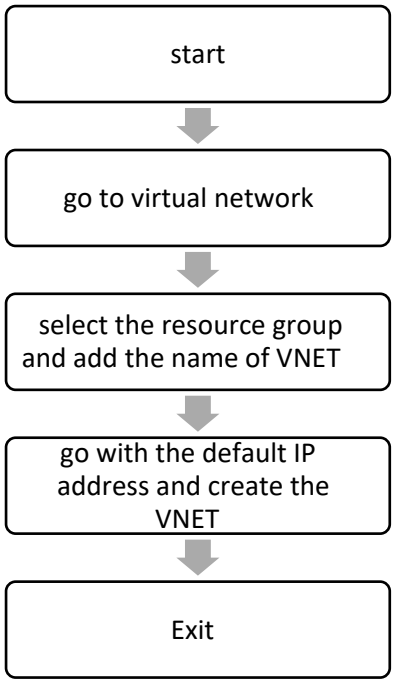


5. As we have denied the Http at subnet level and has allow the http at virtual machine level, Now try to browse the IP address and see you will get your IIS web server or not.
6. It will not show the webserver.

7. Now, go to subnet level security group and allow the http traffic and again try to browse the IP, now it will show the webserver.



Flowchart:



Result: The virtual network has been created and after creating security group we have seen the effect of security group at virtual machine level and at subnet level.

Conclusion: we have successfully created the virtual network and security group, and has attached the security group at virtual machine level and subnet level.

Experiment No.: 9

Name of Experiment: Creation of VNET using ARM management.

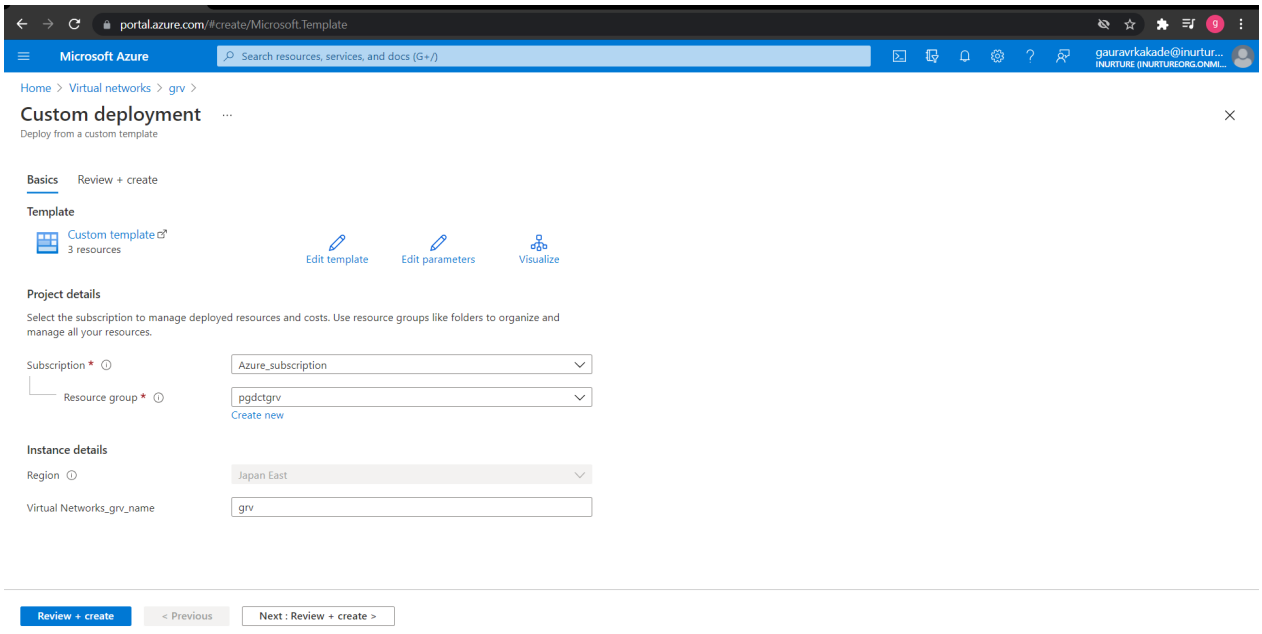
Prerequisites: Azure portal.

Description: ARM Templates are a way to declare the objects you want, the types, names and properties in a JSON file which can be checked into source control and managed like any other code file. ARM Templates are what really gives us the ability to roll out Azure "Infrastructure as code".

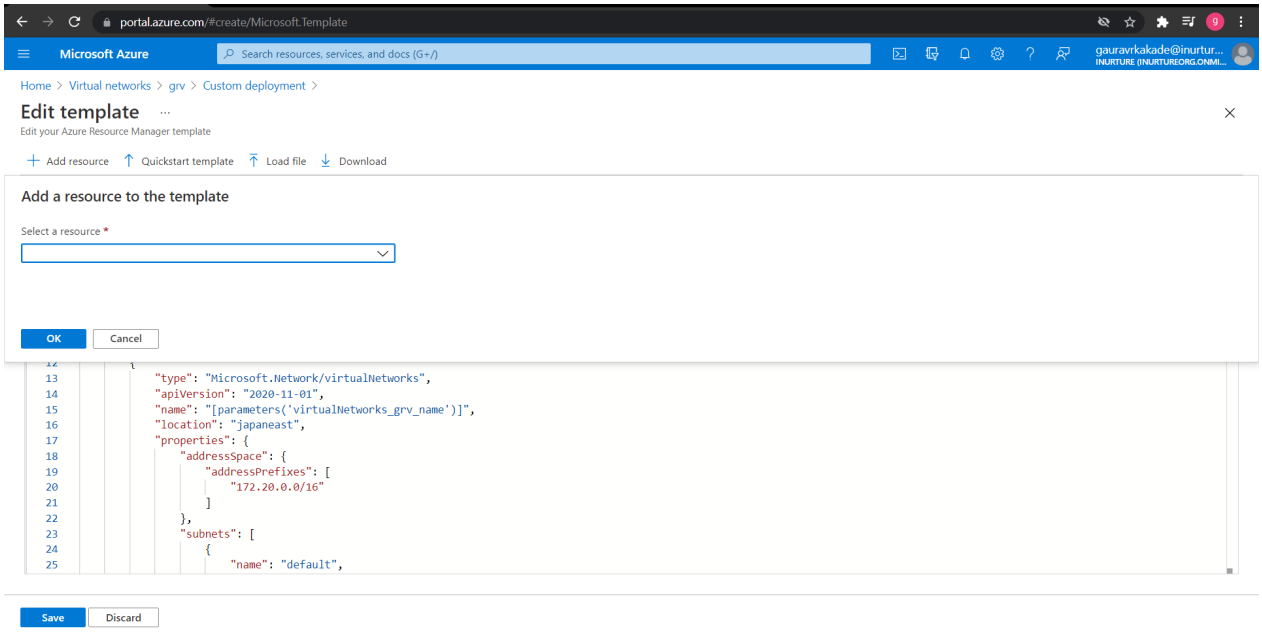
An ARM template can either contain the contents of an entire resource group or it can contain one or more resources from a resource group. When a template is deployed, you have the option of either using 'complete' or 'incremental' mode.

Algorithm:

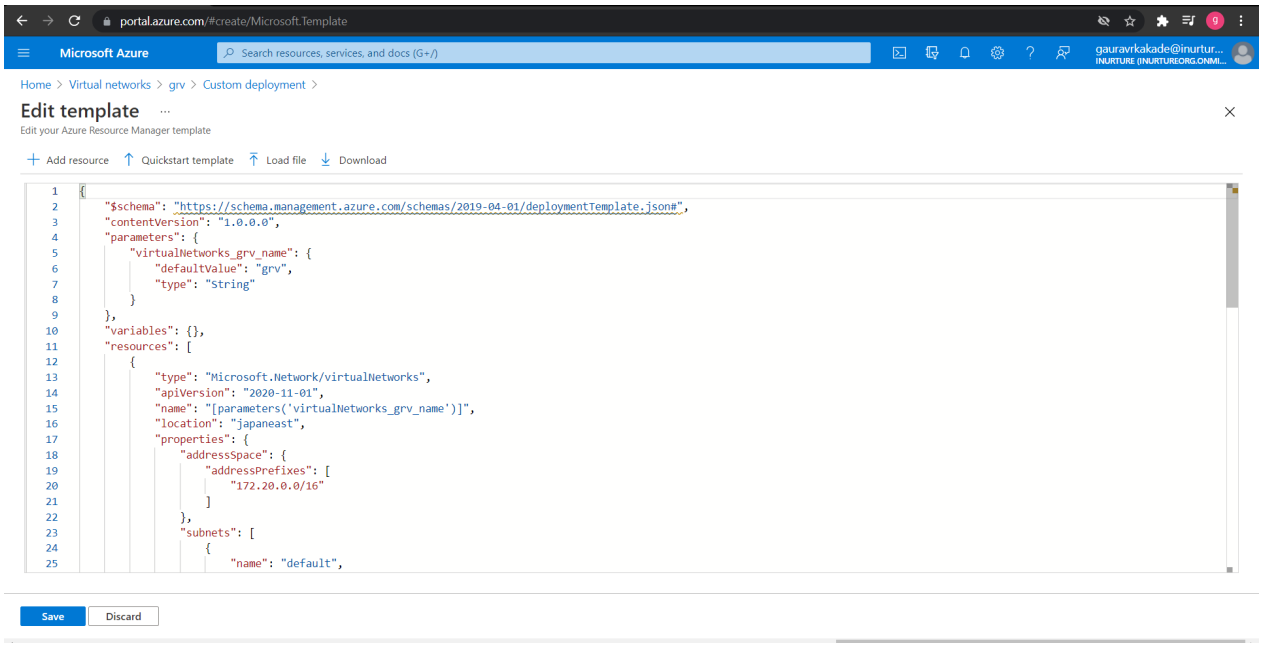
1. Go to deploy a custom template.
2. Click on build your own template with editor.

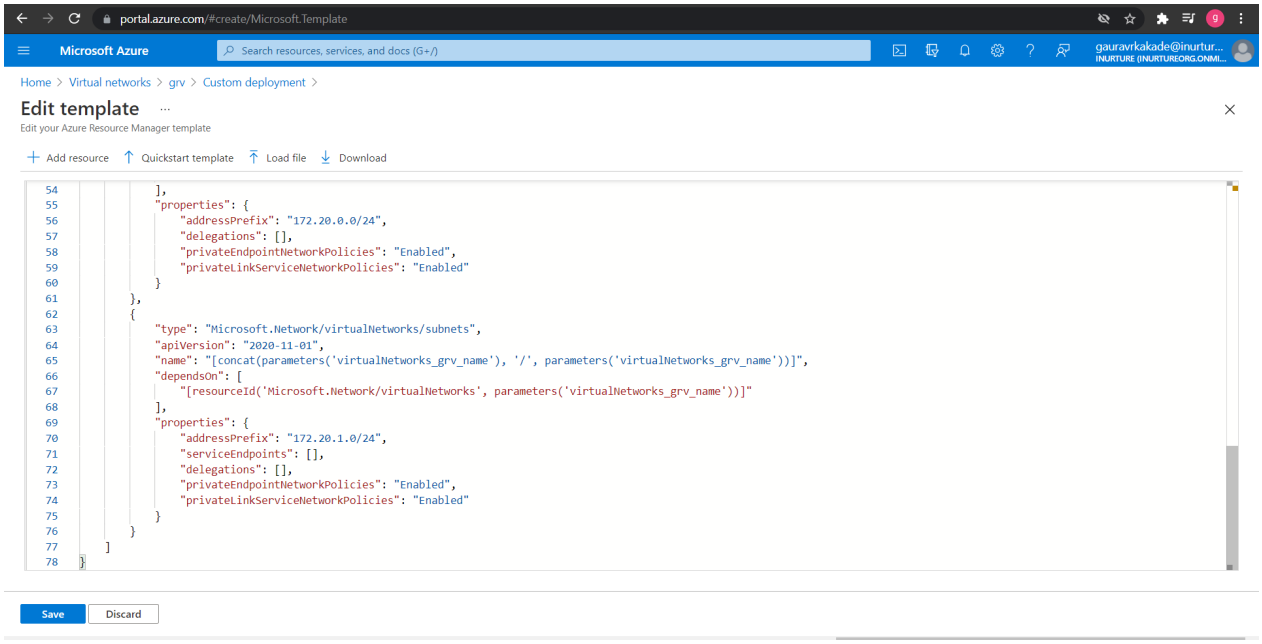


3. There you will see a window where you can add your own code.
4. Other option is that, you can click on add resource.
5. Where you can see various resources.
6. Click on virtual network and give the name.

