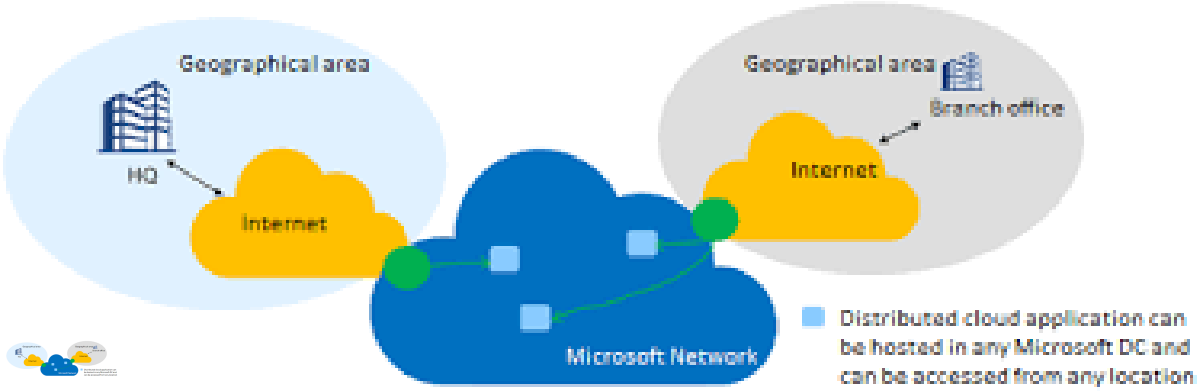
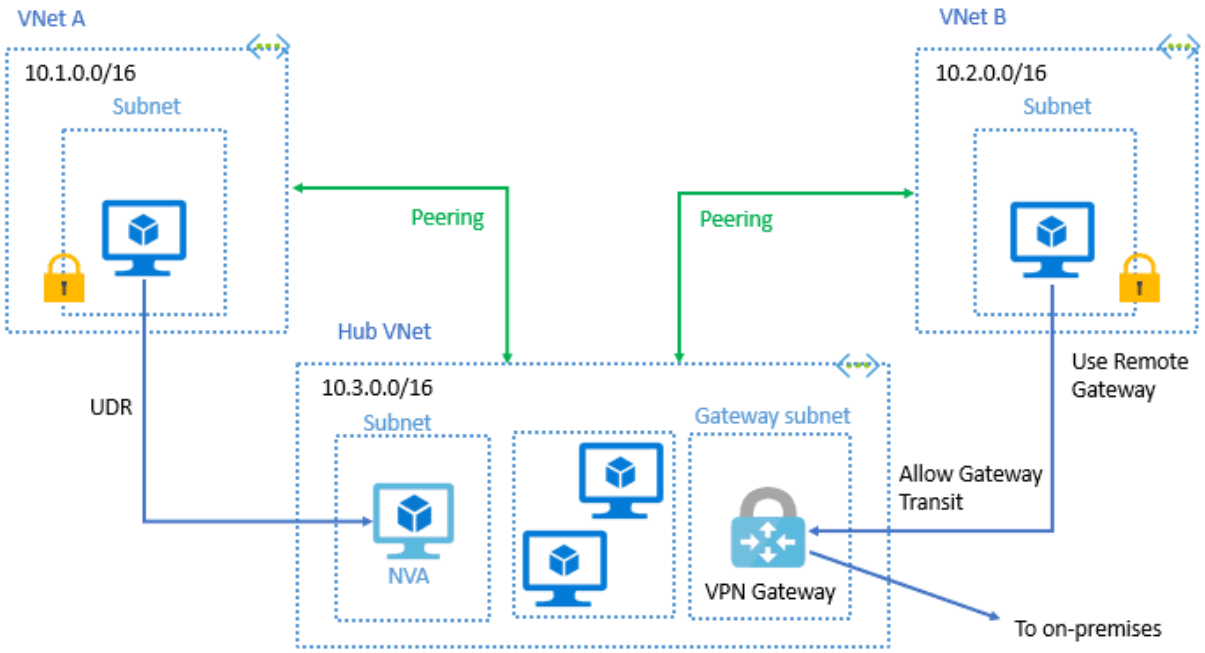


stories 7: Peering two Vnet in Azure

What Is Azure VNet Peering?

