CIDR, VNet, Subnet, and VNet Peering (Azure Virtual Network)

1. Overview of Azure Virtual Network (VNet)

Azure Virtual Network (VNet) is the fundamental building block for private communication between Azure resources. It allows you to create isolated, logically segmented networks in the cloud, similar to traditional networks.

---

2. CIDR (Classless Inter-Domain Routing)

CIDR notation defines IP address ranges for VNets and Subnets.

- **CIDR Format: <IP Address>/<Prefix Length>**
  Example: 10.0.0.0/16
- **Prefix Length defines how many bits are fixed for network address; the rest are for host addressing.**

CIDR	Total IPs	Subnet Mask
/16	65,536	255.255.0.0
/24	256	255.255.255.0

---

3. Azure VNet and Subnet Basics

- **A VNet spans a region and contains multiple subnets.**
- **Subnets allow resource segmentation within a VNet.**

✓ You created:

- **VNet1 → CIDR: 10.0.0.0/16**
  - **Subnet1 (for Linux VM): 10.0.1.0/24**
  - **Subnet2 (for Windows VM): 10.0.2.0/24**
- **VNet2 → CIDR: 10.1.0.0/16**
  - **Subnet3: 10.1.1.0/24**

---

4. VM Deployment (Use Case Execution)

✓ You Deployed:

- **Linux VM in VNet1/Subnet1**
- **Windows VM in VNet1/Subnet2**
- **Enabled ping between both VMs by:**
    - **Adding firewall rules in both OS**
    - **Ensuring NSG allows ICMP (ping)**

---

5. Azure VNet Peering

VNet Peering connects two VNets to enable private communication.

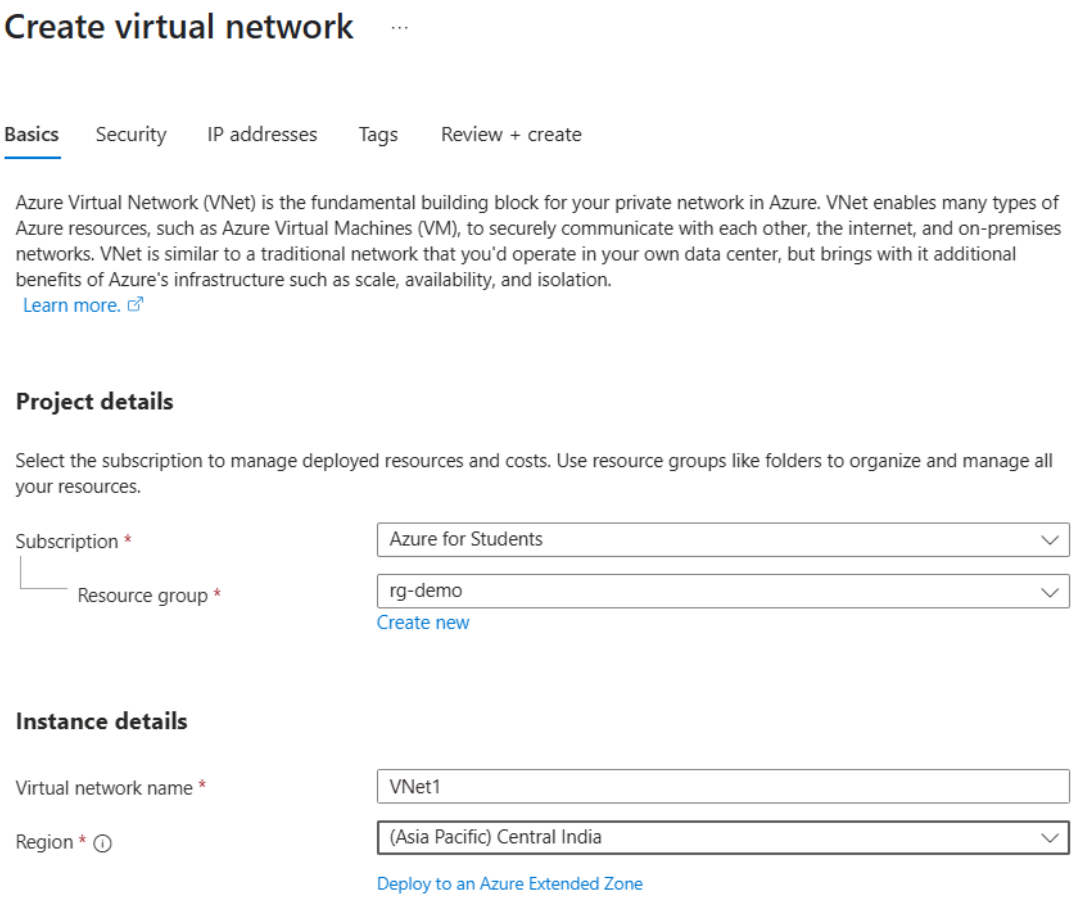
Type	**Direction**	**Notes**
Intra-region	↔	**Between VNets in same region**
Global Peering 🌐		**Between VNets in different regions**

✓ You Configured:

- **Peering: VNet1 ↔ VNet2**
- **Allowed traffic in both directions (no gateway or forwarded traffic)**

STEP 1: Create VNet1 in Central India with 2 subnets

1. Go to Azure Portal
2. Search **Virtual Networks** > **+ Create**
3. In **Basics**:
    - Subscription: your default
    - Resource Group: create new → rg-demo
    - Name: VNet1
    - Region: **Central India**



1. In **IP Addresses**:
    - IP space: 10.0.0.0/16
    - Click **+ Add subnet**:
        - Subnet 1: Name windows-subnet, IP 10.0.1.0/24
        - Subnet 2: Name linux-subnet, IP 10.0.2.0/24

Basics Security **IP addresses** Tags Review + create

Configure your virtual network address space with the IPv4 and IPv6 addresses and subnets you need. Learn more

Define the address space of your virtual network with one or more IPv4 or IPv6 address ranges. Create subnets to segment the virtual network address space into smaller ranges for use by your applications. When you deploy resources into a subnet, Azure assigns the resource an IP address from the subnet. Learn more

+ Add a subnet

10.0.0.0/16 Delete address space

10.0.0.0 /16

10.0.0.0 - 10.0.255.255 65,536 addresses

Subnets	IP address range	Size	NAT gateway
default	10.0.0.0 - 10.0.0.255	/24 (256 addresses)	-
windows-subnet	10.0.1.0 - 10.0.1.255	/24 (256 addresses)	-
linux-subnet	10.0.2.0 - 10.0.2.255	/24 (256 addresses)	-

Add IPv4 address space

Create virtual network ...

