



Azure Application Gateway

Azure Application Gateway is a layer 7 load balancer and web application firewall (WAF) service offered by Microsoft Azure. It enables you to build highly available and scalable web applications by providing features such as SSL termination, cookie-based session affinity, URL-based routing, and WAF protection against common web vulnerabilities.

Here are some key features and use cases of Azure Application Gateway:

1. **Load Balancing:** Application Gateway can distribute incoming HTTP/HTTPS traffic among multiple backend servers based on various criteria such as round-robin, least connections, or session affinity. This helps improve the availability, scalability, and performance of web applications by evenly distributing the workload.
2. **SSL Termination:** Application Gateway can offload SSL/TLS encryption and decryption, reducing the computational overhead on backend servers. It supports multiple SSL certificates, allowing you to secure your web applications with custom domain names and SSL policies.
3. **URL-Based Routing:** With URL-based routing capabilities, Application Gateway can route incoming requests to different backend pools based on the URL path or hostname. This enables you to host multiple web applications or services behind a single Application Gateway instance and route traffic accordingly.
4. **Session Affinity:** Application Gateway supports cookie-based session affinity, also known as sticky sessions, ensuring that subsequent requests from the same client are routed to the same backend server. This is useful for maintaining session state and preserving user sessions in stateful web applications.
5. **Web Application Firewall (WAF):** Application Gateway includes a built-in web application firewall that provides protection against common web vulnerabilities such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF). The WAF rules can be customized to meet specific security requirements, helping to secure your web applications against malicious attacks.
6. **Health Monitoring and Autoscaling:** Application Gateway continuously monitors the health of backend servers and automatically removes or adds servers based on their availability and health status. This ensures high availability and reliability of web applications, even during traffic spikes or server failures.
7. **Integration with Azure Services:** Application Gateway integrates seamlessly with other Azure services such as Azure Load Balancer, Azure Front Door, Azure Kubernetes Service (AKS), and Azure App Service, providing a comprehensive solution for building modern web applications and microservices architectures in Azure.

Overall, Azure Application Gateway simplifies the deployment, management, and scaling of web applications by providing advanced load balancing, SSL termination, URL routing, and web application firewall capabilities in a single, managed service.

Use cases of Gateway Load Balancer:

Here are some common use cases for Azure Application Gateway:

1. **Web Application Hosting:** One of the primary use cases for Azure Application Gateway is hosting web applications. Whether it's a simple website or a complex web application, Application Gateway can efficiently distribute incoming HTTP/HTTPS traffic to multiple backend servers, ensuring high availability and scalability.
2. **SSL Termination:** If you have web applications that require SSL/TLS encryption, Application Gateway can handle SSL termination, offloading the encryption and decryption process from backend servers. This reduces the computational overhead on your servers and simplifies SSL certificate management.
3. **Path-Based Routing:** Application Gateway supports URL-based routing, allowing you to route incoming requests to different backend pools based on the URL path. This feature is useful for hosting multiple web applications or APIs behind a single Application Gateway instance and routing traffic accordingly.
4. **Session Affinity:** In scenarios where maintaining session state is important, such as e-commerce platforms or online banking applications, Application Gateway's cookie-based session affinity ensures that subsequent requests from the same client are routed to the same backend server, preserving user sessions.
5. **Web Application Firewall (WAF):** Security is a critical aspect of web applications. Application Gateway includes a built-in web application firewall (WAF) that provides protection against common web vulnerabilities, such as SQL injection, cross-site scripting (XSS), and more. This helps secure your web applications from malicious attacks.
6. **Microservices Architecture:** For applications built on a microservices architecture, Application Gateway can be used to route traffic to different microservices based on URL paths or headers. Combined with Azure Kubernetes Service (AKS) or Azure App Service, Application Gateway provides a scalable and resilient infrastructure for microservices-based applications.
7. **Hybrid Cloud Scenarios:** In hybrid cloud environments where you have resources both in Azure and on-premises, Application Gateway can act as a reverse proxy, securely exposing on-premises web applications to the internet or routing traffic between Azure and on-premises environments.
8. **Content Delivery:** Application Gateway can be used as a content delivery network (CDN) for serving static content such as images, CSS files, and JavaScript files. By caching content at the edge, Application Gateway reduces latency and improves the performance of web applications for end users.

In this walkthrough, we are setting up Azure Application Gateway to efficiently route incoming web traffic to different backend servers based on URL paths. The end goal is to showcase how Application Gateway can be used to host multiple web applications or serve different types of content (such as images and videos) from separate backend servers, all while providing load balancing, SSL termination, and

web application firewall (WAF) protection. By implementing URL-based routing, we can demonstrate how users accessing the same public IP address are directed to different web content based on their requested URL paths, enhancing the flexibility and scalability of web application hosting in Azure.

