**Azure Virtual Network CIDR Ranges, Subnets, and VNet Peering**

**Introduction**

Microsoft Azure Virtual Networks (VNets) allow you to create isolated network environments in the cloud, enabling secure communication between resources. A key aspect of VNets is the use of Classless Inter-Domain Routing (CIDR) notation to define IP address ranges for VNets and their subnets. VNet Peering further enhances connectivity by linking VNets, allowing resources to communicate as if they were in the same network. This document provides a comprehensive guide to understanding CIDR ranges, subnets, and VNet Peering, along with a step-by-step implementation of a use case involving creating VNets, subnets, virtual machines (VMs), and establishing VNet Peering.

**Key Points**

* **CIDR Ranges**: VNets use private IP ranges (e.g., 10.0.0.0/16) from RFC 1918 or RFC 6598, with subnets carved out as smaller ranges (e.g., 10.0.0.0/24).
* **Subnets**: Each subnet reserves five IP addresses for Azure services, reducing available IPs (e.g., 251 in a /24 subnet).
* **VNet Peering Types**: Regional (same Azure region) and Global (different regions) peering enable private, low-latency communication.
* **Use Case**: Create a VNet with two subnets, deploy a Windows and Linux VM, ensure they can ping each other, and connect two VNets via peering.
* **Prerequisites**: An active Azure subscription and sufficient permissions are required.

**Understanding CIDR Ranges**

CIDR notation defines IP address ranges for VNets and subnets. For example, a VNet with 10.0.0.0/16 provides 65,536 IP addresses, from which subnets like 10.0.0.0/24 (256 IPs, 251 usable) can be created. Azure reserves specific ranges (e.g., 224.0.0.0/4 for multicast) that cannot be used.

**VNet Peering Overview**

VNet Peering connects two VNets, allowing seamless resource communication without public internet or gateways. Regional peering is ideal for low-latency scenarios within the same region, while Global peering supports cross-region connectivity, though with limitations like restricted load balancer support.

**Use Case Implementation**

The use case involves creating a VNet with two subnets, deploying a Windows VM and a Linux VM, verifying they can ping each other, creating a second VNet, and establishing VNet Peering. Below are the detailed steps using the Azure portal.

**Detailed Implementation Guide**

**Prerequisites**

* **Azure Account**: An active subscription is required (Azure Free Account).
* **Permissions**: Contributor or higher role to create and manage VNets, subnets, VMs, and peering.
* **Azure Portal Access**: Familiarity with navigating portal.azure.com.

**Section 1: CIDR Ranges for VNets and Subnets**

**1.1 CIDR Notation**

CIDR specifies an IP address range using a base IP and a subnet mask (e.g., 10.0.0.0/16). The number after the slash indicates the number of bits for the network portion, determining the range of available IPs.

**1.2 Recommended CIDR Ranges**

Azure recommends private IP ranges from:

* **RFC 1918**:
  + 10.0.0.0/8 (10.0.0.0–10.255.255.255)
  + 172.16.0.0/12 (172.16.0.0–172.31.255.255)
  + 192.168.0.0/16 (192.168.0.0–192.168.255.255)
* **RFC 6598**: 100.64.0.0/10 (100.64.0.0–100.127.255.255)
* **Prohibited Ranges**:
  + 224.0.0.0/4 (Multicast)
  + 255.255.255.255/32 (Broadcast)
  + 127.0.0.0/8 (Loopback)
  + 169.254.0.0/16 (Link-local)
  + 168.63.129.16/32 (Internal DNS, DHCP, Azure Load Balancer)

**1.3 Subnets and Reserved IPs**

Subnets are subsets of a VNet’s address space. For example:

* VNet: 10.0.0.0/16
* Subnet1: 10.0.0.0/24 (256 IPs, 251 usable)
* Subnet2: 10.0.1.0/24 (256 IPs, 251 usable) Azure reserves five IPs per subnet:
* First IP (e.g., 10.0.0.0): Network address
* Second IP (e.g., 10.0.0.1): Default gateway
* Third and fourth IPs (e.g., 10.0.0.2, 10.0.0.3): Azure DNS
* Last IP (e.g., 10.0.0.255): Broadcast

|  |  |  |  |
| --- | --- | --- | --- |
| **Subnet CIDR** | **Total IPs** | **Reserved IPs** | **Usable IPs** |
| 10.0.0.0/24 | 256 | 5 | 251 |
| 10.0.1.0/24 | 256 | 5 | 251 |

**1.4 Best Practices**

* Avoid overlapping CIDR ranges with other VNets or on-premises networks.
* Reserve address space for future subnets.
* Prefer fewer, larger VNets for simpler management.