Azure Virtual Network is used for the Virtual Network Peering empowers users to flawlessly communicate with virtual networks in Azure. **VNet Peering in Azure** allows the traffic of one virtual network to communicate to another virtual network. This is basically used for database failover, disaster recovery, or cross-region data replication.VPN gateways are used in an encrypted connection in the region but VNet Peering provides connection sharing in different regions.



Virtual Network (VNet) Peering in Azure

Importance Of VNet Peering

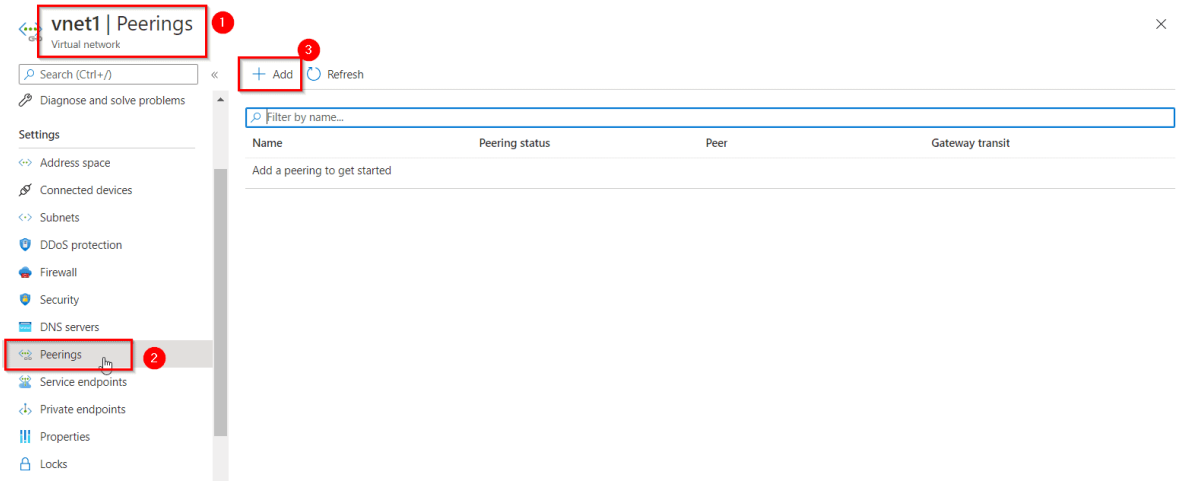
- VNet peering is similar to an inter-VLAN Routing in VLAN of On-premise networks so it works similarly to inter-VLAN connect to one VLAN to another VLAN for communication.
- In Azure infrastructure, need to connect to virtual networks to each other for sharing traffic which can be applications, backup, replication, recovery, or information sharing.
- The virtual machines of virtual network connections to other virtual machines of different Virtual network via connection of VNet Peering in the same region or across the region