😊 To begin with the Lab:

1. First, we are going to create 2 virtual machines based on Windows Server 2022. Then we are going to install a Web Server (IIS) on both of them plus we will also create files on them to differentiate between the servers as when we'll call them using the Public IP address.
2. Below are the snapshots that you will follow to create the VMs and Install the web server on them.
3. Choose the Properties for your VM as shown below.

Project details

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

Subscription *	<input type="text" value="Azure Pass - Sponsorship"/>
Resource group *	<input type="text" value="new-grp"/> Create new

Instance details

Virtual machine name *	<input type="text" value="ImageVM"/> ✓
Region *	<input type="text" value="(Europe) North Europe"/>
Availability options	<input type="text" value="No infrastructure redundancy required"/>
Security type	<input type="text" value="Standard"/>
Image *	<input type="text" value="Windows Server 2022 Datacenter - x64 Gen2"/> ✓ See all images Configure VM generation

🚀 This image is compatible with additional security features. [Click here to swap to the Trusted launch security type.](#)

Run with Azure Spot discount

Size * Standard_DS1_v2 - 1 vcpu, 3.5 GiB memory (₹7,044.92/month) [See all sizes](#)

Enable Hibernation
i Hibernate is not supported by the size that you have selected. Choose a size that is compatible with Hibernate to enable this feature. [Learn more](#)

Administrator account

Username * <input type="checkbox"/>	imagevm ✓
Password *	***** ✓
Confirm password *	***** ✓

Inbound port rules

Select which virtual machine network ports are accessible from the public internet. You can specify more limited or granular network access on the Networking tab.

Public inbound ports * None Allow selected ports

Select inbound ports *

i All traffic from the internet will be blocked by default. You will be able to change inbound port rules in the VM > Networking page.

4. Then in the networking section for the Network Interface you have to create a new Virtual Network.

Basics Disks **Networking** Management Monitoring Advanced Tags Review + create

Define network connectivity for your virtual machine by configuring network interface card (NIC) settings. You can control ports, inbound and outbound connectivity with security group rules, or place behind an existing load balancing solution.
[Learn more](#)

Network interface

When creating a virtual machine, a network interface will be created for you.

Virtual network * <input type="checkbox"/>	(new) Demo-VN <input type="button" value="▼"/> Create new
Subnet * <input type="checkbox"/>	(new) default (10.0.0.0/24) <input type="button" value="▼"/>
Public IP <input type="checkbox"/>	(new) ImageVM-ip <input type="button" value="▼"/> Create new

5. Then directly jump to the Review page and create your Virtual machine.

6. Now as your virtual machine is being deployed go ahead and launch your second virtual machine.

Subscription * ⓘ Azure Pass - Sponsorship ⌄

Resource group * ⓘ new-grp ⌄
[Create new](#)

Instance details

Virtual machine name * ⓘ VideoVM ✓

Region * ⓘ (Europe) North Europe ⌄

Availability options ⓘ No infrastructure redundancy required ⌄

Security type ⓘ Trusted launch virtual machines ⌄
[Configure security features](#)

Image * ⓘ Windows Server 2022 Datacenter - x64 Gen2 ⌄
[See all images](#) | [Configure VM generation](#)

Administrator account

Username * ⓘ videovm ✓

Password * ✓

Confirm password * ✓

Inbound port rules

Select which virtual machine network ports are accessible from the public internet. You can specify more limited or granular network access on the Networking tab.

Public inbound ports * ⓘ None Allow selected ports

Select inbound ports * HTTP (80), RDP (3389) ⌄

7. Just remember that the Virtual Network should be the same as before.

Define network connectivity for your virtual machine by configuring network interface card (NIC) settings. You can control ports, inbound and outbound connectivity with security group rules, or place behind an existing load balancing solution.

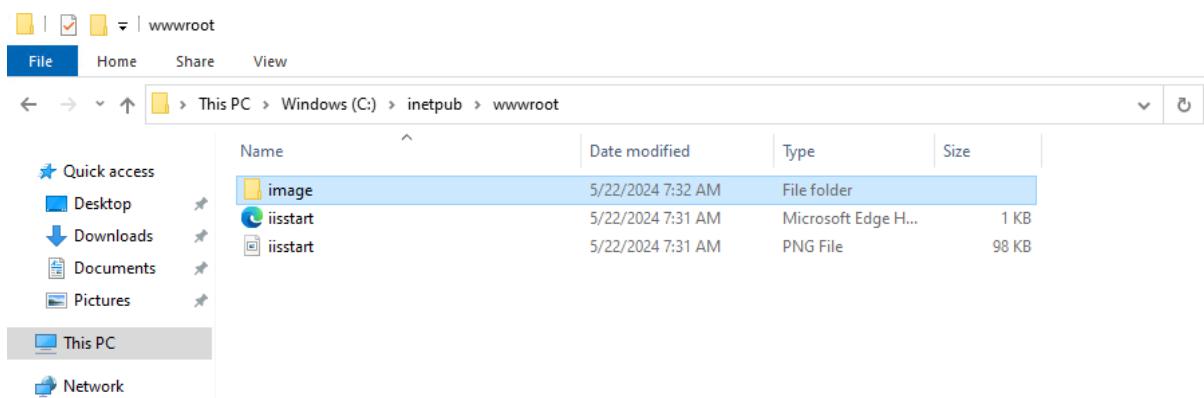
[Learn more](#)

