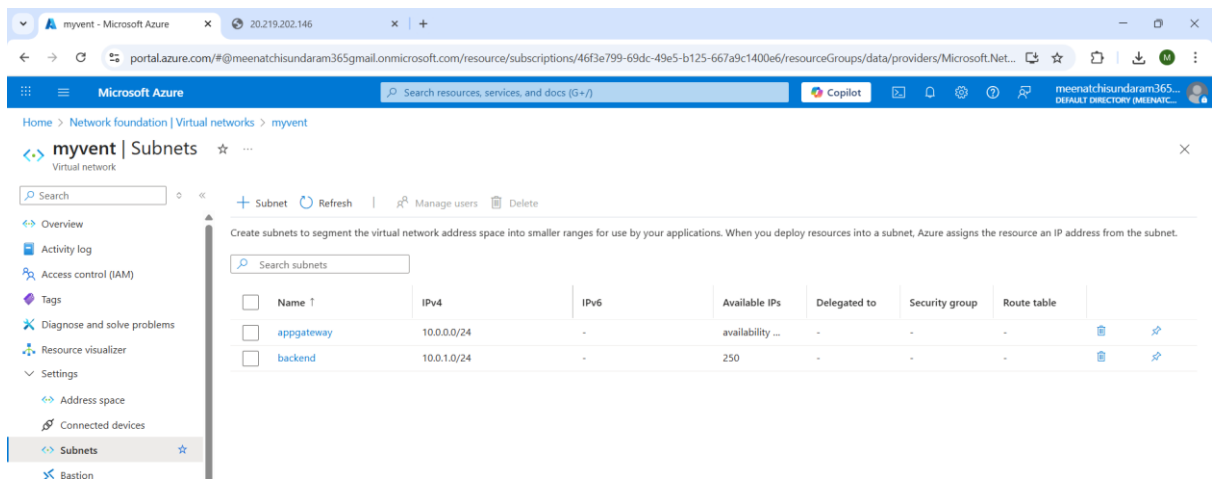


Configure Azure Application Gateway with WAF

Step 1: Create a Virtual Network and Subnets

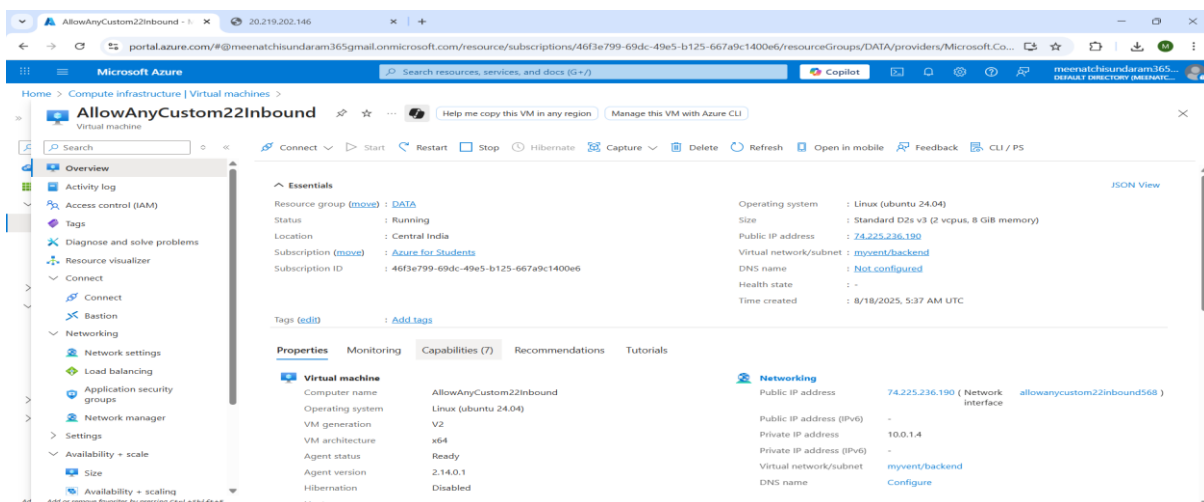
1. In the Azure Portal, search and select **Virtual Networks** > + **Create**.
2. Fill in the details:
 - o Name: MyVNet
 - o Region: Same as your Application Gateway
3. Add two subnets:
 - o **AppGatewaySubnet** (e.g., 10.0.0.0/24)
 - o **BackendSubnet** (e.g., 10.0.1.0/24)
4. Click **Review + Create** > **Create**



Step 2: Deploy Backend Target (VM or Web App)

Option A: Create a Virtual Machine

5. Go to **Virtual Machines** > + **Create**
6. Use the same region and resource group
7. Choose size and credentials
8. Under Networking, place it in **BackendSubnet**
9. Allow **HTTP (port 80)** traffic
10. Deploy and install a basic web server (e.g., IIS on Windows or Apache on Linux)



