😊 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:

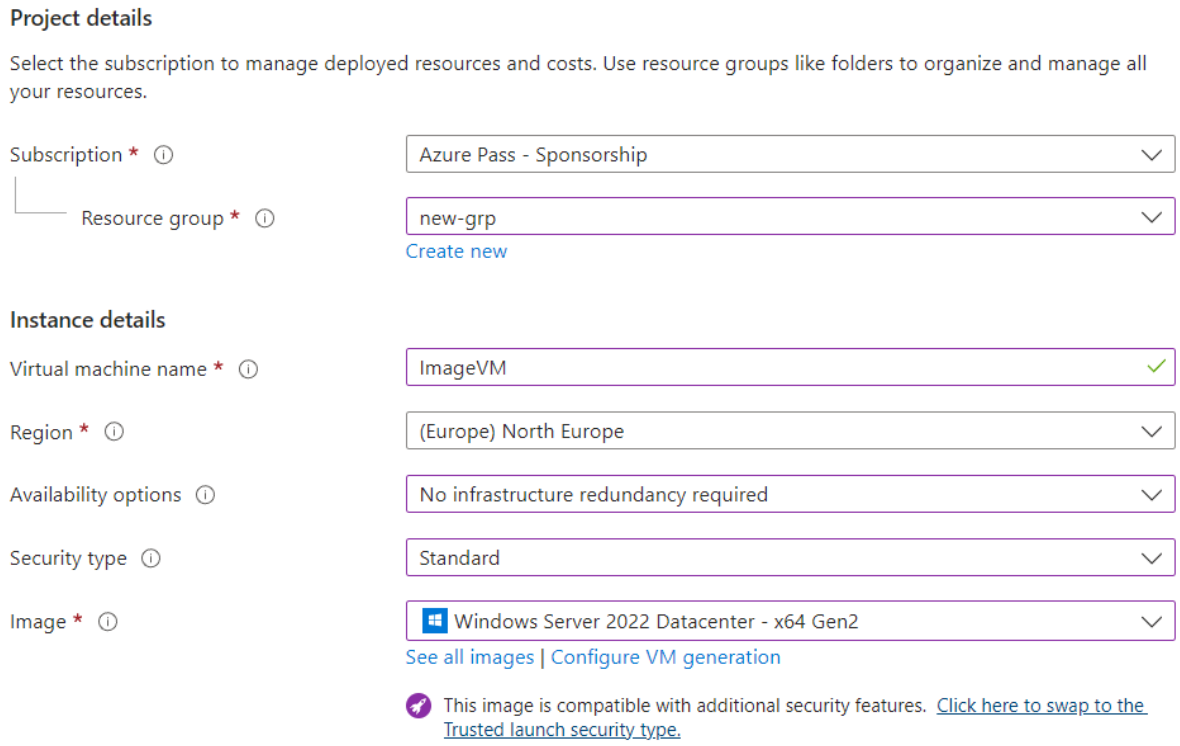
1. **Web Application Hosting:** One of the primary use cases for Azure Application Gateway is hosting web applications. Whether it's a simple website or a complex web application, Application Gateway can efficiently distribute incoming HTTP/HTTPS traffic to multiple backend servers, ensuring high availability 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.

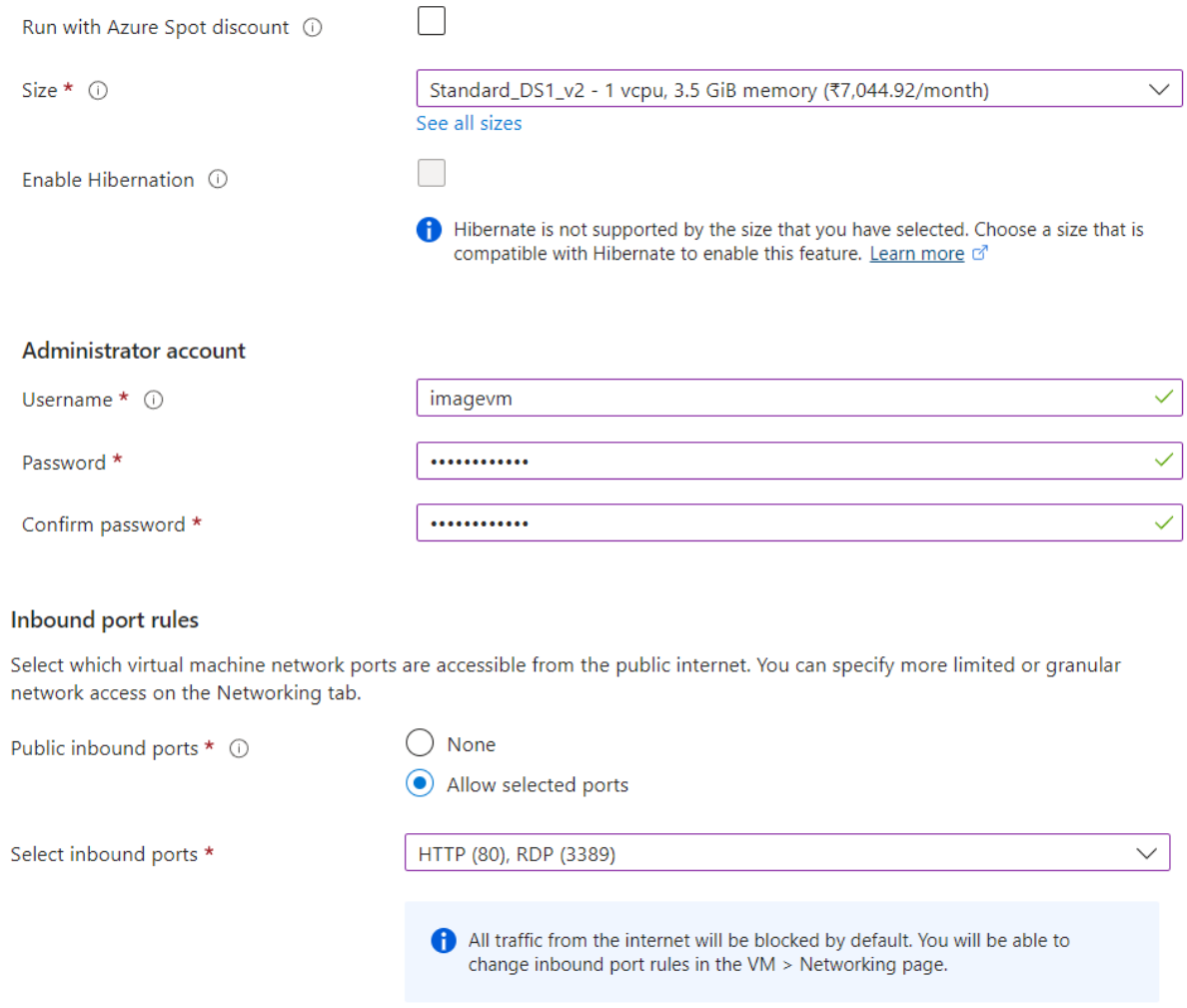
In this walkthrough, we are setting up Azure Application Gateway to efficiently route incoming web traffic to different backend servers based on URL paths. The end goal is to showcase how Application Gateway can be used to host multiple web applications or serve different types of content (such as images 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.

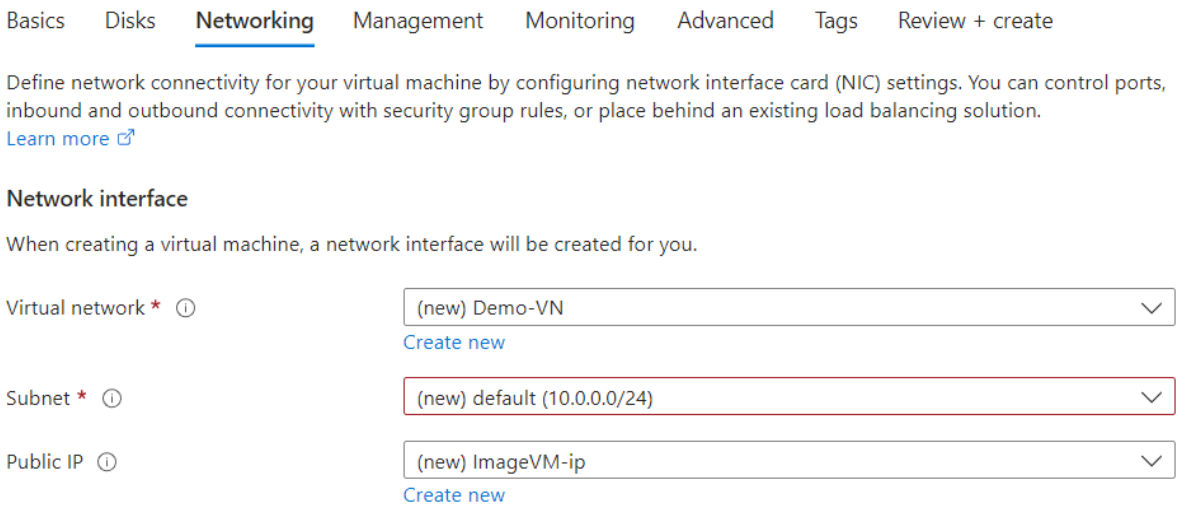
😄 To begin with the Lab:

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.



