R&D Document: CIDR Ranges of VNet, Subnet, VNet Peering and Its Types with Azure Use Case

Title: CIDR Ranges of a VNet, Subnet, Vnet Peering and it's Types.

Presented to: Celebal Technologies

Prepared by: Tejasvi Avhad – Intern (Cloud Infra and Security)

1. Introduction

In cloud computing, networks are created virtually. Microsoft Azure provides something called a Virtual Network (VNet) which acts like a personal network in the cloud. Inside a VNet, we can create smaller divisions called Subnets. These help in organizing and securing our resources better.

We also use CIDR ranges to assign IP addresses in the network. Additionally, we can connect two different VNets using a concept called VNet Peering.

This document explains all of this in a simple way and also includes the steps performed by me practically on the Azure Portal.

2. CIDR

CIDR stands for Classless Inter-Domain Routing. It is a way to write IP address ranges. For example:

- 10.0.0.0/16 means we are giving a big range of IPs.
- 10.0.1.0/24 means a smaller range inside the bigger one.

CIDR helps Azure know how many IP addresses we want to reserve.

3. VNet (Virtual Network)

A VNet is like a private network that we create inside Azure. It is completely isolated and helps us connect different services securely. For example, we can launch virtual machines inside it and make them talk to each other.

4. Subnet

A Subnet is a smaller part of a VNet. We create subnets to group different types of resources. For example, we can keep Windows VMs in one subnet and Linux VMs in another.

Each subnet also has its own IP range, which comes from the VNet’s CIDR range.

5. VNet Peering

Sometimes, we need two different VNets to talk to each other. For this, we use VNet Peering. It connects two VNets so that resources inside them can communicate as if they are inside the same network.

There are two types:

- Same region peering – Both VNets are in the same Azure region.
- Global peering – The VNets are in different regions.

6. Prerequisites to Perform This Use Case

Before we begin, we need:

- A valid Azure subscription
- Internet connection
- Basic knowledge of how to use the Azure portal

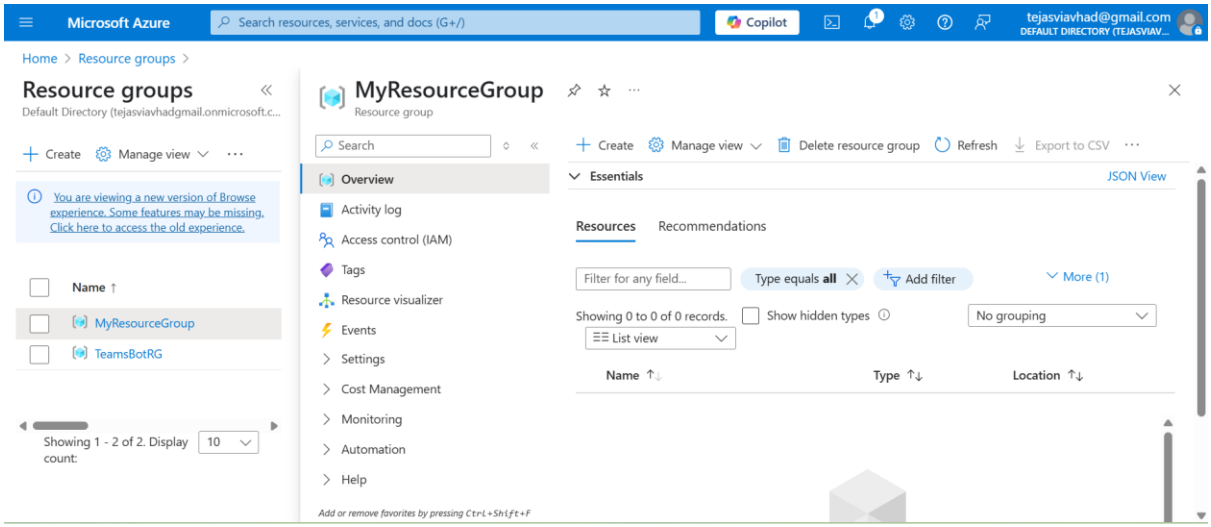
7. Use Case Overview

Create two VNets. Inside the first VNet, create two subnets. Launch a Windows VM in the first subnet and a Linux VM in the second. Then, create a second VNet and connect it with the first using VNet peering. Finally, ensure that the VMs can ping each other.