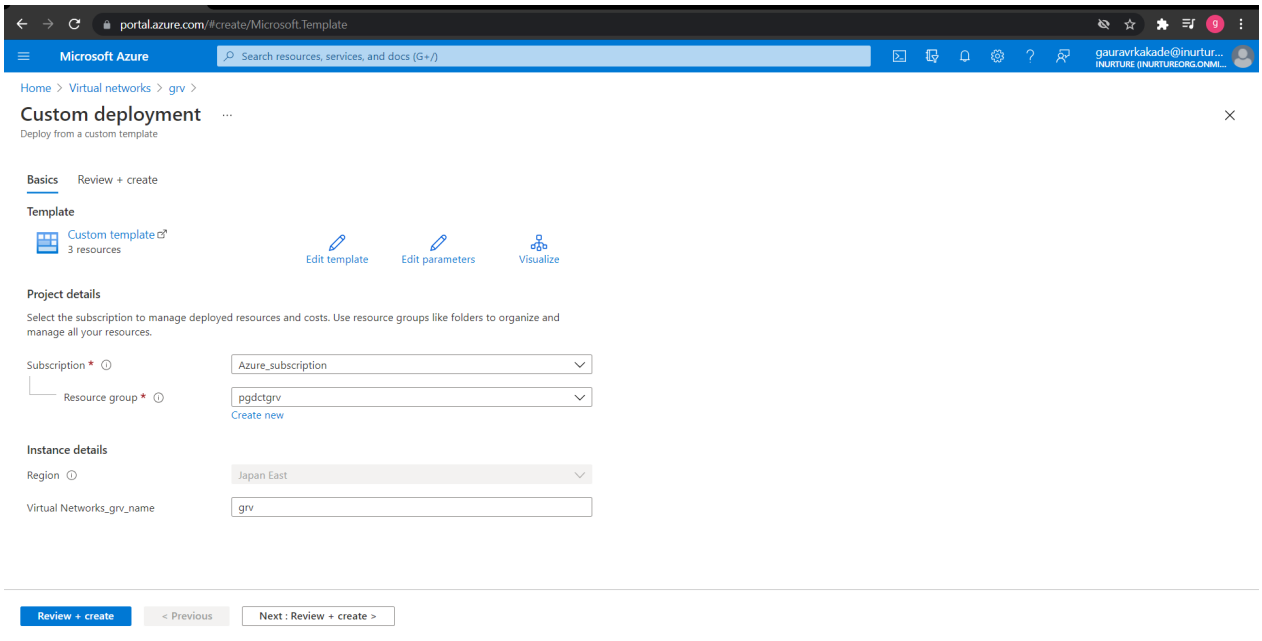


7. Then you can see azure automatically add the code for virtual network.
8. Then you can save the template.

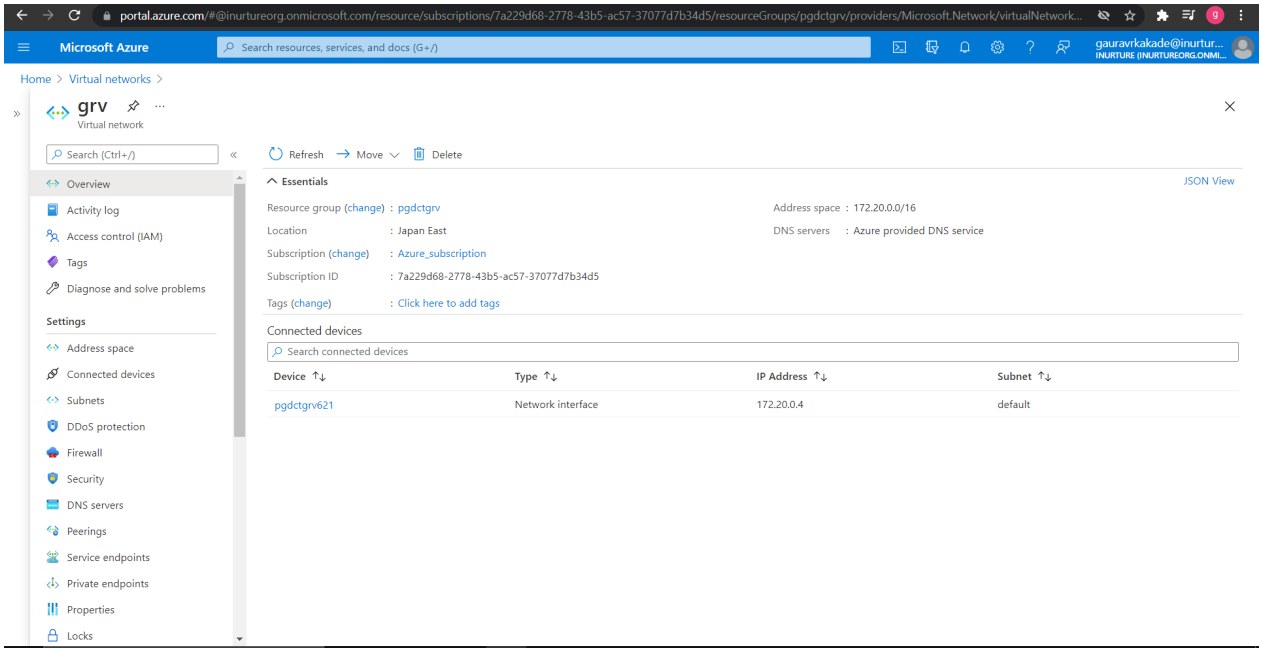




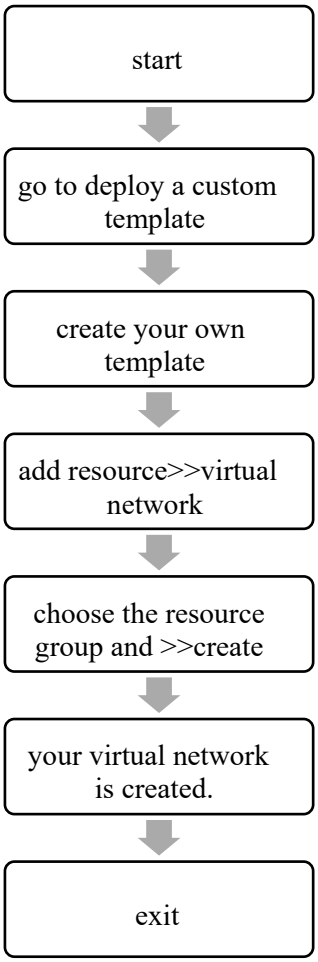
9. Select your resource group and region and then create.



10. You will see your virtual network is created.


portal.azure.com/#@inurtureorg.onmicrosoft.com/resource/subscriptions/7a229d68-2778-43b5-ac57-37077d7b34d5/resourceGroups/pgdctgrv/providers/Microsoft.Network/virtualNetwork...
Microsoft Azure
Search resources, services, and docs (G+/)
gauravrkakade@inurtur...
INURTURE (INURTUREORG.ONMI...
Home > Virtual networks >
grv
Virtual network
Search (Ctrl+/)
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Settings
Address space
Connected devices
Subnets
DDoS protection
Firewall
Security
DNS servers
Peerings
Service endpoints
Private endpoints
Properties
Locks
Refresh
Move
Delete
Essentials
JSON View
Resource group (change) : pgdctgrv
Location : Japan East
Subscription (change) : Azure_subscription
Subscription ID : 7a229d68-2778-43b5-ac57-37077d7b34d5
Tags (change) : Click here to add tags
Address space : 172.20.0.0/16
DNS servers : Azure provided DNS service
Connected devices
Search connected devices
Device
Type
IP Address
Subnet
pgdctgrv621
Network interface
172.20.0.4
default

Flowchart:



Result: It has been seen that we can also create various resources using ARM template, where we have to just select the resource and give name, the code is given by azure or we can also deploy our code and create the resource.

Conclusion: we have successfully created the virtual network using ARM template.

Experiment No. 10

Name of Experiment: Creating load balancer in azure- Basic and Standard.

Prerequisites: Azure portal, RDP, putty.

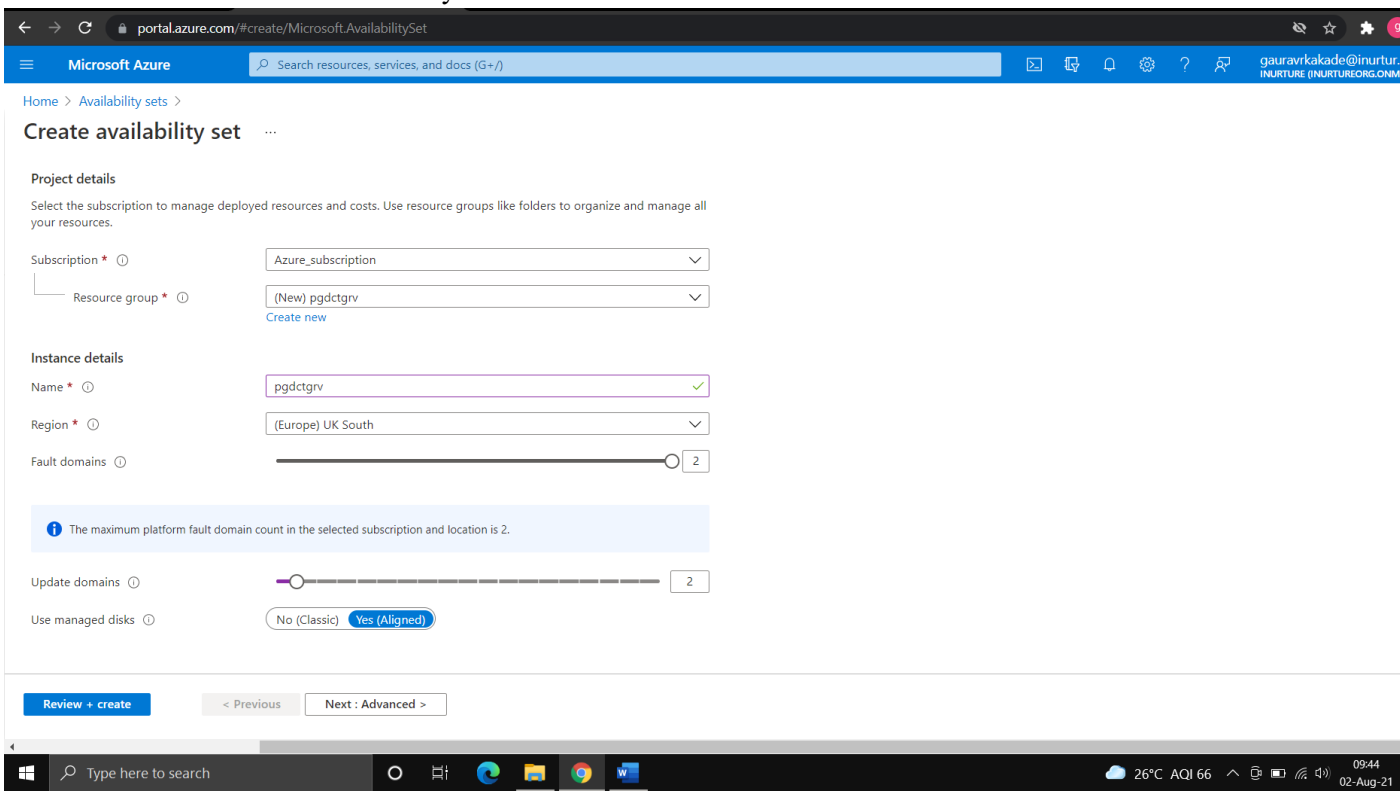
Description: Load balancing refers to evenly distributing load (incoming network traffic) across a group of backend resources or servers. Azure Load Balancer operates at layer 4 of the Open Systems Interconnection (OSI) model. It's the single point of contact for clients. Load balancer distributes inbound flows that arrive at the load balancer's front end to backend pool instances. These flows are according to configured loadbalancing rules and health probes. The backend pool instances can be Azure Virtual Machines or instances in a virtual machine scale set.

Load balancer supports both Standard and Basic SKUs. These SKUs differ in scenario scale, features, and pricing. Any scenario that's possible with Basic load balancer can be created with Standard load balancer.