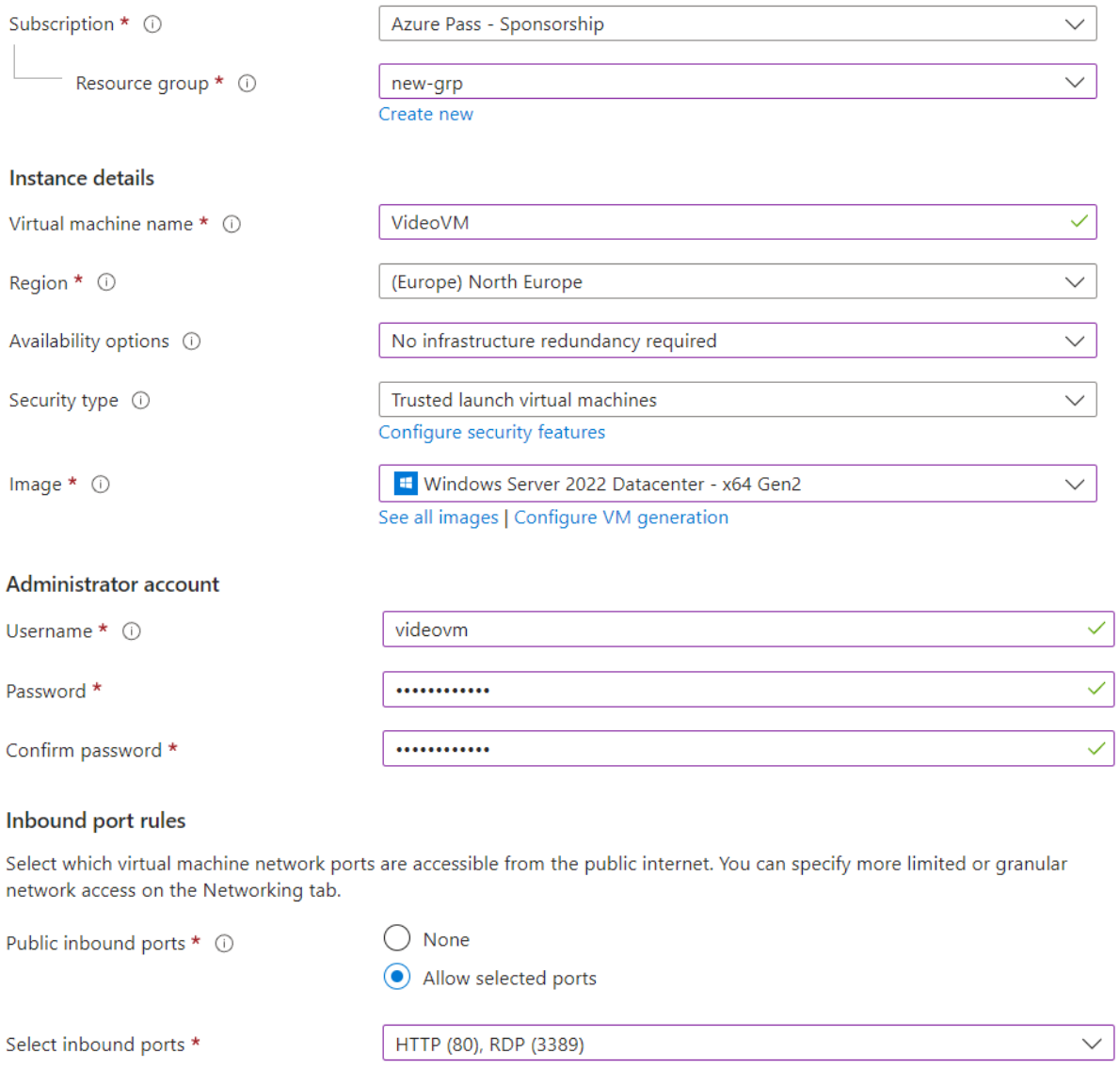


4. Then in the networking section for the Network Interface you have to create a new Virtual Network.

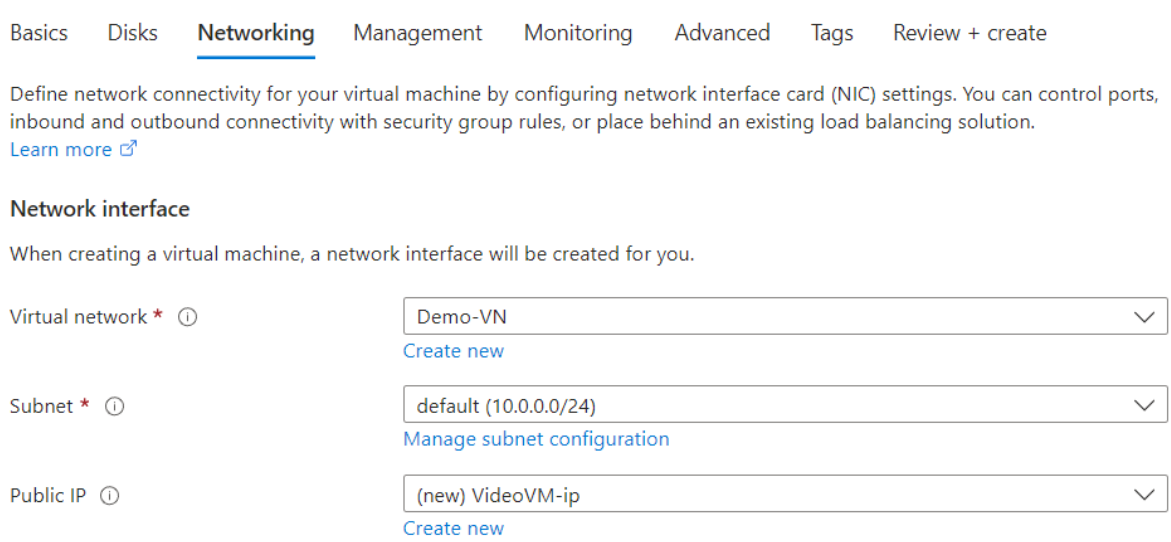


5. Then directly jump to the Review page and create your Virtual machine.

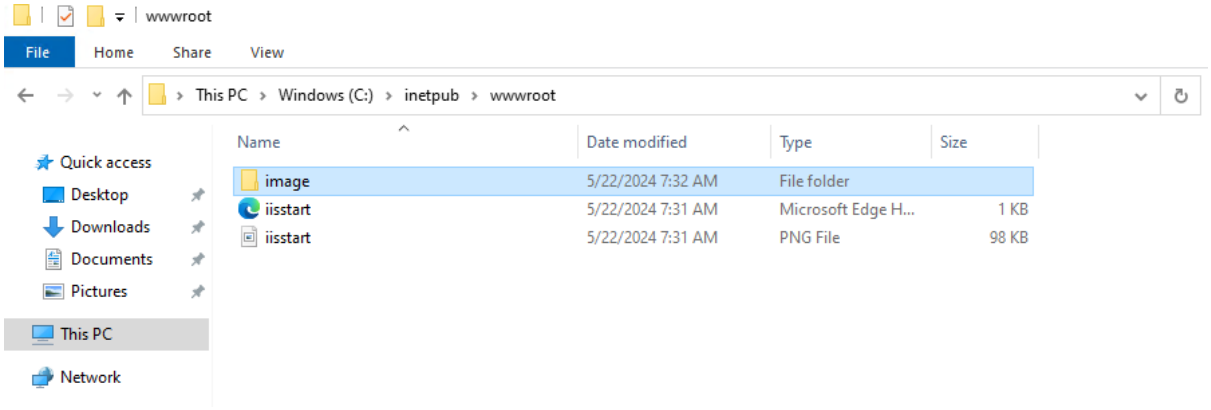
6. Now as your virtual machine is being deployed go ahead and launch your second virtual machine.