Network interface

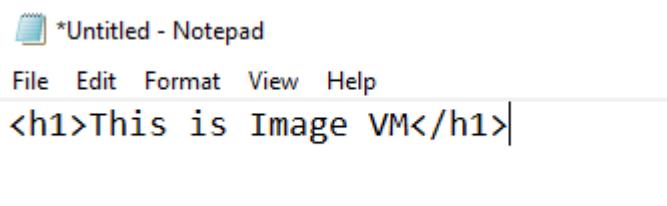
When creating a virtual machine, a network interface will be created for you.

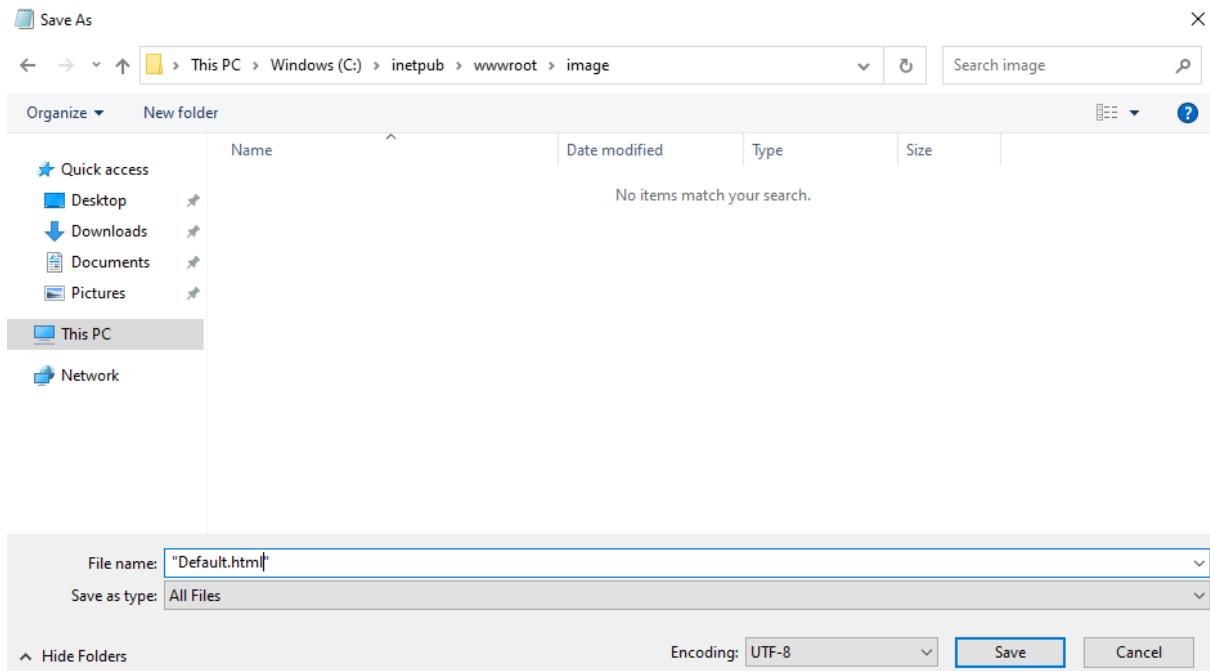
Virtual network *	<input type="text" value="Demo-VN"/> ▼ Create new
Subnet *	<input type="text" value="default (10.0.0.0/24)"/> ▼ Manage subnet configuration
Public IP	<input type="text" value="(new) VideoVM-ip"/> ▼ Create new

8. Then just move to the review page and create your VM.
9. Now you are going to login to your First VM which is your image VM and install Web Server in it.
10. Once your Web Server is installed then you need to go to this folder and create a folder with the image name in it.