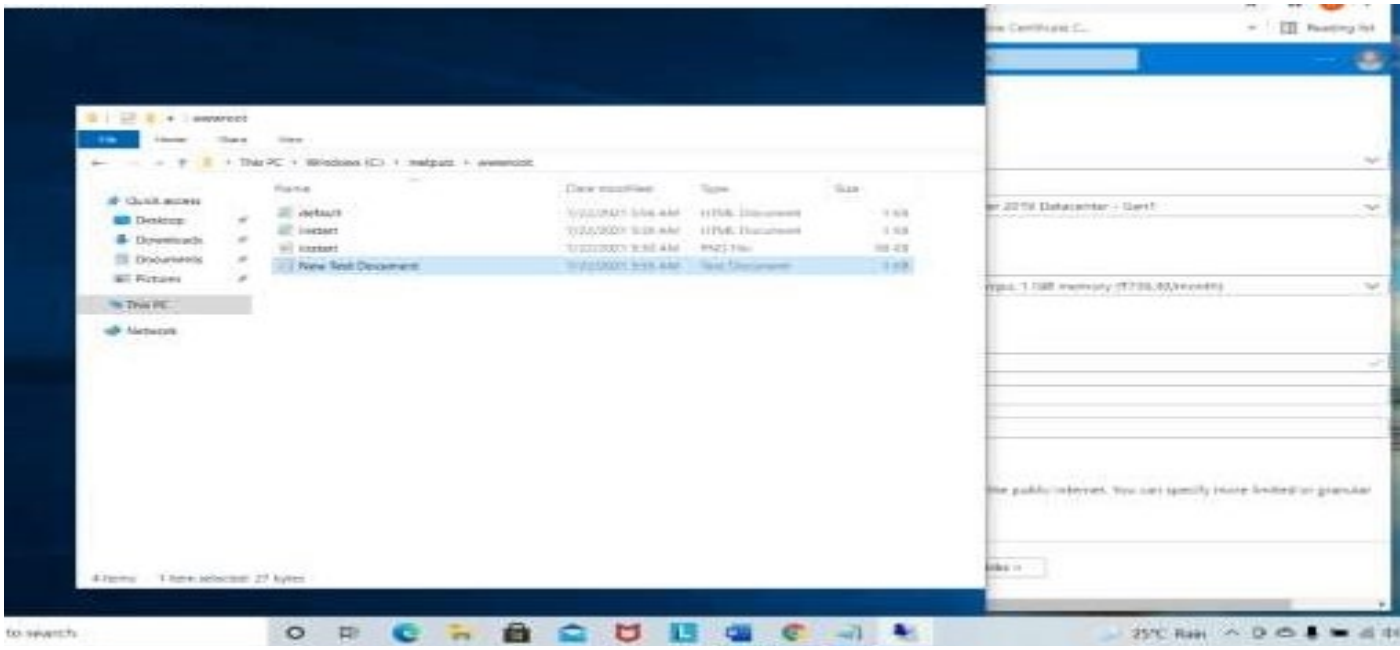
Algorithm:

Creating load balancer with basic SKUs:

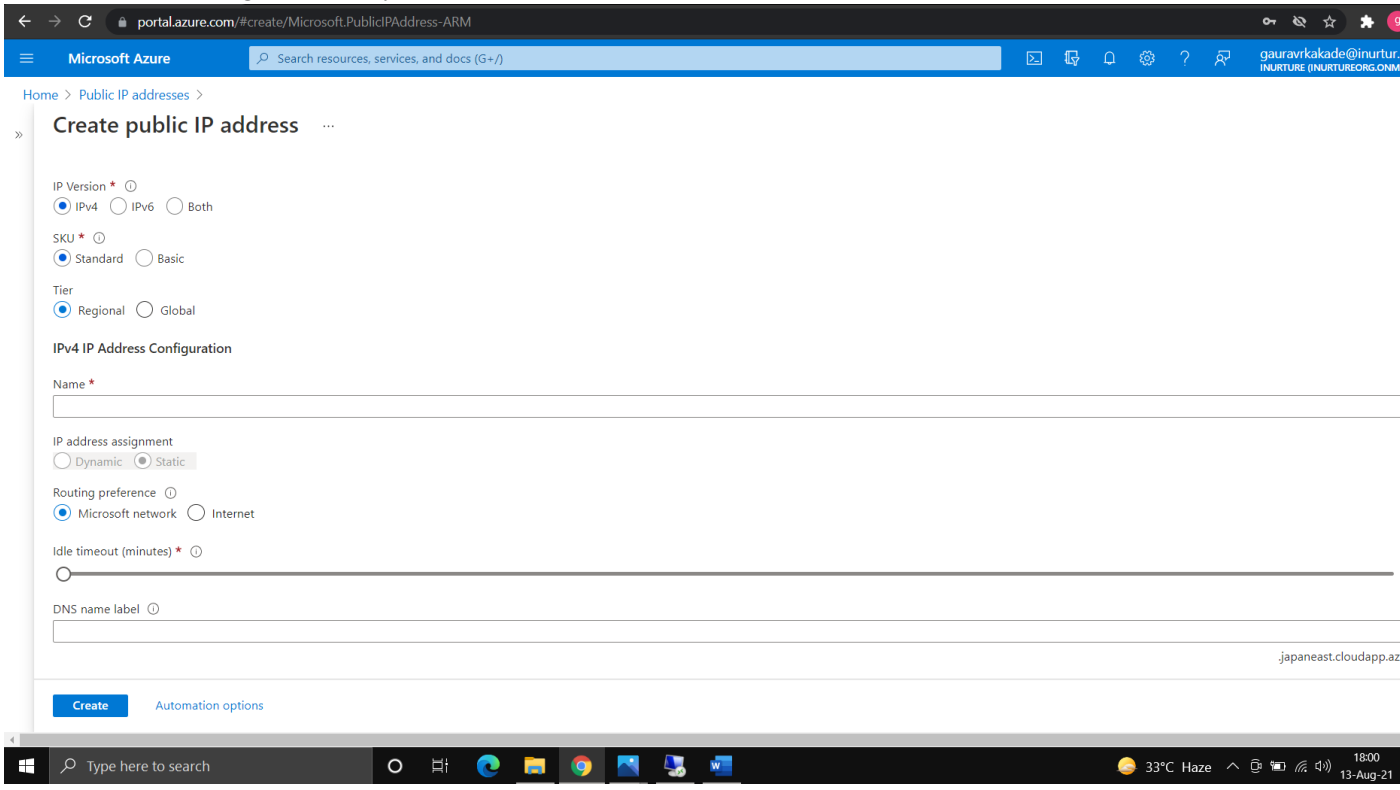
1. Create two virtual machines.
2. Create the availability sets in both virtual machines.



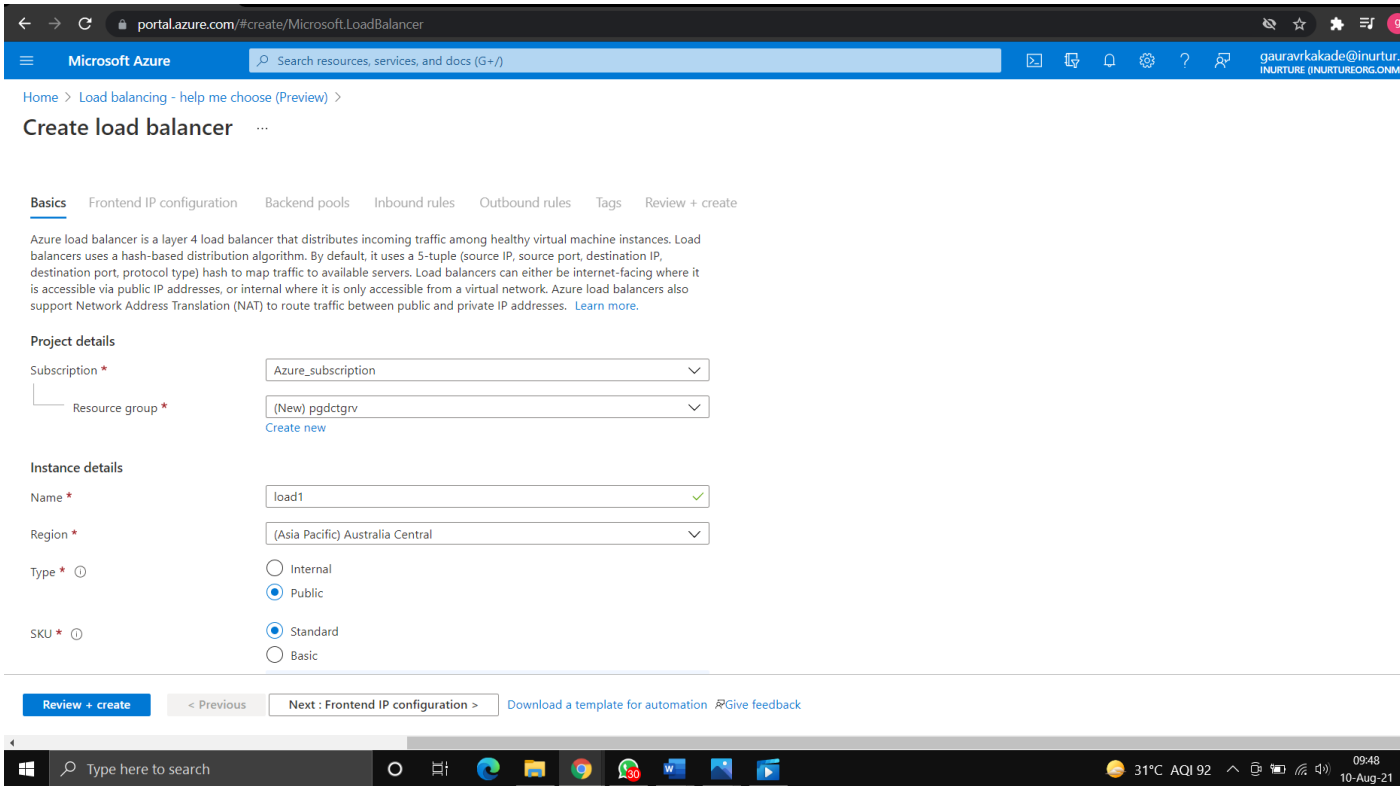
3. Install IIS server on both the virtual machines.
4. Create a html file in c drive>>inetpub>>wwwroot>>default.hmtl (do this in both the virtual machines).