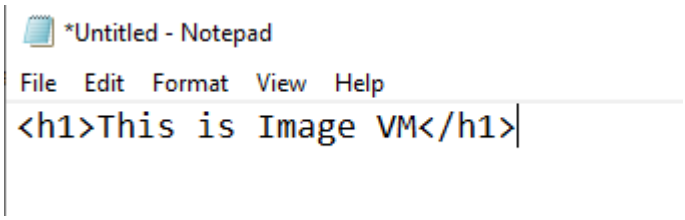
7. Just remember that the Virtual Network should be the same as before.

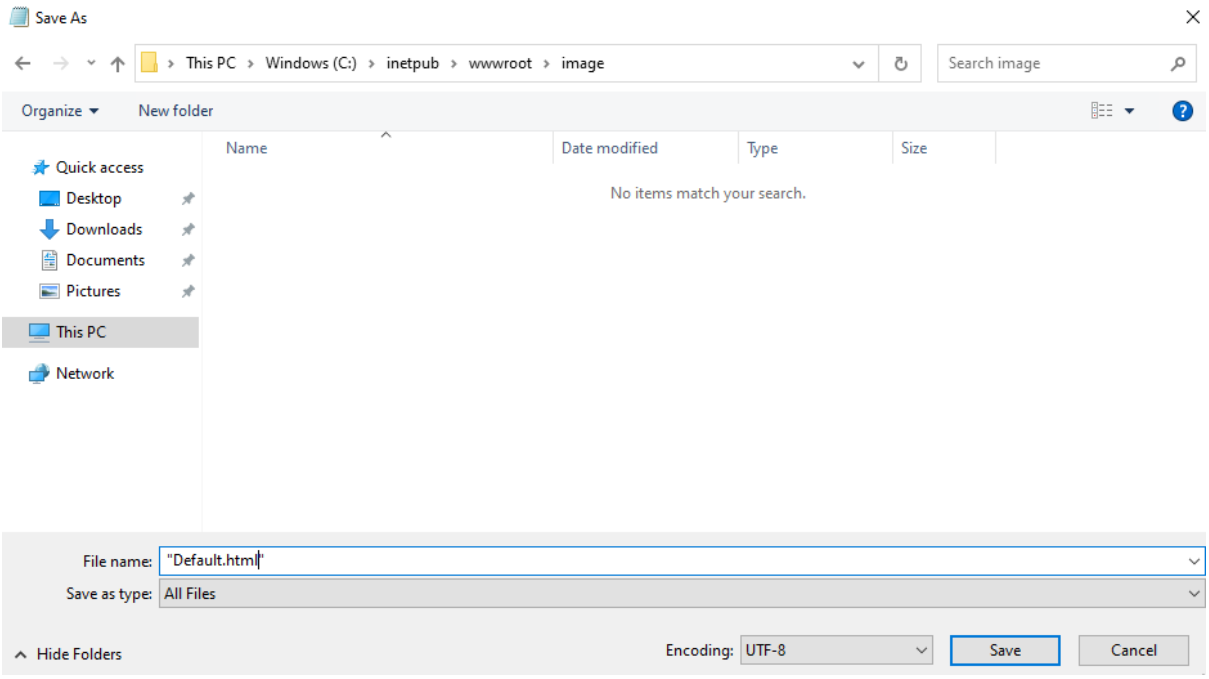


8. Then just move to the review page and create your VM.
9. Now you are going to login to your First VM which is your image VM and install Web Server in it.
10. Once your Web Server is installed then you need to go to this folder and create a folder with the image name in it.

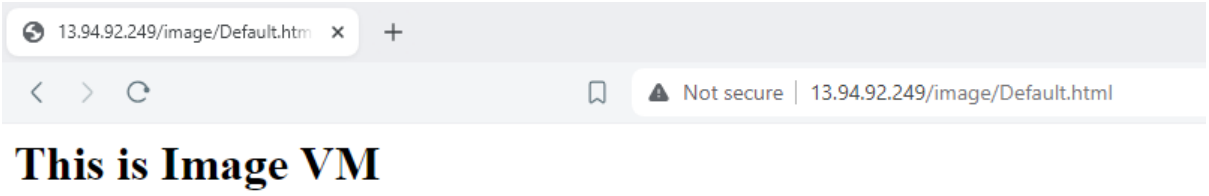


11. After that open the notepad and create a basic HTML page. Then save it to the image folder as shown below.

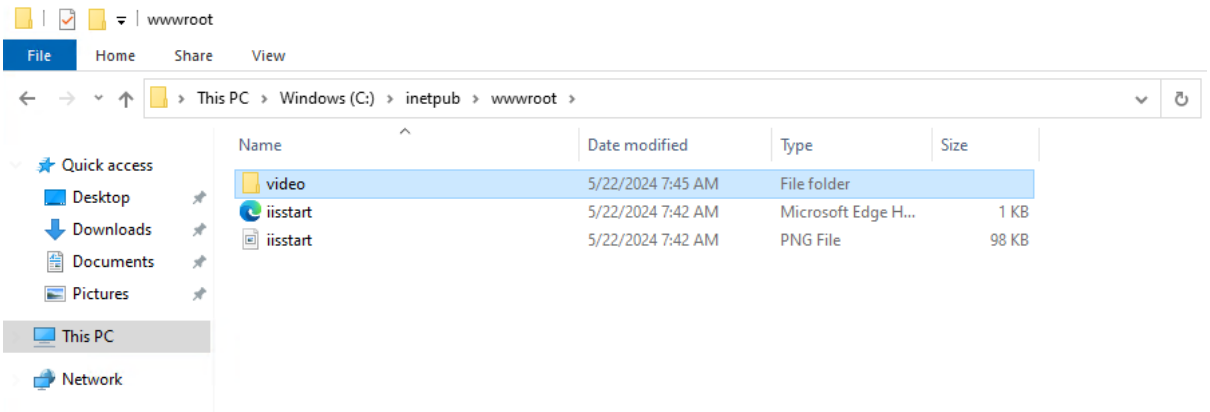




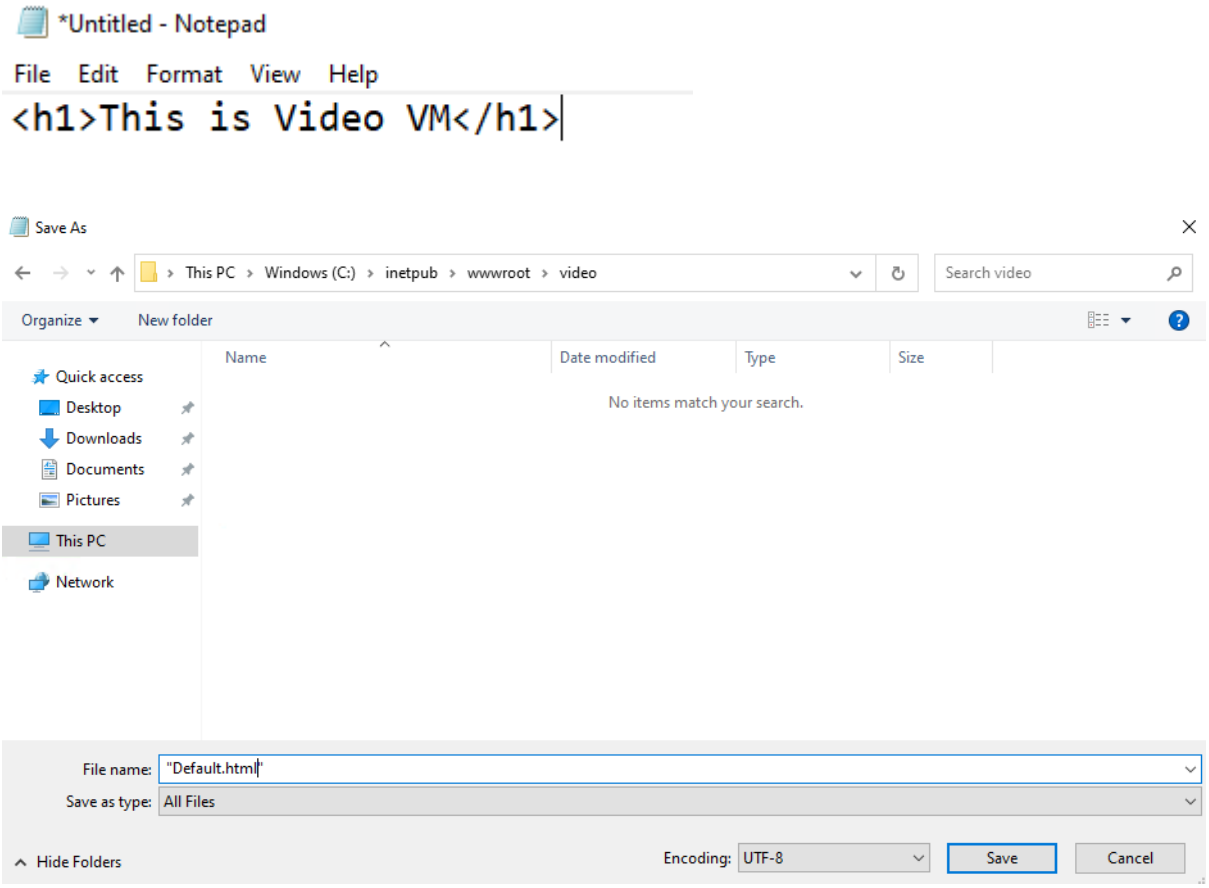
12. Also, if you try to access the web page for the Image VM and append it with the folder and file name then you will see the results as expected.



