
🔹 Configure Networking Settings for Azure App Service

Networking settings in App Service control **how your app connects to the internet, other Azure resources, and on-premises systems**, as well as **who can access it**.

Step 1: Open Networking Settings

1. Log in to Azure Portal.
2. Go to your **App Service**.
3. In the left-hand menu → click **Networking**.
   Here, you'll see all networking options grouped in one place.

Step 2: Configure Inbound Traffic (Who Can Access Your App)

- **Access Restrictions**
    1. In Networking → select **Access Restrictions**.
    2. Add rules to allow or deny specific **IP addresses, ranges, or subnets**.
    3. Example: Allow only traffic from your company IP range, block all others.
- **Private Endpoints**
    1. In Networking → select **Private Endpoint Connections**.
    2. Add a private endpoint to place your App Service inside your **Azure Virtual Network (VNET)**.

3. This makes your app accessible only via private IP, not public internet.

Step 3: Configure Outbound Traffic (How App Service Connects to Others)

- **VNET Integration** (for accessing databases, APIs, or on-prem systems)
    1. In Networking → select **VNET Integration**.
    2. Choose a Virtual Network + Subnet.
    3. Your app can now securely reach resources inside that VNET (e.g., Azure SQL Database, VM, On-Prem via VPN/ExpressRoute).
- **Hybrid Connections**
    1. In Networking → select **Hybrid Connections**.
    2. Add a Hybrid Connection linked to an **Azure Relay**.
    3. This allows your app to connect securely to on-prem servers without full VNET integration.

Step 4: Configure Custom Domains & TLS

- In Networking → **Custom Domains**.
- Map a custom DNS domain (e.g., `www.myapp.com`).
- Secure it with SSL/TLS (upload cert or use free Azure-managed cert).

Step 5: Configure HTTPS and Minimum TLS Version

1. In **TLS/SSL Settings**.
2. Enforce **HTTPS Only** = On.
3. Set **Minimum TLS version** → Recommended **1.2**.

Step 6: Configure Outbound Restrictions (Optional)

- App Service uses a pool of outbound IP addresses.
- You can check them in **Properties** (important for firewall whitelisting).
- If you want a dedicated outbound IP → use **NAT Gateway** with VNET integration.

Step 7: Test Networking

- Access your app via browser with the public URL or custom domain.
- Test restricted access (e.g., try accessing from a blocked IP).
- If VNET integrated, test access to private resources (e.g., SQL Server inside VNET).

✅ Summary

- **Access Restrictions** → Control inbound traffic (who can reach the app).
- **Private Endpoints** → Lock app to private IP (no public access).
- **VNET Integration** → Allow outbound traffic to Azure/on-prem resources.
- **Hybrid Connections** → Lightweight connection to on-prem systems.
- **TLS/SSL** → Enforce secure connections.

- **Outbound IPs** → Use NAT Gateway if fixed outbound IP is needed.

🔹 Lab Guide: Configure Networking Settings for Azure App Service

---

Lab Objectives

By the end of this lab, students will:

1. Restrict access to an App Service using **Access Restrictions**.
2. Configure **VNET Integration** to allow App Service to connect to a private resource.
3. Test and validate the networking configuration.

Prerequisites

- An active **Azure subscription**.
- One **App Service** already deployed (any runtime stack).
- A **Virtual Network** (VNET) with at least one subnet.
- A sample private resource inside the VNET (e.g., Azure SQL Database or a VM with private IP).

Exercise 1: Restrict Inbound Access with Access Restrictions

1. In Azure Portal → go to your **App Service**.
2. In the left menu, select **Networking** → **Access Restrictions**.
3. Click **Add Rule**:
    - **Name**: `Allow-My-IP`.
    - **Action**: Allow.
    - **Priority**: 100.
    - **Source**: `IPv4`.
    - **Address Range**: Enter your current public IP (find at whatismyip.com).
    - Click **Add Rule**.
4. Add a **Deny All Rule** (priority 200).
5. Save changes.

👉 Now only your IP should be able to access the web app. Test from your browser.

Exercise 2: Integrate App Service with a VNET

1. In the App Service menu → select **Networking** → **VNET Integration**.
2. Click **Add VNET**.
3. Choose your existing VNET and a **subnet**.
4. Save changes.

👉 This allows the App Service to connect to private resources in that subnet.

Exercise 3: Test Private Resource Access

1. In App Service → go to **Configuration** → **Application Settings**.
2. Add a setting:
    - Name: `DB_CONNECTION_STRING`.
    - Value: Connection string of your Azure SQL Database (private endpoint enabled).
    - Save & Restart App Service.
3. Update your web app code (or test script) to connect to the private database using this connection string.
4. Test: Try accessing the private resource from your App Service.

Exercise 4: Enforce HTTPS and TLS

1. In App Service → **TLS/SSL Settings**.
2. Enable **HTTPS Only**.
3. Set **Minimum TLS Version** to **1.2**.
4. Save changes.

👉 Test by accessing `http://<appname>.azurewebsites.net` → it should redirect to **https**.

Validation

- ✅ App Service is only accessible from your IP.

- ✅ App Service can connect to a private resource inside VNET.
- ✅ HTTP is redirected to HTTPS.
- ✅ TLS version is enforced.

Cleanup (Optional)

- Remove access restrictions.
- Disconnect VNET integration.
- Delete resource group if no longer needed.