Basics Security IP addresses Tags **Review + create**

View automation template

Basics

Subscription	Azure for Students
Resource Group	rg-demo
Name	VNet1
Region	Central India

Security

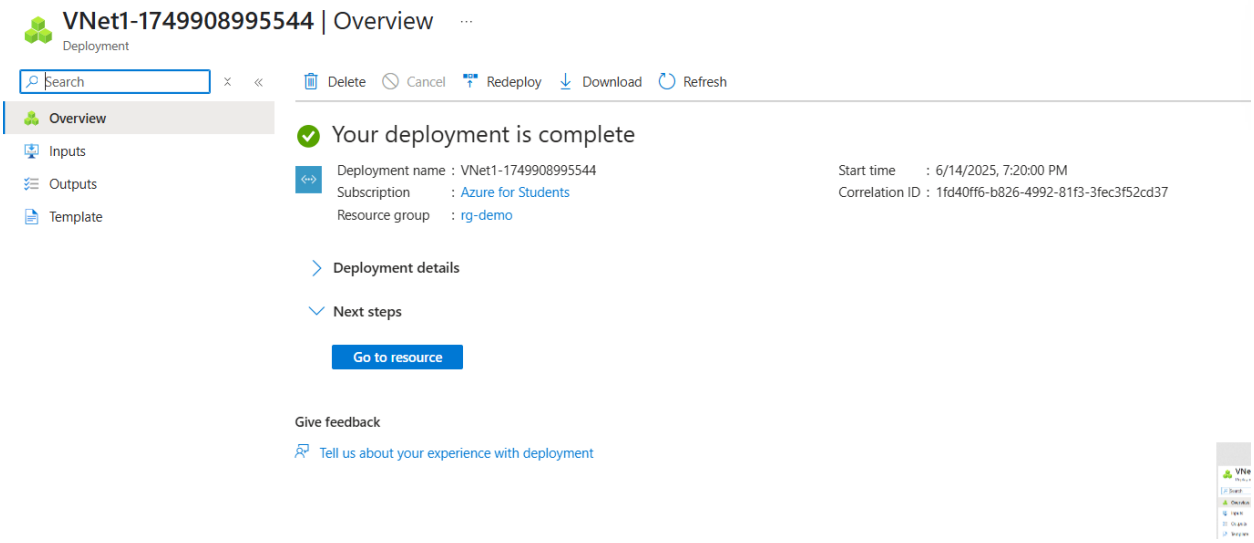
Azure Bastion	Disabled
Azure Firewall	Disabled
Azure DDoS Network Protection	Disabled

IP addresses

Address space	10.0.0.0/16 (65,536 addresses)
Subnet	default (10.0.0.0/24) (256 addresses)
Subnet	windows-subnet (10.0.1.0/24) (256 addresses)
Subnet	linux-subnet (10.0.2.0/24) (256 addresses)

