😊 Azure Basic Load Balancer

Azure Load Balancer is a networking service provided by Microsoft Azure that distributes incoming network traffic across multiple servers or virtual machines (VMs) to ensure the high availability and reliability of applications and services. It operates at the Transport layer (Layer 4) of the OSI model and can balance both TCP and UDP traffic.

Here are some key features and functionalities of Azure Load Balancer:

1. **Traffic Distribution:** Azure Load Balancer distributes incoming traffic evenly among healthy instances of VMs or services based on configured load balancing rules.
2. **High Availability:** It enhances the availability of applications by providing redundancy and failover capabilities. If one instance or VM becomes unavailable, traffic is automatically rerouted to healthy instances.
3. **Health Probes:** Azure Load Balancer continuously monitors the health of backend instances by sending health probes. If an instance fails health checks, it is automatically removed from the load-balanced pool.
4. **Backend Pool Configuration:** You can define backend pools, which are groups of VMs or services that will receive traffic from the load balancer.
5. **Session Persistence:** Azure Load Balancer supports session persistence, ensuring that all requests from the same client are sent to the same backend instance.
6. **Public and Private Load Balancers:** You can deploy Azure Load Balancers as either public or internal load balancers, depending on whether they are accessible from the internet or limited to internal network traffic.
7. **Integration with Azure Services:** Azure Load Balancer seamlessly integrates with other Azure services such as Virtual Machines, Virtual Machine Scale Sets, Azure Kubernetes Service (AKS), and Azure Virtual Network Gateways.
8. **Security:** It offers security features such as Network Security Groups (NSGs) to control traffic flow and access to backend instances.

Overall, Azure Load Balancer plays a crucial role in ensuring the scalability, reliability, and performance of applications and services hosted on the Azure cloud platform.

😄 Use cases of Load Balancer:

Azure Load Balancer serves various use cases across different scenarios and industries. Here are some common ones:

1. **Web Applications:** Load balancing web traffic across multiple instances of web servers ensures high availability and scalability for web applications. Whether it's a simple website or a complex web application, Azure Load Balancer can distribute incoming traffic effectively.
2. **Microservices Architecture:** In a microservices architecture, different components of an application are deployed as separate services. Azure Load Balancer helps in distributing traffic across these microservices, ensuring each component is responsive and scalable.

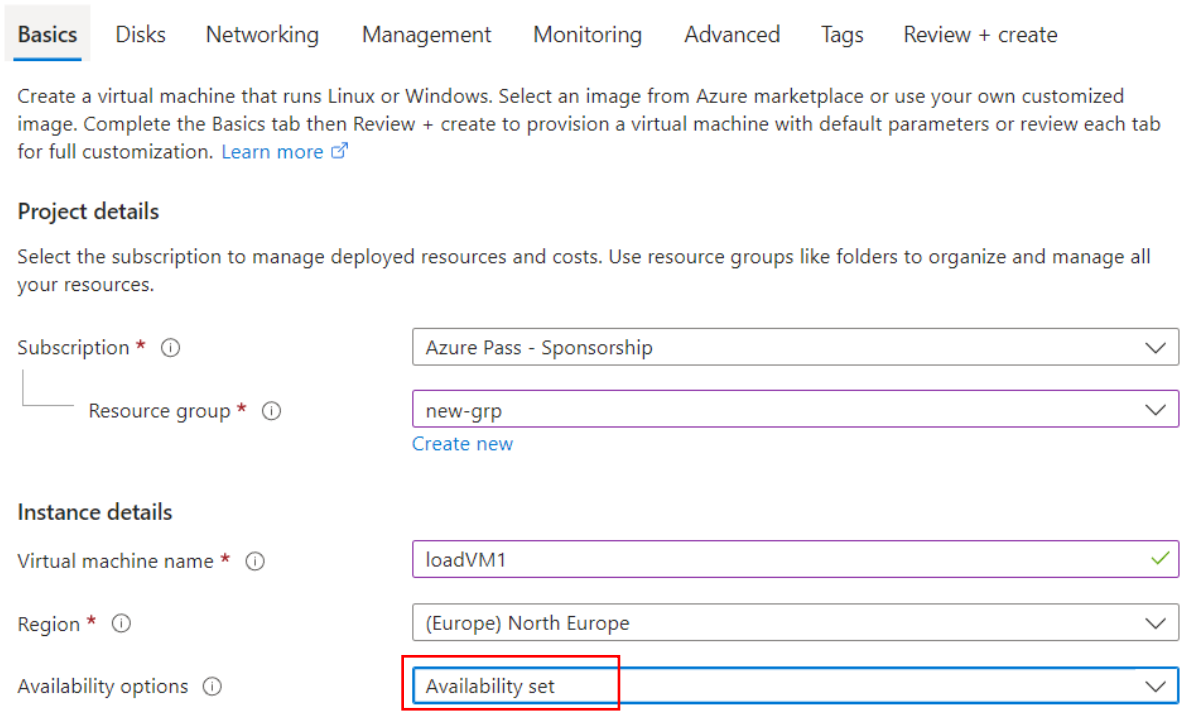
3. **Highly Available Services:** Critical services such as databases, messaging queues, and APIs can be made highly available by distributing traffic across multiple instances or replicas. Azure Load Balancer ensures that these services remain accessible even if some instances fail.
4. **Hybrid Cloud Scenarios:** In hybrid cloud setups where applications span across on-premises data centers and Azure, Azure Load Balancer facilitates seamless traffic distribution between on-premises and cloud-based resources.
5. **Disaster Recovery:** Load balancing traffic between primary and secondary data centers or between regions helps in disaster recovery scenarios. In case of a failure in one location, Azure Load Balancer can automatically redirect traffic to the secondary location.
6. **Internet-Facing Applications:** Public-facing applications such as e-commerce websites, gaming platforms, and media streaming services benefit from Azure Load Balancer to handle incoming internet traffic efficiently and ensure a smooth user experience.
7. **Internal Applications:** Internal line-of-business applications, intranet portals, and backend services within an organization can leverage Azure Load Balancer as an internal load balancer to distribute traffic across backend servers or services.
8. **DevOps and Continuous Integration/Continuous Deployment (CI/CD):** Azure Load Balancer can be integrated into DevOps pipelines to automate the deployment and scaling of applications. It ensures that newly deployed instances are seamlessly added to the load-balanced pool.
9. **Testing and Staging Environments:** Load balancing traffic in testing and staging environments helps in simulating real-world conditions and ensures that applications perform optimally before being deployed to production.

In this setup, we're configuring Azure Load Balancer to evenly distribute incoming web traffic across two virtual machines running Windows Server 2022 with Internet Information Services (IIS) installed. The end goal is to ensure high availability and scalability for web applications by automatically routing traffic to healthy instances and providing redundancy in case of failures. This setup enhances application reliability and performance by distributing the workload across multiple servers.