**Section 2: Types of VNet Peering**

VNet Peering enables private communication between VNets. The two main types are:

**2.1 Regional VNet Peering**

* Connects VNets in the same Azure region.
* Offers low-latency, high-bandwidth connectivity.
* Ideal for applications requiring frequent resource interaction.

**2.2 Global VNet Peering**

* Connects VNets across different Azure regions.
* Supports global architectures, such as disaster recovery.
* Limitations: Cannot use basic load balancers across globally peered VNets.

**2.3 Cross-Subscription Peering**

* VNets in different subscriptions can be peered if within the same Azure AD tenant.
* Useful for organizations managing multiple subscriptions.

**Section 3: Step-by-Step Guide for Use Case**

**3.1 Create a VNet with Subnets**

1. **Sign into Azure Portal**:
   * Navigate to portal.azure.com and log in.
2. **Create a Resource Group**:
   * Click “Create a resource” in the top-left corner.
   * Search for “Resource group” and select it.
   * Enter:
     + Name: myResourceGroup
     + Region: East US
   * Click “Review + create”, then “Create”.
3. **Create First VNet (myVNet)**:
   * Click “Create a resource” and search for “Virtual network”.
   * In the “Create virtual network” blade:
     + **Project details**:
       - Subscription: Select your subscription.
       - Resource group: Select myResourceGroup.
     + **Instance details**:
       - Name: myVNet
       - Region: East US
     + **IP addresses**:
       - IPv4 address space: 10.0.0.0/16
       - Subnet:
         * Name: Subnet1
         * Address range: 10.0.0.0/24
   * Click “Review + create”, then “Create”.
4. **Add Second Subnet (Subnet2)**:
   * Navigate to myVNet in the portal.
   * Click “Subnets” under Settings.
   * Click “+ Subnet”.
   * Enter:
     + Name: Subnet2
     + Address range: 10.0.1.0/24
   * Click “OK”.

**3.2 Launch VMs in Subnets**

1. **Create Windows VM in Subnet1**:
   * Click “Create a resource” and search for “Windows Server 2022 Datacenter”.
   * In the “Create a virtual machine” blade:
     + **Basics**:
       - VM name: myWindowsVM
       - Resource group: myResourceGroup
       - Region: East US
     + **Networking**:
       - Virtual network: myVNet
       - Subnet: Subnet1
     + Complete other settings (e.g., size, credentials).
     + Click “Review + create”, then “Create”.
2. **Create Linux VM in Subnet2**:
   * Click “Create a resource” and search for “Ubuntu Server 20.04 LTS”.
   * In the “Create a virtual machine” blade:
     + **Basics**:
       - VM name: myLinuxVM
       - Resource group: myResourceGroup
       - Region: East US
     + **Networking**:
       - Virtual network: myVNet
       - Subnet: Subnet2
     + Complete other settings.
     + Click “Review + create”, then “Create”.
3. **Verify Connectivity**:
   * Connect to myWindowsVM via RDP and myLinuxVM via SSH.
   * From myWindowsVM, ping myLinuxVM’s private IP (e.g., 10.0.1.4).
   * From myLinuxVM, ping myWindowsVM’s private IP (e.g., 10.0.0.4).
   * VMs in the same VNet should communicate without additional configuration.

**3.3 Create Second VNet and Set Up Peering**

1. **Create Second VNet (myVNet2)**:
   * Repeat VNet creation steps:
     + Name: myVNet2
     + Address space: 10.1.0.0/16
     + Subnet:
       - Name: Subnet3
       - Address range: 10.1.0.0/24
2. **Set Up VNet Peering**:
   * Navigate to myVNet -> “Peerings” under Settings.
   * Click “+ Add”.
   * Enter:
     + Name: toMyVNet2
     + Remote virtual network: myVNet2
     + Leave defaults (e.g., allow forwarded traffic).
   * Click “OK”.
   * Navigate to myVNet2 -> “Peerings”.
   * Click “+ Add”.
   * Enter:
     + Name: toMyVNet
     + Remote virtual network: myVNet
   * Click “OK”.
3. **Verify Peering**:
   * Create a VM in Subnet3 (e.g., myLinuxVM2).
   * From myWindowsVM in myVNet, ping myLinuxVM2’s IP (e.g., 10.1.0.4).
   * Confirm bidirectional connectivity.

**Conclusion**

This guide covers CIDR ranges, subnets, and VNet Peering in Azure, with a practical use case demonstrating VNet creation, VM deployment, and peering setup. By following best practices, such as avoiding overlapping CIDR ranges and reserving address space, you can build scalable and secure network architectures.

**Key Citations**

* Azure Virtual Network FAQ
* Azure Virtual Network Concepts and Best Practices
* Azure Virtual Network Peering Overview
* Create, Change, or Delete an Azure Virtual Network
* Quickstart: Create an Azure Virtual Network