**Step 1: Create a Resource Group**

First, create a resource group to organize your resources.

**Azure CLI:**

az group create --name HubSpokeResourceGroup --location eastus

### Step 2: Create the Hub Virtual Network

Create the central hub virtual network (VNet) which will host shared services like VPN gateway, firewall, etc.

**Azure CLI:**

az network vnet create \

--resource-group HubSpokeResourceGroup \

--name HubVNet \

--address-prefix 10.0.0.0/16 \

--subnet-name GatewaySubnet \

--subnet-prefix 10.0.0.0/24

### Step 3: Create the Spoke Virtual Networks

Create multiple spoke VNets that will connect to the hub VNet.

**Azure CLI:**

# Spoke VNet 1

az network vnet create \

--resource-group HubSpokeResourceGroup \

--name SpokeVNet1 \

--address-prefix 10.1.0.0/16 \

--subnet-name SpokeSubnet1 \

--subnet-prefix 10.1.0.0/24

# Spoke VNet 2

az network vnet create \

--resource-group HubSpokeResourceGroup \

--name SpokeVNet2 \

--address-prefix 10.2.0.0/16 \

--subnet-name SpokeSubnet2 \

--subnet-prefix 10.2.0.0/24

### Step 4: Create Virtual Network Gateways

If you need connectivity to on-premises networks or other VNets, create virtual network gateways in the hub VNet.

**Azure CLI:**

az network vnet-gateway create \

--resource-group HubSpokeResourceGroup \

--name HubVNetGateway \

--vnet HubVNet \

--public-ip-address HubVNetGatewayPublicIP \

--gateway-type Vpn \

--vpn-type RouteBased \

--sku VpnGw1 \

--no-wait

### Step 5: Peer the Hub and Spoke VNets

Create VNet peering connections between the hub VNet and each spoke VNet.

**Azure CLI:**

# Peer Hub VNet to Spoke VNet 1

az network vnet peering create \

--name HubToSpoke1 \

--resource-group HubSpokeResourceGroup \

--vnet-name HubVNet \

--remote-vnet SpokeVNet1 \

--allow-vnet-access

az network vnet peering create \

--name Spoke1ToHub \

--resource-group HubSpokeResourceGroup \

--vnet-name SpokeVNet1 \

--remote-vnet HubVNet \

--allow-vnet-access

# Peer Hub VNet to Spoke VNet 2

az network vnet peering create \

--name HubToSpoke2 \

--resource-group HubSpokeResourceGroup \

--vnet-name HubVNet \

--remote-vnet SpokeVNet2 \

--allow-vnet-access

az network vnet peering create \

--name Spoke2ToHub \

--resource-group HubSpokeResourceGroup \

--vnet-name SpokeVNet2 \

--remote-vnet HubVNet \

--allow-vnet-access

### Step 6: Route Traffic Through the Hub

Configure routes to ensure traffic from spoke VNets can route through the hub VNet, often for centralized security or inspection.

**Azure CLI:**

# Create a route table

az network route-table create \

--resource-group HubSpokeResourceGroup \

--name HubRouteTable

# Add a route to the route table

az network route-table route create \

--resource-group HubSpokeResourceGroup \

--route-table-name HubRouteTable \

--name RouteToSpokeVNets \

--address-prefix 10.0.0.0/8 \

--next-hop-type VirtualNetworkGateway

# Associate the route table with the subnets

az network vnet subnet update \

--vnet-name SpokeVNet1 \

--name SpokeSubnet1 \

--resource-group HubSpokeResourceGroup \

--route-table HubRouteTable

az network vnet subnet update \

--vnet-name SpokeVNet2 \

--name SpokeSubnet2 \

--resource-group HubSpokeResourceGroup \

--route-table HubRouteTable

### Step 7: Test Connectivity

1. **Deploy VMs in Hub and Spoke VNets:** Deploy virtual machines in the hub and each spoke VNet to test connectivity.
2. **Ping from Hub VM to Spoke VMs:** Ensure you can ping and access resources between the hub and spoke VNets.

**Azure CLI Example to Create VMs:**

# Create a VM in Hub VNet

az vm create \

--resource-group HubSpokeResourceGroup \

--name HubVM \

--vnet-name HubVNet \

--subnet GatewaySubnet \

--image UbuntuLTS \

--admin-username azureuser \

--generate-ssh-keys

# Create a VM in Spoke VNet 1

az vm create \

--resource-group HubSpokeResourceGroup \

--name SpokeVM1 \

--vnet-name SpokeVNet1 \

--subnet SpokeSubnet1 \

--image UbuntuLTS \

--admin-username azureuser \

--generate-ssh-keys

# Create a VM in Spoke VNet 2

az vm create \

--resource-group HubSpokeResourceGroup \

--name SpokeVM2 \

--vnet-name SpokeVNet2 \

--subnet SpokeSubnet2 \

--image UbuntuLTS \

--admin-username azureuser \

--generate-ssh-keys

### Step 8: Create Internal and External Load Balancers

For high availability and load distribution, create internal and external load balancers.

**Azure CLI:**

# Create Public IP for the External Load Balancer

az network public-ip create \

--resource-group HubSpokeResourceGroup \

--name MyPublicIP \

--sku Standard

# Create External Load Balancer

az network lb create \

--resource-group HubSpokeResourceGroup \

--name MyExternalLB \

--frontend-ip-name MyFrontendIP \

--backend-pool-name MyBackendPool \

--public-ip-address MyPublicIP

# Create Internal Load Balancer

az network lb create \

--resource-group HubSpokeResourceGroup \

--name MyInternalLB \

--frontend-ip-name MyInternalFrontendIP \

--backend-pool-name MyInternalBackendPool \

--vnet-name HubVNet \

--subnet GatewaySubnet \

--private-ip-address 10.0.0.10