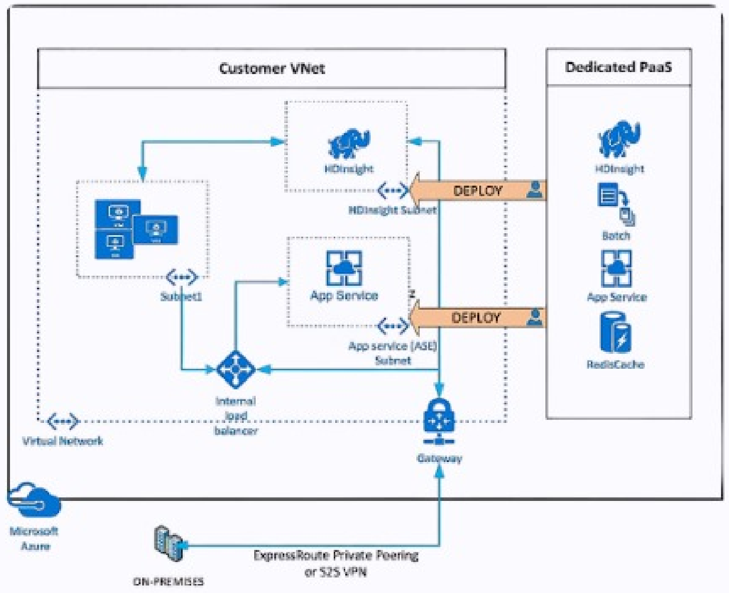
Virtual Network and Virtual Machine Overview

This document provides a comprehensive overview of Azure Virtual Networks (VNets) and Virtual Machines (VMs), covering key concepts, deployment scenarios, and hands-on assignments. It explores CIDR ranges, subnet configurations, different types of VNet peering, and the deployment of both Windows and Linux VMs. The guide also includes a section dedicated to Azure Virtual Network research and prerequisite gathering, ensuring a thorough understanding of the topic before diving into the practical assignments.

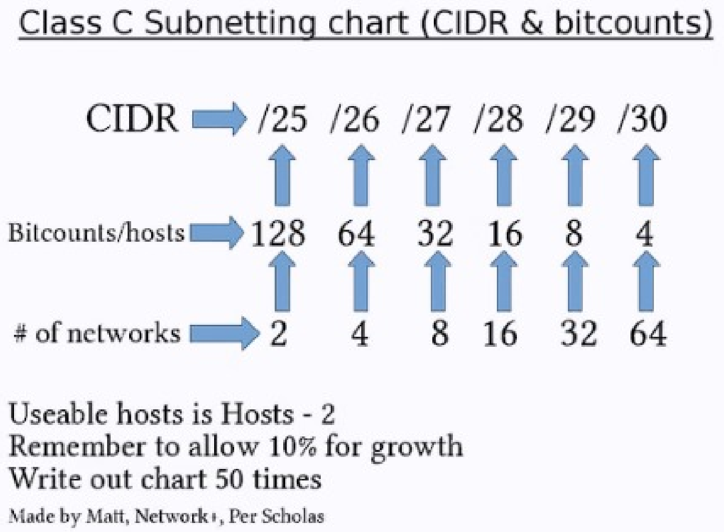
By Aniket Das



CIDR Range and Subnet Concepts

In the context of virtual networking, Classless Inter-Domain Routing (CIDR) is a method for efficiently allocating and managing IP address spaces. CIDR notation allows for the creation of subnets by dividing a larger network into smaller, more manageable segments. This is particularly important when designing Azure Virtual Networks, as it enables you to organize and control the IP address space within your virtual network.

Subnets, on the other hand, are logical subdivisions of a VNet, each with its own IP address range. Subnets help to segment the network, improve security, and optimize routing. By carefully planning the CIDR ranges and subnet configurations, you can effectively manage the IP address space and ensure efficient communication within and between your Azure resources.



