Project - Secure Hosting of Web App via Azure Application Gateway

Scenario:

Implement Hub and Spoke topology where Hub contains the centralized components like Azure Firewall, Application Gateway, DNS Forwarding VM, Azure Bastion etc. and one spoke has Web App and another spoke has a Storage account with no public access.

1. Establishes a secure connection between the on-premises data centre and the hub VNet and Spoke VNets.

2. Provides custom DNS on the top of Spoke VNets as DNS Forwarding VM.

3. Resolve all the DNS queries for Azure to On-premises, Azure to Azure and On-premises to Azure.

4. All the traffic should be routed through Azure Firewall.

5. Internet traffic will land on Application Gateway Public Fronted IP.

6. On-Premises traffic will land on Application Gateway Private Fronted IP.

7. SSL Offloading will be implemented on top of Application Gateway.

8. Set up multiple listeners to route traffic to the backend with their respective set of configurations.

Solution:

To implement a Hub and Spoke topology in Azure that includes the specified requirements, follow these detailed steps:

Step 1: Set Up Resource Groups

Create a resource group to organize your resources.

1. **Create HubSpokeRG**

Step 2: Set Up the Hub and Spoke VNets

2. **Create Hub VNet**:
    - Name: **HubVNet**
    - Address Space: **10.0.0.0/16**
3. **Create Spoke VNets**:
    - **SpokeVNet1** (for Web App):
        - Name: **SpokeVNet1**
        - Address Space: **10.1.0.0/16**

- **SpokeVNet2** (for Storage Account):
    - Name: **SpokeVNet2**
    - Address Space: **10.2.0.0/16**

Step 3: Set Up Peering

1. **Peer Hub to SpokeVNet1**:
    - Allow traffic to/from SpokeVNet1
2. **Peer Hub to SpokeVNet2**:
    - Allow traffic to/from SpokeVNet2
3. **Peer SpokeVNet1 to SpokeVNet2**:
    - Allow traffic to/from each other

Step 4: Set Up Azure Firewall in the Hub VNet

1. **Deploy Azure Firewall** in **HubVNet**.
2. **Create Firewall Policy**:
    - Configure rules to control traffic between VNets and to/from on-premises.
3. **Set Up Route Tables**:
    - Create route tables in both spoke VNets and route all traffic (0.0.0.0/0) to Azure Firewall.

Step 5: Deploy Azure Bastion in the Hub VNet

1. **Deploy Azure Bastion** for secure RDP/SSH access to VMs in Hub and Spoke VNets.

Step 6: Deploy Application Gateway in the Hub VNet

1. **Create Application Gateway**:
    - Configure with both Public and Private Frontend IPs.
    - Enable SSL offloading.
    - Create listeners for different traffic types (internet, on-premises).
2. **Set Up Backend Pools**:
    - Configure backend pools to route traffic to Web App and Storage Account.

Step 7: Set Up Custom DNS with DNS Forwarding VM

1. **Deploy a VM for DNS Forwarding** in **HubVNet**.

- Configure it to forward DNS queries to on-premises DNS servers and Azure DNS.

2. **Configure DNS Settings** in all VNets to use the DNS Forwarding VM.

Step 8: Secure On-Premises Connectivity

1. **Set Up Azure VPN Gateway** or **ExpressRoute** in **HubVNet** to establish a secure connection to the on-premises data center.
2. **Configure Route Tables** in Hub VNet to route on-premises traffic through VPN Gateway/ExpressRoute.

Step 9: Integrate Services in Spoke VNets

1. **Deploy Web App** in **SpokeVNet1**.
2. **Deploy Storage Account** in **SpokeVNet2** with private endpoint enabled and public access disabled.

Step 10: DNS Resolution Configuration

1. **Configure DNS Forwarding VM**:
    - Forwards Azure DNS queries to Azure-provided DNS.
    - Forwards on-premises queries to on-premises DNS.
    - Configure conditional forwarding rules as needed.

Step 11: Route All Traffic Through Azure Firewall

1. **Update UDRs (User Defined Routes)** in both spoke VNets to ensure all outbound traffic goes through the Azure Firewall.