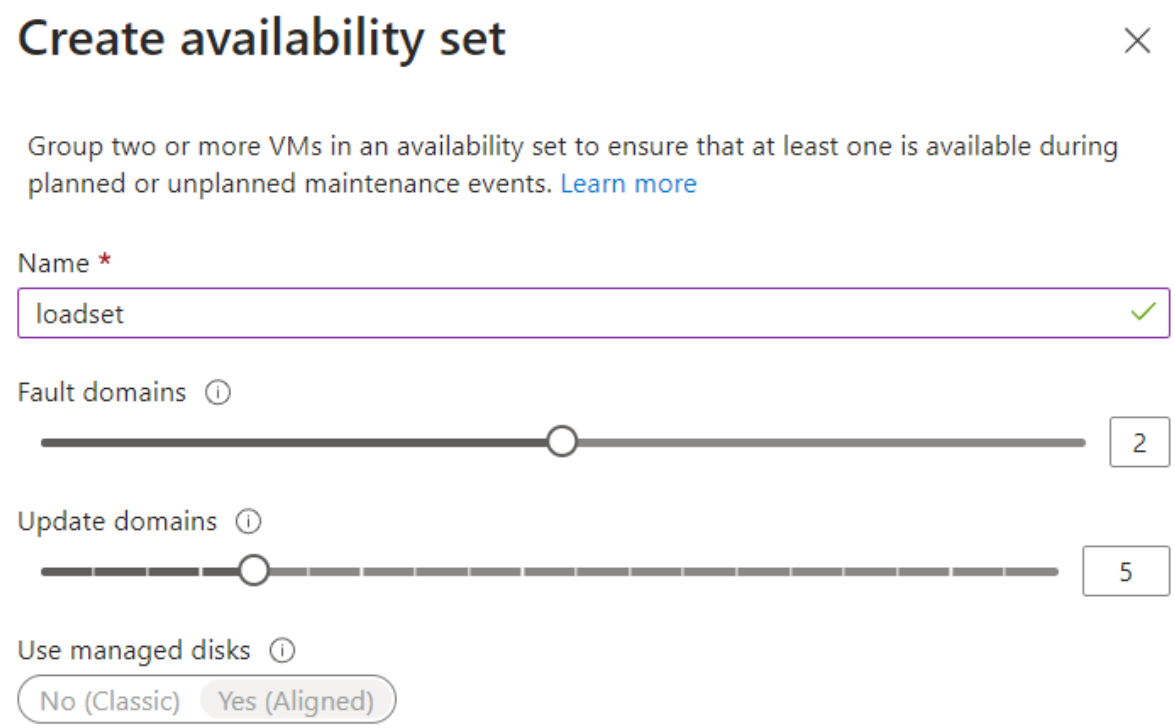
😄 To begin with the Lab:

1. Now we're going to move on to the implementation of the load balancer. Before that, we are going to have a simple setup in place.
2. We are first going to look at the load balancer when it comes to the basic SKU.
3. For this, we are going to create two virtual machines based on Windows Server 2022. We are going to make both machines part of an availability set, that'll be part of a virtual network.
4. We'll install internet information services on both of the machines. We'll have sort of a default HTML page in place.
5. Now in your Azure Portal navigate to Virtual Machine and create a new virtual machine. Here you need to choose a resource group and then give your VM a name

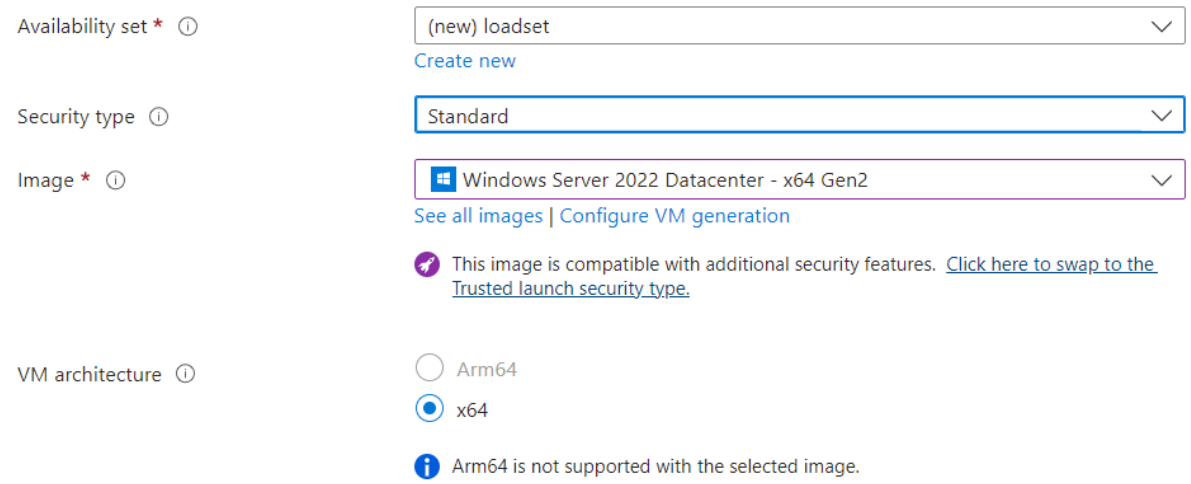
and then choose your region. After that in the availability options, you have to choose Availability Set.



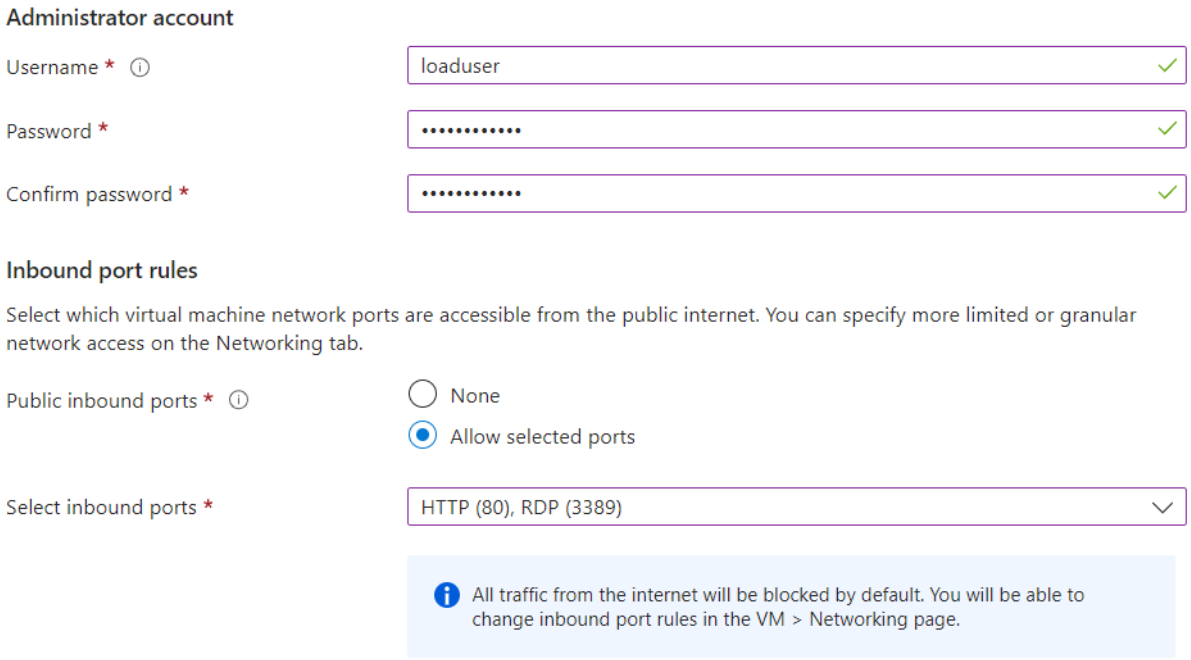
6. Then you have to create your availability set keep things to default and you have to give it name then just create it.