Execute this commands in Linux

sudo apt update

```
azureuser@AllowAnyCustom22Inbound:~$ sudo apt update
sudo: apt: command not found
azureuser@AllowAnyCustom22Inbound:~$ sudo apt update
Hit:1 http://azure.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://azure.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://azure.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://azure.archive.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://azure.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://azure.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://azure.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:8 http://azure.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:9 http://azure.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
```

sudo apt install apache2 -y

```
azureuser@AllowAnyCustom22Inbound:~$ sudo apt install apache2 -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1t64
  liblua5.4-0 ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1t64
  liblua5.4-0 ssl-cert
```

echo "Welcome to the secure app!" | sudo tee /var/www/html/index.html

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
azureuser@AllowAnyCustom22Inbound:~$ echo "Welcome the source app!" | sudo tee /var/www/html/index.html
Welcome the source app!
azureuser@AllowAnyCustom22Inbound:~$ sudo apt install apache2 -y
```

Step 3: Create Azure Application Gateway with WAF

11. Go to **Application Gateways > + Create**

12. Fill in the **Basics** tab:

- Subscription and resource group
- Name: MyAppGateway
- Region: Same as VNet
- Tier: **WAF V2**
- Enable autoscaling (optional)

Changes you make on this tab may affect any configuration you've done on other tabs. Review all options prior to creating the application gateway.

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *

Resource group * [Create new](#)

Instance details

Application gateway name *

Region *

Tier

Enable autoscaling ☒ Yes ☐ No

Minimum instance count *

Maximum instance count

IP address type ☒ IPv4 only ☐ Dual stack (IPv4 & IPv6)

HTTP2 ☐ Disabled ☒ Enabled

[Previous](#) [Next: Frontends >](#)

13. Frontend configuration:

- Choose **Public IP**
- Click + **Add new** to create a new IP (MyAppGatewayIP)

The screenshot shows the 'Create application gateway' page in the Microsoft Azure portal, specifically the 'Frontends' tab. The page has a blue header with the Microsoft Azure logo and a search bar. Below the header, there's a breadcrumb trail: 'Home > Load balancing and content delivery | Application gateways >'. The main title is 'Create application gateway'. A warning message states: 'Changes you make on this tab may affect any configuration you've done on other tabs. Review all options prior to creating the application gateway.' The 'Frontends' tab is selected, showing a progress bar with 'Basics', 'Frontends', 'Backends', 'Configuration', 'Tags', and 'Review + create'. Below the progress bar, a description reads: 'Traffic enters the application gateway via its frontend IP address(es). An application gateway can use a public IP address, private IP address, or one of each type.' The 'Frontend IP address type' section has three radio buttons: 'Public' (selected), 'Private', and 'Both'. The 'Public IPv4 address' section shows a dropdown menu with '(New) myappgatewayip' selected and an 'Add new' link below it.

14. Backend pool:

- Name: BackendPool
- Add the IP address of your VM or FQDN of your web app

The screenshot shows the 'Add a backend pool' dialog in the Microsoft Azure portal. The dialog has a blue header with the title 'Add a backend pool.' and a close button. Below the header, a description reads: 'A backend pool is a collection of resources to which your application gateway can send traffic. A backend pool can contain virtual machines, virtual machines scale sets, IP addresses, domain names, or an App Service.' The 'Name' field is empty. There are two buttons: 'Yes' and 'No'. Below the buttons, the 'Backend targets' section shows '0 items'. The 'Target type' dropdown menu is set to 'IP address or FQDN'. The 'Target' field is empty. In the background, the 'Create application gateway' page is visible, showing the 'Backends' tab selected in the progress bar. The 'Backends' tab description reads: 'A backend pool is a collection of resources to which your application gateway can send traffic. A backend pool can contain virtual machines, virtual machine scale sets, app services, IP addresses, or fully qualified domain names (FQDN).'. Below the description, there's a table with two columns: 'Backend pool' and 'Targets'. The 'Backend pool' column has a link 'Add a backend pool' and a row with 'backendpool'. The 'Targets' column has a link '> 1 target'.

15. Routing rules:

- **Listener:**
 - Name: AppListener
 - Protocol: **HTTP**
 - Port: 80
 - Frontend IP: Public
- **Rule:**
 - Name: HTTPRule
 - Backend Pool: BackendPool
 - Backend HTTP Settings:
 - Port: 80
 - Protocol: HTTP
 - Affinity: Disabled

16. Click **Next** through remaining tabs > **Review + Create** > **Create**

Step 4: Enable and Configure WA

17. Go to the **Application Gateway** resource

18. Under **Settings**, select **Web application firewall**

19. Click **Edit**

20. Set the following:

- **WAF status:** Enabled
- **Firewall mode:** Prevention (or Detection for testing)
- **Rule set:** OWASP 3.2 or 3.1
- **Request size:** Leave default unless needed

21. Click **Save**

Output :