8. Steps to Perform the Use Case (on Azure Portal)

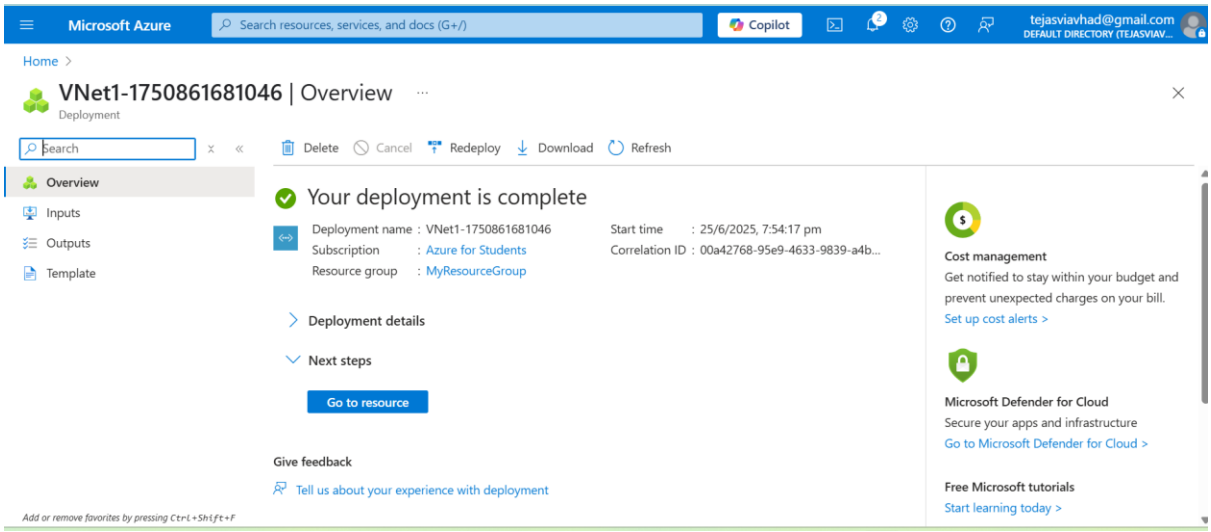
Step 1: Create a Resource Group

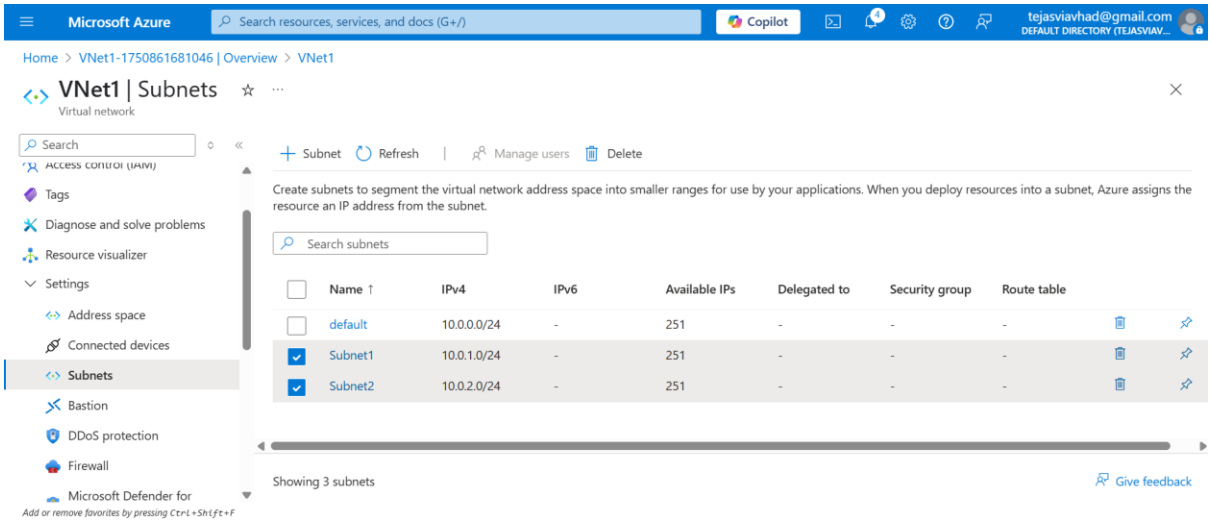
- Go to Azure Portal
- Search and open Resource Groups
- Click on + Create
- Name the group (MyResourceGroup)
- Select Region
- Click Create



Step 2: Create First VNet and Two Subnets

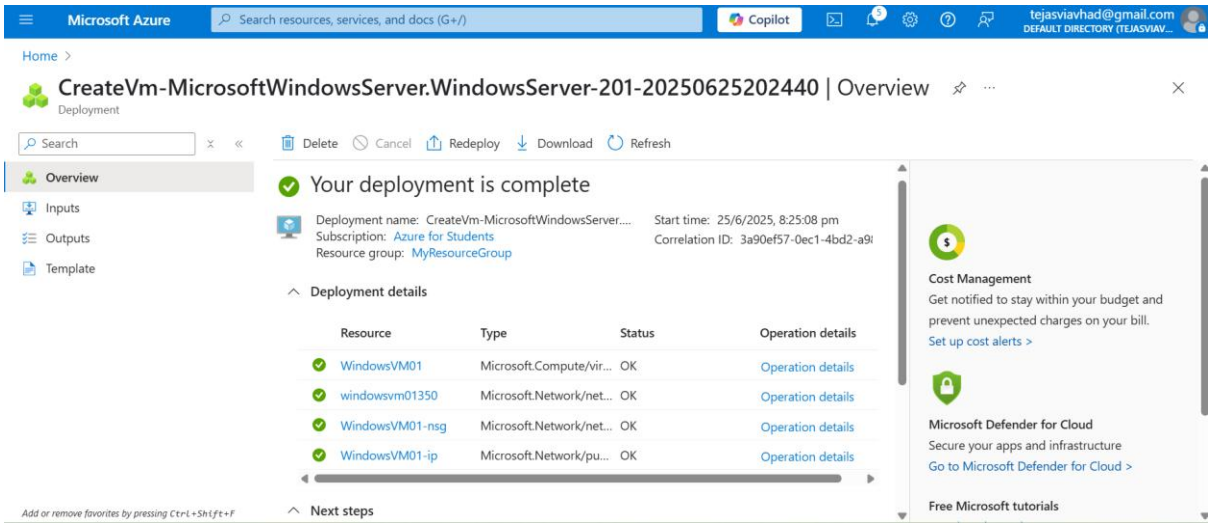
- Search and open Virtual Networks
- Click + Create
- Name it VNet1
- Set address space to 10.0.0.0/16
- Create two subnets:
    - Subnet1: 10.0.1.0/24 (for Windows VM)
    - Subnet2: 10.0.2.0/24 (for Linux VM)
- Click Create