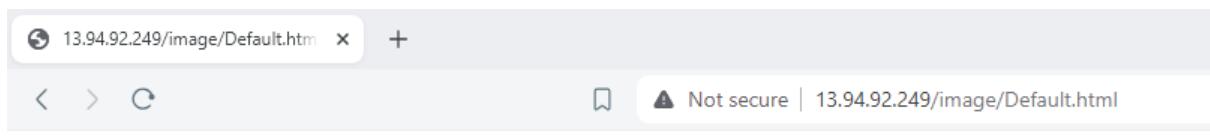


11. After that open the notepad and create a basic HTML page. Then save it to the image folder as shown below.

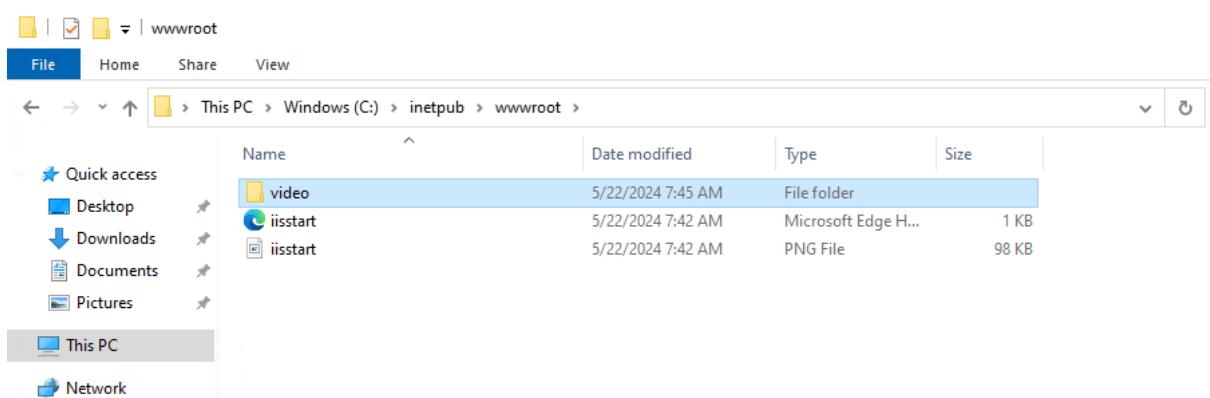




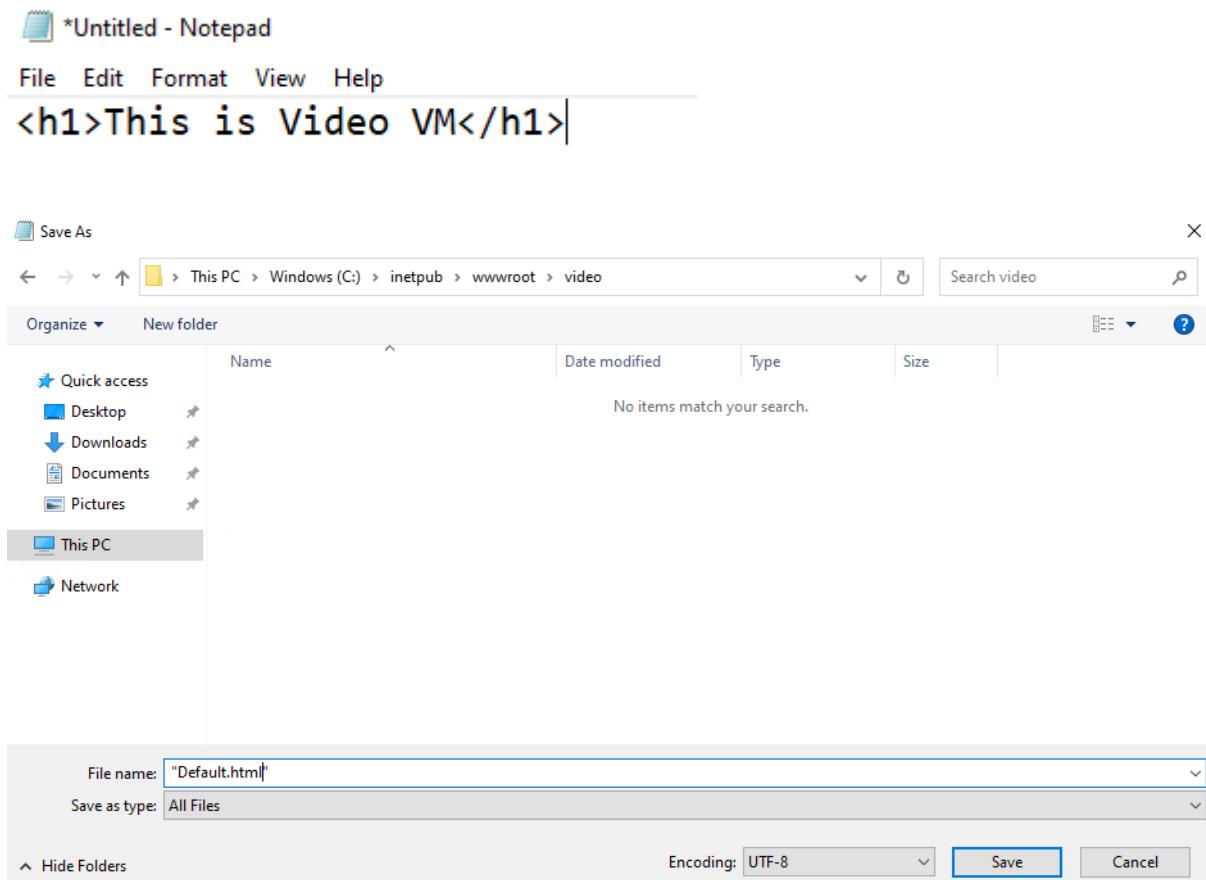
12. Also, if you try to access the web page for the Image VM and append it with the folder and file name then you will see the results as expected.