Step 12: Verify and Test Configuration

1. **Validate Connectivity**:
    - Ensure VMs in Spoke VNets can resolve DNS queries via the DNS Forwarding VM.
    - Ensure traffic from on-premises to Azure and Azure to on-premises is routed correctly and securely.
    - Test access to the Web App via the Application Gateway.
2. **Monitor Traffic and Logs**:
    - Use Azure Firewall logs, Application Gateway access logs, and Network Watcher for monitoring.

Command Line Solution

Step 1: Set Up Resource Groups

 # Create Resource Group

az group create --name HubSpokeRG --location eastus

Step 2: Create VNets and Subnets

 # Create Hub VNet

az network vnet create --name HubVNet --resource-group HubSpokeRG --address-prefix 10.0.0.0/16 --subnet-name AzureFirewallSubnet --subnet-prefix 10.0.1.0/24

 # Create Spoke VNet 1 (Web App)

az network vnet create --name SpokeVNet1 --resource-group HubSpokeRG --address-prefix 10.1.0.0/16 --subnet-name default --subnet-prefix 10.1.1.0/24

 # Create Spoke VNet 2 (Storage Account)

az network vnet create --name SpokeVNet2 --resource-group HubSpokeRG --address-prefix 10.2.0.0/16 --subnet-name default --subnet-prefix 10.2.1.0/24

Step 3: Set Up Peering

 # Peer Hub VNet with SpokeVNet1

az network vnet peering create --name HubToSpoke1 --resource-group HubSpokeRG --vnet-name HubVNet --remote-vnet SpokeVNet1 --allow-vnet-access

 # Peer Hub VNet with SpokeVNet2

az network vnet peering create --name HubToSpoke2 --resource-group HubSpokeRG --vnet-name HubVNet --remote-vnet SpokeVNet2 --allow-vnet-access

 # Peer SpokeVNet1 with SpokeVNet2

az network vnet peering create --name Spoke1ToSpoke2 --resource-group HubSpokeRG --vnet-name SpokeVNet1 --remote-vnet SpokeVNet2 --allow-vnet-access

Step 4: Deploy Azure Firewall in Hub VNet

 # Create Public IP for Firewall

az network firewall create --name HubFirewall --resource-group HubSpokeRG --vnet-name HubVNet

 # Create Azure Firewall

az network public-ip create --resource-group HubSpokeRG --name HubFirewallPublicIP --sku Standard

 # Configure Firewall IP Configuration

az network firewall ip-config create --firewall-name HubFirewall --name FWConfig --public-ip-address HubFirewallPublicIP --vnet-name HubVNet

Step 5: Set Up Azure Bastion in Hub VNet

 # Create Public IP for Bastion

az network public-ip create --resource-group HubSpokeRG --name HubBastionPublicIP --sku Standard

 # Create Bastion Subnet

az network vnet subnet create --resource-group HubSpokeRG --vnet-name HubVNet --name AzureBastionSubnet --address-prefix 10.0.2.0/24

 # Deploy Bastion

az network bastion create --name HubBastion --resource-group HubSpokeRG --vnet-name HubVNet --public-ip-address HubBastionPublicIP

Step 6: Deploy Application Gateway in Hub VNet

 # Create Public IP for Application Gateway

az network public-ip create --resource-group HubSpokeRG --name AppGatewayPublicIP --sku Standard

 # Create Subnet for Application Gateway

az network vnet subnet create --resource-group HubSpokeRG --vnet-name HubVNet --name AppGatewaySubnet --address-prefix 10.0.3.0/24

 # Create Application Gateway

az network application-gateway create --name HubAppGateway --resource-group HubSpokeRG --vnet-name HubVNet --subnet AppGatewaySubnet --capacity 2 --sku WAF_v2 --public-ip-address AppGatewayPublicIP

Step 7: Configure DNS Forwarding VM

 # Create VM for DNS Forwarding

az vm create --resource-group HubSpokeRG --name DNSForwardingVM --image UbuntuLTS --vnet-name HubVNet --subnet default --admin-username azureuser --generate-ssh-keys

 # Install and Configure DNS Forwarding (Example for Ubuntu)

az vm extension set --resource-group HubSpokeRG --vm-name DNSForwardingVM --name customScript --publisher Microsoft.Azure.Extensions --settings '{"commandToExecute":"apt-get update && apt-get install -y bind9 && echo \"forwarders { 168.63.129.16; };\" >> /etc/bind/named.conf.options && service bind9 restart"}'