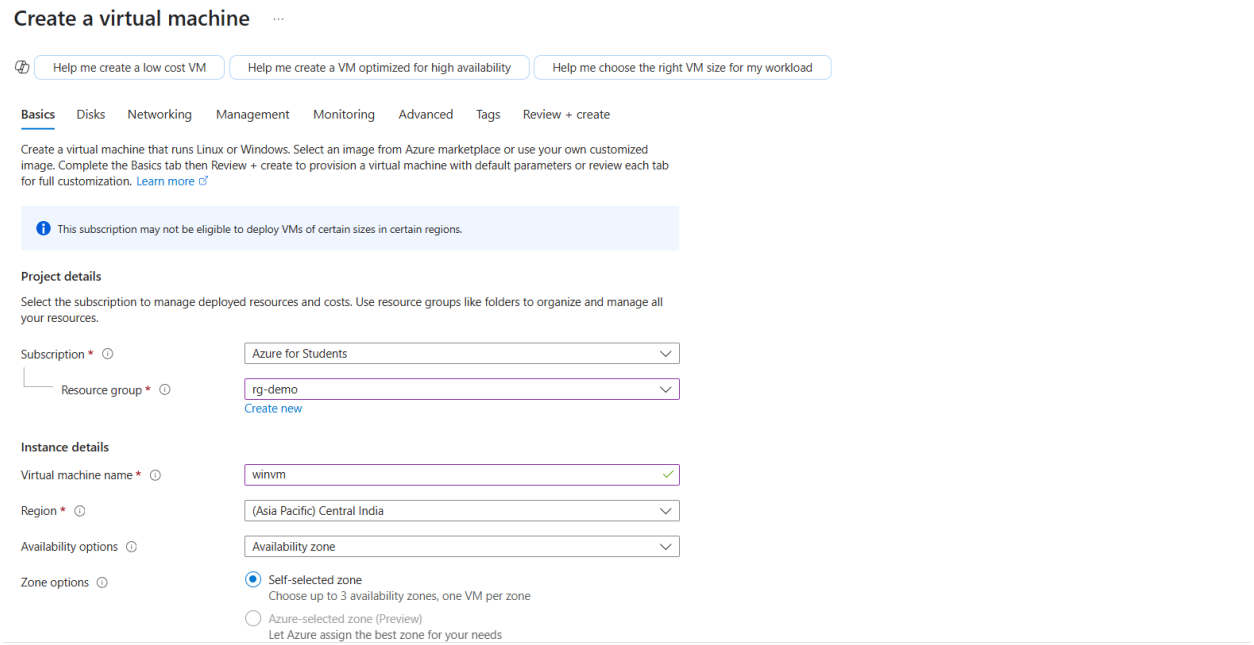
Tags

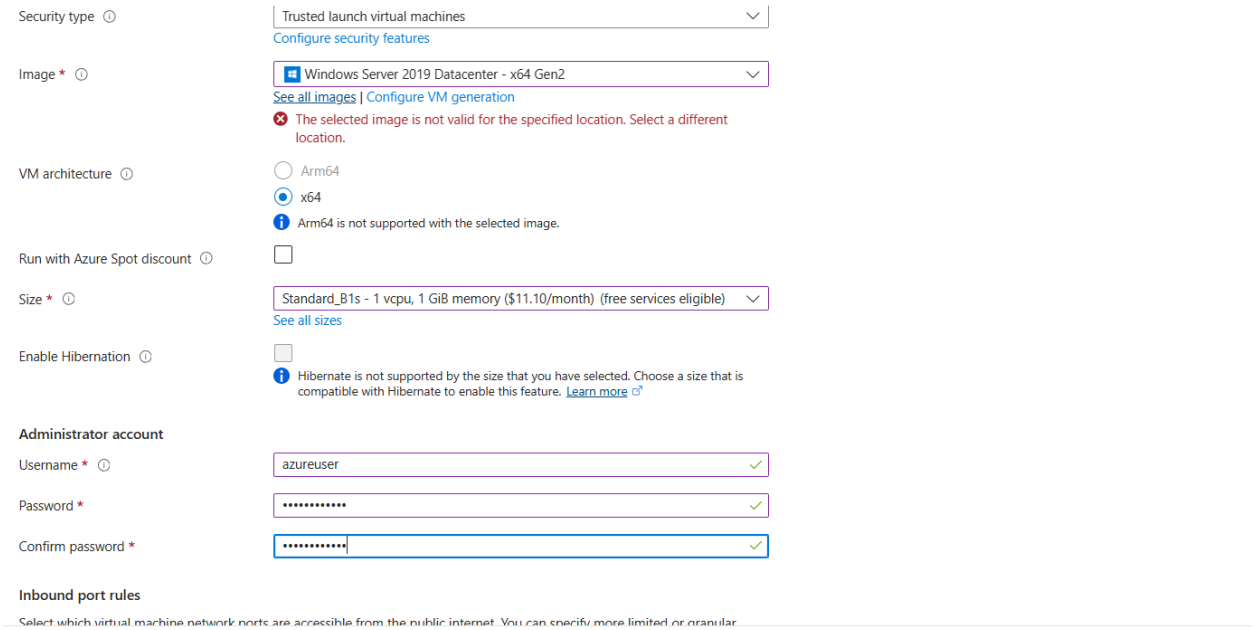
1. Click **Review + Create → Create**



STEP 2: Create Cheapest Windows VM in windows-subnet

1. Go to **Virtual Machines** > **+ Create**
2. In **Basics**:
    - Name: winvm
    - Region: **Central India**
    - Image: **Windows Server 2019 Datacenter**
    - Size: Click "See all sizes" → choose **B1s** (cheapest)
    - Username: azureuser
    - Password: your strong password
    - Inbound ports: **RDP (3389)**
3. In **Networking** tab:
    - VNet: VNet1
    - Subnet: windows-subnet
4. Click **Review + Create → Create**