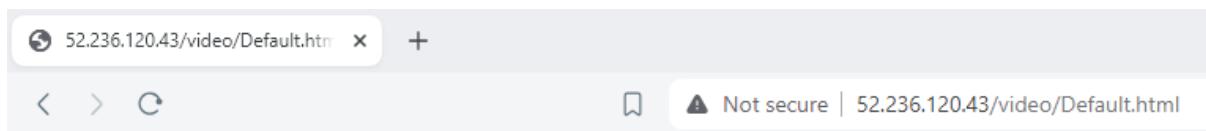
13. Now you have to do the same thing in your Video VM. Install the Web Server in it then create a video folder in this location shown below.



14. Then open your Notepad, create an HTML page, and save it at the same location as you can see down below.



15. Also, if you try to access the web page for the Video VM and append it with the folder and file name then you will see the results as expected.



This is Video VM



URL Routing Implementation

1. First, we need to create an empty subnet that will store the Application Gateway's own compute machines or say its own resources in this empty subnet.
2. So, these resources are required for the routing of traffic onto machines in your backend pool.
3. So, in the Azure Load Balancer, this was a very simple network routing service. It just looked at the network layer and it would route the request onto the machines in the backend pool.
4. But here in the Azure Application Gateway, remember, this is a layer 7 routing service. That means it needs to process the application request, the HTTP request, and then do the routing accordingly.

- So, for this, it is going to have its own infrastructure. This is completely managed by the service itself. All you need to do is to ensure that you have an empty subnet in place in an Azure virtual network.
- So, for any VM we will go onto the virtual network and from here go to subnets then click on add subnet as shown below.

Demo-VN | Subnets

Virtual network

+ Subnet + Gateway subnet Refresh Manage users Delete

Address space Connected devices Subnets Bastion

Search subnets

Name ↑↓	IPv4 ↑↓	IPv6 ↑↓	Available IPs ↑↓	Delegated to ↑↓
default	10.0.0.0/24	-	249	-

- Here we just need to give it a name and we'll accept the subnet range it is providing us. Then just click on Add.

Add a subnet

Select an address space and configure your subnet. You can customize a default subnet or select from subnet templates if you plan to add select services later. [Learn more](#)

Subnet purpose ① Default

Name * ① demosubnet

IPv4

Include an IPv4 address space

IPv4 address range * ① 10.0.0.0/16
10.0.0.0 - 10.0.255.255

Starting address * ① 10.0.1.0

Size ① /24 (256 addresses)

Subnet address range ① 10.0.1.0 - 10.0.1.255

IPv6

Include an IPv6 address space This virtual network has no IPv6 address ranges.

Private subnet PREVIEW

Private subnets enhance security by not providing default outbound access. To enable outbound connectivity for virtual machines to access the internet, it is necessary to explicitly grant outbound access. A NAT gateway is the recommended way to provide outbound connectivity for virtual machines in the subnet. [Learn more](#)

- Now in the marketplace you need to search for application gateway and choose this service accordingly.

Application Gateway

Microsoft

Application Gateway Add to Favorites

Microsoft | Azure Service

★ 4.4 (219 ratings)

Plan

Application Gateway ▼ Create

9. First you need to select your resource group.

Project details

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

Subscription *	<input type="text" value="Azure Pass - Sponsorship"/>
Resource group *	<input type="text" value="new-grp"/> Create new

10. Then give a name to your application gateway and choose the tier as Standard V2.
After that to enable autoscaling say NO.

11. Then for instance count keep it to 1. For the availability zones choose all 3. Then disable HTTP2.

Instance details

Application gateway name *	<input type="text" value="demo-gateway"/> ✓
Region *	<input type="text" value="North Europe"/> ▼
Tier	<input type="text" value="Standard V2"/> ▼
Enable autoscaling	<input type="radio"/> Yes <input checked="" type="radio"/> No
Instance count *	<input type="text" value="1"/> ✓
Availability zone *	<input type="text" value="Zones 1, 2, 3"/> ▼
HTTP2	<input checked="" type="radio"/> Disabled <input type="radio"/> Enabled
IP address type	<input checked="" type="radio"/> IPv4 only <input type="radio"/> Dual stack (IPv4 & IPv6)

12. Then for configuring Virtual Network choose your virtual network and the empty subnet. If you try to choose the subnet which has our VMs then you will instantly get an error.