Step 8: Set Up Secure Connectivity with On-Premises

 # Create Public IP for VPN Gateway

az network public-ip create --resource-group HubSpokeRG --name VpnGatewayPublicIP --sku Standard

 # Create VPN Gateway

az network vnet-gateway create --name HubVpnGateway --resource-group HubSpokeRG --vnet HubVNet --public-ip-address VpnGatewayPublicIP --gateway-type Vpn --vpn-type RouteBased --sku VpnGw1 --no-wait

Step 9: Deploy and Configure Spoke Resources

 # Create Web App in SpokeVNet1

az webapp create --resource-group HubSpokeRG --plan AppServicePlan --name WebAppInSpoke1 --vnet-name SpokeVNet1

 # Create Storage Account in SpokeVNet2 with Private Endpoint

az storage account create --name SpokeStorage --resource-group HubSpokeRG --location eastus --sku Standard_LRS --enable-large-file-share --allow-blob-public-access false

 # Create Private Endpoint for Storage Account

az network private-endpoint create --name StoragePrivateEndpoint --resource-group HubSpokeRG --vnet-name SpokeVNet2 --subnet StorageSubnet --private-connection-resource-id /subscriptions/{subscription-id}/resourceGroups/HubSpokeRG/providers/Microsoft.Storage/storageAccounts/SpokeStorage --group-id blob

Step 10: Route All Traffic Through Azure Firewall

 # Create Route Table

az network route-table create --name HubRouteTable --resource-group HubSpokeRG

 # Add Route to Route Table

az network route-table route create --resource-group HubSpokeRG --route-table-name HubRouteTable --name DefaultRoute --address-prefix 0.0.0.0/0 --next-hop-type VirtualAppliance --next-hop-ip-address {Azure Firewall Private IP}

 # Associate Route Table with Subnets

az network vnet subnet update --vnet-name SpokeVNet1 --name WebAppSubnet --route-table HubRouteTable --resource-group HubSpokeRG

az network vnet subnet update --vnet-name SpokeVNet2 --name StorageSubnet --route-table HubRouteTable --resource-group HubSpokeRG

Step 11: Configure Application Gateway for SSL Offloading and Traffic Routing

 # Add SSL Certificate to Application Gateway

az network application-gateway ssl-cert create --resource-group HubSpokeRG --gateway-name HubAppGateway --name SslCert --cert-file "/path/to/certfile.pfx" --cert-password "yourpassword"

 # Create HTTP Settings with SSL

az network application-gateway http-settings create --resource-group HubSpokeRG --gateway-name HubAppGateway --name HttpsSetting --port 443 --protocol Https --cert-name SslCert

 # Create Listeners

az network application-gateway http-listener create --resource-group HubSpokeRG --gateway-name HubAppGateway --name PublicListener --frontend-port 443 --ssl-cert SslCert --frontend-ip AppGatewayPublicIP

az network application-gateway http-listener create --resource-group HubSpokeRG --gateway-name HubAppGateway --name PrivateListener --frontend-port 443 --ssl-cert SslCert --frontend-ip AppGatewayPrivateIP

 # Configure Backend Pools and Rules

az network application-gateway address-pool create --resource-group HubSpokeRG --gateway-name HubAppGateway --name WebAppPool --servers WebAppInSpoke1.azurewebsites.net

az network application-gateway address-pool create --resource-group HubSpokeRG --gateway-name HubAppGateway --name StoragePool --servers storageaccount.blob.core.windows.net

 # Create Routing Rules

az network application-gateway rule create --resource-group HubSpokeRG --gateway-name HubAppGateway --name WebAppRule --http-listener PublicListener --rule-type Basic --address-pool WebAppPool --http-settings HttpsSetting

az network application-gateway rule create --resource-group HubSpokeRG --gateway-name HubAppGateway --name StorageRule --http-listener PrivateListener --rule-type Basic --address-pool StoragePool --http-settings HttpsSetting

Step 12: Test and Validate Configuration

1. **Validate Connectivity**:
    - Ensure VMs in Spoke VNets can resolve DNS queries via the DNS Forwarding VM.
    - Ensure traffic from on-premises to Azure and Azure to on-premises is routed correctly and securely.
    - Test access to the Web App via the Application Gateway.
2. **Monitor Traffic and Logs**:

Use Azure Firewall logs, Application Gateway access logs, and Network Watcher for monitoring