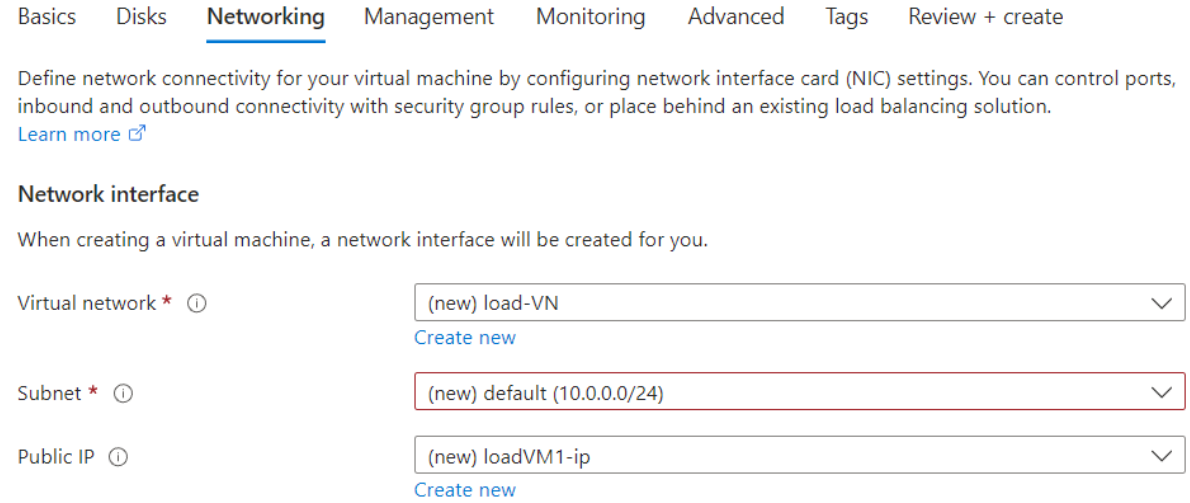
7. Then you have to choose your Image as Windows server 2022 datacenter.



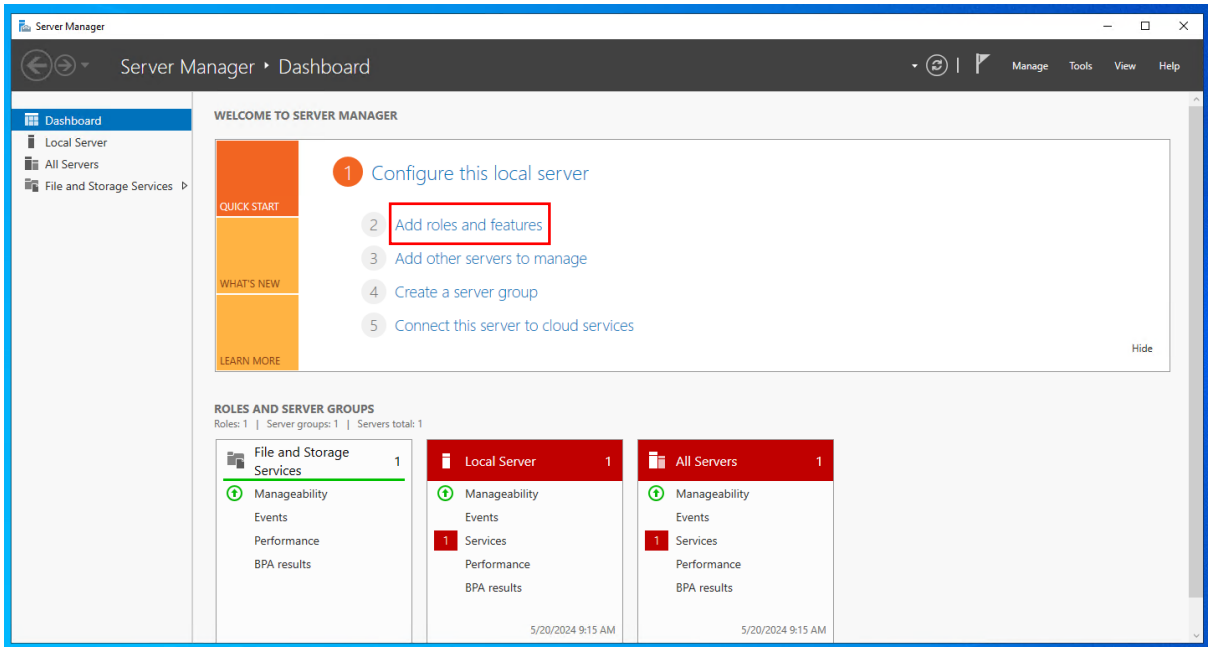
8. Now give a username and password then in the inbound rules you have to choose HTTP as your inbound rule.