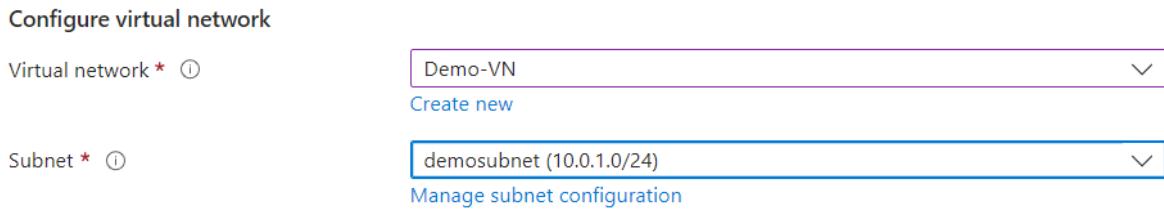
Configure virtual network

Virtual network * ⓘ Demo-VN

Create new

Subnet * ⓘ demosubnet (10.0.1.0/24)

Manage subnet configuration



13. Then for the frontend IP you have to create a new Public Address.

✓ Basics 2 Frontends 3 Backends 4 Configuration 5 Tags 6 Review + create

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. ↗

Frontend IP address type ⓘ Public Private Both

Public IPv4 address *

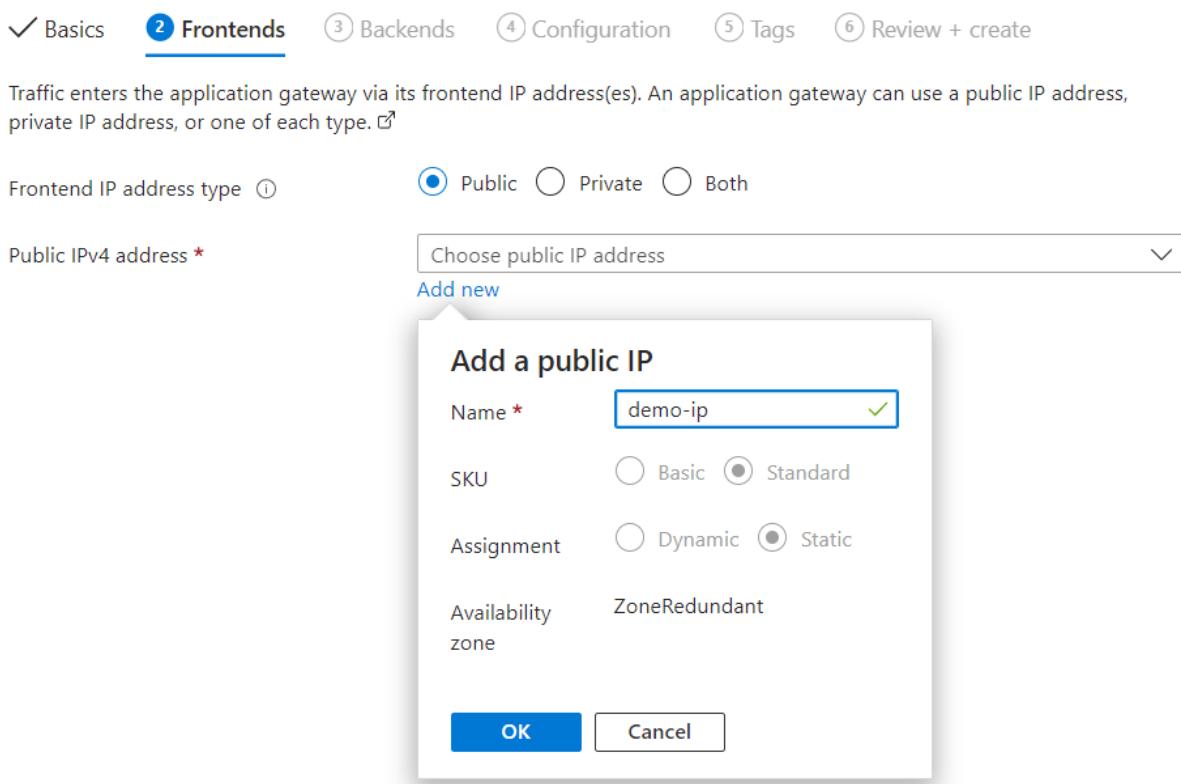
Choose public IP address

Add new

Add a public IP

Name *	demo-ip
SKU	<input type="radio"/> Basic <input checked="" type="radio"/> Standard
Assignment	<input type="radio"/> Dynamic <input checked="" type="radio"/> Static
Availability zone	ZoneRedundant