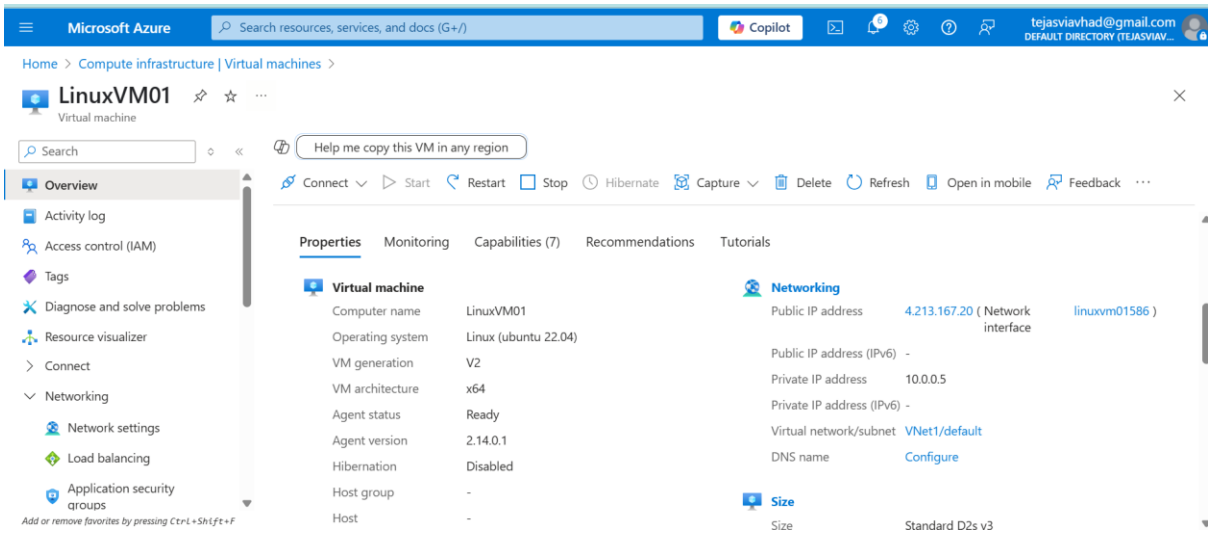




Step 3: Create Windows and Linux Virtual Machines

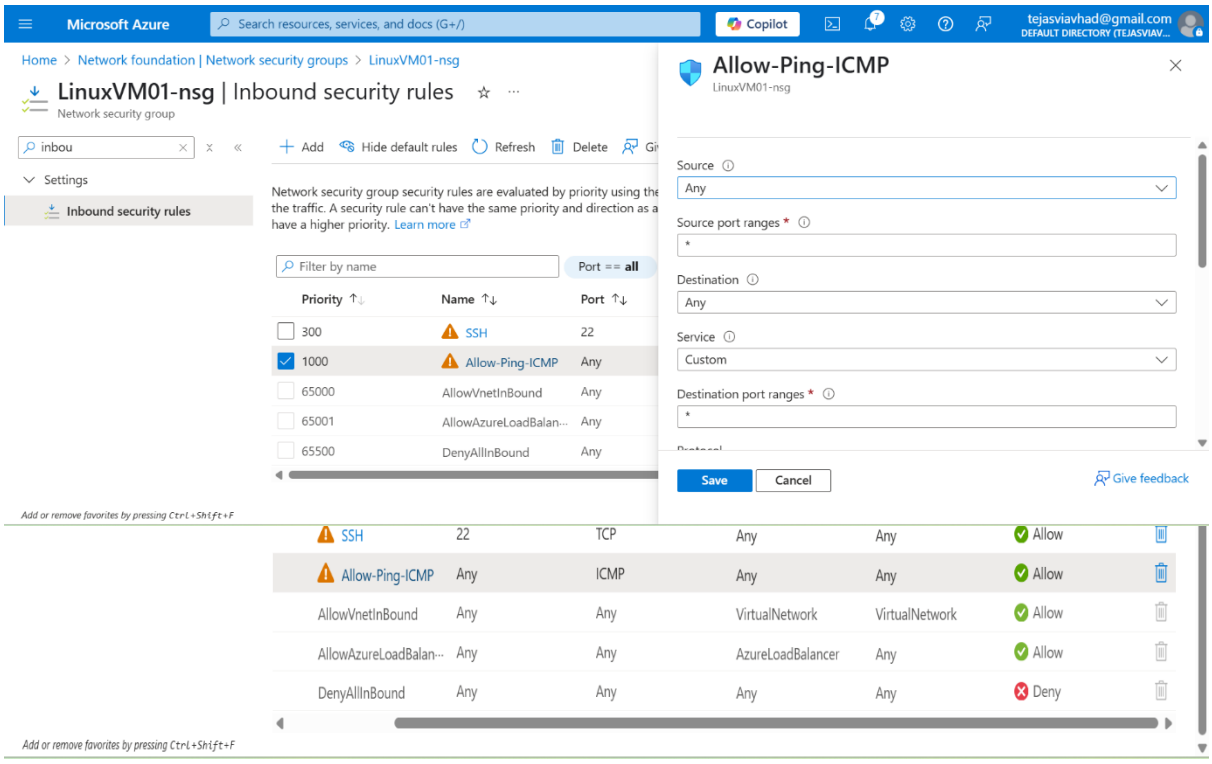
- Go to Virtual Machines click + Create
- Choose Windows Server 2019 for the first VM
- Place it in Subnet1 of VNet1
- Choose Ubuntu Linux for the second VM
- Place it in Subnet2 of VNet1
- Set login credentials
- Allow required ports (RDP for Windows, SSH for Linux)