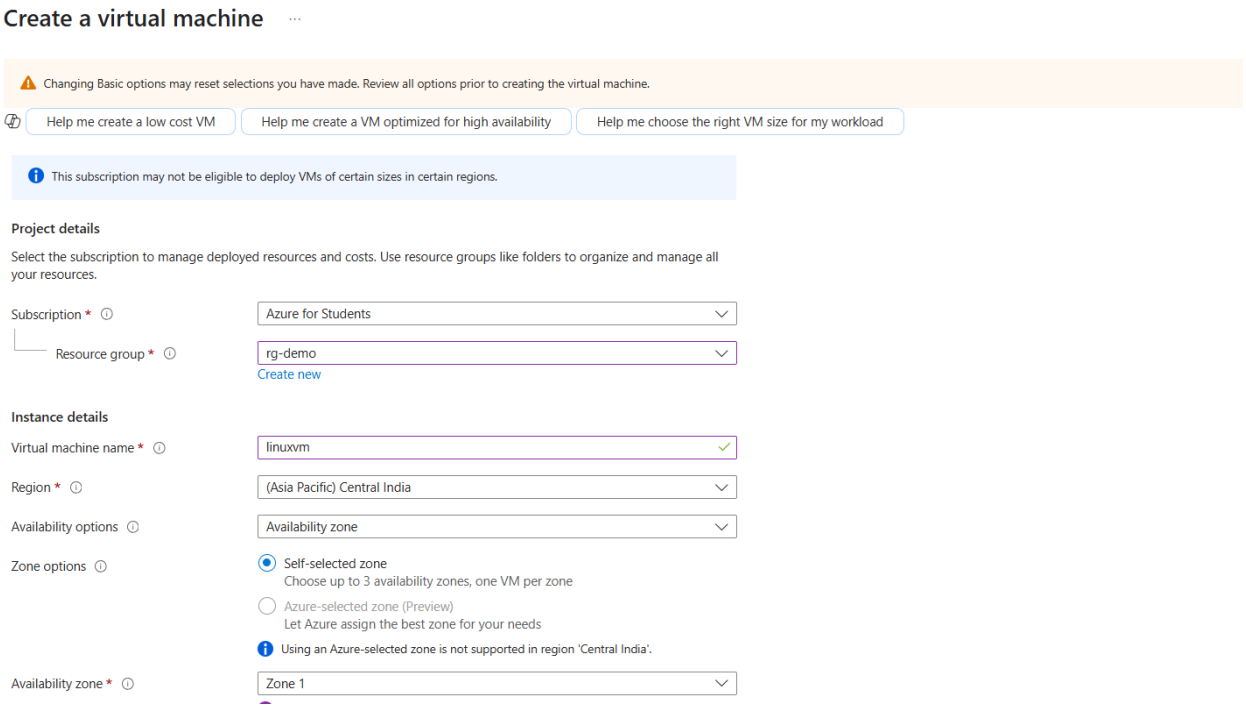


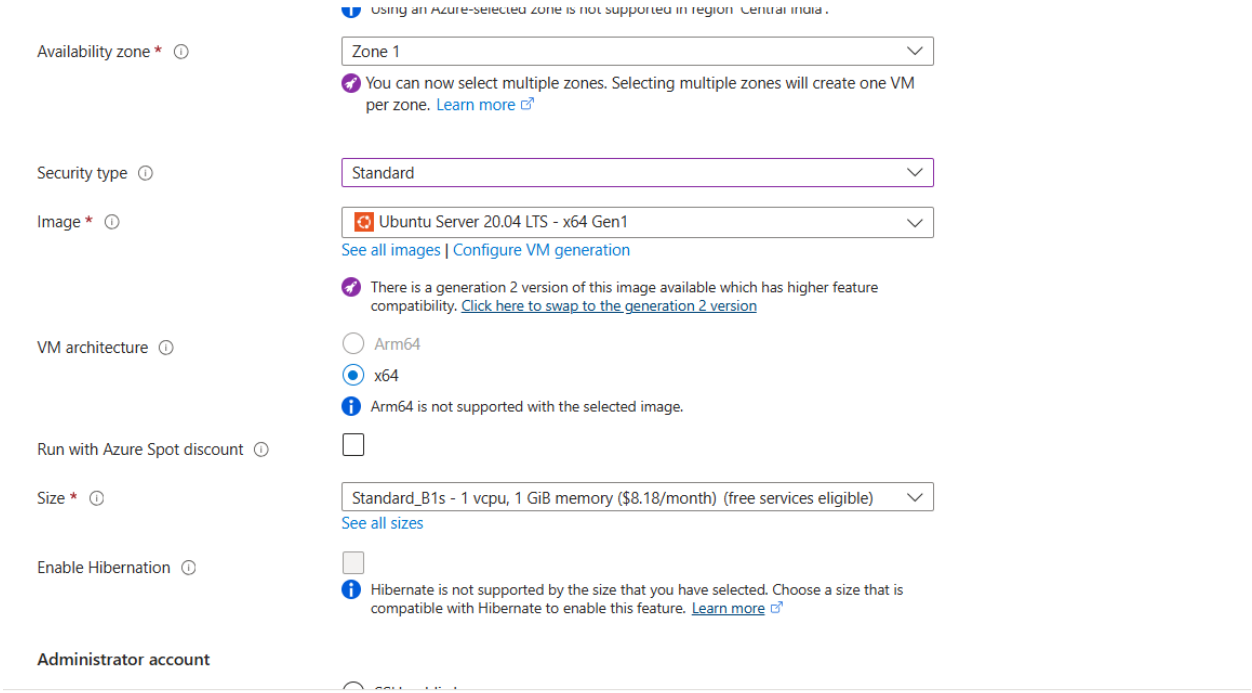
STEP 3: Create Cheapest Linux VM in linux-subnet

1. Same as above but:
    - Name: linuxvm
    - Image: **Ubuntu Server 20.04 LTS (Gen1)** → Free tier

- Size: **B1s**
- Inbound ports: **SSH (22)**
- Subnet: linux-subnet

2. Create it





STEP 4: Allow Ping (ICMP) Between VMs

You have:

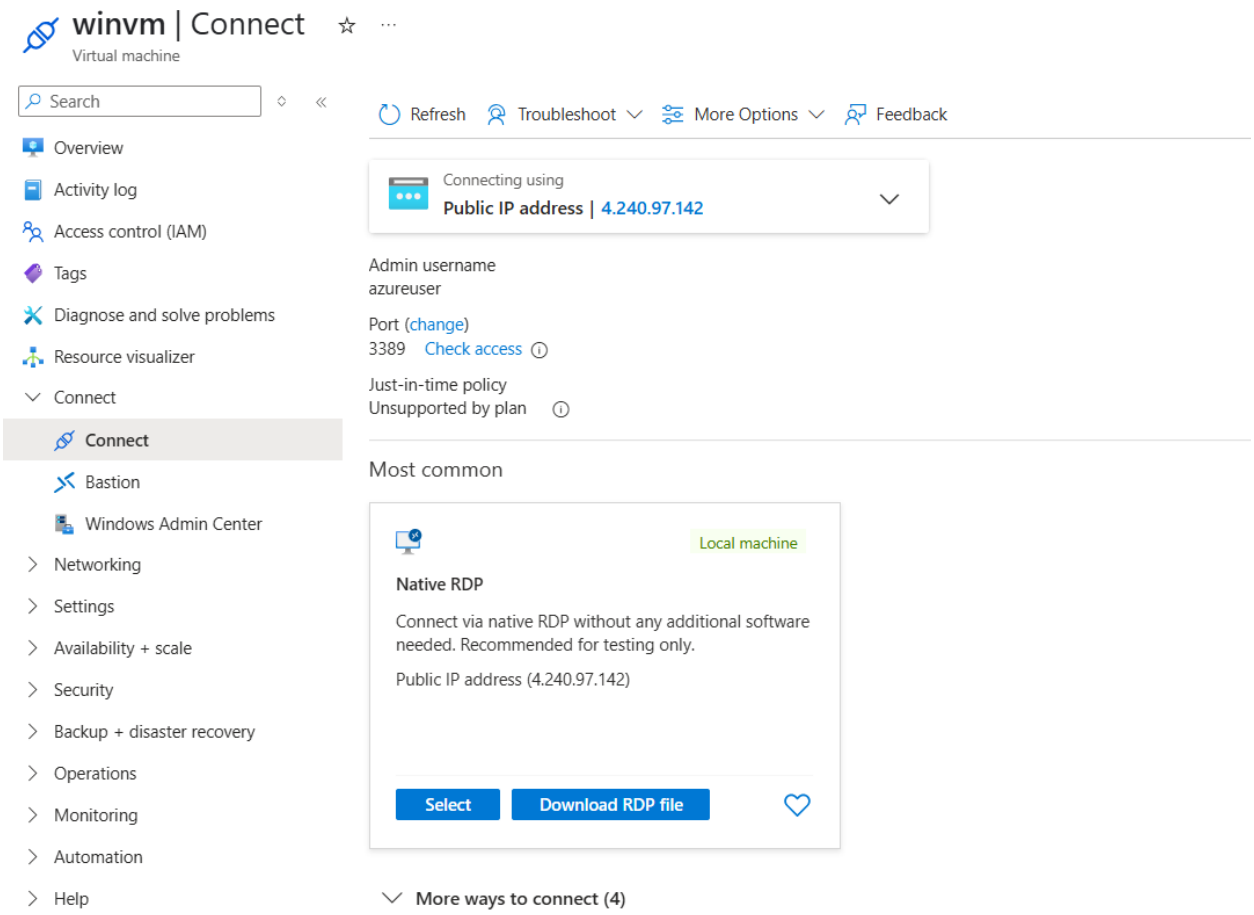
- A **Windows VM** in windows-subnet
- A **Linux VM** in linux-subnet
- Both inside the **same VNet1**

