Lab Document: Deploy Azure Application Gateway and Route Traffic to VMs

Table of Contents

1. Introduction

Azure Application Gateway is a **Layer 7 (Application Layer) load balancer** that enables you to manage traffic to your web applications. It supports advanced routing features such as **URL-based routing**, **SSL termination**, and **Web Application Firewall (WAF)** capabilities.

---

2. Key Components of Azure Application Gateway

Component	Description
Frontend IP	The IP address (Public or Private) where clients send requests
Listener	Listens for incoming connection requests on the frontend IP and port (HTTP/HTTPS)
Backend Pool	Group of backend servers (VMs, NICs, IP addresses, or App Services) that receive traffic
HTTP Settings	Configuration for backend communication (protocol, port, session affinity, cookie-based)
Rules	Define how requests are routed from listener to backend pools based on URL path or hostname
Health Probes	Used to monitor the health of backend servers

---

3. Lab Objectives

- Create a Virtual Network and Subnets
- Deploy two Linux VMs running a simple web server
- Create an Azure Application Gateway
- Configure backend pool with VMs
- Set up HTTP settings, listener, and routing rules
- Route traffic through Application Gateway to VMs

- Verify load balancing and routing

---

4. Prerequisites

- Azure subscription
- Basic knowledge of Azure Portal or Azure CLI
- SSH key pair for VM authentication or password

---

5. Step-by-Step Lab Guide

Step 1: Create a Resource Group

- Navigate to Azure Portal
- Search **Resource groups** > **Create**
- Enter:
  - Name: `AppGatewayLabRG`
  - Region: e.g., `East US`
- Click **Review + Create** > **Create**

Step 2: Create Virtual Network and Subnets

- Go to **Virtual networks** > **Create**
- Fill details:

    - Name: `AppGatewayVNet`
    - Address space: `10.0.0.0/16`
- Add two subnets:
    - **Subnet1:** `AppGatewaySubnet` with `10.0.1.0/24` (for Application Gateway - **must** be named this exactly)
    - **Subnet2:** `BackendSubnet` with `10.0.2.0/24` (for VMs)
- Create the VNet

Step 3: Deploy Virtual Machines

- Create two Ubuntu VMs in `BackendSubnet`
- Use SSH key or password authentication
- Install Apache Web Server on each VM via SSH:

```
sudo apt update
sudo apt install apache2 -y
echo "<h1>VM1 - Welcome to Azure Application Gateway Demo</h1>" | sudo tee
/var/www/html/index.html
sudo systemctl start apache2
```

- Repeat on VM2 but change the welcome message to “VM2 - Welcome...”

Step 4: Deploy Azure Application Gateway

- Go to **Create a resource** > Search **Application Gateway** > **Create**
- Fill in the basics:

    - Name: `AppGatewayDemo`
    - Region: Same as VNet
    - Tier: Standard V2 (recommended)
    - SKU size: Small or Medium
- **Virtual Network:** Select `AppGatewayVNet`
- **Subnet:** Select `AppGatewaySubnet` (created earlier)
- Assign a **Public IP** to frontend IP configuration
- Leave other settings as default and create the gateway

Step 5: Configure Backend Pool

- Navigate to the Application Gateway resource
- Under **Backend pools**, click **Add**
- Name: `AppGatewayBackendPool`
- Add the **IP addresses** or **NICs** of the two VMs from `BackendSubnet`
- Save

Step 6: Create HTTP Settings

- Go to **HTTP settings** > **Add**
- Name: `AppGatewayHTTPSettings`
- Backend protocol: HTTP

- Backend port: 80
- Leave cookie-based affinity disabled (or enable if session stickiness required)
- Use default settings for probes or create a new probe targeting `/` on port 80
- Save

Step 7: Create Listeners and Rules

- Go to **Listeners** > **Add listener**
- Name: `AppGatewayListener`
- Frontend IP: Public IP created earlier
- Protocol: HTTP
- Port: 80
- Save
- Go to **Rules** > **Add basic rule**
- Name: `AppGatewayRule`
- Listener: Select `AppGatewayListener`
- Backend pool: Select `AppGatewayBackendPool`
- HTTP settings: Select `AppGatewayHTTPSettings`
- Save

Step 8: Test Application Gateway

- Copy the **Public IP address** of the Application Gateway frontend
- Open a browser and navigate to: `http://<AppGatewayPublicIP>`

You should see the web page served by either VM1 or VM2 (round-robin load balancing). Refresh multiple times to confirm traffic distribution.

---

6. Summary

Component	Purpose
Resource Group	Logical container for all resources
Virtual Network/Subnets	Network segmentation for Application Gateway & VMs
Virtual Machines	Backend servers hosting web applications
Application Gateway	Layer 7 load balancer routing HTTP traffic
Backend Pool	Group of VMs to route traffic
HTTP Settings	Connection settings for backend communication
Listener	Entry point for client requests
Rule	Maps listener to backend pool and HTTP settings

---

7. Cleanup

- Delete the resource group `AppGatewayLabRG` to remove all resources and avoid charges.