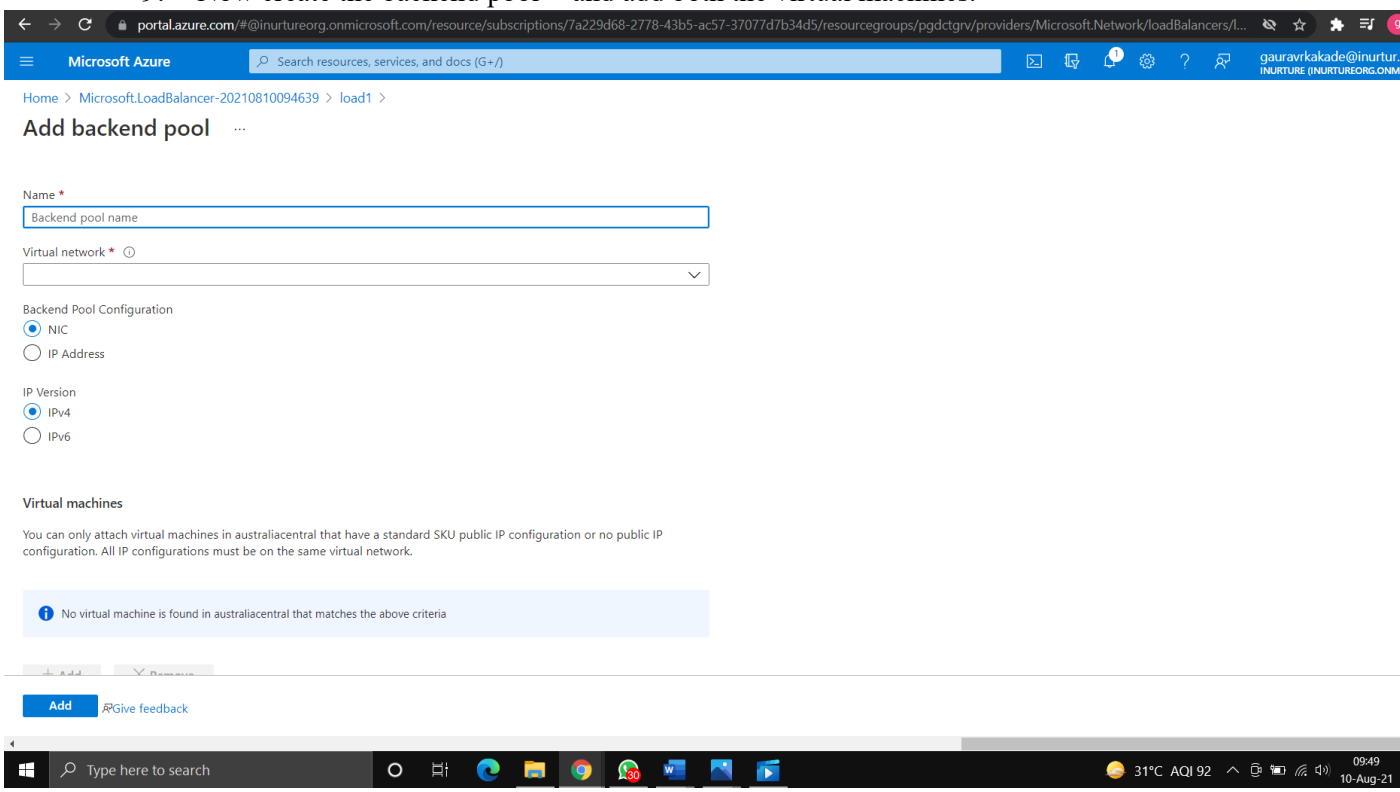
5. Disable the public IP of both virtual machine, to disable the IP go to networking>>go to NIC>>IP configuration>>disassociate>>save.
6. Create the public IP address>>give the name of public IP and keep IP address assignment as dynamic and click on create.



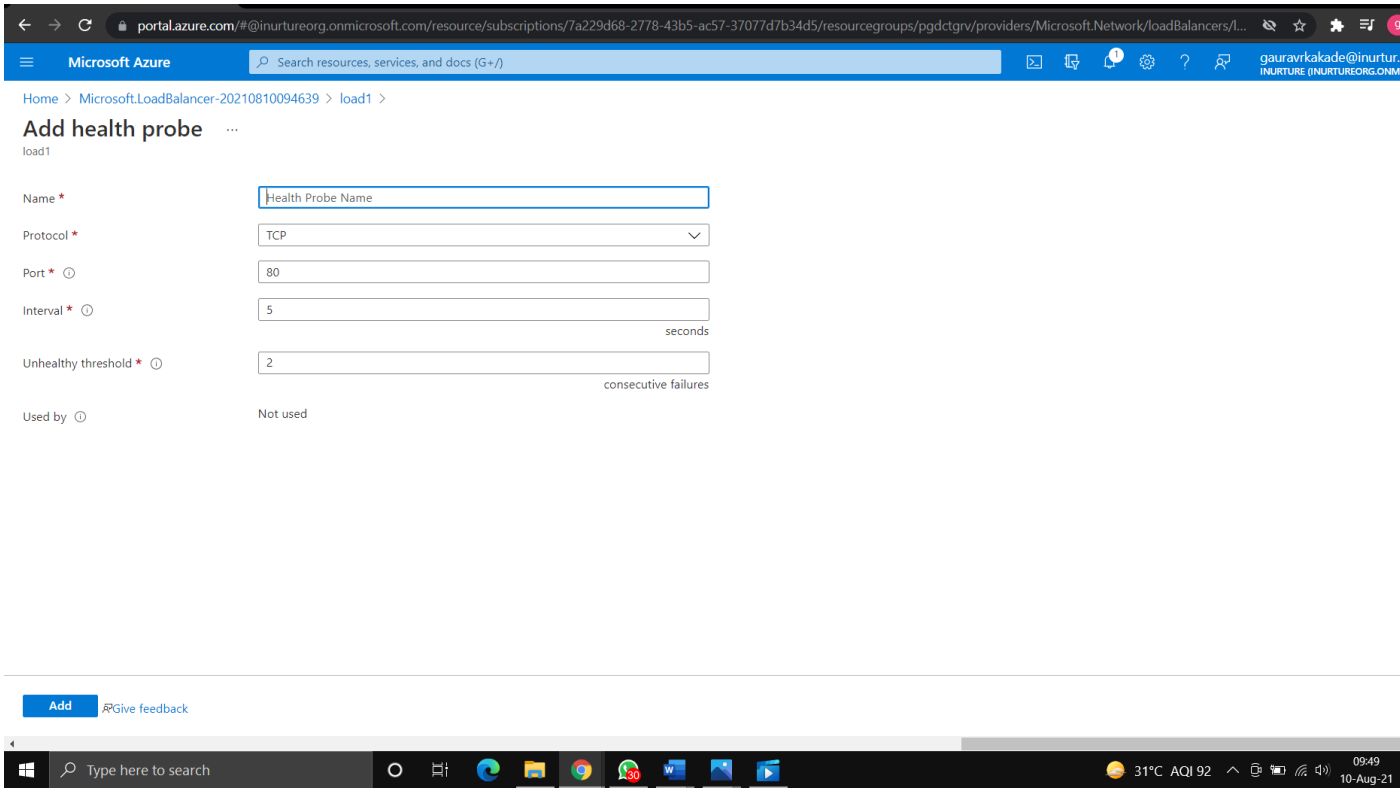
7. Go to load balancer>>give the name of load balancer>>keep type as public and SKUs as basic.
8. Choose the public IP address that you have created and create the load balancer.