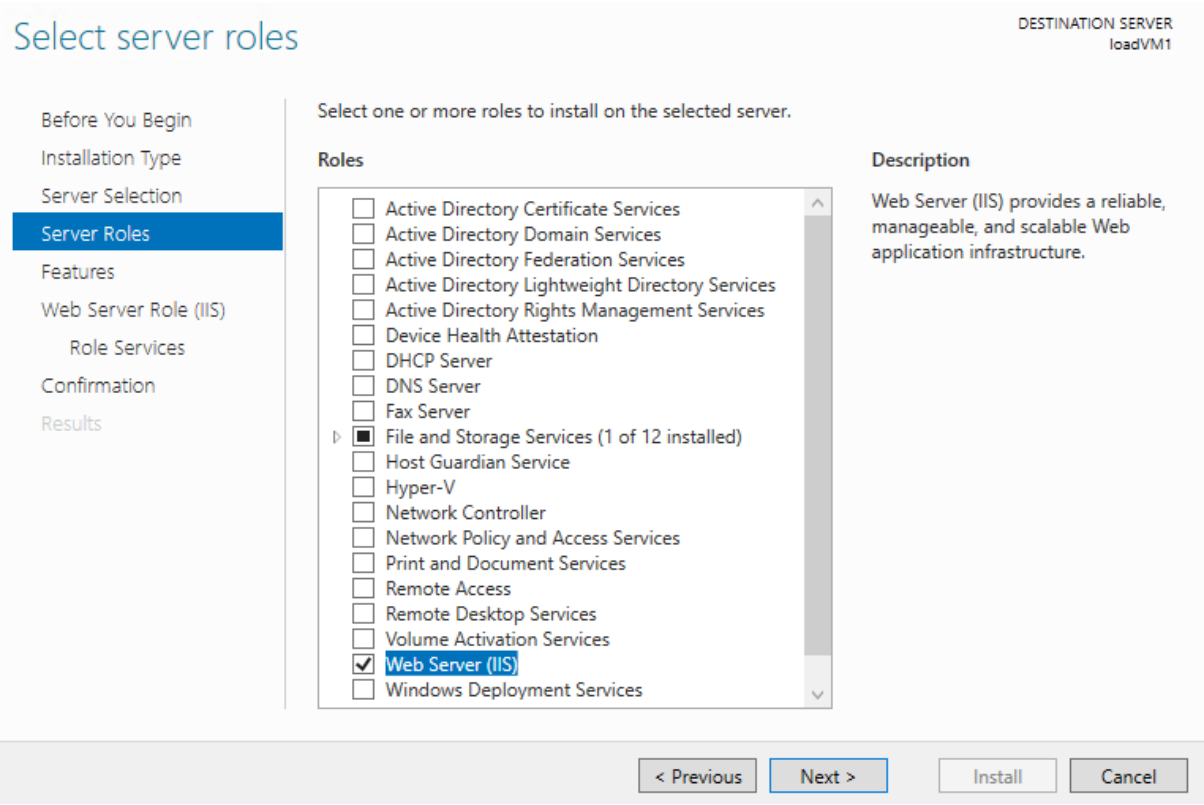
9. Now in the Networking tab you have to create a new Virtual Network, for that click on new and just change the name of your VN.



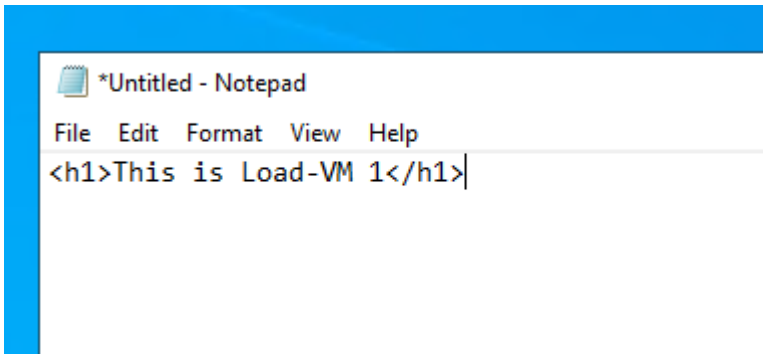
10. After that go ahead and create your Virtual Machine. Once your resources are deployed then you need to RDP into your VM.
11. Now in your VM you have to install IIS. For that you need to click on Add roles and features in your Server Manager.



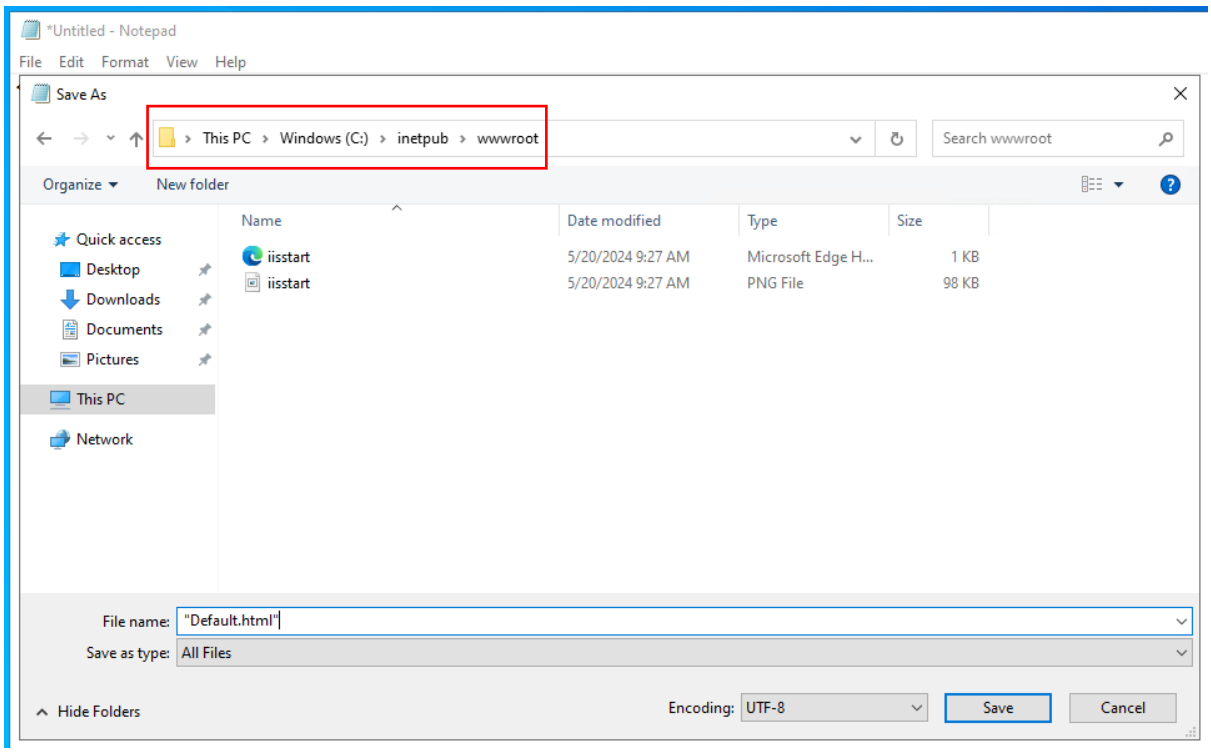
12. Then you have to click on next until you reach to Server Roles and from here you have to add Web Server (IIS) as shown in the snapshot.
13. After that move to the installation and install it.



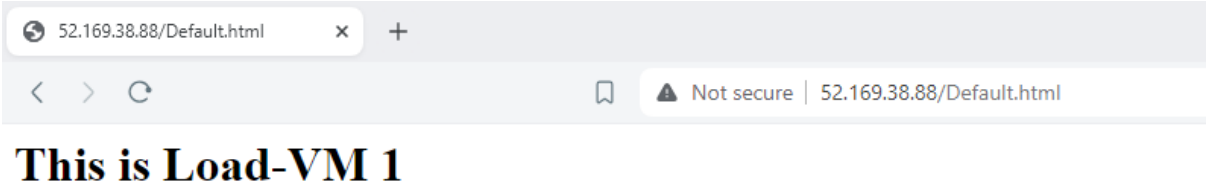
14. Once the installation is completed then you have to open the Note Pad in your VM and you have to write this.