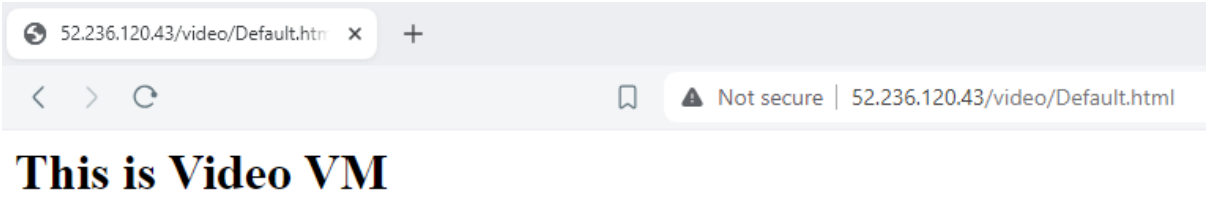
13. Now you have to do the same thing in your Video VM. Install the Web Server in it then create a video folder in this location shown below.



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



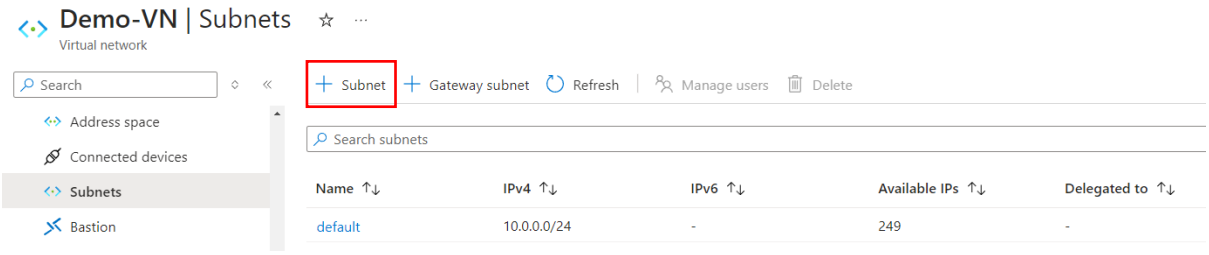
15. Also, if you try to access the web page for the Video VM and append it with the folder and file name then you will see the results as expected.



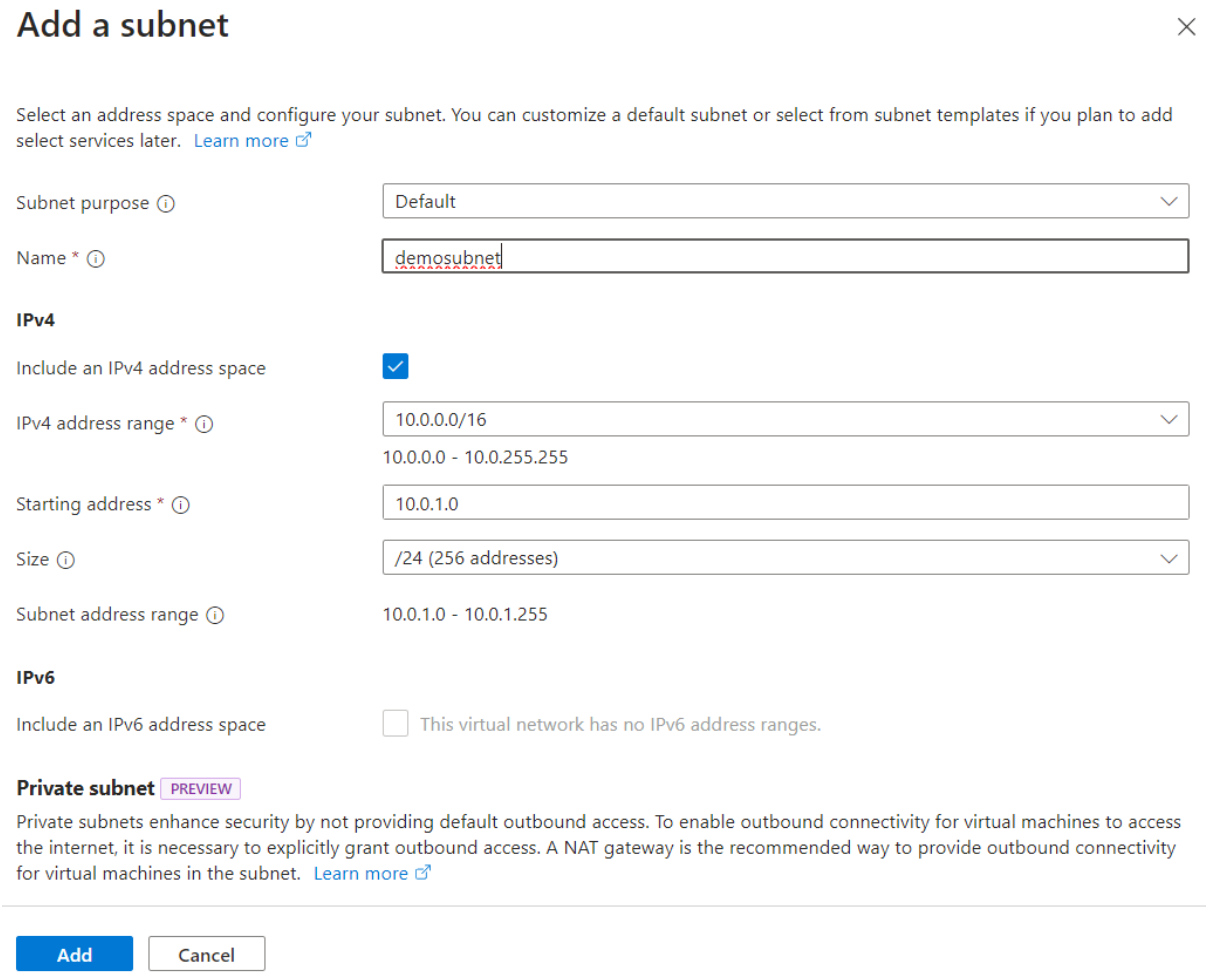
😊 URL Routing Implementation

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.

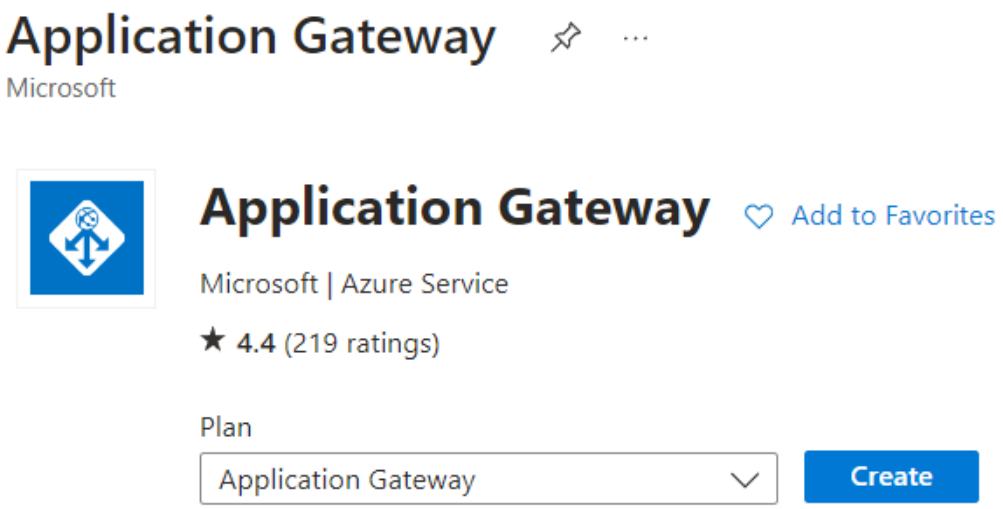
5. So, for this, it is going to have its own infrastructure. This is completely managed by the service itself. All you need to do is to ensure that you have an empty subnet in place in an Azure virtual network.
6. So, for any VM we will go onto the virtual network and from here go to subnets then click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



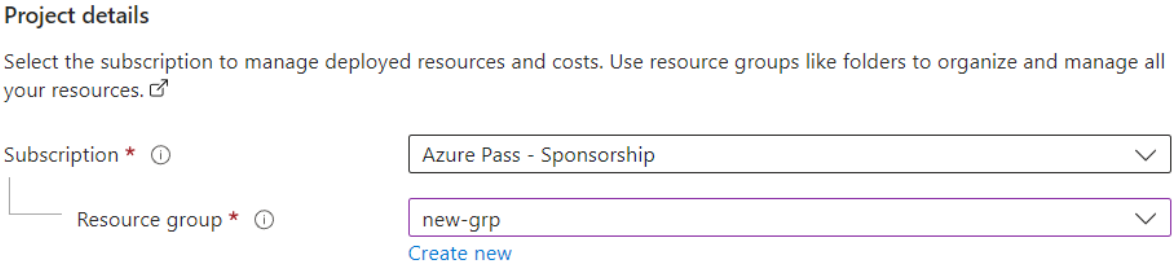
7. Here we just need to give it a name and we'll accept the subnet range it is providing us. Then just click on Add.