Class C Subnetting chart (CIDR & bitcounts)

CIDR	/25	/26	/27	/28	/29	/30
Bitcounts/hosts	128	64	32	16	8	4
# of networks	2	4	8	16	32	64

Useable hosts is Hosts - 2
Remember to allow 10% for growth
Write out chart 50 times
Made by Matt, Network+, Per Scholas

Azure Virtual Network (VNet) and Peering Types

Azure Virtual Network (VNet) is a fundamental component of Azure's networking services. It provides a logically isolated and secure environment for your Azure resources to communicate with each other, as well as with on-premises resources and the internet. VNets can be configured with custom IP address spaces, subnets, route tables, and other advanced networking features.

VNet peering is a mechanism that allows two or more VNets to communicate with each other using private IP addresses. There are several types of VNet peering, including:

1 Virtual Network Peering

This type of peering allows communication between VMs and other resources within the same Azure region.

2 Global VNet Peering

This type of peering enables communication between VMs and other resources across different Azure regions.

3 Hub-and-Spoke Peering

This configuration involves a central hub VNet that is connected to multiple spoke VNets, allowing for centralized management and control.

Windows and Linux Virtual Machine Deployment

Azure Virtual Machines (VMs) are a versatile compute service that allows you to deploy and manage Windows or Linux-based virtual machines in the cloud. When deploying VMs, you can choose from a wide range of operating system images, VM sizes, and configurations to meet your specific workload requirements.

Windows Virtual Machines

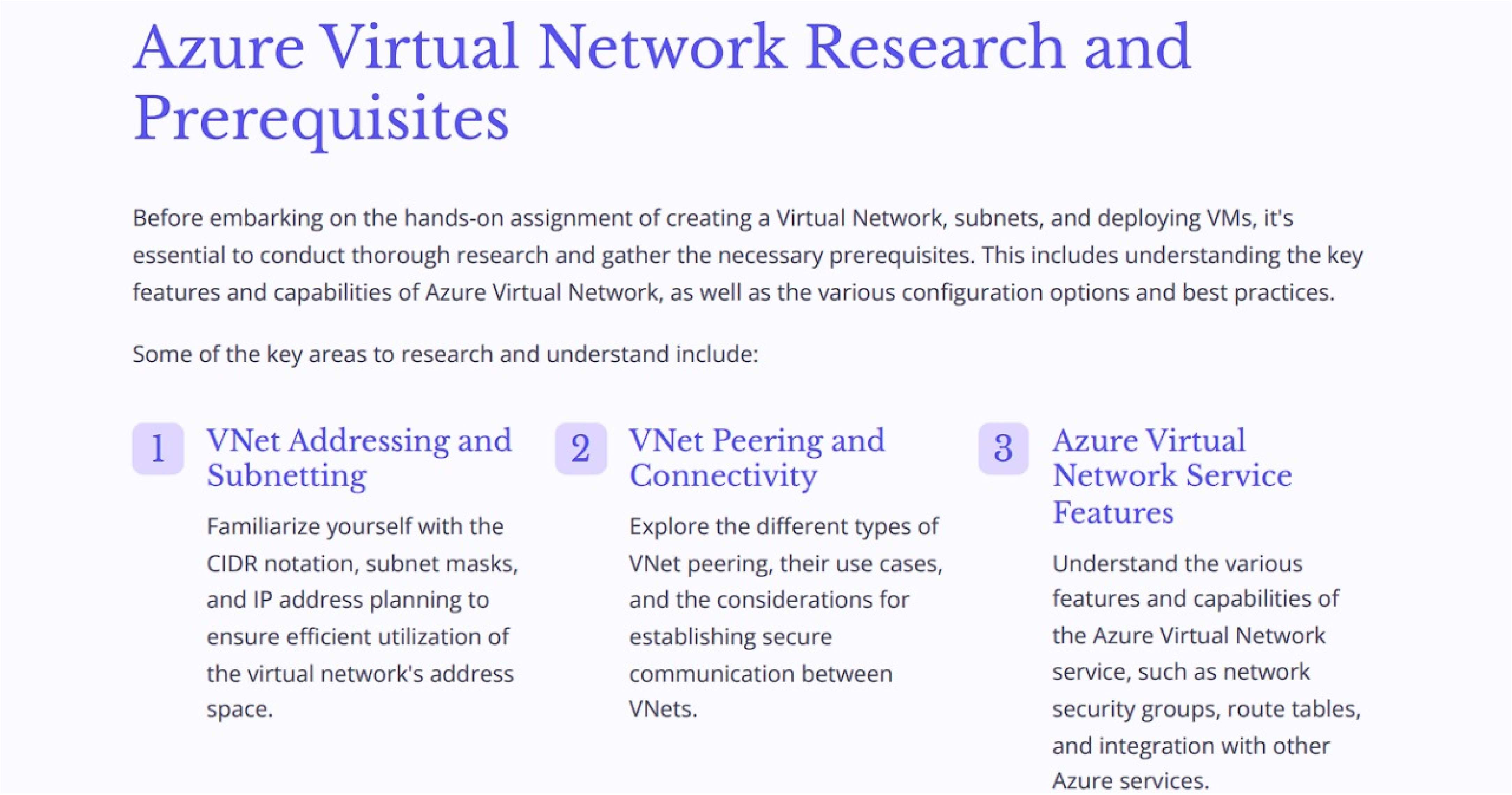
Windows VMs offer a familiar and robust platform for hosting a variety of Windows-based applications, services, and development environments. They provide seamless integration with other Microsoft technologies and tools, making them a popular choice for enterprises and organizations already leveraging the Microsoft ecosystem.