15. And you have to save this file in this highlighted path as shown in the snapshot.

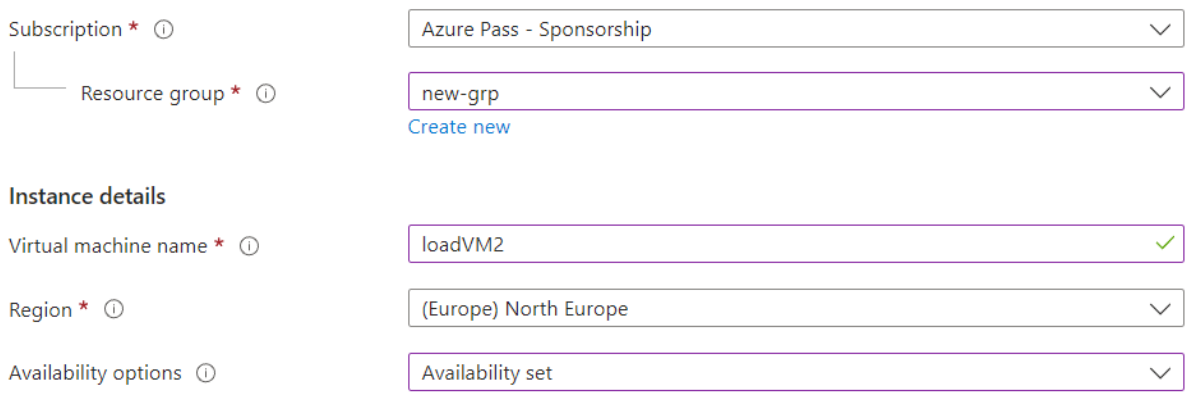


16. Now to see that it is working as expected you have to copy the Public IP address of your VM and paste it in a new tab and append it with the file name which you saved in your VM. You will see that result as expect.

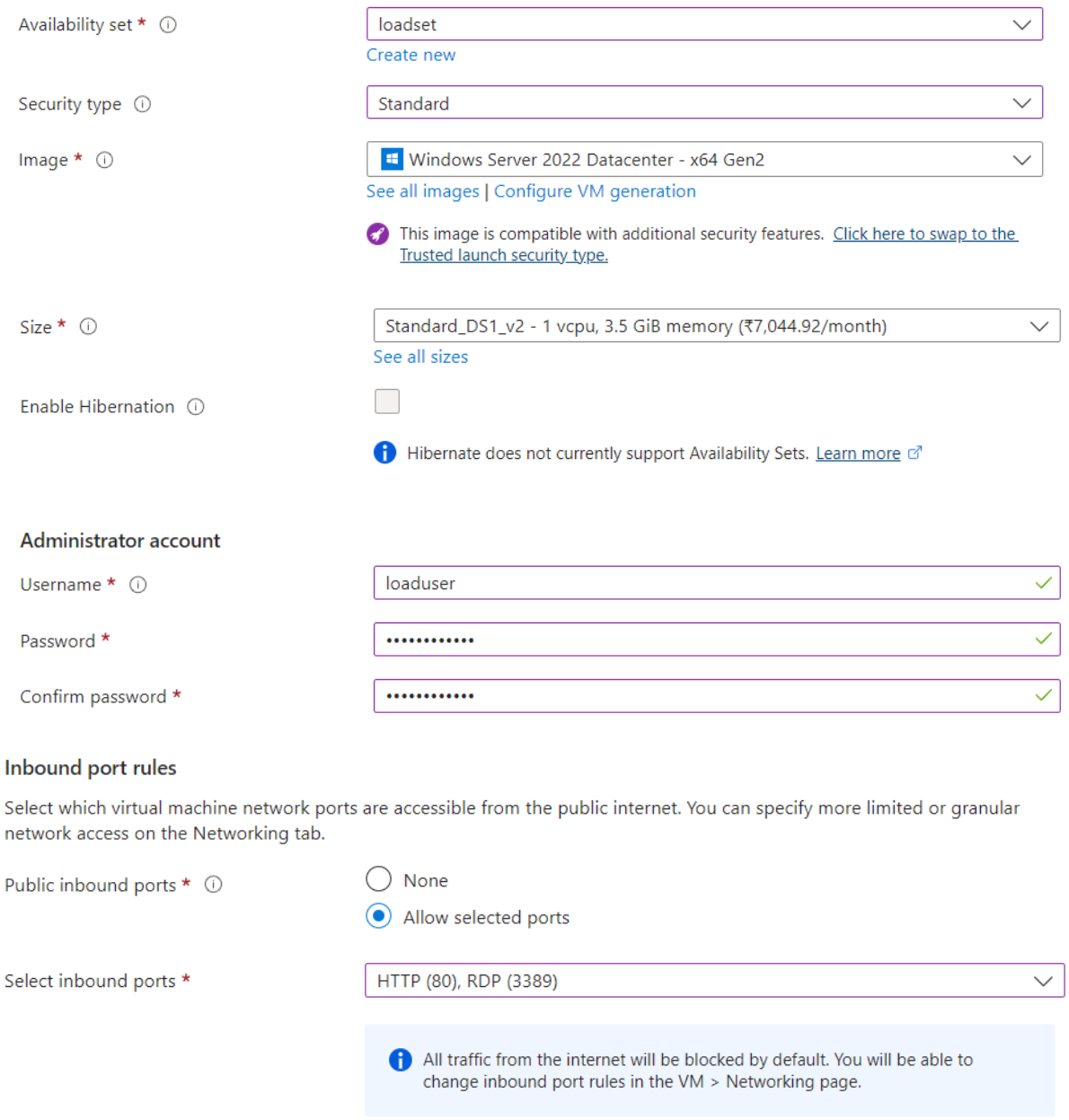


This is Load-VM 1

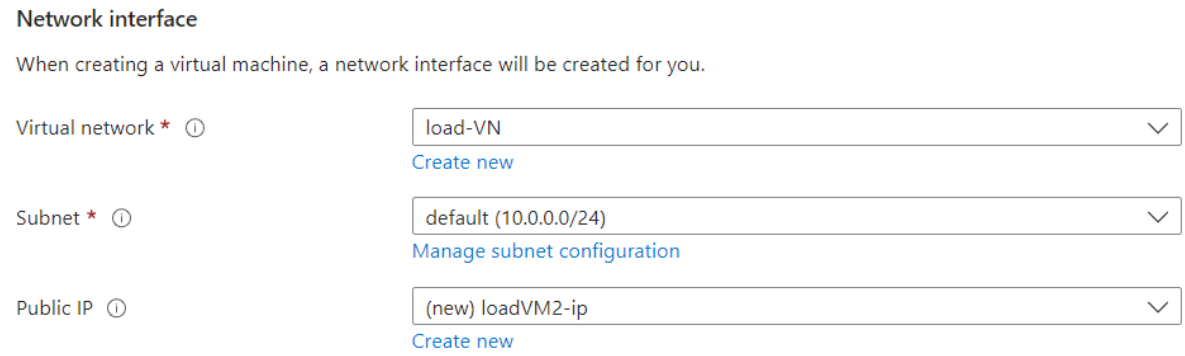
17. After that you are going to create a new Windows VM and do the same thing as you did in the first VM.
18. Again choose your resource group then give your VM a name then choose availability set in availability options.



