A R&D report
on

Azure Virtual Networks, CIDR Ranges, Subnets, and VNet Peering

Submitted to **Celebal Technologies** in partial fulfillment of the internship task

Week 4

in

Cloud Infra and Security

By

Mr. Ganesh Ghadge

StudentId: CT_CSI_CI_6124

Objective

This document provides a comprehensive understanding of networking in Microsoft Azure, focusing on core components such as Virtual Networks (VNets), CIDR ranges, subnets, and VNet Peering. Using a real-world scenario, it demonstrates how to create a secure and connected network topology across multiple VNets.
The objectives are:

The document's objectives are as follows:

- Understand the role and structure of Azure Virtual Networks.
- Learn how to design and assign CIDR blocks to VNets and subnets.
- Deploy and configure Windows and Linux Virtual Machines (VMs) in different subnets and ensure mutual communication.
- Establish secure and performant connections between two VNets using VNet Peering.
- Explore use cases where network segmentation and VNet peering are crucial.
- Provide a portal-based walkthrough (with screenshots) for deploying resources and verifying connectivity.

Introduction

Microsoft Azure Virtual Network (VNet) is the cornerstone of Azure's networking architecture. It allows organizations to build isolated, logically segmented cloud networks that resemble on-premises data center networks, but with greater flexibility, scalability, and global reach.

Azure VNets support custom IP addressing, subnetting, routing, and connectivity features that make them suitable for complex workloads. Whether you're running a single VM or a multi-tier application spanning multiple regions, Azure Virtual Network provides the essential networking backbone.

This document focuses on how to design and deploy CIDR-based address spaces, configure subnets, and implement VNet peering to establish secure communication between isolated networks. It also explores how to provision Windows and Linux Virtual Machines (VMs) in different subnets and enable communication between them.

The networking scenario in this document is a common one in real-world enterprise deployments:

- Deploy a Windows VM and a Linux VM in different subnets within the same VNet.
- Configure both machines to communicate (ping) with each other securely.
- Create a second VNet and connect it to the first using VNet Peering, enabling cross-VNet traffic.

The goal is to provide a practical understanding of Azure's networking capabilities through guided deployment, portal configuration, and hands-on examples.

Azure Virtual Networks (VNets)

An Azure Virtual Network (VNet) is a core component of Azure's infrastructure services. It acts as a logically isolated network within the Azure cloud that enables secure communication between Azure resources such as Virtual Machines, web apps, databases, containers, and on-premises environments.

Key Features

- Custom IP addressing using CIDR (Classless Inter-Domain Routing) blocks
- Subnetting to organize and control traffic between resource groups
- Traffic filtering through Network Security Groups (NSGs)
- Routing control, both within Azure and to the internet or on-premises networks
- DNS configuration, using Azure DNS or custom DNS servers

VNet Scope and Limits

- A VNet is regional, meaning it spans all availability zones within that region.
- Resources in a VNet can communicate with each other across subnets unless access is explicitly restricted.
- Azure VNets cannot span regions, but cross-region connectivity can be achieved using VNet Peering (Global).

Basic VNet Configuration

Property	Value
Name	VNet-Primary
Region	East US
Address Space	10.0.0.0/16
Subnets	Subnet-Windows → 10.0.1.0/24Subnet-Linux → 10.0.2.0/24

Azure Portal Walkthrough – Create a VNet

1. Go to Azure Portal > Virtual Networks > + Create.

2. On the Basics tab:
    a. Choose Subscription and Resource Group (or create one).
    b. Name: VNet-Primary
    c. Region: e.g., East US
3. On the IP Addresses tab:
    a. Set the Address Space to 10.0.0.0/16
4. Add two subnets:
    a. Subnet-Windows: 10.0.1.0/24
    b. Subnet-Linux: 10.0.2.0/24
5. Keep other options at default for now (Security, Tags).
6. Review and click Create.

Why Use VNets?

- Isolation and Control: VNets keep resources private unless configured otherwise.
- Security: Fine-grained traffic control with NSGs, user-defined routes, and service endpoints.
- Scalability: VNets support thousands of resources and large IP address spaces.
- Integration: VNets can connect to each other, the internet, or on-prem environments.