Linux Virtual Machines

Linux VMs offer a highly customizable and open-source platform for hosting a wide range of applications, from web servers to data analytics pipelines. They provide flexibility, scalability, and cost-effectiveness, making them a popular choice for developers, DevOps teams, and organizations with a diverse technology stack.

VM Deployment and Connectivity

Regardless of the operating system, Azure provides a streamlined process for deploying and managing VMs, including the ability to configure virtual networks, network security groups, and other networking settings to ensure secure connectivity between your VMs and other Azure resources.



Azure Virtual Network Research and Prerequisites

Before embarking on the hands-on assignment of creating a Virtual Network, subnets, and deploying VMs, it's essential to conduct thorough research and gather the necessary prerequisites. This includes understanding the key features and capabilities of Azure Virtual Network, as well as the various configuration options and best practices.

Some of the key areas to research and understand include:

1 VNet Addressing and Subnetting

Familiarize yourself with the CIDR notation, subnet masks, and IP address planning to ensure efficient utilization of the virtual network's address space.

2 VNet Peering and Connectivity

Explore the different types of VNet peering, their use cases, and the considerations for establishing secure communication between VNets.

3 Azure Virtual Network Service Features

Understand the various features and capabilities of the Azure Virtual Network service, such as network security groups, route tables, and integration with other Azure services.

Hands-on Assignment: VNet Creation, Subnets, VM Deployment, and VNet Peering

In this hands-on assignment, you will put your understanding of Azure Virtual Networks and Virtual Machines into practice. You will create a Virtual Network (VNet) with multiple subnets, deploy both Windows and Linux Virtual Machines (VMs) within those subnets, and establish a VNet peering connection between two VNets.

1 VNet Creation and Subnets

Begin by creating an Azure Virtual Network, defining the CIDR range, and configuring multiple subnets within the VNet. Ensure that the IP address space and subnet allocations are planned effectively to accommodate your deployment requirements.

2 Windows and Linux VM Deployment

Next, deploy a Windows VM and a Linux VM within different subnets of the VNet. Configure the necessary network settings to ensure that the VMs can communicate with each other using their private IP addresses.

3 VNet Peering

Finally, create a second VNet and establish a peering connection between the two VNets. This will allow resources (such as VMs) in one VNet to communicate with resources in the other VNet, using private IP addresses and without the need for additional gateways or network appliances.