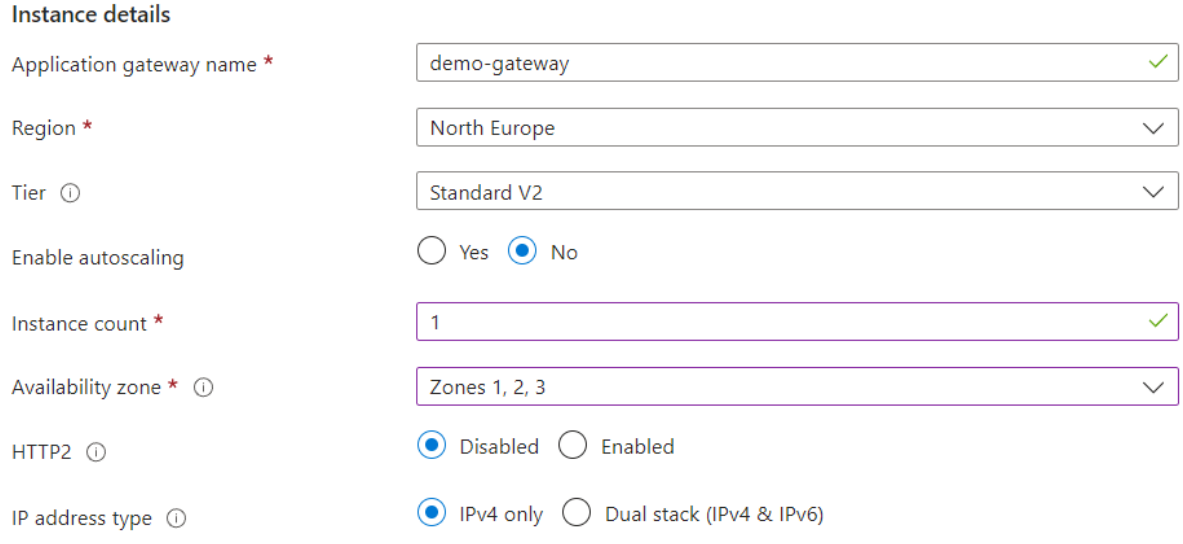
8. Now in the marketplace you need to search for application gateway and choose this service accordingly.



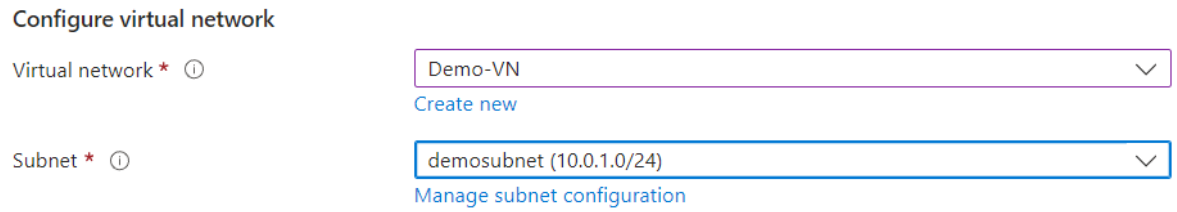
9. First you need to select your resource group.



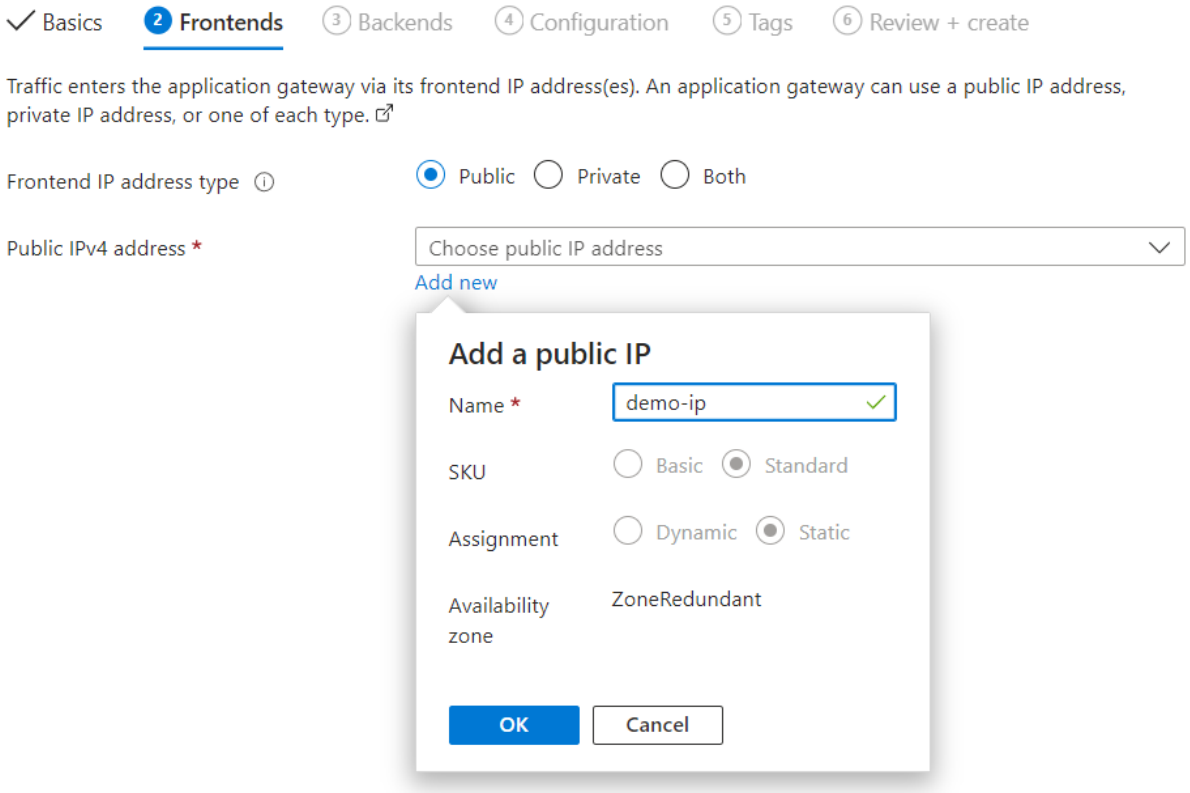
10. Then give a name to your application gateway and choose the tier as Standard V2. After that to enable autoscaling say NO.
11. Then for instance count keep it to 1. For the availability zones choose all 3. Then disable HTTP2.



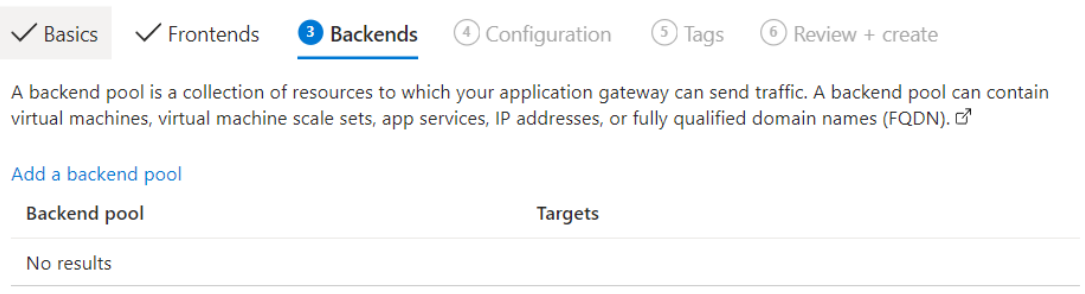
12. Then for configuring Virtual Network choose your virtual network and the empty subnet. If you try to choose the subnet which has our VMs then you will instantly get an error.



