Azure Virtual Network (VNet), Subnets, CIDR Ranges, and VNet Peering

- Rajus Kandpal (Cloud Infra Security)

Table of Contents

1.Introduction

This R&D document delves into the fundamental ideas of Azure Virtual Network (VNet), Subnets, CIDR ranges, and VNet Peering on the Microsoft Azure cloud platform. Understanding these core networking services is critical for developing and implementing secure, scalable, and efficient cloud infrastructures.

2.Azure Virtual Network (VNet)

The Azure Virtual Network service serves as the foundation for your private network in Azure. An instance of the service (a virtual network) allows many different types of Azure services to securely communicate with one another, the internet, and on-premises networks. These Azure resources include virtual machines (VMs).

A virtual network is analogous to a physical network that you would run in your own datacenter. However, it provides additional benefits of the Azure architecture, such as scale, availability, and isolation.
An Azure Virtual Network (VNet) is a logically isolated segment of the Azure cloud that is specific to your subscription. It functions similarly to a traditional network in your own data center, but with the added scalability and flexibility of Azure infrastructure.

A VNet allows you to deploy and manage Azure resources such as Virtual Machines (VMs), App Services, and Azure SQL Databases within a secure and controlled network environment.

A VNet's primary features include private IP address allocation, subnetting, routing, network security groups (NSGs), service endpoints, and VNet peering.
The primary advantages of VNets include secure communication, resource isolation, traffic control, scalability, and the ability to interface with on-premises networks via VPN Gateway or ExpressRoute.

3. Understanding CIDR Ranges in Azure Networking

CIDR (Classless Inter-Domain Routing) is a more efficient technique of assigning IP addresses than standard class-based IP addressing.

When constructing a VNet in Azure, you must use CIDR notation to describe the IP address space.
The CIDR notation combines an IP address and a network prefix length. For example, the CIDR block "10.0.0.0/16" specifies that the first 16 bits represent the network part, with the remaining bits reserved for host addresses.

Planning your VNet IP space is crucial. Once defined, a VNet's CIDR range cannot be easily altered if resources are already present on the network.
For example, you can create a VNet with the address space 10.0.0.0/16, which allows for up to 65,536 IP addresses. This allows you to further divide the area into smaller subnets, such as 10.0.1.0/24 or 10.0.2.0/24.

If you intend to implement VNet Peering later, it is advised that you use non-overlapping IP ranges. Overlapping CIDR ranges between VNets preclude effective peering.

4. Subnets in Azure

After creating a VNet, divide its address space into one or more subnets.
Subnets are used to divide the VNet into smaller address ranges in order to organize and separate Azure resources.
For example, you can designate a "Web" subnet for front-end web servers and a "DB" subnet for backend databases.
Each subnet is assigned its own unique address range derived from the parent VNet CIDR block.
Subnets also enable you to implement fine-grained traffic controls via Network Security Groups (NSGs), which function as virtual firewalls at the subnet or NIC level.
Azure requires that each subnet within the VNet have a distinct, non-overlapping address range.
Subnet design is also important in deploying service endpoints, private endpoints, and user-defined routing (UDR).

5. VNet Peering in Azure

VNet Peering is a feature that connects two or more Azure Virtual Networks directly and discreetly via the Azure backbone network.

VNet Peering allows resources in both VNets to communicate with one another using private IP addresses, eliminating the need for VPN Gateways, public IPs, or Internet access.

This enables low-latency, high-bandwidth connection between VNets.

VNet Peering is useful in situations where you want to:

- Connect several environments (development, testing, and production).
- Expand network connection across multiple regions (using Global Peering).
- Enable shared services, such as DNS or monitoring systems, across VNets.

It is crucial to note that, by default, VNet Peering does not enable transitive routing. This means that if VNet1 peers with VNet2 and VNet2 peers with VNet3, VNet1 will not automatically interact with VNet3.

6. Types of VNet Peering

There are two major types of VNet Peering in Azure:

Intra-Region VNet Peering enables communication between VNets inside a single Azure region. This is commonly used when multiple VNets in the same region need to communicate securely.

Global VNet Peering connects VNets across Azure regions via the private backbone network. This is useful for worldwide enterprises that require cross-regional connectivity between VNets.

While both modes provide private communication, global peering may have slightly greater latencies due to geographical distance.

7. Use Cases of VNet Peering

Several real-world conditions make VNet Peering necessary:

In large businesses, various departments may have their own VNets for isolation. Peering enables them to securely share data as needed.

To ensure security and compliance, development and production environments can be deployed in distinct VNets. VNet Peering enables regulated communication between them when necessary.

Global VNet Peering allows global enterprises with resources spread across multiple Azure regions to do cross-region replication, backup, and failover.

Another use case is to use a hub-and-spoke network structure, in which a central hub VNet links to several spoke VNets to centralize shared functions such as logging, monitoring, and DNS.

8. Prerequisites and Limitations of VNet Peering

Before implementing VNet Peering, certain conditions must be met:

First, the VNets involved must have unique, non-overlapping IP address ranges.

Second, you need appropriate Azure RBAC permissions, such as Contributor or Network Administrator roles on both VNets.

Third, any Network Security Groups (NSGs) or User Defined Routes (UDRs) attached to subnets should allow traffic from the peered VNet, or else traffic may get blocked.

Additionally, note that VNet Peering does not support transitive routing unless configured using third-party network virtual appliances.

Another limitation is that VNet Peering does not support resource-level filtering. Once peered, all resources can potentially communicate unless further restricted by NSGs or routing rules.

9. Step-by-Step Implementation Guide (Azure Portal)

Here is a simple step-by-step guide to implement VNet Peering between two VNets in Azure:

Step 1: Create Two VNets
Go to the Azure Portal and create two separate Virtual Networks.

For example:

- VNet1 with an address space like 10.0.0.0/16
- VNet2 with an address space like 10.1.0.0/16

Step 2: Create Subnets in Each VNet
Within each VNet, create at least one subnet to host virtual machines or other resources.

Step 3: Deploy Virtual Machines
Launch a virtual machine in each subnet. For example, create a Linux VM in VNet1 and a Windows VM in VNet2.

Step 4: Configure Peering from VNet1 to VNet2
Navigate to VNet1 in the Azure Portal.
Go to the Peerings section and click "Add."
Provide a peering name and select VNet2 from the directory and subscription.
Enable "Allow Virtual Network Access" if you want resources in both VNets to communicate.

Step 5: Configure Peering from VNet2 to VNet1
Repeat the same process from the VNet2 side, pointing the peering to VNet1.

Step 6: Verify Peering Status
After configuration, both peering connections should show a status of "Connected."

Step 7: Test Connectivity
Log into the virtual machines and try to ping each other using private IP addresses.
You can use SSH to connect to the Linux VM and Remote Desktop (RDP) for the Windows VM.

Step 8: Adjust NSGs and Firewalls (if needed)
If the ping or connection fails, check and update Network Security Groups and VM firewalls to allow ICMP (for ping) or specific port traffic.

10. Best Practices and Considerations

When planning VNets and Peering, always allocate IP address ranges carefully to avoid overlaps.

Use tagging and proper naming conventions to manage multiple peered VNets in larger deployments.

Document peering relationships for future maintenance.

Consider implementing Network Security Groups (NSGs) and Route Tables (UDRs) for additional control and isolation.

For production workloads, monitor peered network traffic for performance and security compliance.

In Global Peering scenarios, keep an eye on data transfer costs as inter-region traffic is billed.

If planning to implement transitive routing between VNets, deploy a virtual network appliance or configure Azure Route Server.