OK Cancel



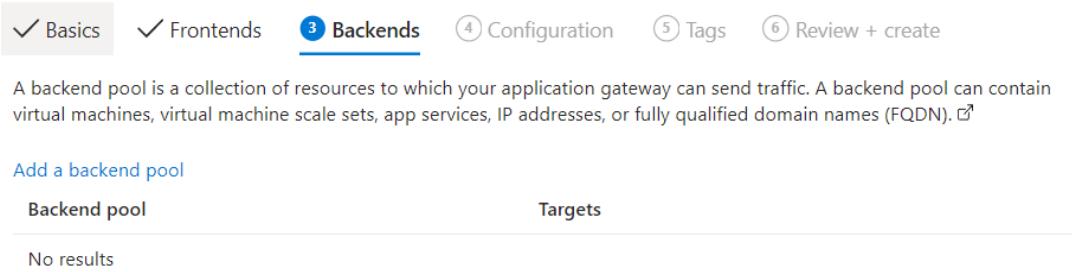
14. After that for the backend pool you have to click on add and you are going to add both of your VMs separately. Which means that you have to create two Pools.

✓ Basics ✓ Frontends 3 Backends 4 Configuration 5 Tags 6 Review + create

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). ↗

Add a backend pool

Backend pool	Targets
No results	



15. Below you can see that first we created a pool for our image VM and chose the properties for it as shown below.

Add a backend pool.

X

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.

Name *	imagepool ✓		
Add backend pool without targets	Yes	No	
Backend targets			
1 item			
Target type	Target		
Virtual machine	imagevm842 (10.0.0.4)	⋮	✖ ...
IP address or FQDN			

16. Then created a pool for our Video VM.

Add a backend pool.

X

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.

Name *	videopool ✓		
Add backend pool without targets	Yes	No	
Backend targets			
1 item			
Target type	Target		
Virtual machine	videovm367 (10.0.0.5)	⋮	✖ ...
IP address or FQDN			

17. Below you can see that we added both of our VMs in backend pool separately.

✓ Basics ✓ Frontends 3 Backends ④ Configuration ⑤ Tags ⑥ Review + create

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). ↗

Add a backend pool

Backend pool	Targets	...
imagepool	> 1 target	...
videopool	> 1 target	...

18. Then in configuration you can see that we have our frontend and backend ready. Now we need to create routing rules for our VM. For that click on add routing rule.

✓ Basics ✓ Frontends ✓ Backends ① Configuration ② Tags ③ Review + create

Create routing rules that link your frontend(s) and backend(s). You can also add more backend pools, add a second frontend IP configuration if you haven't already, or edit previous configurations. ↗

The screenshot shows the Azure Application Gateway configuration interface. On the left, there's a 'Frontends' section with a 'demo-ip' entry and a '+ Add a frontend IP' button. In the center, a 'Routing rules' section is highlighted with a red box; it contains a '+ Add a routing rule' button. On the right, a 'Backend pools' section lists 'imagepool' and 'videopool' with their respective details and edit buttons.

19. Here you can see that first we need to give our rule a name then we need to set the priority. After that we need to give a listener name and then move to backend targets.

Add a routing rule

X

Configure a routing rule to send traffic from a given frontend IP address to one or more backend targets. A routing rule must contain a listener and at least one backend target.

Rule name *	RuleA
Priority *	1
* Listener	* Backend targets

A listener "listens" on a specified port and IP address for traffic that uses a specified protocol. If the listener criteria are met, the application gateway will apply this routing rule.[↗]

Listener name *	Listner
Frontend IP *	Public IPv4
Protocol	<input checked="" type="radio"/> HTTP <input type="radio"/> HTTPS
Port *	80
Listener type	<input checked="" type="radio"/> Basic <input type="radio"/> Multi site

Custom error pages

Show customized error pages for different response codes generated by Application Gateway. This section lets you configure Listener-specific error pages. [Learn more ↗](#)

Bad Gateway - 502	Enter Html file URL
Forbidden - 403	Enter Html file URL

[Show more status codes](#)

20. Then we need to choose our backend target which will be our image pool and then click on add new for backend settings.

* Listener * Backend targets

Choose a backend pool to which this routing rule will send traffic. You will also need to specify a set of Backend settings that define the behavior of the routing rule.[↗]

Target type	<input checked="" type="radio"/> Backend pool <input type="radio"/> Redirection
Backend target *	imagepool
Backend settings *	Add new

21. For the backend settings you just need to give it a name and leave the rest of the settings as they are and click on save.