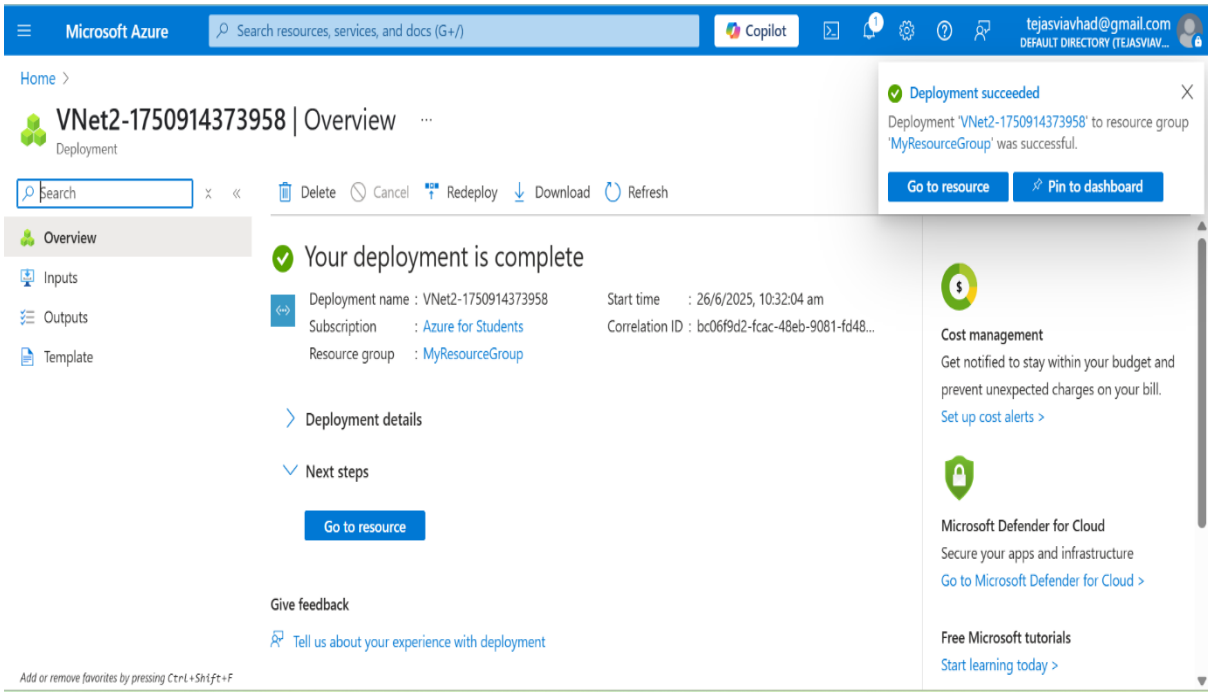
Step 4: Allow Ping (ICMP) in Network Security Group