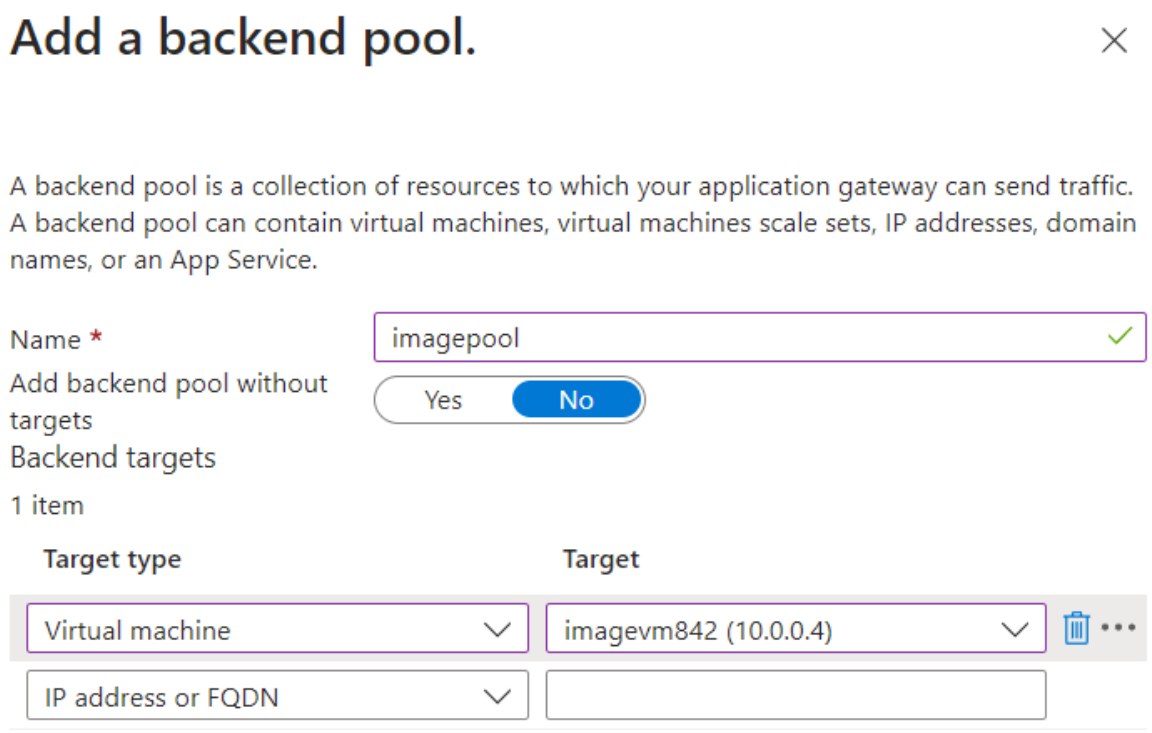
13. Then for the frontend IP you have to create a new Public Address.



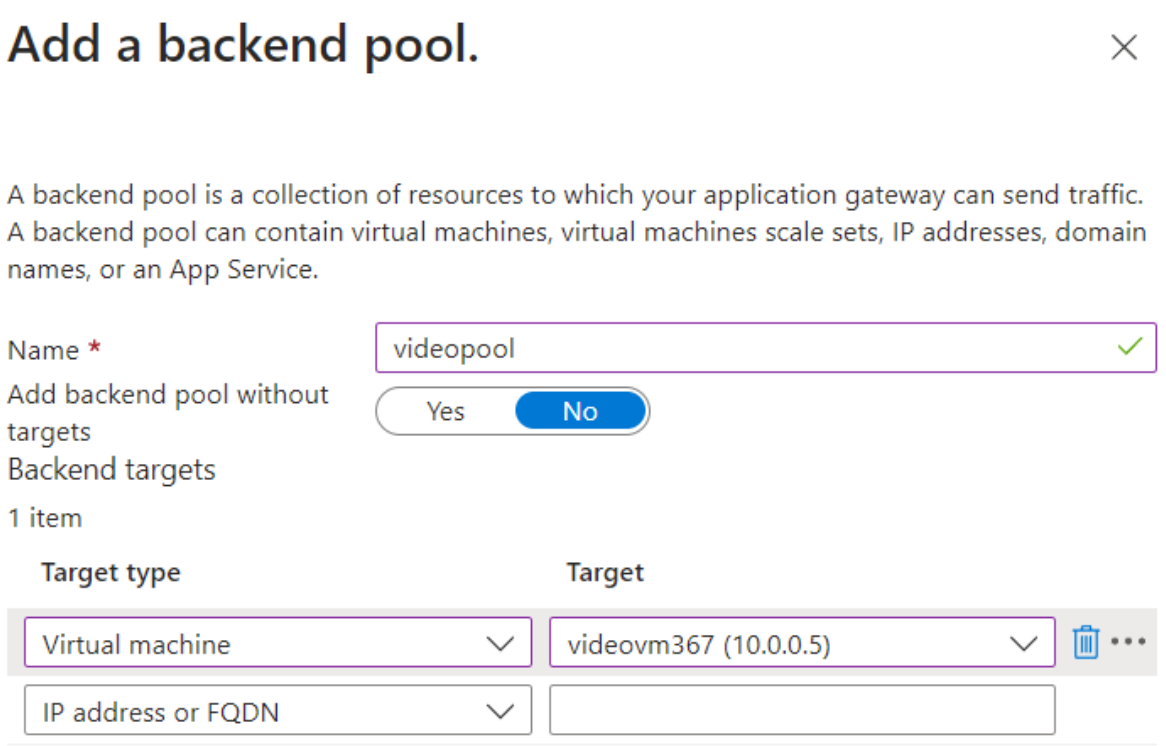
14. After that for the backend pool you have to you have to click on add and you are going to add both of your VMs separately. Which means that you have to create two Pools.



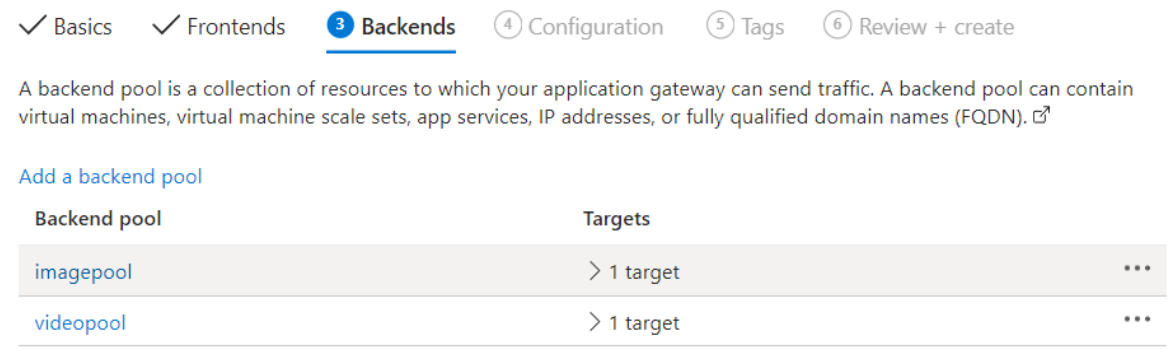
15. Below you can see that first we created a pool for our image VM and chose the properties for it as shown below.



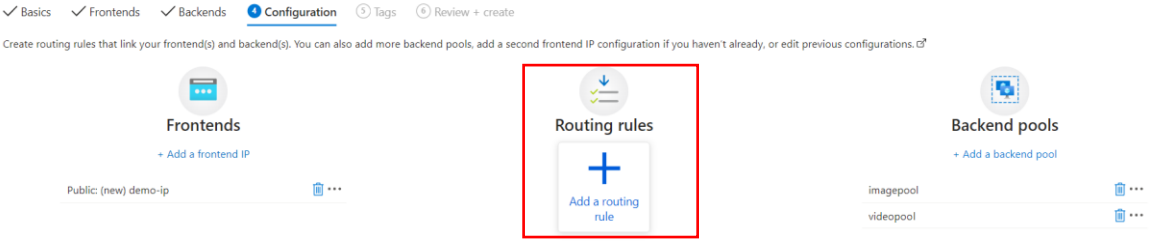
16. Then created a pool for our Video VM.