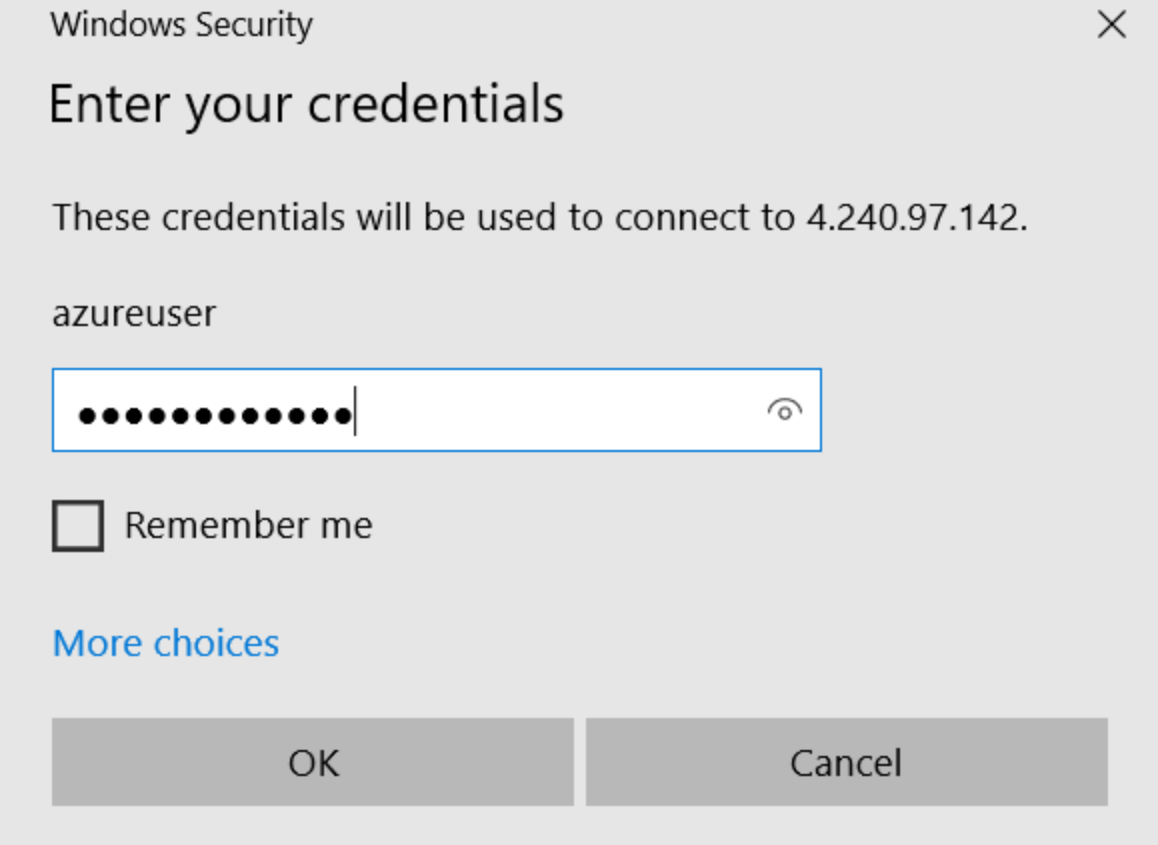
Now, follow these two sub-steps:

□ A. On the Windows VM

1. Connect to Windows VM using RDP:

- Go to the Azure Portal → **Virtual Machines**
- Click your WindowsVM → Click **Connect** → Choose **RDP**
- Download the .rdp file and **open it**
- Login with the username and password you gave while creating the VM

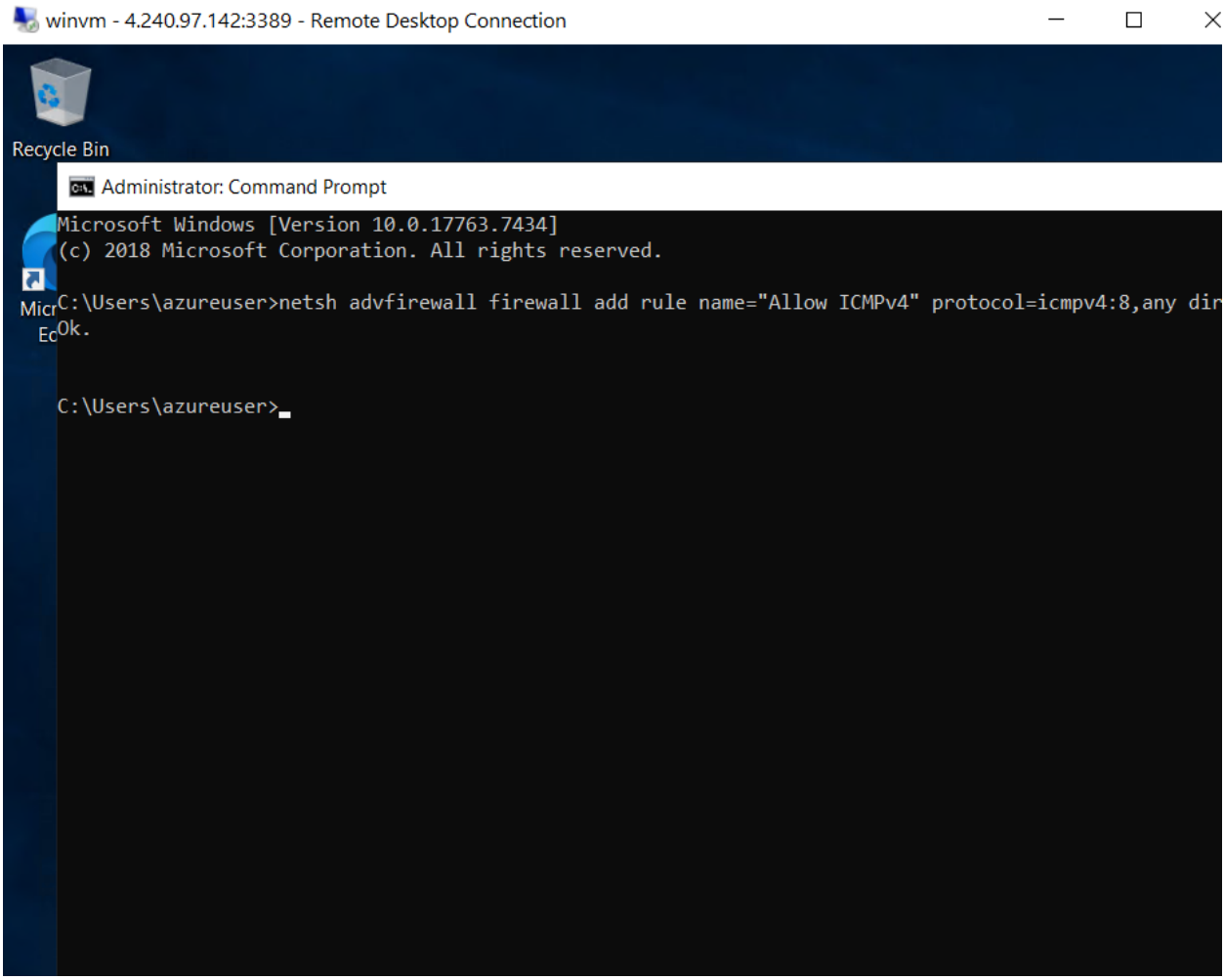

winvm | Connect
Virtual machine
Search
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Resource visualizer
Connect
Connect
Bastion
Windows Admin Center
Networking
Settings
Availability + scale
Security
Backup + disaster recovery
Operations
Monitoring
Automation
Help
Refresh
Troubleshoot
More Options
Feedback
Connecting using
Public IP address | 4.240.97.142
Admin username
azureuser
Port (change)
3389
Check access
Just-in-time policy
Unsupported by plan
Most common
Local machine
Native RDP
Connect via native RDP without any additional software needed. Recommended for testing only.
Public IP address (4.240.97.142)
Select
Download RDP file
More ways to connect (4)