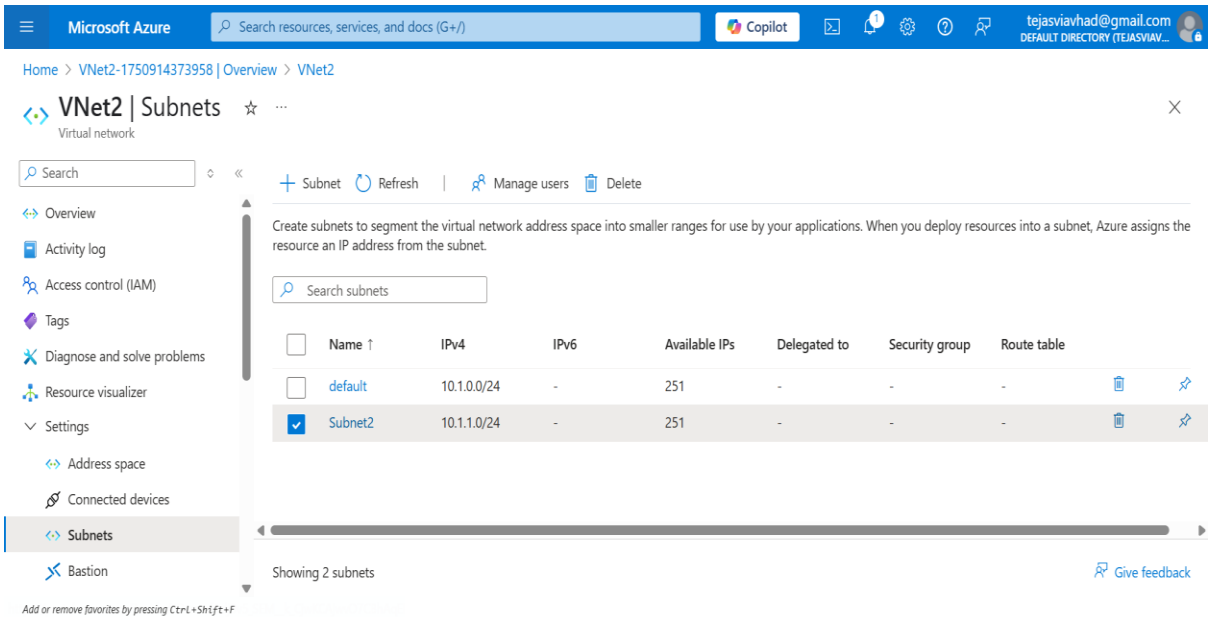
- Go to each VM's Networking settings
- Click on Network Security Group
- Add a rule to allow ICMP (Ping)



Step 5: Create Second VNet

- Go to Virtual Networks click on + Create
- Name it VNet2
- Set address space to 10.1.0.0/16
- Create one subnet inside it: 10.1.1.0/24
- Click Create