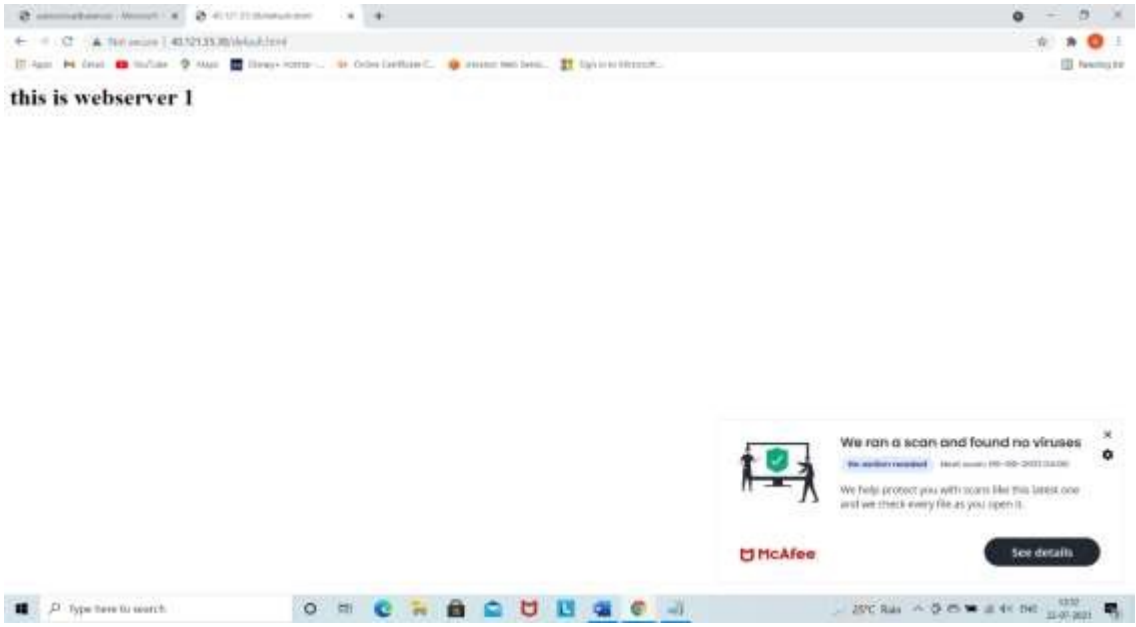
9. Now create the backend pool>>and add both the virtual machines.



10. Now create the load balancing rule>>give the name of the rule>>select the frontend IP address.
11. Add port 80>>add backend port as 80>>add your backend pool.
12. Create the health probe>>give the name of health probe>>give port 80 and create>>add the load balancing rule.



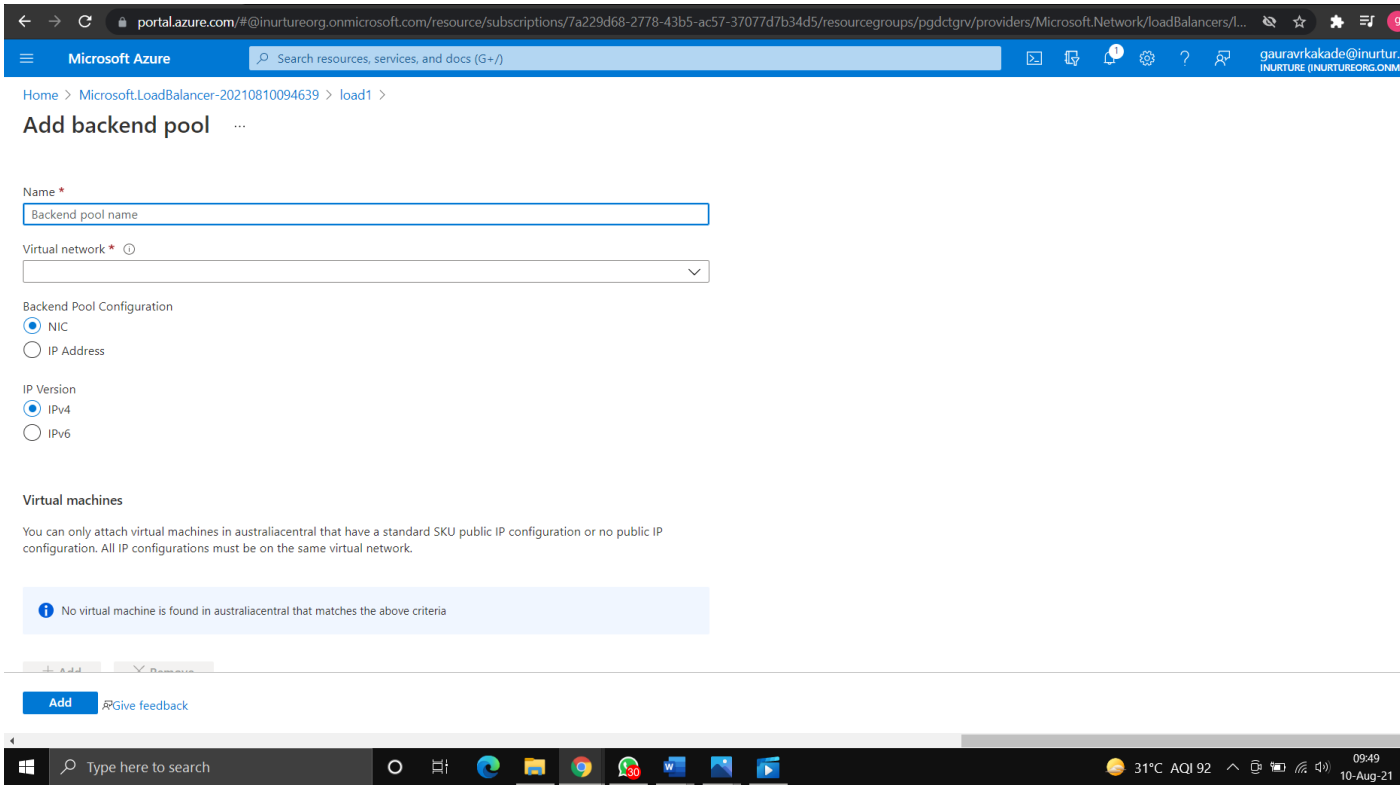
13. Now copy the IP address of the load balancer and browse(40.121.22.38/default.html).
14. You will see you webpage1, after refreshing in sometime you will see your another webpage.



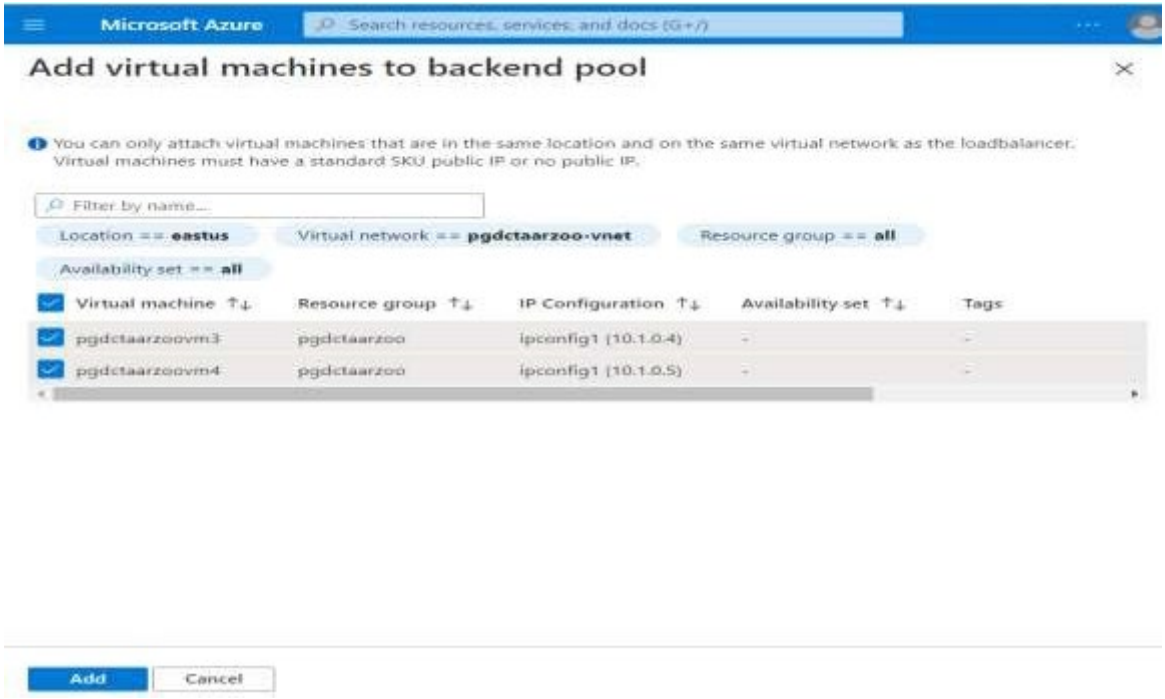
Creating load balancer using standard SKUs:

Here we will see a use case in which we will configure multiple backend pool and will see how to handle both backend pool through frontend IP configuration.