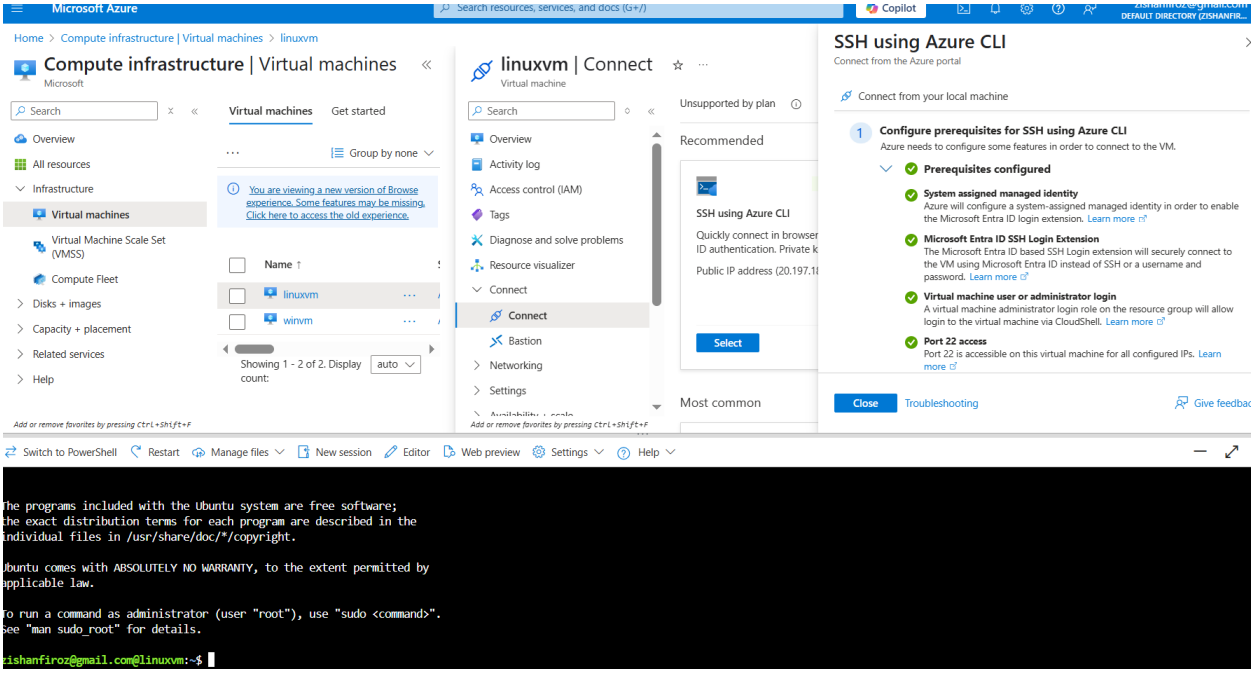
1. On your Windows VM:
    - Press **Start**
    - Type cmd
    - **Right-click** on **Command Prompt → Run as Administrator**
2. Then paste this:

```
netsh advfirewall firewall add rule name="Allow ICMPv4" protocol=icmpv4:8,any dir=in action=allow
```

This command adds a firewall rule to allow **ping (ICMPv4)** traffic.


winvm - 4.240.97.142:3389 - Remote Desktop Connection
Recycle Bin
Administrator: Command Prompt
Microsoft Windows [Version 10.0.17763.7434]
(c) 2018 Microsoft Corporation. All rights reserved.
C:\Users\azureuser>netsh advfirewall firewall add rule name="Allow ICMPv4" protocol=icmpv4:8,any dir
Ok.
C:\Users\azureuser>

For Linux VM:



Command to Allow ICMP (ping) in UFW:

Run these one by one in your **Linux VM**:

Use iptables (Advanced way)

If you want to **allow only ICMP** without disabling UFW:

sudo iptables -A INPUT -p icmp --icmp-type echo-request -s 10.0.0.0/16 -j ACCEPT

sudo iptables -A INPUT -p icmp --icmp-type echo-reply -s 10.0.0.0/16 -j ACCEPT

Then save it (so it persists):