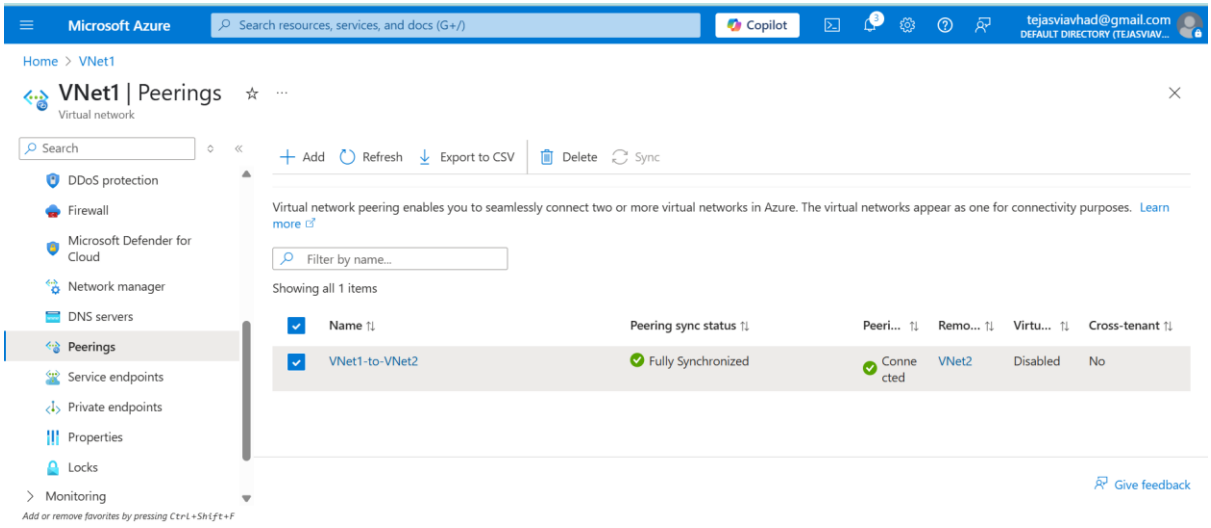
Step 6: Configure VNet Peering

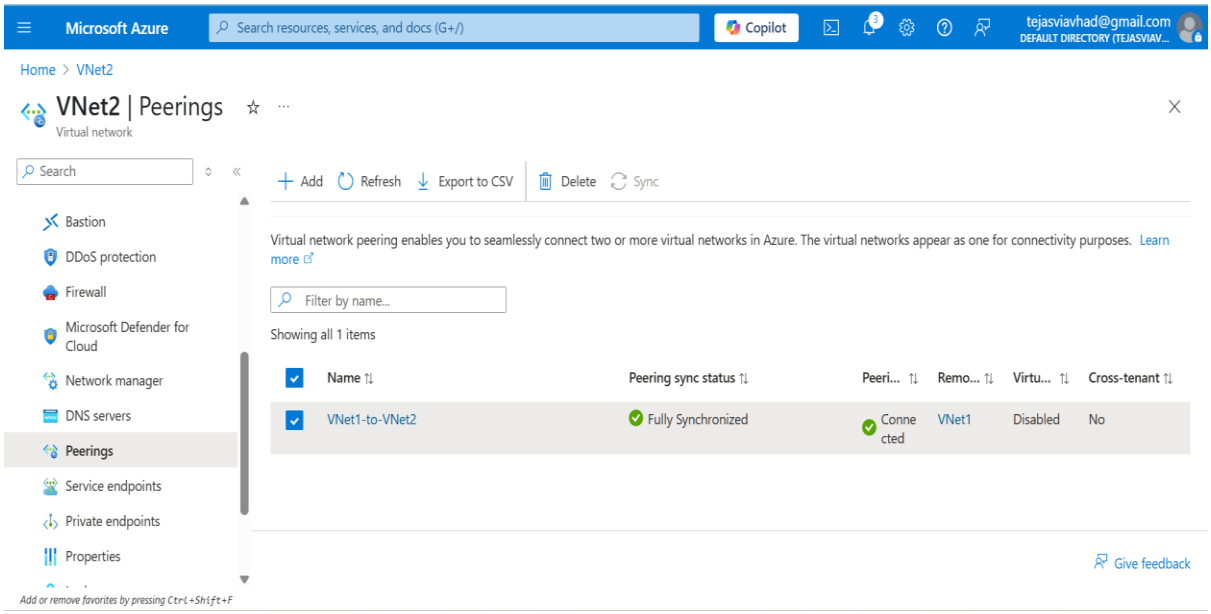
From VNet1 to VNet2:

- Go to VNet1 > Peerings > + Add
- Give name: VNet1-To-VNet2
- Select VNet2 as the peer
- Enable "Allow VNet access"
- Click Add

From VNet2 to VNet1:

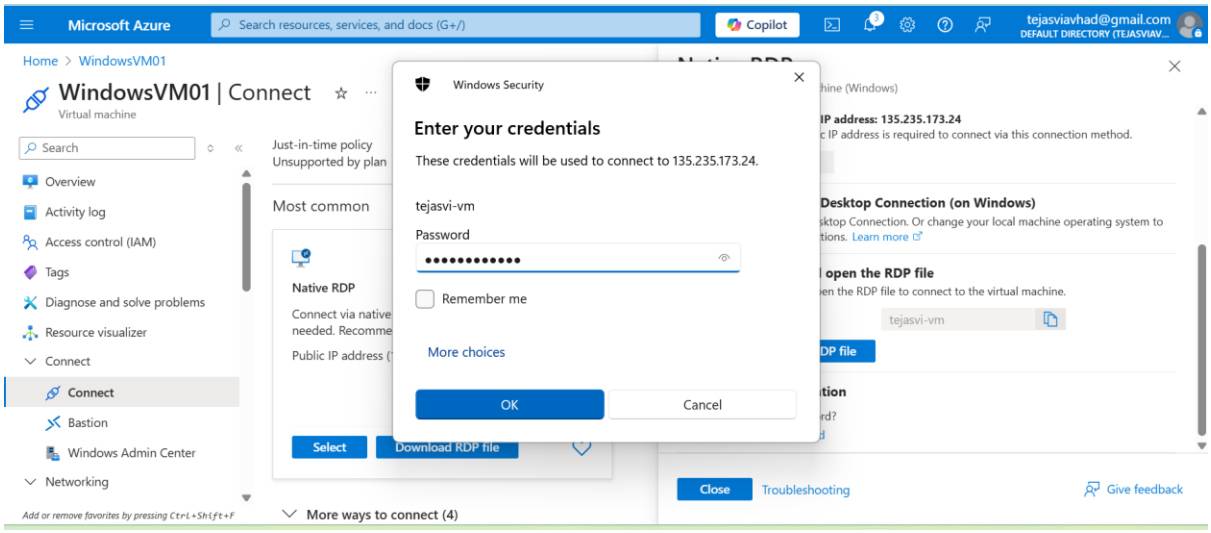
- Go to VNet2 > Peerings > + Add
- Give name: VNet2-To-VNet1
- Select VNet1 as the peer
- Enable "Allow VNet access"
- Click Add