Benefits

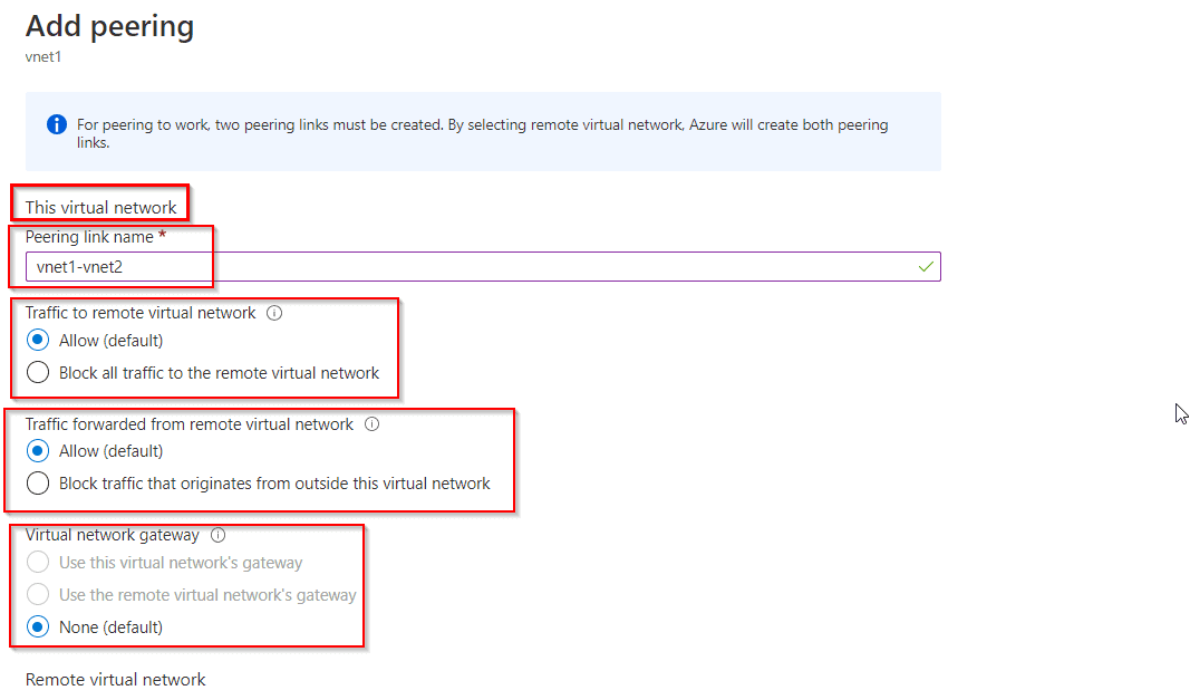
- Network traffic of peered Virtual networks become private.
- Virtual network peering in Azure allows transferring data across Azure deployment models, subscriptions, and other regions.
- No downtime issues in global Azure virtual network peering.
- It configures the connection with high bandwidth Low latency in the VNet region.

Step-by-Step Configuration

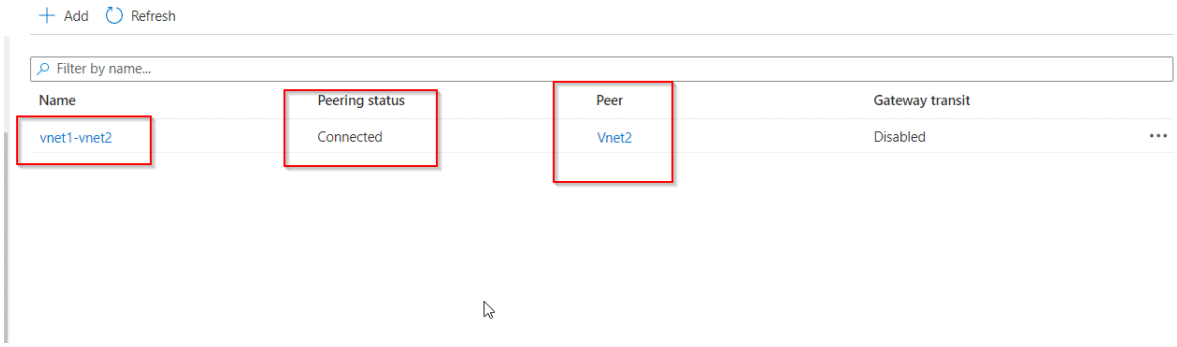
1. Log in to the Azure portal at https://portal.azure.com.
2. Create two Virtual networks in the same or Different regions like ***Vnet1***, ***Vnet2*****,** etc.
3. Now go to any one of the two **Virtual Networks** and select **Peerings**, under **Settings**, and then select **Add.**



4. Configuring the peering for the two virtual networks and select, **Add.This virtual network:** means the vnet 1**Remote virtual network:** means the vnet (here vnet2) which you want to peer the vnet1 with.