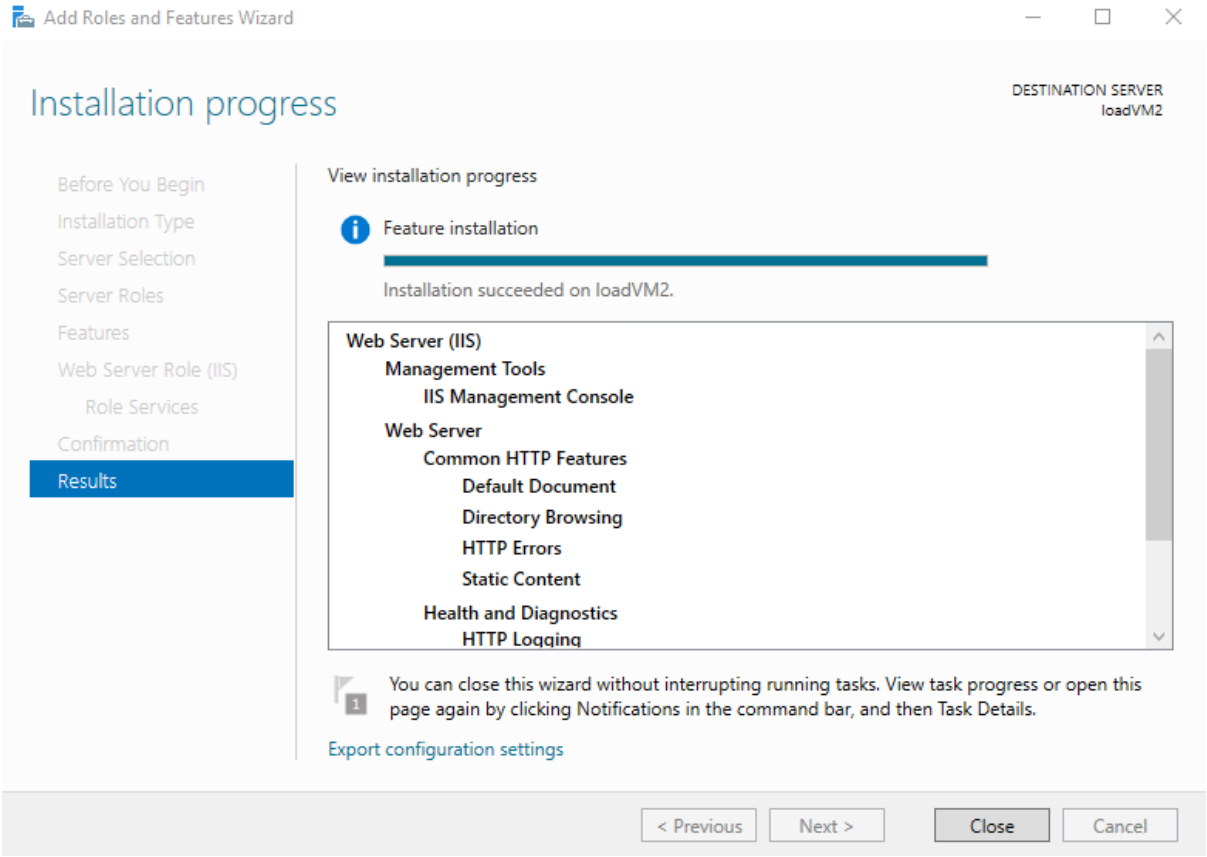
19. Then choose your availability set, after that choose your image and size. Then give your machine a username and password. Also, choose HTTP as your inbound rule.



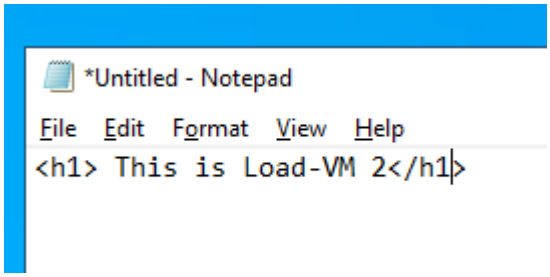
20. Then in the network interface choose the same virtual network, the same subnet, and the new public IP address. After that just move to the Review page and create your Virtual machine.



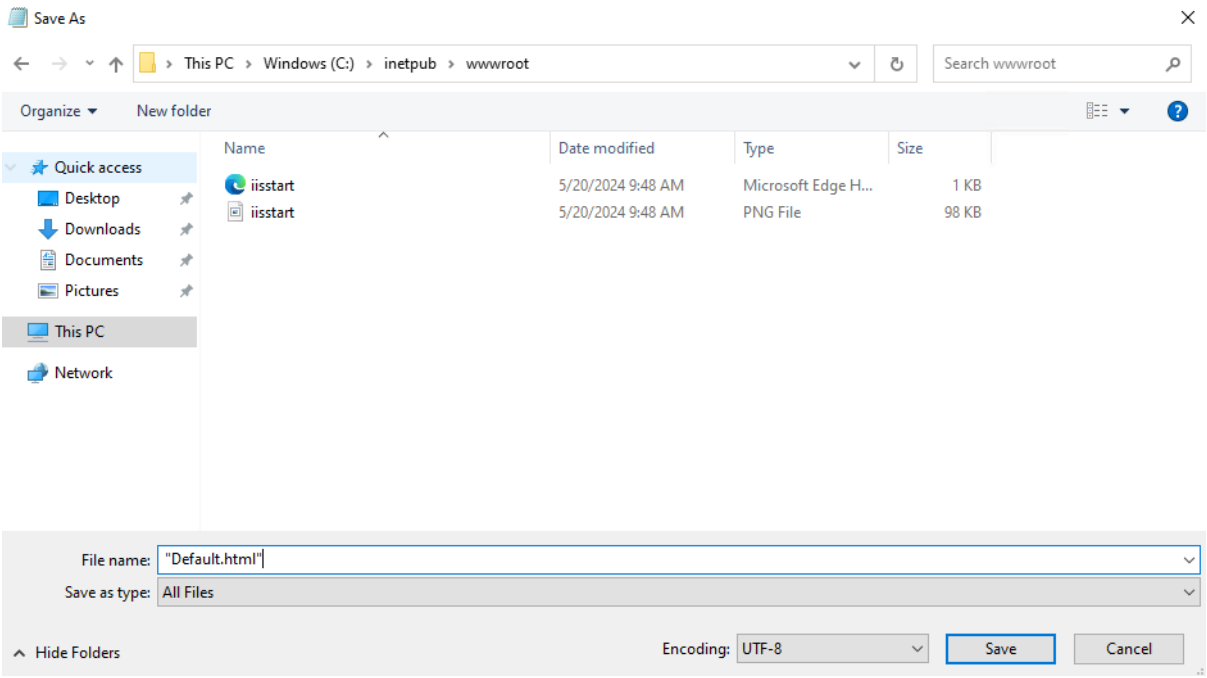
21. Once the deployment is complete for your VM then just RDP into it and install the web server (IIS) as you did for the first VM.
22. Once your Web Server is installed then open your Note Pad.