[← Discard changes and go back to routing rules](#)

Backend settings name *	setting
Backend protocol	<input checked="" type="radio"/> HTTP <input type="radio"/> HTTPS
Backend port *	80
Additional settings	
Cookie-based affinity	<input type="radio"/> Enable <input checked="" type="radio"/> Disable
Connection draining	<input type="radio"/> Enable <input checked="" type="radio"/> Disable
Request time-out (seconds) *	20
Override backend path	
Host name	
By default, the Application Gateway sends the same HTTP host header to the backend as it receives from the client. If your backend application/service requires a specific host value, you can override it using this setting.	
<input type="radio"/> Yes <input checked="" type="radio"/> No	
Override with new host name	<input type="radio"/> Yes <input checked="" type="radio"/> No
Create custom probes	

22. After that scroll down to backend targets and we need to add our path-based routing. We need to click on add multiple targets.

Path-based routing

You can route traffic from this rule's listener to different backend targets based on the URL path of the request. You can also apply a different set of Backend settings based on the URL path. ↗

Path based rules

Path	Target name	Backend setting name	Backend pool
No additional targets to display			

[Add multiple targets to create a path-based rule](#)

23. Here you need to give the path for your image folder as shown below. Just remember in the path the name should be same as you defined in your virtual machine while creating it. Then you need to give a target name, after that you have to choose the backend setting and the backend target for you is image pool. After that just click on save.

Target type	<input checked="" type="radio"/> Backend pool <input type="radio"/> Redirection
Path *	/image/*
Target name *	imagetarget
	setting
Backend settings *	<input type="radio"/> Add new <input checked="" type="radio"/> imagepool
Backend target *	<input type="radio"/> Add new <input checked="" type="radio"/> imagepool

24. Then you are going to do the same for video and click on save.

Target type	<input checked="" type="radio"/> Backend pool <input type="radio"/> Redirection
Path *	/video/*
Target name *	videotarget
	setting
Backend settings *	Add new
Backend target *	videopool
	Add new

25. Then in the path-based routing you can see both of the paths.

Path-based routing

You can route traffic from this rule's listener to different backend targets based on the URL path of the request. You can also apply a different set of Backend settings based on the URL path. ☰

Path based rules

Path	Target name	Backend setting name	Backend pool	...
/image/*	imagetarget	setting	imagepool	...
/video/*	videotarget	setting	videopool	...

Add multiple targets to create a path-based rule

26. After that just click on ADD and your rule will be saved then you need to go to the review page and create your Application Gateway.
27. Now this might take around 4-5 mins. Once your deployment is complete go to the resource group and choose your application gateway.

The screenshot shows the 'Overview' tab of an Application Gateway deployment named 'Microsoft.ApplicationGateway-20240522134943'. The deployment status is marked as 'complete' with a green checkmark. Deployment details include:

- Deployment name: Microsoft.ApplicationGateway-20240522134943
- Subscription: Azure Pass - Sponsorship
- Resource group: new-grp
- Start time: 22/5/2024, 2:15:03 pm
- Correlation ID: 35774ebe-b965-4c0f-a5b8-c2f928fbdd6

Navigation links include 'Deployment details' and 'Next steps'. A blue button at the bottom right says 'Go to resource group'.

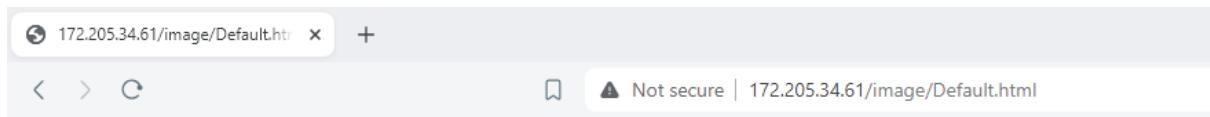
28. Then copy the public IP address from the dashboard of the application gateway.

The screenshot shows the Azure portal interface for the 'demo-gateway' application gateway. The top navigation bar includes 'Search', 'Delete', 'Refresh', and 'Feedback' buttons. Below the navigation is a sidebar with 'Overview' selected, followed by 'Activity log', 'Access control (IAM)', 'Tags', 'Diagnose and solve problems', 'Settings', 'Monitoring', 'Automation', and 'Help'. The main content area is titled 'Essentials' and displays the following details:

Resource group	(move) : new-grp	Virtual network/subnet	: Demo-VN/demosubnet
Location	: North Europe (Zone 1, 2, 3)	Frontend public IP address	: 172.205.34.61 (demo-ip)
Subscription	(move) : Azure Pass - Sponsorship	Frontend private IP addr...	: -
Subscription ID	: 3541d15a-44aa-4f6e-a120-1b7a6d5925bf	Tier	: Standard V2
Tags	Tags (edit) : Add tags	Availability zone	: 1, 2, 3

Below this, there is a 'Show data for last' dropdown with options: 1 hour, 6 hours, 12 hours, 1 day, 7 days, and 30 days. At the bottom of the essentials section are two buttons: 'Sum Total Requests' and 'Sum Failed Requests'.

29. Now if you append the IP with the image folder then you will see the image web page.



30. And if you append it with the video folder then it will show you the video web page.

