**Project: Multi-Tier Secure Web Application Deployment in Azure**

**Scenario**

You’ve been hired by a **fictitious e-commerce company** called **ShopSphere**.  
They want to migrate their **3-tier application** (Web → App → Database) to Azure, ensuring **high security, proper network segmentation, and controlled traffic flow**.

Ddesign and deploy the Azure network & compute infrastructure.

**Business Requirements**

1. **Network Design**
   * Create a **hub-and-spoke architecture**:
     + **Hub VNet**: Contains shared services like the firewall and jump server.
     + **Spoke 1 VNet**: Hosts the **Web Tier** (public-facing VMs).
     + **Spoke 2 VNet**: Hosts the **App Tier** (internal VMs).
     + **Spoke 3 VNet**: Hosts the **Database Tier** (internal VMs, no internet access).
   * Configure **VNet Peering** between hub and spokes.
2. **Subnet Structure**
   * In each spoke, create dedicated subnets:
     + Web Tier Subnet
     + App Tier Subnet
     + DB Subnet (in Spoke 3 only)
   * In the hub VNet, create:
     + Azure Firewall Subnet (AzureFirewallSubnet)
     + Jumpbox Subnet
3. **Security Rules**
   * Use **NSGs** to:
     + Allow HTTP/HTTPS from the internet only to the Web Tier.
     + Allow only App Tier to talk to DB Tier on SQL port (1433).
     + Block all other unnecessary traffic between subnets.
   * Use **Azure Firewall** to:
     + Control outbound internet traffic from all spokes.
     + Allow App Tier to access an external API endpoint.
4. **Virtual Machines**
   * Deploy **2 Web VMs** in Spoke 1 (load-balanced in the future).
   * Deploy **2 App VMs** in Spoke 2.
   * Deploy **1 DB VM** in Spoke 3 (can use SQL Server image).
   * Deploy **1 Jumpbox VM** in the Hub for admin access (use Bastion for extra security).
5. **Routing**
   * Configure **User Defined Routes (UDR)** so that all traffic between spokes passes through the Azure Firewall in the hub.
6. **Testing**
   * Verify:
     + Public can reach the web servers.
     + Web servers can reach app servers.
     + App servers can reach DB server.
     + DB server cannot reach the internet.
     + Outbound traffic is filtered by the firewall.

**Deliverables**

* Azure Resource Group(s) with all resources deployed.
* Network diagram showing VNets, Subnets, Peering, NSGs, and Firewall.
* Screenshots of:
  + NSG rules
  + Peering settings
  + Firewall rules
  + Successful connectivity tests (ping, telnet, etc.)
* Documentation of setup steps.

**Learning Outcome**

By completing this project, you’ll get hands-on practice with:

* **VNet planning & peering**
* **Subnet segmentation**
* **NSG rule creation & traffic restriction**
* **Azure Firewall deployment & route control**
* **VM creation & connectivity testing**
* **Hub-and-spoke network design** (a common real-world Azure pattern)

**Project Documentation – Multi-Tier Secure Web Application in Azure**

**Project Name: ShopSphere – Secure Hub-and-Spoke Network Deployment**

**Author: *Your Name***

**Date: *Today’s Date***

**1. Project Overview**

This project implements a **hub-and-spoke** network architecture for an e-commerce application hosted in Azure.  
It provides **segmentation**, **security controls**, and **controlled connectivity** using VNets, Subnets, NSGs, Azure Firewall, and VNet peering.

**2. Architecture Diagram**

*(You can draw this in tools like draw.io, Lucidchart, or even Azure Diagram Tool)*  
The diagram should show:

* **Hub VNet** with Firewall, Jumpbox
* **Spoke 1** (Web Tier), **Spoke 2** (App Tier), **Spoke 3** (DB Tier)
* VNet Peering lines
* NSG & UDR placements
* Traffic flow arrows

**3. Phase 1 – Resource Groups**

**Objective:** Organize resources logically.

**Steps:**

1. In Azure Portal, go to **Resource groups → + Create**.
2. Create the following RGs:
   * RG-Network-Hub
   * RG-Network-Spoke1
   * RG-Network-Spoke2
   * RG-Network-Spoke3
   * RG-Compute
3. Choose a single region (e.g., **East US**) for all resources.