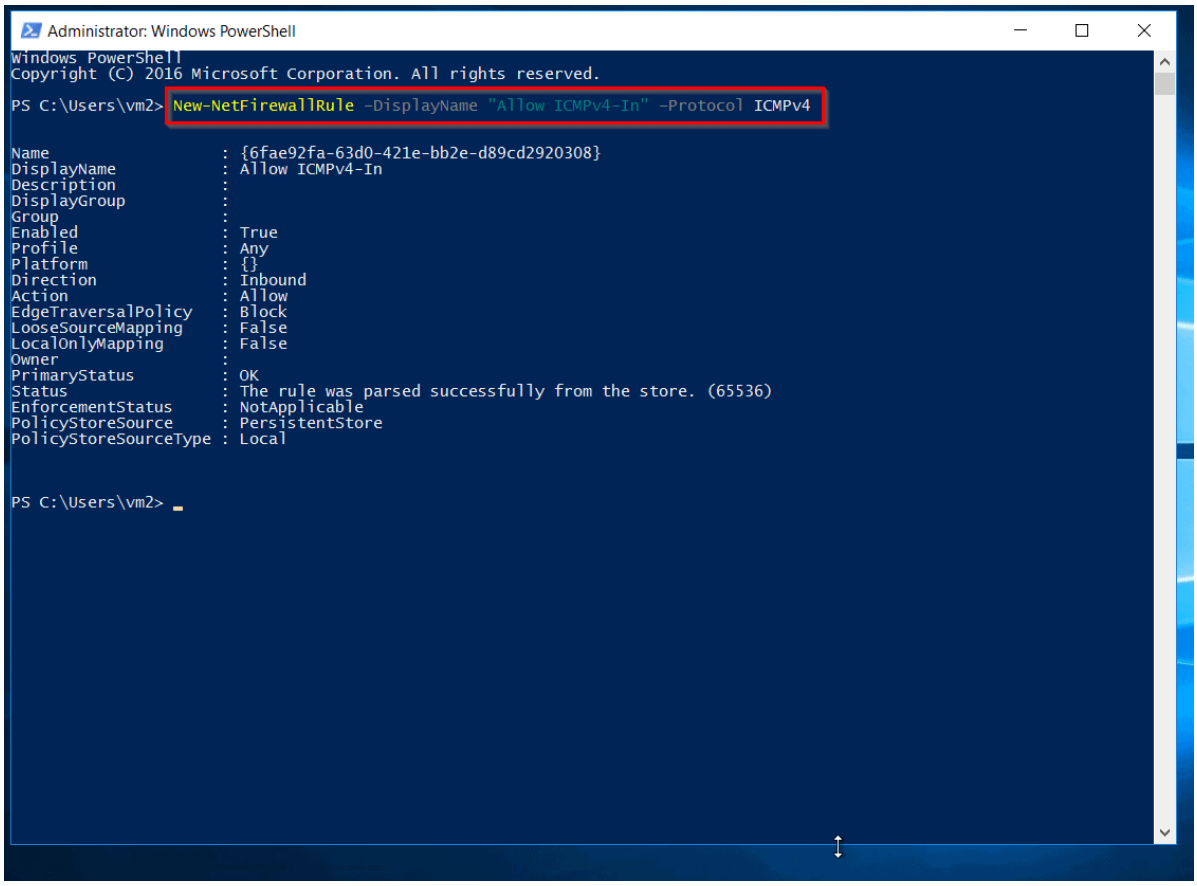


5. The **PEERING STATUS** is Connected, as shown in the following picture:If you don't see the status, refresh your browser.***Note:*** *Configuring peering on anyone VNet will configure the peering automatically on the other VNet as well.*



6. Now connect anyone of your VM and then try to ping the **Public IP** of the second Virtual Machine to test the peering.***Note:*** *If you are using a Windows Server VM, the* ***ping*** *will fail, because ping uses the Internet Control Message Protocol (ICMP). By default, ICMP isn't allowed through the Windows firewall.*

7. To allow VM1 to ping VM2 in a later step, enter this command in the **VM2** Powershell.***New-NetFirewallRule –DisplayName "Allow ICMPv4-In" –Protocol ICMPv4NOTE:*** *You have to enter this command on the other VM. (Here VM2).*



8. Now, close the remote desktop connection to VM2 and connect to the VM1, then again ping the **Public IP** of the second VM.

Also Check: Azure VPN Gateway vs ExpressRoute, to know the major differences between them

Now you will see that your Virtual Machines is connected as it has 100% packages received on pinging to VM2.

```
PS C:\Users\vm1> ping 10.1.0.4

Pinging 10.1.0.4 with 32 bytes of data:
Reply from 10.1.0.4: bytes=32 time=6ms TTL=128
Reply from 10.1.0.4: bytes=32 time=5ms TTL=128
Reply from 10.1.0.4: bytes=32 time=5ms TTL=128
Reply from 10.1.0.4: bytes=32 time=5ms TTL=128

Ping statistics for 10.1.0.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 5ms, Maximum = 6ms, Average = 5ms
PS C:\Users\vm1>
```