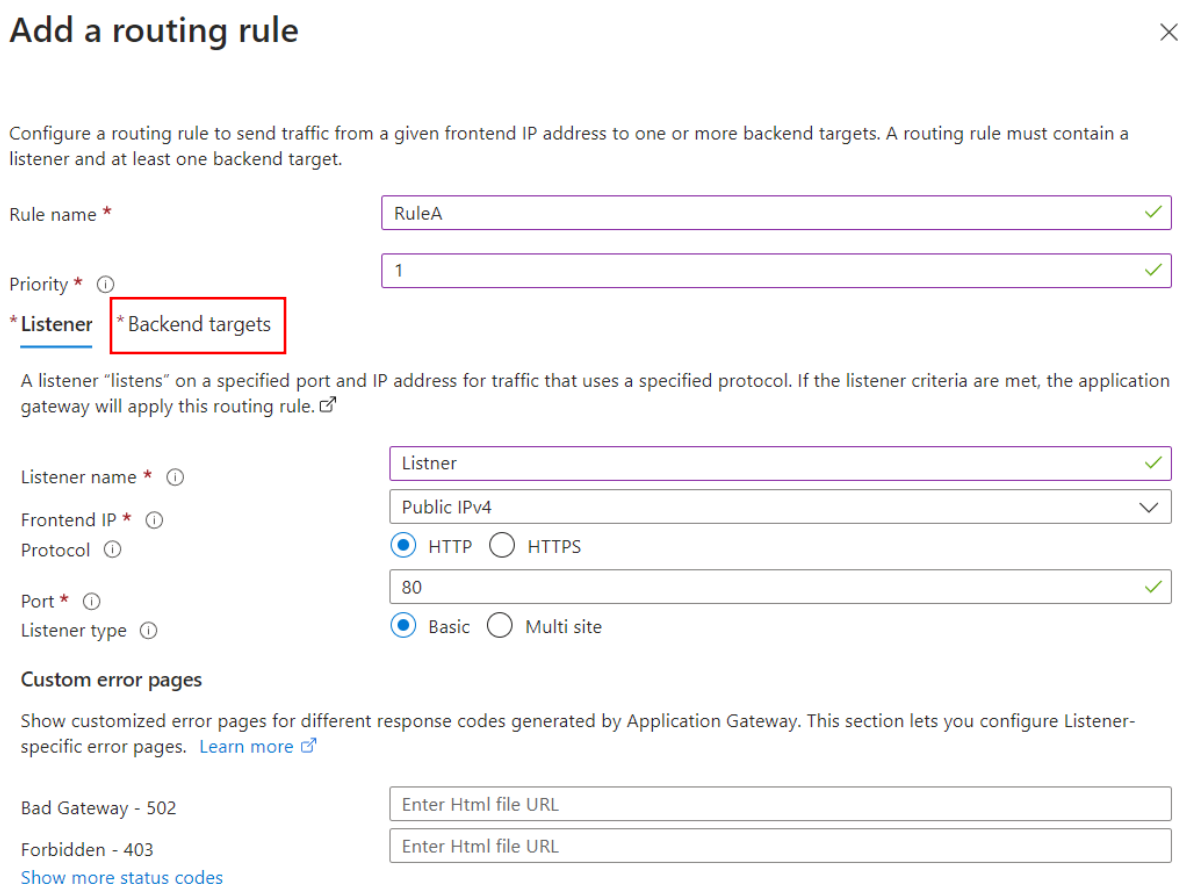
17. Below you can see that we added both of our VMs in backend pool separately.



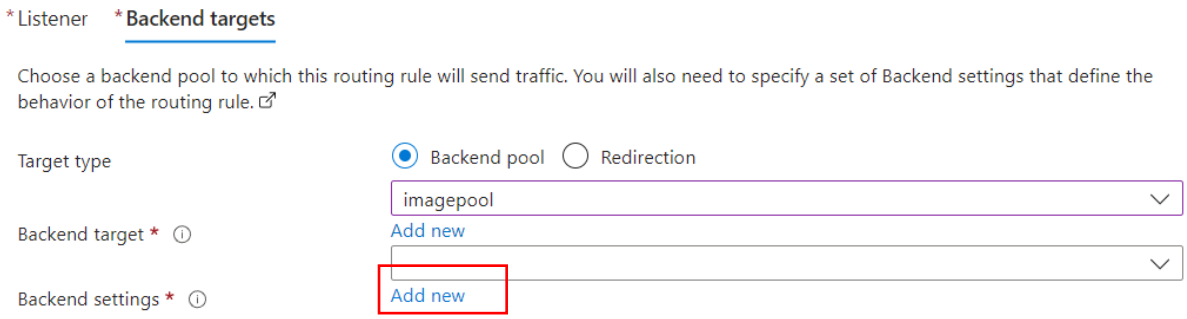
18. Then in configuration you can see that we have our frontend and backend ready. Now we need to create routing rules for our VM. For that click on add routing rule.



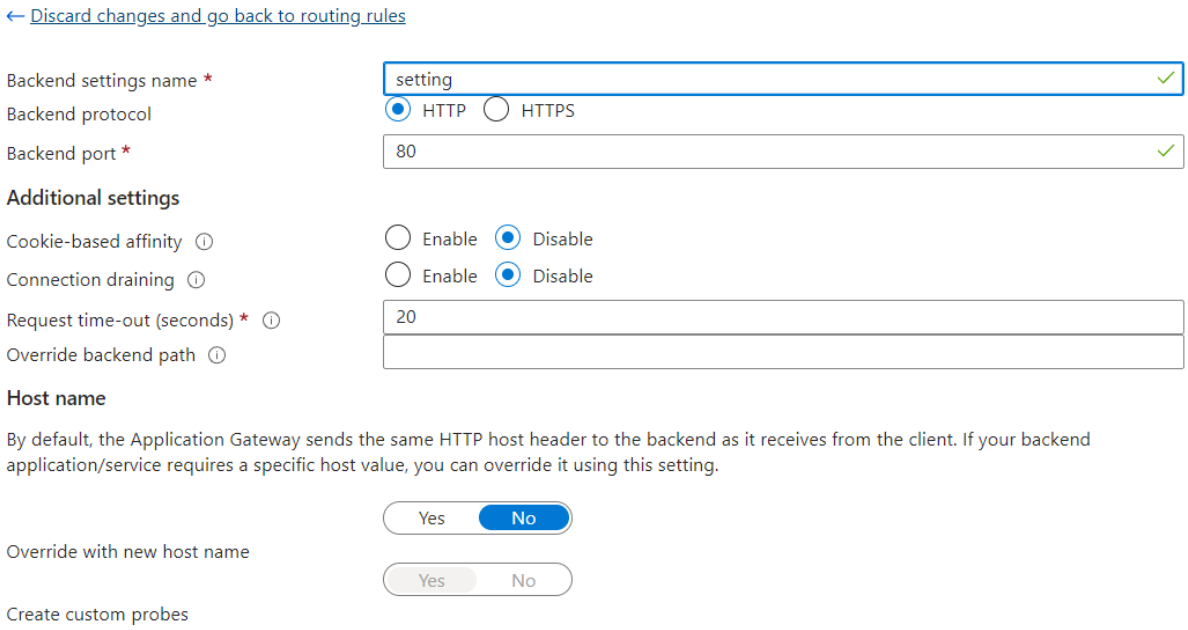
19. Here you can see that first we need to give our rule a name then we need to set the priority. After that we need to give a listener name and then move to backend targets.



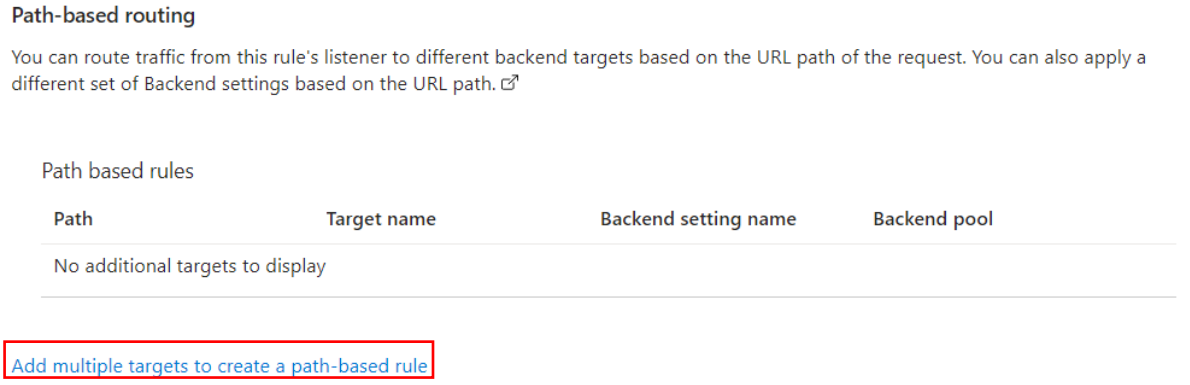
20. Then we need to choose our backend target which will be our image pool and then click on add new for backend settings.