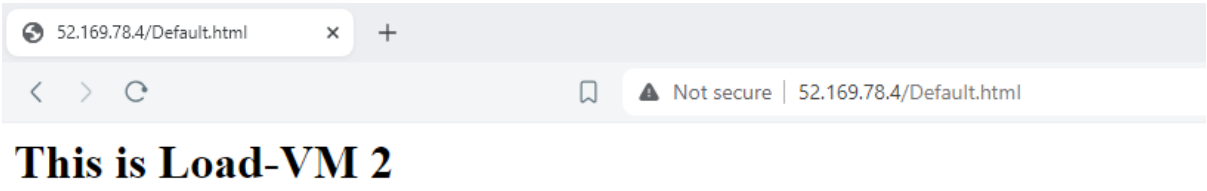
23. Then write this on your pad and save it in the same location as before.



24. Below is the location where you have to save it.



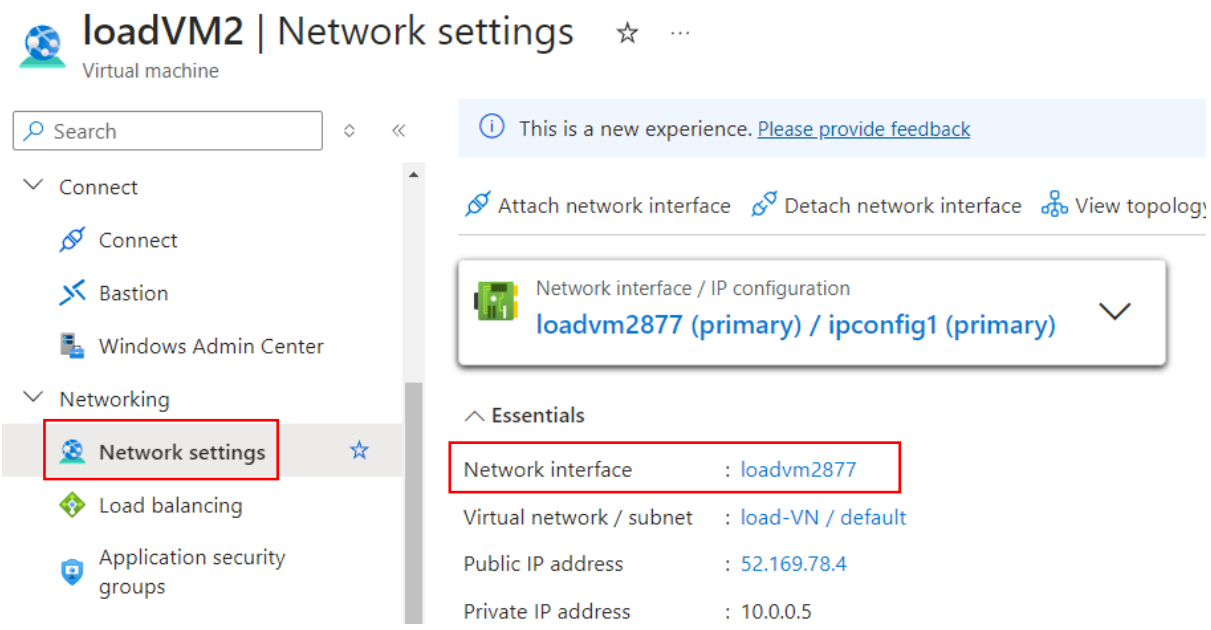
25. Then copy the public IP of your 2nd VM and append it with the file name.



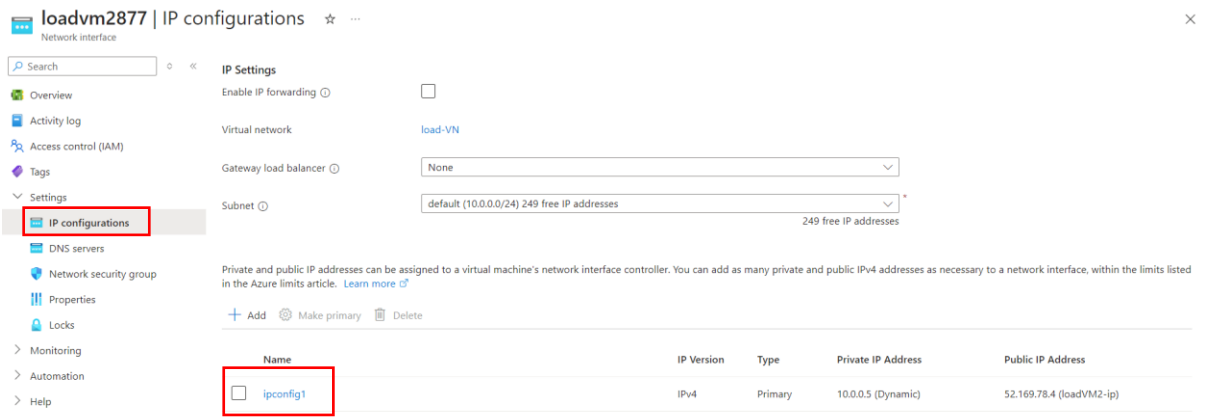
😊 Basic Load Balancer Deployment

In this process, we're refining the network configuration by removing public IP addresses from the virtual machines and consolidating network security groups (NSGs) to enhance security and streamline management. The end goal is to deploy an Azure Load Balancer with basic SKU, configuring it to evenly distribute incoming traffic to the backend pool of virtual machines. This setup aims to improve application availability, reliability, and scalability by efficiently managing incoming traffic across multiple instances.