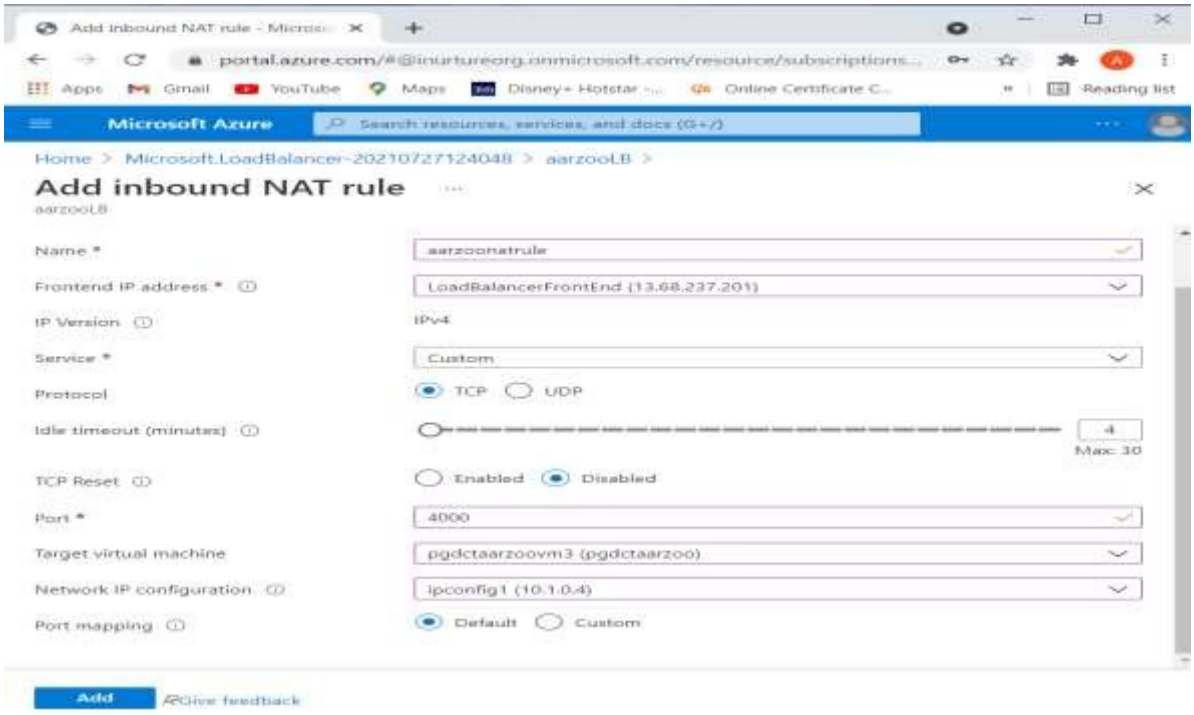
1. Create two virtual machines.
2. Disassociate public IP of both the virtual machines.
3. Create the load balancer with standard SKUs



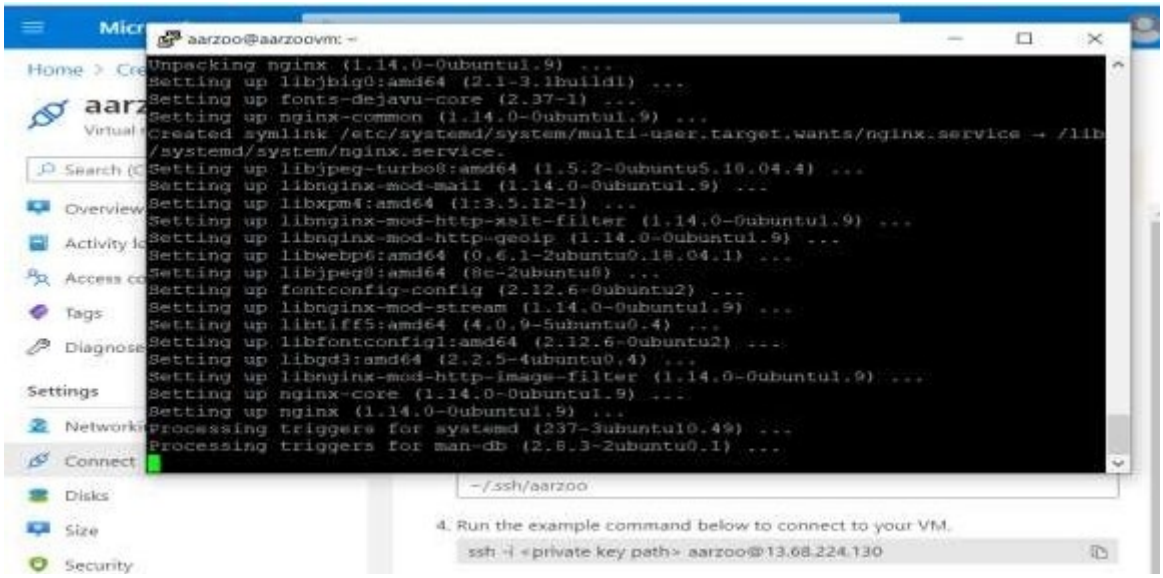
4. Create the public IP address>>create the load balancer.
5. Create the backend pool and select both the virtual machines.



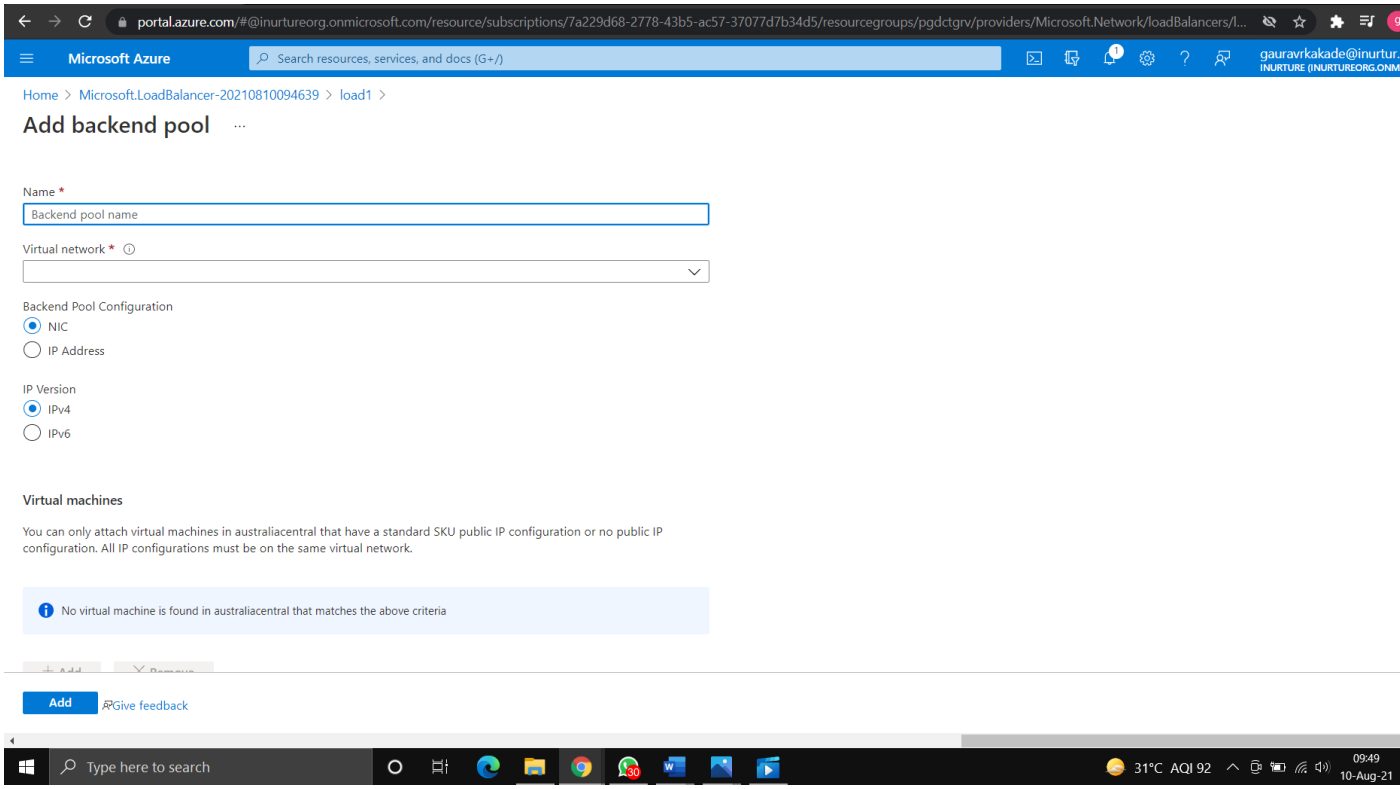
6. Create health probe and create load balancing rule.
7. Create NAT rule>>give the name of rule>>select the frontend IP address>>In service choose custom>>give port 4000>>select the target virtual machine>>add.



