**Azure Application Gateway: Step-by-Step**

**Objective:**

To create and test an **Azure Application Gateway** that distributes HTTP traffic across backend web servers.

**Prerequisites:**

* Azure Subscription (Free or Paid)
* At least two virtual machines (VMs) running a web server (e.g., IIS or Apache)
* Azure Resource Group

**Step 1: Create a Resource Group**

1. Go to **Azure Portal**: <https://portal.azure.com>
2. Search for **Resource groups** > **+ Create**.
3. Fill in:
   * Subscription
   * Resource Group Name: AppGwResourceGroup
   * Region: Select your desired region
4. Click **Review + Create** > **Create**.

**Step 2: Create Virtual Network and Subnets**

1. Search for **Virtual networks** > **+ Create**.
2. Fill in:
   * Name: AppGwVNet
   * Region: Same as Resource Group
   * Subnet 1 (Frontend): frontend-subnet
   * Subnet 2 (Backend): backend-subnet
3. Click **Review + Create** > **Create**.

**Step 3: Create Backend VMs**

1. Search for **Virtual Machines** > **+ Add**.
2. Create **2 VMs** in the backend-subnet:
   * Name: webvm1, webvm2
   * Image: Windows Server or Ubuntu
   * Open port: HTTP (port 80)
3. Install a web server (e.g., IIS):
   * For Windows:

Install-WindowsFeature -name Web-Server -IncludeManagementTools

* + Place a simple HTML file with text: “Hello from webvm1” and “Hello from webvm2”

**Step 4: Create Public IP Address**

1. Search for **Public IP addresses** > **+ Create**.
2. Name: AppGwPublicIP
3. SKU: **Standard**
4. Assignment: **Static**
5. Click **Review + Create** > **Create**.

**Step 5: Create Azure Application Gateway**

1. Search for **Application Gateway** > **+ Create**.
2. Basics:
   * Subscription & Resource Group
   * Name: MyAppGateway
   * Region: Same as VNet
   * Tier: **Standard V2**
3. Frontend IP:
   * Select Public IP: AppGwPublicIP
4. Backend pool:
   * Add VMs webvm1 and webvm2 (via NICs or IPs)
5. Routing Rules:
   * Listener: HTTP
   * Backend target: the pool created
   * HTTP Settings: Port 80, pick “override with host name” if needed
6. Review + Create > Wait for deployment

**Step 6: Test the Application Gateway**

1. Go to the **Public IP** of the Application Gateway (found in the App Gateway overview).
2. Open it in a browser.
3. Refresh the page multiple times.
4. You should see responses alternate between:
   * "Hello from webvm1"
   * "Hello from webvm2"

**Verification:**

* Open browser: http://<AppGwPublicIP>
* Confirm load balancing works
* Use Azure Monitor for metrics like:
  + Backend health
  + Throughput
  + Response status

**Cleanup (Optional):**

1. Delete the Resource Group to avoid charges:
   * Go to Resource Groups > AppGwResourceGroup > Delete