**4. Phase 2 – Virtual Networks & Subnets**

**Objective:** Create one Hub VNet and three Spoke VNets with proper subnets.

**Steps:**

**4.1 Hub VNet**

* Name: VNet-Hub
* Address Space: 10.0.0.0/16
* Subnets:
  + AzureFirewallSubnet → 10.0.1.0/26
  + Jumpbox-Subnet → 10.0.2.0/24

**4.2 Spoke 1 (Web Tier)**

* Name: VNet-Spoke1-Web
* Address Space: 10.1.0.0/16
* Subnets:
  + Web-Subnet → 10.1.1.0/24

**4.3 Spoke 2 (App Tier)**

* Name: VNet-Spoke2-App
* Address Space: 10.2.0.0/16
* Subnets:
  + App-Subnet → 10.2.1.0/24

**4.4 Spoke 3 (DB Tier)**

* Name: VNet-Spoke3-DB
* Address Space: 10.3.0.0/16
* Subnets:
  + DB-Subnet → 10.3.1.0/24

**5. Phase 3 – VNet Peering**

**Objective:** Connect the hub and spokes.

**Steps:**

1. Go to VNet-Hub → **Peerings** → **+ Add**.
2. Peer Hub to Spoke 1, allowing:
   * **Allow virtual network access** ✅
   * **Use remote gateways** (Hub → Yes, Spoke → No)
3. Repeat for Spoke 2 and Spoke 3.
4. Verify that all hub-to-spoke and spoke-to-hub peerings are active.

**6. Phase 4 – Network Security Groups**

**Objective:** Control traffic at subnet level.

**Steps:**

**6.1 Web Tier NSG**

* Allow inbound HTTP (80) & HTTPS (443) from **Internet**.
* Allow inbound App Tier access on port 8080 (example app port).
* Deny all other inbound traffic.

**6.2 App Tier NSG**

* Allow inbound traffic only from Web Tier Subnet.
* Allow outbound to DB Tier Subnet on port 1433 (SQL).
* Deny all internet access (force via Firewall).

**6.3 DB Tier NSG**

* Allow inbound SQL (1433) only from App Tier Subnet.
* Deny all outbound internet traffic.

**7. Phase 5 – Azure Firewall**

**Objective:** Centralized outbound and inbound control.

**Steps:**

1. In **RG-Network-Hub**, create **Azure Firewall**:
   * Name: FW-Hub
   * SKU: Standard
   * Public IP: FW-PIP
   * Subnet: AzureFirewallSubnet
2. Configure firewall rules:
   * Application Rule: Allow App Tier to access external API (api.example.com).
   * Network Rule: Allow Web Tier outbound to internet for updates.
3. Create **Route Tables (UDR)** for each spoke:
   * Route: 0.0.0.0/0 → Next Hop: Firewall Private IP
4. Associate UDRs with subnets.

**8. Phase 6 – Virtual Machines**

**Objective:** Deploy VMs for each tier.

**Steps:**

1. In **RG-Compute**, deploy:
   * WebVM1, WebVM2 in Web Subnet (Ubuntu/Windows)
   * AppVM1, AppVM2 in App Subnet
   * DBVM1 in DB Subnet (SQL Server image)
   * JumpboxVM in Hub Jumpbox Subnet
2. Assign appropriate NSGs.
3. Access VMs via **Bastion** or Jumpbox (no direct public IP for internal VMs).

**9. Phase 7 – Testing & Validation**

**Tests:**

* Public can reach Web VMs on HTTP/HTTPS.
* Web VMs can reach App VMs (custom port).
* App VMs can query DB VM via SQL port.
* DB VM cannot ping google.com (blocked by firewall).
* Firewall logs show outbound traffic inspection.

**10. Phase 8 – Documentation & Handover**

* Save:
  + Network diagram
  + NSG screenshots
  + Peering settings screenshots
  + Firewall rules & UDR screenshots
  + Test results
* Deliver as a PDF/Word document.