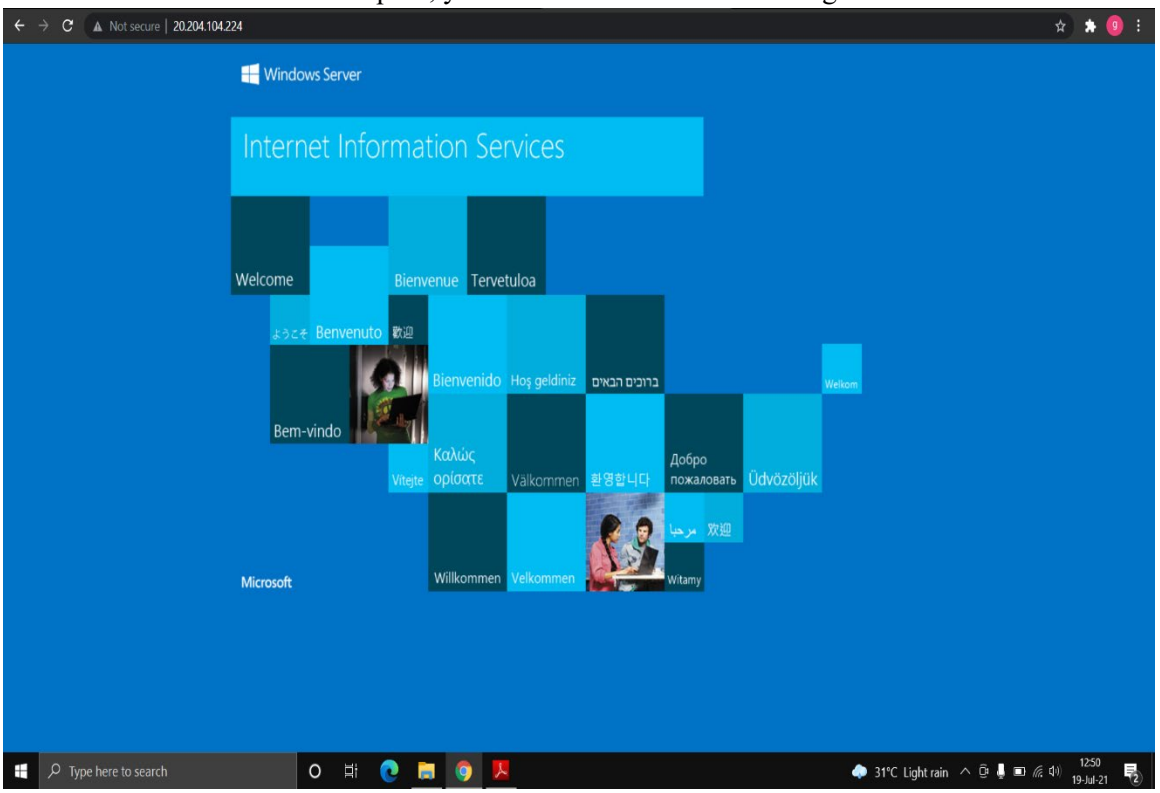
8. Now open command prompt and type the command **mstsc /v: IP address:4000.**
9. Connect both the virtual machines with RDP and install IIS server.
10. Now, Create a new linux virtual machine.
11. Connect with putty.
12. Install nginx server on machine: **sudo apt-get install nginx.**

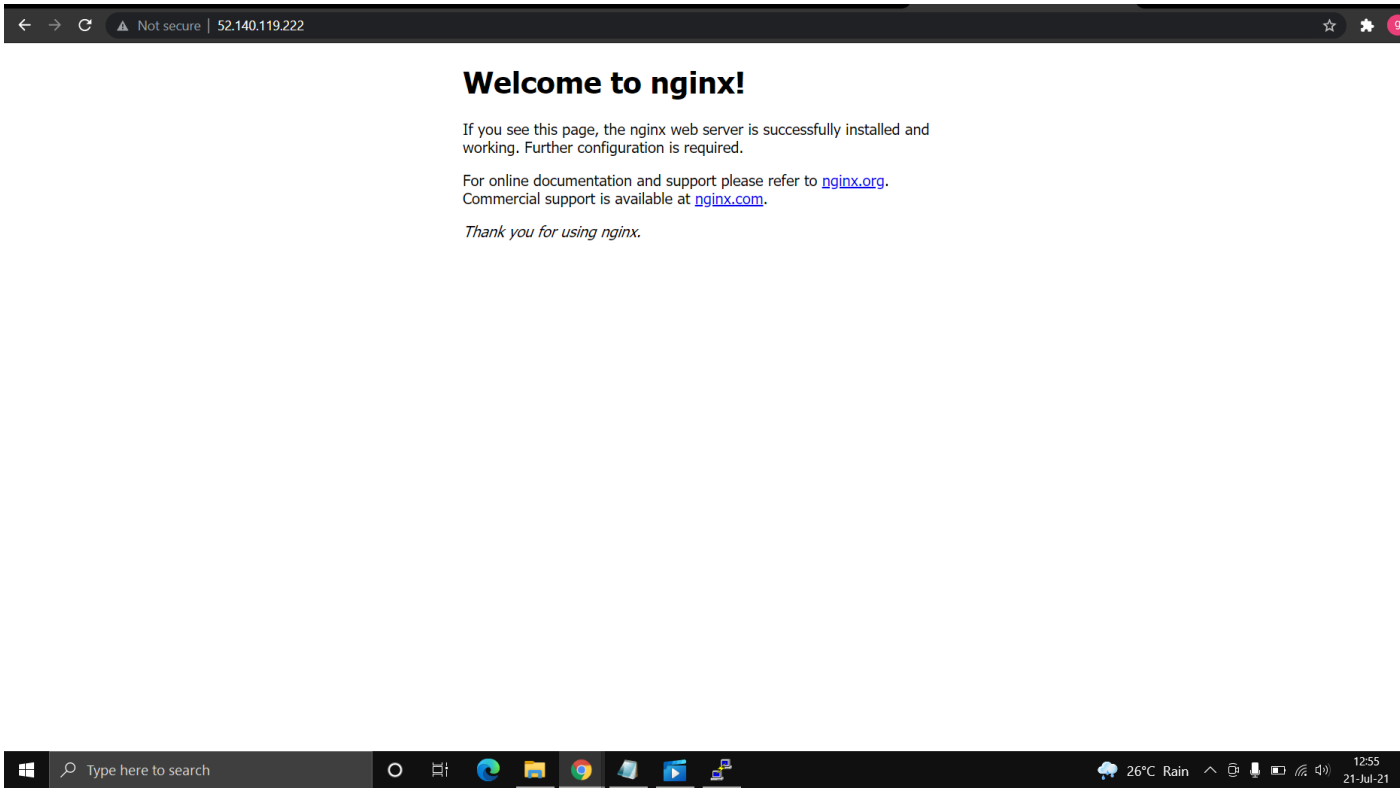


13. Disassociate the public IP.
14. Now go to load balancer and create a new backend pool>>add the linux virtual machine.



15. Add frontend IP address.
16. Create new load balancing rule and select the health probe.
17. Now browse the IP of both the pool, you will see the IIS server and nginx server.





Result: The load balancer has been created with basic and standard SKUs, and it has been seen that while creating load balancer with basic SKU the virtual machine must be in an availability set and with standard SKU the multiple virtual machine can be in load balancer backend pool without creating availability set. We have also seen the configuration of multiple backend pools.

Conclusion: we have successfully created the basic and standard load balancer and have also seen how to created multiple backend pools.