CIDR Ranges and Subnets

CIDR (Classless Inter-Domain Routing) is a method for allocating IP addresses and routing IP packets efficiently. In Azure, all VNets and subnets use CIDR notation to define their IP ranges.
CIDR notation uses a **prefix (network portion)** and a **mask length**.
For example: 10.0.0.0/16

- 10.0.0.0 is the network address.
- /16 indicates the number of bits reserved for the network (the rest are for host addresses).

A **/16** network gives you **65,536 IP addresses** (2^{16}). This range can be divided into multiple **/24** subnets (each with 256 addresses).

Subnetting in Azure

Once you define a VNet's address space, you can create subnets within that space to:

- Segment traffic
- Apply separate NSG policies
- Host different services (web tier, app tier, DB tier)
- Isolate workloads (Linux in one, Windows in another, etc.)

Examples subnet plan

Subnet Name	CIDR Block	Purpose
Subnet-Windows	10.0.1.0/24	Host Windows VM
Subnet-Linux	10.0.2.0/24	Host Linux VM

These subnets are **non-overlapping** and fully contained within the parent VNet (10.0.0.0/16).

CIDR Size Considerations

CIDR Range	IPs Available	Use Case
/24	256	Typical for a single subnet
/20	4096	For larger workloads
/16	65,536	Very large VNets with many subnets

Azure reserves **5 IPs in every subnet** for internal usage (e.g., gateway address, DHCP, etc.), so always plan for **slightly fewer usable addresses**.

Azure Portal – Subnet Configuration Steps

1. During VNet creation (or after), go to the IP Addresses tab.
2. Click + Add Subnet.
3. Name your subnets:
    a. Name: Subnet-Windows
    b. Address range: 10.0.1.0/24
    c. Repeat for Subnet-Linux with 10.0.2.0/24
4. Leave NSG and route table blank for now (we'll configure them later).
5. Click Add, then Review + Create.

Best Practices

- Avoid overlapping address spaces across VNets if you plan to use peering.
- Always leave room for subnet expansion — don't assign the full VNet space upfront.
- Use naming conventions like subnet-app and subnet-db for clarity.
- Consider /28 or /29 subnets for small services like Bastion or Gateway.

Azure Virtual Network Peering

VNet Peering connects two Azure Virtual Networks to allow traffic between them using private IP addresses, effectively extending a single network across two VNets.
Peering provides:

- Low latency, high bandwidth connectivity: Since traffic remains on Azure's backbone network.
- Resource communication: Enables VMs and services in different VNets to communicate seamlessly.
- No gateways required: Simplifies network design and reduces costs.

There are two main types of VNet Peering:

- Regional Peering: Between VNets in the same Azure region.
- Global Peering: Between VNets in different Azure regions.

Peering can also differ based on:

- Subscription context: Peering within the same subscription or across subscriptions.
- Traffic flow: One-way or two-way peering (most common is two-way).

Implementation

Step 1: Create the First Virtual Network (VNet) and Subnets

1. Navigate to the Azure Portal
    a. Log in to the Azure Portal.
    b. On the left-hand menu, click “Create a resource”.
2. Search for “Virtual Network” and Select It
    a. In the search bar, type “Virtual Network”.
    b. Select Virtual Network from the results and click Create.
3. Configure Basic Settings
    a. Subscription: Choose the appropriate subscription.
    b. Resource Group: Select an existing resource group or create a new one (e.g., RG-VNetDemo).
    c. Name: Provide a name for the VNet, e.g., VNet-Demo1.
    d. Region: Select your preferred Azure region (e.g., East US).
4. Define the Address Space
    a. In IP Addresses, specify the CIDR block for the VNet, e.g., 10.0.0.0/16.
5. Create Subnets
    a. Click + Add subnet.
    b. Enter Subnet name: e.g., Subnet-WinVM.
    c. Enter Subnet address range: e.g., 10.0.1.0/24.
    d. Click Add.
    e. Add another subnet, e.g.,
        i. Name: Subnet-LinuxVM
        ii. Address range: 10.0.2.0/24.
6. Review and Create
7. Review your configuration and click Create.