sudo apt install iptables-persistent -y

sudo netfilter-persistent save

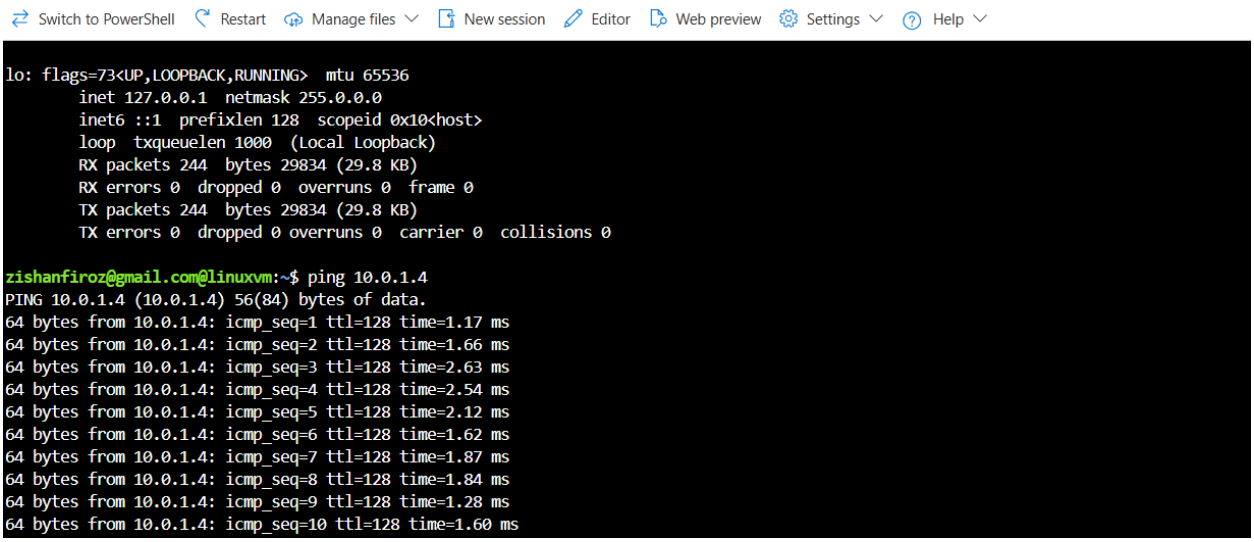
```
Switch to PowerShell  Restart  Manage files  New session  Editor  Web preview  Settings  Help
zishanfiroz@gmail.com@linuxvm:~$ sudo ufw allow from 10.0.0.0/16 to any proto icmp
ERROR: Unsupported protocol 'icmp'
zishanfiroz@gmail.com@linuxvm:~$ sudo ufw disable
Firewall stopped and disabled on system startup
zishanfiroz@gmail.com@linuxvm:~$ sudo iptables -A INPUT -p icmp --icmp-type echo-request -s 10.0.0.0/16 -j ACCEPT
zishanfiroz@gmail.com@linuxvm:~$ sudo iptables -A INPUT -p icmp --icmp-type echo-reply -s 10.0.0.0/16 -j ACCEPT
zishanfiroz@gmail.com@linuxvm:~$ sudo apt install iptables-persistent -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  netfilter-persistent
The following NEW packages will be installed:
  iptables-persistent netfilter-persistent
0 upgraded, 2 newly installed, 0 to remove and 26 not upgraded.
Need to get 13.8 kB of archives.
After this operation, 89.1 kB of additional disk space will be used.
Get:1 http://azure.archive.ubuntu.com/ubuntu focal-updates/universe amd64 netfilter-persistent all 1.0.14ubuntu1 [7268 B]
Get:2 http://azure.archive.ubuntu.com/ubuntu focal-updates/universe amd64 iptables-persistent all 1.0.14ubuntu1 [6552 B]
Fetched 13.8 kB in 1s (17.9 kB/s)
Preconfiguring packages ...
Selecting previously unselected package netfilter-persistent.
```

Now Testing Ping:

Ping <private linux ip>

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.17763.7434]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\azureuser>ping 20.197.18.73

Pinging 20.197.18.73 with 32 bytes of data:
Request timed out.

Ping statistics for 20.197.18.73:
    Packets: Sent = 1, Received = 0, Lost = 1 (100% loss),
Control-C
^C
C:\Users\azureuser>ping 10.0.2.4

Pinging 10.0.2.4 with 32 bytes of data:
Reply from 10.0.2.4: bytes=32 time=2ms TTL=64
Reply from 10.0.2.4: bytes=32 time=1ms TTL=64
Reply from 10.0.2.4: bytes=32 time=1ms TTL=64
Reply from 10.0.2.4: bytes=32 time=1ms TTL=64

Ping statistics for 10.0.2.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 2ms, Average = 1ms