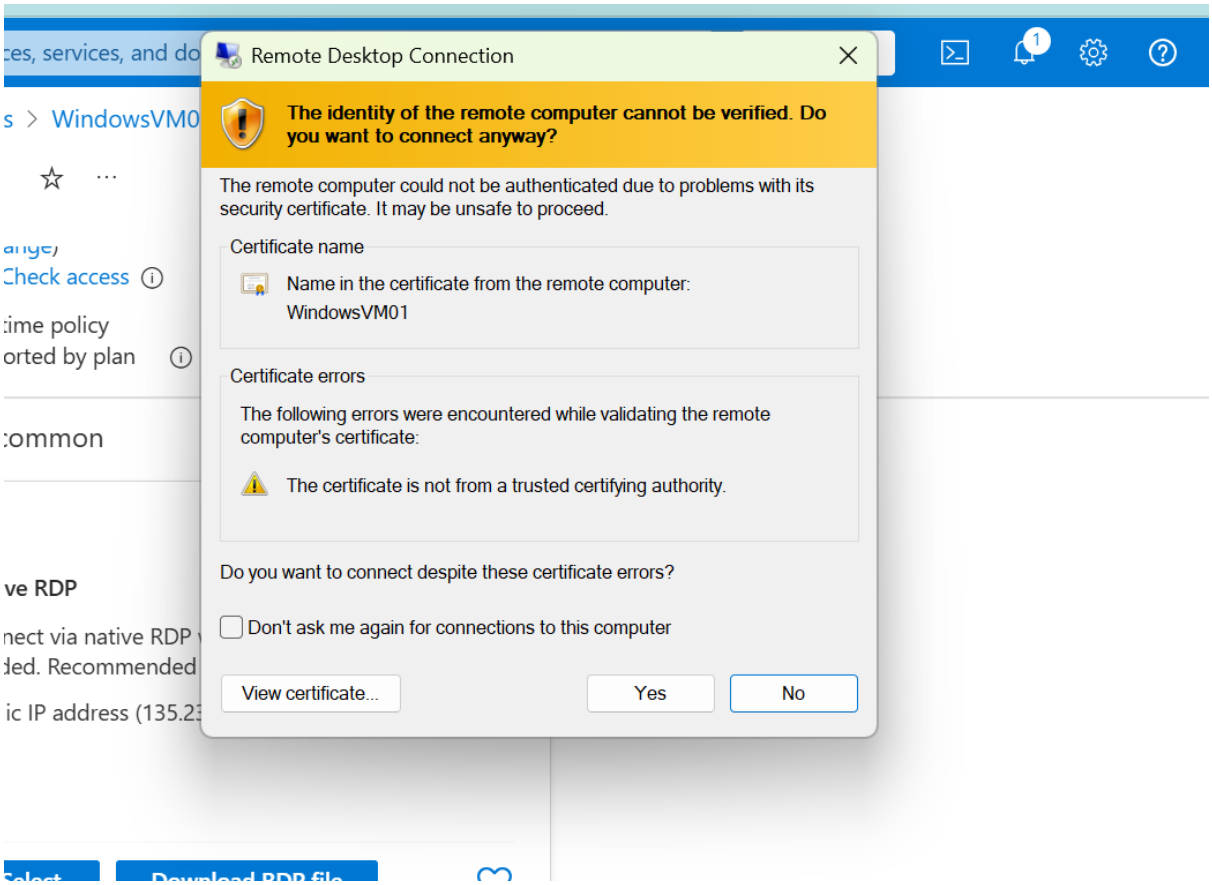


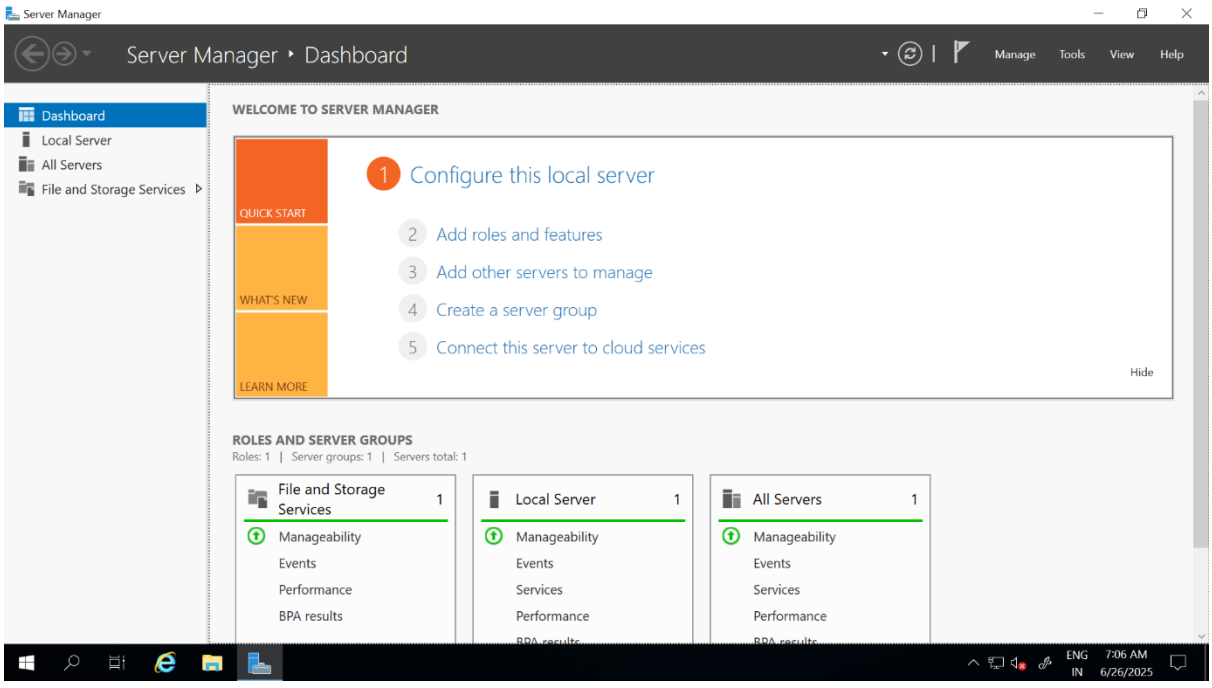


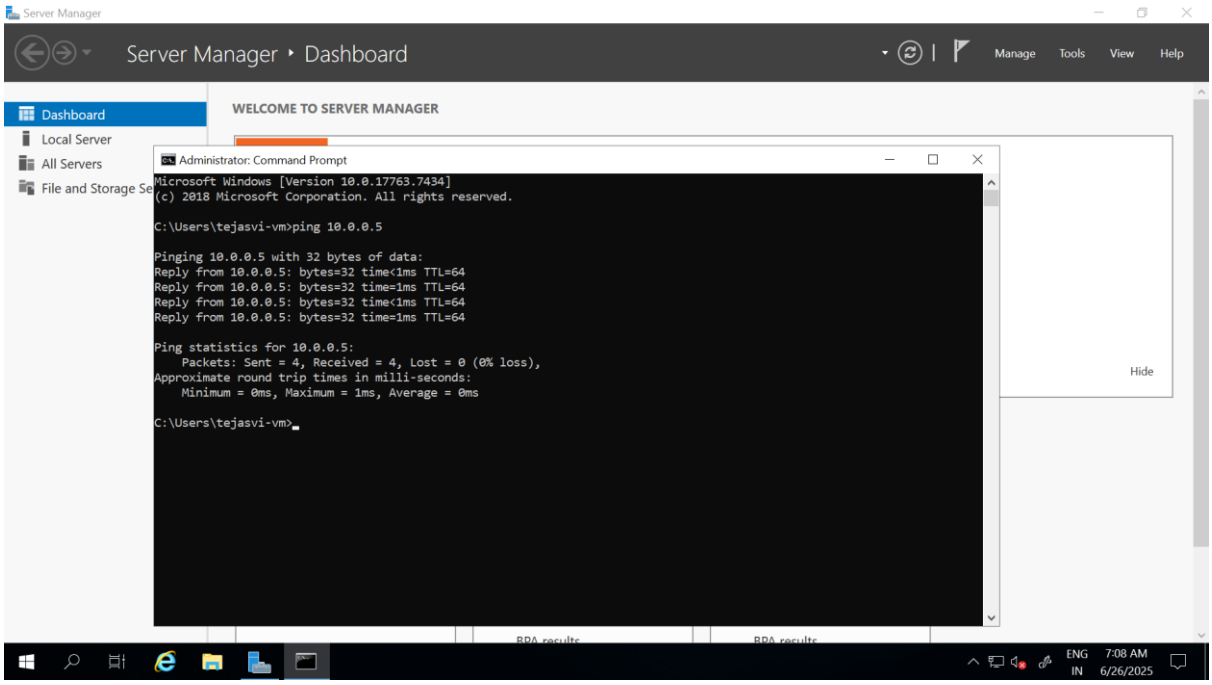
Step 7: Ping Test Between VMs

- From Windows VM, open Command Prompt:
  - Run: ping 10.0.0.5 (Linux VM private IP)
- Expected Output:
  - Replies with TTL=64, 0% loss




Remote Desktop Connection
The identity of the remote computer cannot be verified. Do you want to connect anyway?
The remote computer could not be authenticated due to problems with its security certificate. It may be unsafe to proceed.
Certificate name
Name in the certificate from the remote computer:
WindowsVM01
Certificate errors
The following errors were encountered while validating the remote computer's certificate:
The certificate is not from a trusted certifying authority.
Do you want to connect despite these certificate errors?
Don't ask me again for connections to this computer
View certificate...
Yes
No


Server Manager
Server Manager • Dashboard
Manage
Tools
View
Help
Dashboard
Local Server
All Servers
File and Storage Services
WELCOME TO SERVER MANAGER
QUICK START
WHAT'S NEW
LEARN MORE
1 Configure this local server
2 Add roles and features
3 Add other servers to manage
4 Create a server group
5 Connect this server to cloud services
Hide
ROLES AND SERVER GROUPS
Roles: 1 | Server groups: 1 | Servers total: 1
File and Storage Services 1
Manageability
Events
Performance
BPA results
Local Server 1
Manageability
Events
Services
Performance
All Servers 1
Manageability
Events
Services
Performance
ENG
IN
7:06 AM
6/26/2025



Peering done successfully.

9. Conclusion

This task helped in understanding how networks are managed in Azure. I learned:

- What CIDR is and how IP ranges work
- How to create and divide a VNet into subnets
- How to launch Windows and Linux VMs in different subnets
- How to connect two different VNets using VNet Peering
- How to test communication between VMs

9. References

- Azure Virtual Netwok Overview
- CIDR Notation Basics
- Azure Documentation and Portal