Step 2: Deploy Virtual Machines in Subnets

Deploy a Windows VM in Subnet-WinVM

1. Navigate to the Azure Portal
    a. Click on Create a resource > Search for Windows Server 2022 Datacenter (or your preferred Windows Server image).
    b. Click Create.
2. Configure Basics
    a. Subscription: Select the appropriate subscription.
    b. Resource Group: Use the same as VNet (RG-VNetDemo).

    c. Virtual machine name: e.g., WinVM.
    d. Region: Same as VNet (e.g., East US).
    e. Image: Select Windows Server 2022 Datacenter.
    f. Size: Choose an appropriate size (e.g., Standard_DS1_v2).
    g. Authentication type: Password or SSH public key.
    h. Username and password: Set credentials.
3. Configure Networking
    a. Under Networking, select:
        i. Virtual Network: VNet-Demo1
        ii. Subnet: Subnet-WinVM
4. Public IP: Select Create new or None (if you want only a private IP).
5. Network security group: Use default or create new (ensure ICMP allowed for ping).
6. Review + Create
    a. Verify the settings and click Create.

Deploy a Windows VM in Subnet-WinVM

1. Create a Linux VM
2. Create a resource > Search for Ubuntu Server 22.04 LTS (or preferred distro).
3. Click Create.
4. Configure Basics
    a. Use the same subscription and resource group.
    b. VM name: e.g., LinuxVM.
    c. Region: Same as VNet.
    d. Size: e.g., Standard_DS1_v2.
    e. Authentication: SSH public key or password.
    f. Set the username and authentication method.
5. Networking
    a. Virtual Network: VNet-Demo1
    b. Subnet: Subnet-LinuxVM
    c. Public IP: Create new or none.
    d. Network security group: Ensure rules allow SSH and ICMP (for ping).
6. Review + Create
    a. Validate and deploy.

Step 3: Validate Intra-VNet Connectivity (Ping Test)

1. After both VMs are running, connect to each VM:
    a. Windows VM: Use RDP.
    b. Linux VM: Use SSH.

2. From Windows VM, open PowerShell or Command Prompt, and ping the Linux VM's private IP.
3. From the Linux VM, use the ping command to ping the Windows VM's private IP.

Troubleshooting and Best Practices

Common Issues and Troubleshooting Tips

- Check Network Security Groups (NSGs):
    - Ensure that inbound and outbound rules allow ICMP (ping) traffic. By default, ICMP is blocked, so you need to add custom NSG rules permitting it between subnets or VNets.
- Verify Subnet Address Ranges:
    - Confirm there is no overlap between subnet CIDR blocks across VNets, especially before peering.
- Peering Status:
    - Confirm VNet peering status is Connected and that peering is configured both ways for full communication.
- Routing and Gateways:
    - For global peering, verify that forwarding traffic and gateway transit settings are correct if using VPN or ExpressRoute gateways.

Best Practices for VNet, Subnets, and Peering

- Plan IP Addressing Carefully
    - Use non-overlapping CIDR blocks across VNets and subnets.
    - Allocate subnet sizes based on expected workload growth.
- Secure Networks with NSGs
    - Apply NSGs at the subnet or VM NIC levels.
    - Create explicit rules to restrict or allow traffic, especially for cross-subnet or cross-VNet communication.
- Use VNet Peering Wisely
    - Use regional peering when VNets are in the same region for lower latency.
    - Use global peering for multi-region architectures.
    - Avoid transitive peering (A peered with B, B peered with C, but A cannot talk to C).
- Monitor and Audit Network Traffic
    - Use Azure Network Watcher to monitor network traffic and diagnose connectivity issues.
    - Enable NSG flow logs for auditing traffic patterns.

References

1. Microsoft Azure Global Infrastructure
   https://azure.microsoft.com/en-us/explore/global-infrastructure
2. Azure Virtual Network Documentation
   https://learn.microsoft.com/en-us/azure/virtual-network/