C:\Users\azureuser>
```

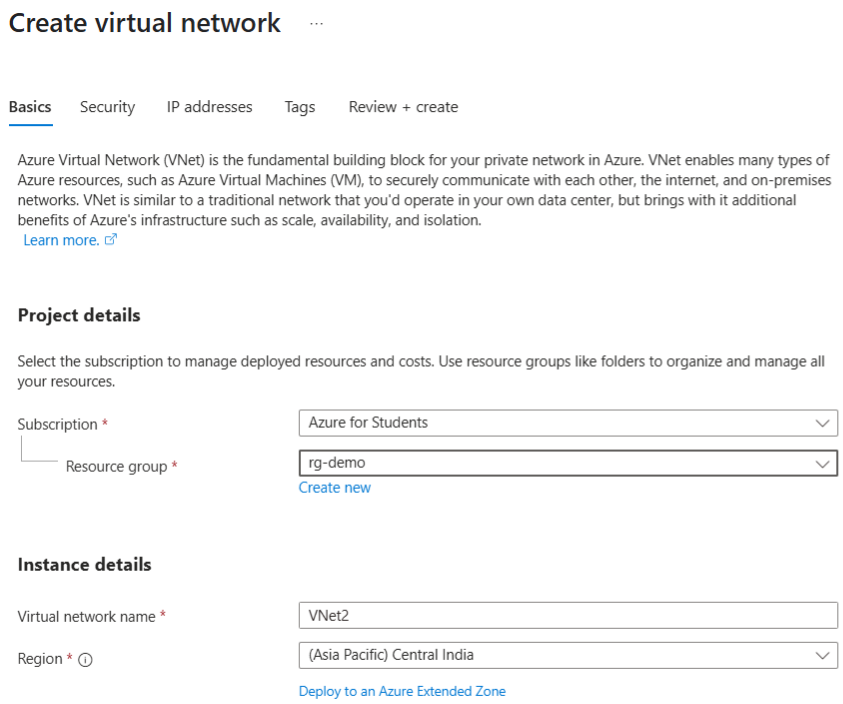
Now same test from linux to windows



Step 5: Create VNet2 in Central India (or same region as VNet1)

🔷 1. Create VNet2

1. Go to **Azure Portal** → Search **Virtual Networks** → Click **+ Create**
2. In **Basics**:
    - Resource Group: rg-demo (or same as VNet1)
    - Name: VNet2
    - Region: **Central India** (same as VNet1)
3. In **IP Addresses** tab:
    - IP space: 10.1.0.0/16
    - Subnet name: vnet2-subnet
    - Subnet range: 10.1.1.0/24
4. Click **Review + Create** → **Create**



🔗 2. Set Up VNet Peering Between VNet1 and VNet2

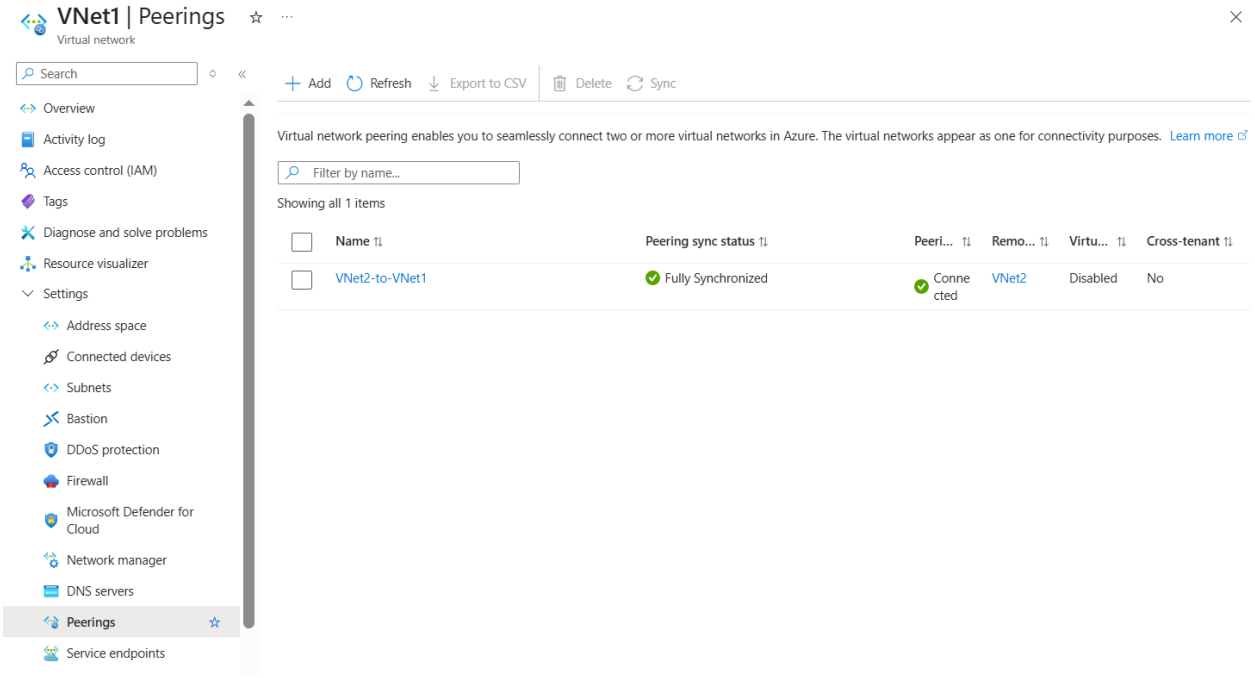
You'll create **two peering connections** — one from VNet1 → VNet2, and one from VNet2 → VNet1.

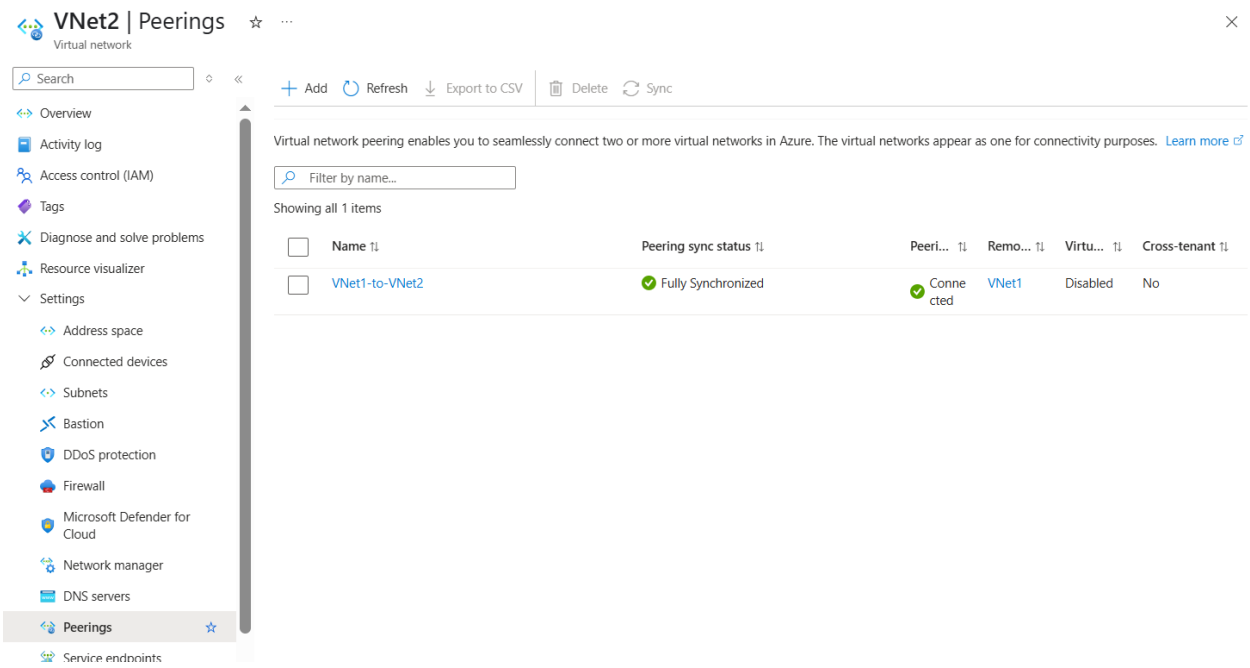
🔁 A. Peering from VNet1 to VNet2

1. Go to **Virtual Networks** → Click on **VNet1**
2. Go to **Peerings** (left menu) → Click **+ Add**
3. Fill the form:
    - Peering link name: VNet1-to-VNet2
    - Remote virtual network: Select **VNet2**
    - Allow traffic between virtual networks: **Yes**
    - Allow forwarded traffic: **No**
    - Use remote gateway: **No**
4. Click **Add**

B. Peering from VNet2 to VNet1

1. Fill the form:
    - Peering link name: VNet2-to-VNet1
    - Remote virtual network: Select **VNet1**
    - Allow traffic between virtual networks: **Yes**
    - Allow forwarded traffic: **No**
    - Use remote gateway: **No**
2. Click **Add**




VNet2 | Peerings
Virtual network
Search
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Resource visualizer
Settings
Address space
Connected devices
Subnets
Bastion
DDoS protection
Firewall
Microsoft Defender for Cloud
Network manager
DNS servers
Peerings
Service endpoints
Add
Refresh
Export to CSV
Delete
Sync
Virtual network peering enables you to seamlessly connect two or more virtual networks in Azure. The virtual networks appear as one for connectivity purposes. Learn more
Filter by name...
Showing all 1 items
Name
Peering sync status
Peeri...
Remo...
Virtu...
Cross-tenant
VNet1-to-VNet2
Fully Synchronized
Connected
VNet1
Disabled
No