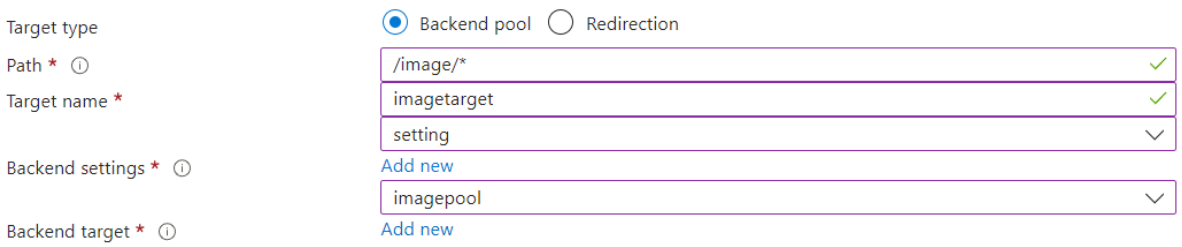
21. For the backend settings you just need to give it a name and leave the rest of the settings as they are and click on save.



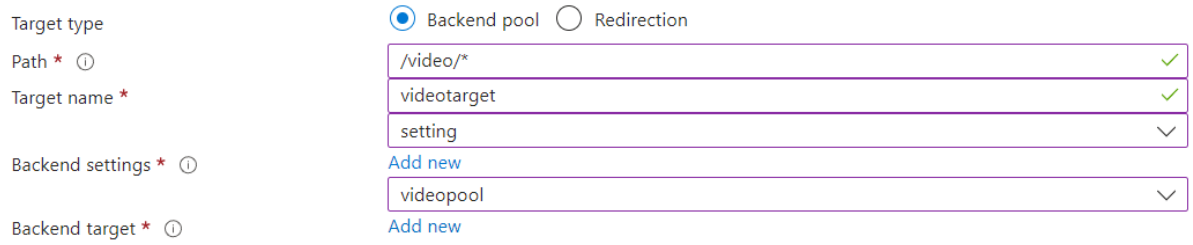
22. After that scroll down to backend targets and we need to add our path-based routing. We need to click on add multiple targets.



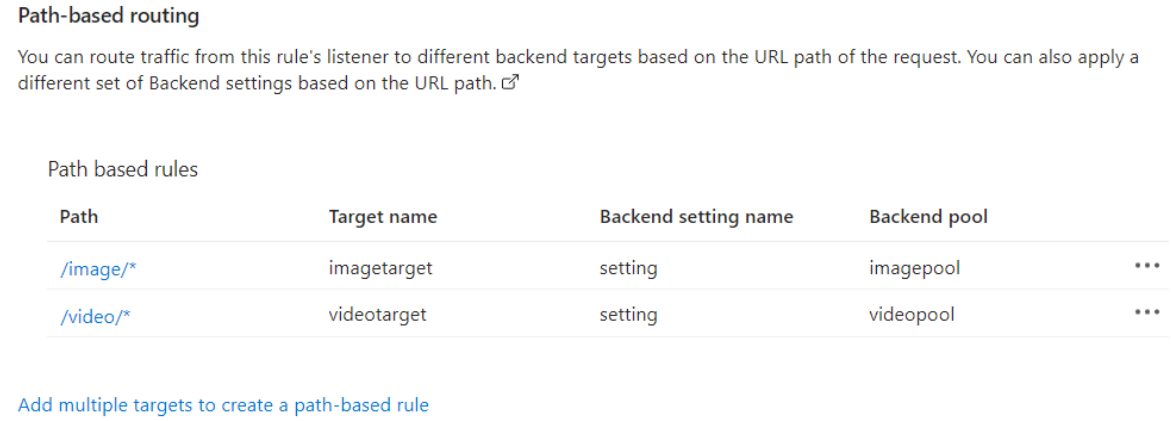
23. Here you need to give the path for your image folder as shown below. Just remember in the path the name should be same as you defined in your virtual machine while creating it. Then you need to give a target name, after that you have to choose the backend setting and the backend target for you is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



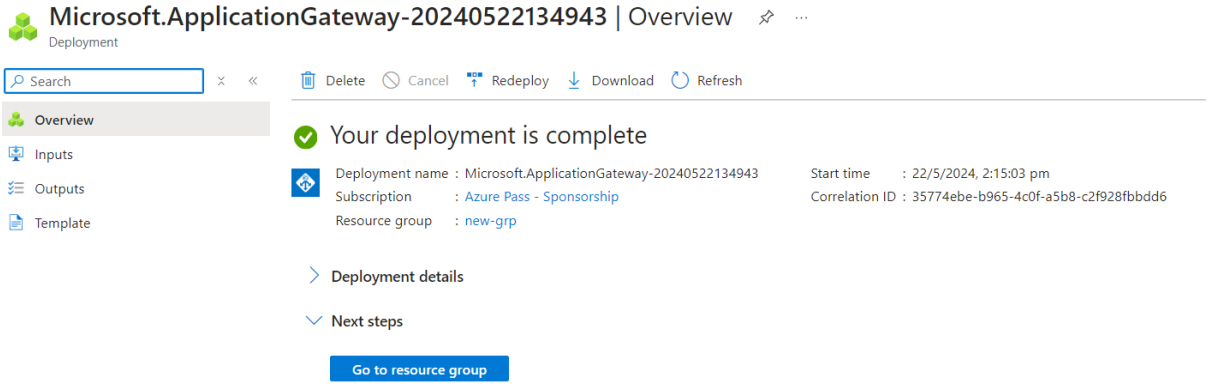
24. Then you are going to do the same for video and click on save.



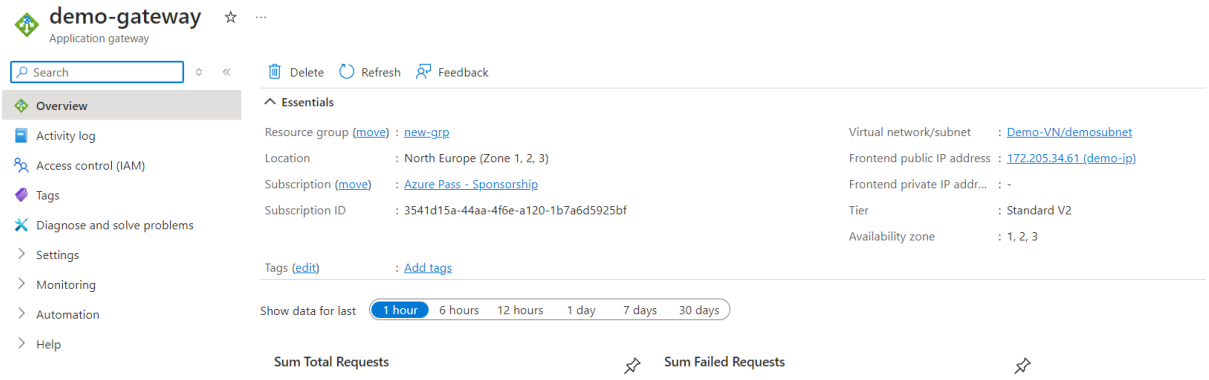
25. Then in the path-based routing you can see both of the paths.



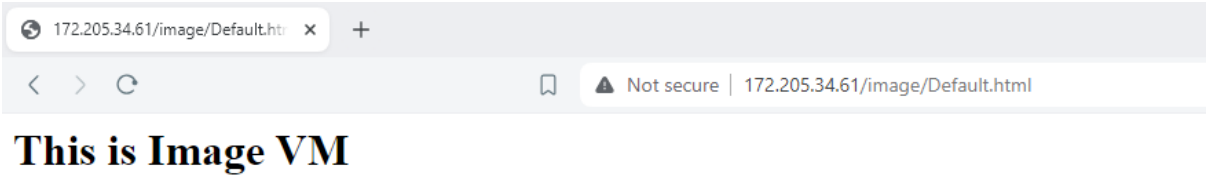
26. After that just click on ADD and your rule will be saved then you need to go to the review page and create your Application Gateway.
27. Now this might take around 4-5 mins. Once your deployment is complete go to the resource group and choose your application gateway.



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

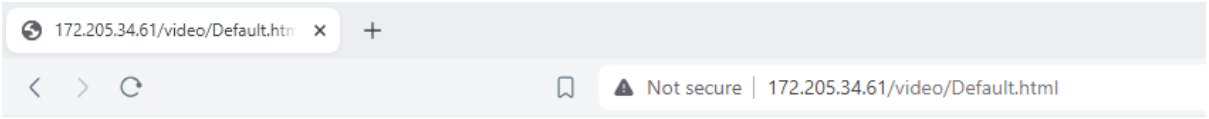


29. Now if you append the IP with the image folder then you will see the image web page.



This is Image VM

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



This is Video VM