**Week – 4**

**Task:**

1. **Create a Vnet, 2 Subnets**

**Subnet-1: Linux VM, WindowsVM**

**Subnet-2: SQL DB, B. Create 4 VNets**

**1. Management Vnet (HUB)**

**2. Production Vnet**

**3. Testing Vnet**

**4. Developing Vnet And Configure Hub and Spoke Architecture and verify it's working by launching VM in each VNet and ping from Managemnent VM to every other VM**

**Create a Internal and External Load Balancer**

**Create and test Azure Application gateway**

**Part A: Create a VNet and Subnets**

1. **Create a VNet with 2 Subnets**:

**az group create --name MyResourceGroup --location eastus**

**az network vnet create \**

**--resource-group MyResourceGroup \**

**--name MyVNet \**

**--address-prefix 10.0.0.0/16 \**

**--subnet-name Subnet1 \**

**--subnet-prefix 10.0.1.0/24**

**az network vnet subnet create \**

**--resource-group MyResourceGroup \**

**--vnet-name MyVNet \**

**--name Subnet2 \**

**--address-prefix 10.0.2.0/24**

1. **Create VMs in Subnet1**:

**# Create a Linux VM**

**az vm create \**

**--resource-group MyResourceGroup \**

**--name MyLinuxVM \**

**--vnet-name MyVNet \**

**--subnet Subnet1 \**

**--image UbuntuLTS \**

**--admin-username azureuser \**

**--generate-ssh-keys**

**# Create a Windows VM**

**az vm create \**

**--resource-group MyResourceGroup \**

**--name MyWindowsVM \**

**--vnet-name MyVNet \**

**--subnet Subnet1 \**

**--image Win2019Datacenter \**

**--admin-username azureuser \**

**--admin-password MyP@ssw0rd123**

**Create SQL Database in Subnet2**:

* SQL databases can't be directly created in a subnet, so typically a SQL Server VM or Azure SQL Database with VNet integration would be used.

### Part B: Create 4 VNets and Configure Hub and Spoke Architecture

1. **Create VNets**:

**# Management VNet (Hub)**

**az network vnet create \**

**--resource-group MyResourceGroup \**

**--name ManagementVNet \**

**--address-prefix 10.1.0.0/16 \**

**--subnet-name ManagementSubnet \**

**--subnet-prefix 10.1.1.0/24**

**# Production VNet**

**az network vnet create \**

**--resource-group MyResourceGroup \**

**--name ProductionVNet \**

**--address-prefix 10.2.0.0/16 \**

**--subnet-name ProductionSubnet \**

**--subnet-prefix 10.2.1.0/24**

**# Testing VNet**

**az network vnet create \**

**--resource-group MyResourceGroup \**

**--name TestingVNet \**

**--address-prefix 10.3.0.0/16 \**

**--subnet-name TestingSubnet \**

**--subnet-prefix 10.3.1.0/24**

**# Developing VNet**

**az network vnet create \**

**--resource-group MyResourceGroup \**

**--name DevelopingVNet \**

**--address-prefix 10.4.0.0/16 \**

**--subnet-name DevelopingSubnet \**

**--subnet-prefix 10.4.1.0/24**

**Configure Peering (Hub and Spoke Architecture)**:

**# Peer ProductionVNet with ManagementVNet**

**az network vnet peering create \**

**--resource-group MyResourceGroup \**

**--name ProdToHub \**

**--vnet-name ProductionVNet \**

**--remote-vnet ManagementVNet \**

**--allow-vnet-access**

**az network vnet peering create \**

**--resource-group MyResourceGroup \**

**--name HubToProd \**

**--vnet-name ManagementVNet \**

**--remote-vnet ProductionVNet \**

**--allow-vnet-access**

**# Peer TestingVNet with ManagementVNet**

**az network vnet peering create \**

**--resource-group MyResourceGroup \**

**--name TestToHub \**

**--vnet-name TestingVNet \**

**--remote-vnet ManagementVNet \**

**--allow-vnet-access**

**az network vnet peering create \**

**--resource-group MyResourceGroup \**

**--name HubToTest \**

**--vnet-name ManagementVNet \**

**--remote-vnet TestingVNet \**

**--allow-vnet-access**

**# Peer DevelopingVNet with ManagementVNet**

**az network vnet peering create \**

**--resource-group MyResourceGroup \**

**--name DevToHub \**

**--vnet-name DevelopingVNet \**

**--remote-vnet ManagementVNet \**

**--allow-vnet-access**

**az network vnet peering create \**

**--resource-group MyResourceGroup \**

**--name HubToDev \**

**--vnet-name ManagementVNet \**

**--remote-vnet DevelopingVNet \**

**--allow-vnet-access**

**Create VMs in each VNet**:

**# VM in ProductionVNet**

**az vm create \**

**--resource-group MyResourceGroup \**

**--name ProdVM \**

**--vnet-name ProductionVNet \**

**--subnet ProductionSubnet \**

**--image UbuntuLTS \**

**--admin-username azureuser \**

**--generate-ssh-keys**

**# VM in TestingVNet**

**az vm create \**

**--resource-group MyResourceGroup \**

**--name TestVM \**

**--vnet-name TestingVNet \**

**--subnet TestingSubnet \**

**--image UbuntuLTS \**

**--admin-username azureuser \**

**--generate-ssh-keys**

**# VM in DevelopingVNet**

**az vm create \**

**--resource-group MyResourceGroup \**

**--name DevVM \**

**--vnet-name DevelopingVNet \**

**--subnet DevelopingSubnet \**

**--image UbuntuLTS \**

**--admin-username azureuser \**

**--generate-ssh-keys**

**# VM in ManagementVNet**

**az vm create \**

**--resource-group MyResourceGroup \**

**--name ManagementVM \**

**--vnet-name ManagementVNet \**

**--subnet ManagementSubnet \**

**--image UbuntuLTS \**

**--admin-username azureuser \**

**--generate-ssh-keys**

**Verify Connectivity**:

* SSH into the Management VM:

**ssh azureuser@<ManagementVM\_Public\_IP>**

Ping the VMs in other VNets:

**ping <ProdVM\_Private\_IP>**

**ping <TestVM\_Private\_IP>**

**ping <DevVM\_Private\_IP>**

### Part C: Create Internal and External Load Balancer

1. **Create an External Load Balancer**:

**az network public-ip create \**

**--resource-group MyResourceGroup \**

**--name MyPublicIP \**

**--allocation-method Static**

**az network lb create \**

**--resource-group MyResourceGroup \**

**--name MyExternalLB \**

**--public-ip-address MyPublicIP \**

**--frontend-ip-name MyFrontEnd \**

**--backend-pool-name MyBackEndPool**

**az network lb probe create \**

**--resource-group MyResourceGroup \**

**--lb-name MyExternalLB \**

**--name MyHealthProbe \**

**--protocol tcp \**

**--port 80**

**az network lb rule create \**

**--resource-group MyResourceGroup \**

**--lb-name MyExternalLB \**

**--name MyLoadBalancerRule \**

**--protocol tcp \**

**--frontend-port 80 \**

**--backend-port 80 \**

**--frontend-ip-name MyFrontEnd \**

**--backend-pool-name MyBackEndPool \**

**--probe-name MyHealthProbe**

Associate VMs to the Load Balancer Backend Pool.

**Create an Internal Load Balancer**:

**az network lb create \**

**--resource-group MyResourceGroup \**

**--name MyInternalLB \**

**--vnet-name ManagementVNet \**

**--subnet ManagementSubnet \**

**--private-ip-address 10.1.1.100 \**

**--frontend-ip-name MyFrontEndInternal \**

**--backend-pool-name MyBackEndPoolInternal**

**az network lb probe create \**

**--resource-group MyResourceGroup \**

**--lb-name MyInternalLB \**

**--name MyHealthProbeInternal \**

**--protocol tcp \**

**--port 80**

**az network lb rule create \**

**--resource-group MyResourceGroup \**

**--lb-name MyInternalLB \**

**--name MyLoadBalancerRuleInternal \**

**--protocol tcp \**

**--frontend-port 80 \**

**--backend-port 80 \**

**--frontend-ip-name MyFrontEndInternal \**

**--backend-pool-name MyBackEndPoolInternal \**

**--probe-name MyHealthProbeInternal**

### Part D: Create and Test Azure Application Gateway

1. **Create an Application Gateway**:

**az network public-ip create \**

**--resource-group MyResourceGroup \**

**--name MyAppGatewayPublicIP**

**az network vnet subnet create \**

**--resource-group MyResourceGroup \**

**--vnet-name MyVNet \**

**--name AppGatewaySubnet \**

**--address-prefix 10.0.3.0/24**

**az network application-gateway create \**

**--name MyAppGateway \**

**--location eastus \**

**--resource-group MyResourceGroup \**

**--capacity 2 \**

**--sku Standard\_v2 \**

**--public-ip-address MyAppGatewayPublicIP \**

**--vnet-name MyVNet \**

**--subnet AppGatewaySubnet**

 **Configure Backend Pools and Rules**:

* Add VMs to the backend pool and configure routing rules accordingly.

 **Test the Application Gateway**:

* Access the public IP of the Application Gateway and verify the routing and load balancing to the backend VMs.