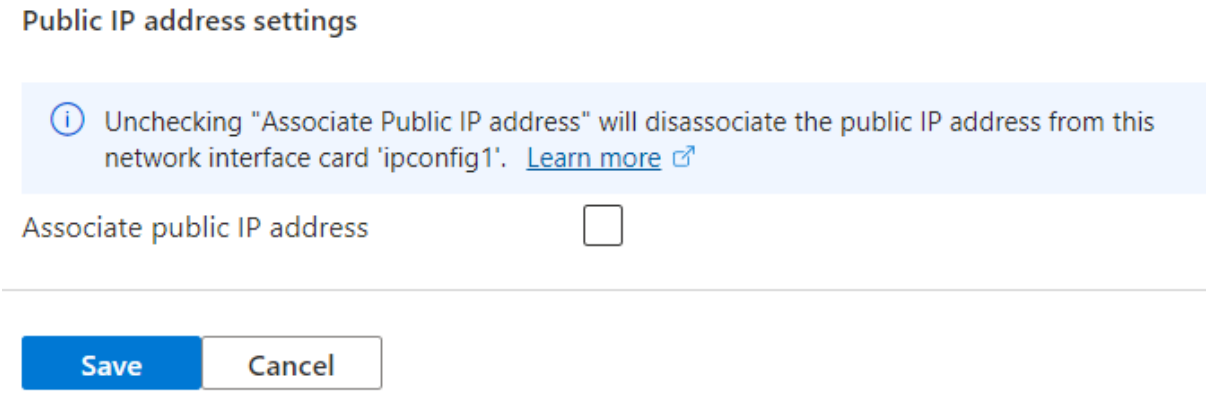
1. We're going to make some changes first. I'm going to ensure that we don't have a public IP address assigned to both machines. We'll remove that because that's not required. We'll also just have one NSG in place and one network security group.
2. Now in your VM you need to go to network settings and then go to network interface.



3. Then in the network interface go to IP configuration and the open the ip config which you will see inside it.

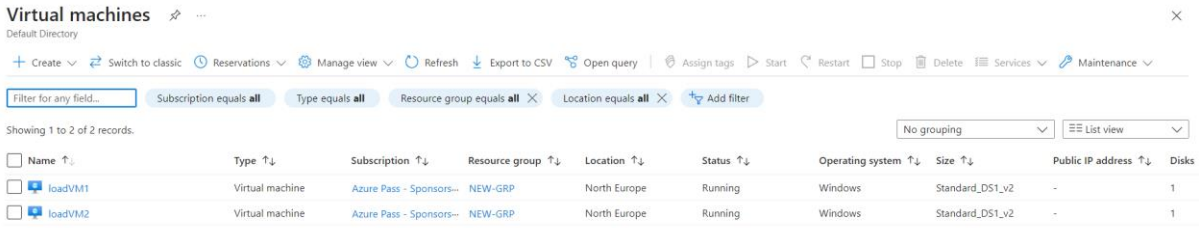


4. Then you have to uncheck the associate public IP address part and then click on save.

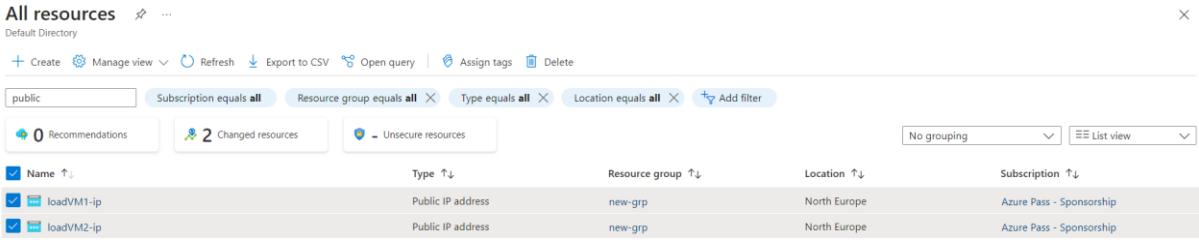


5. Then you have to do the same for the other VM to remove the Public IP address.

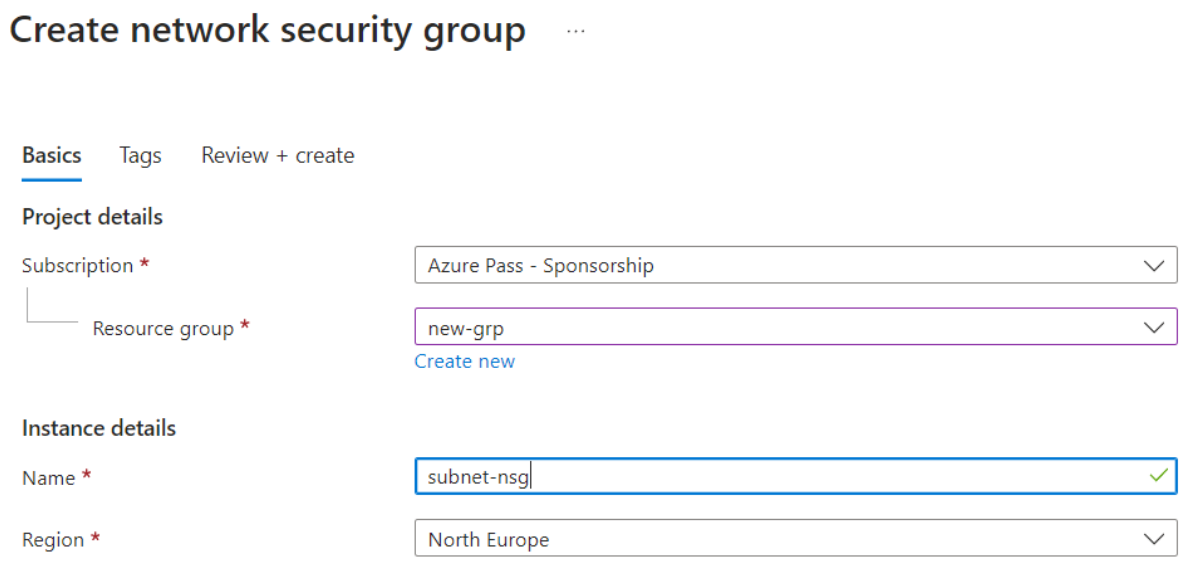
6. Now if you go to virtual machines, you will see that the Public IP address has been removed.



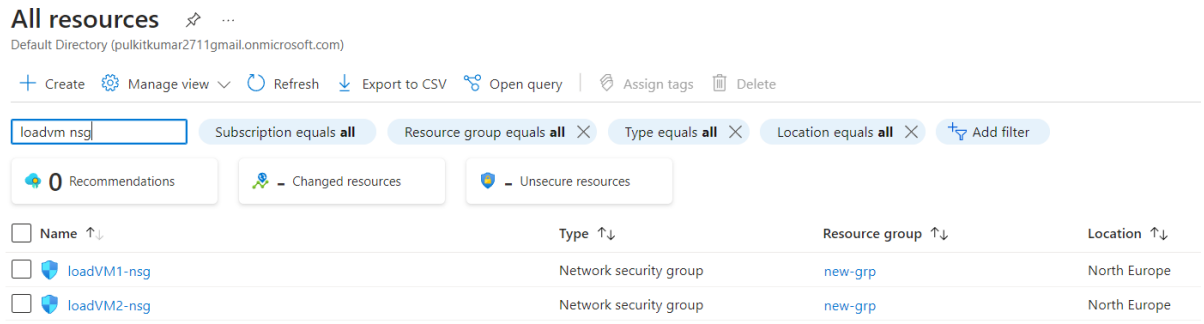
7. Now go to the All Resources tab and delete your public IP address because we don't need it.



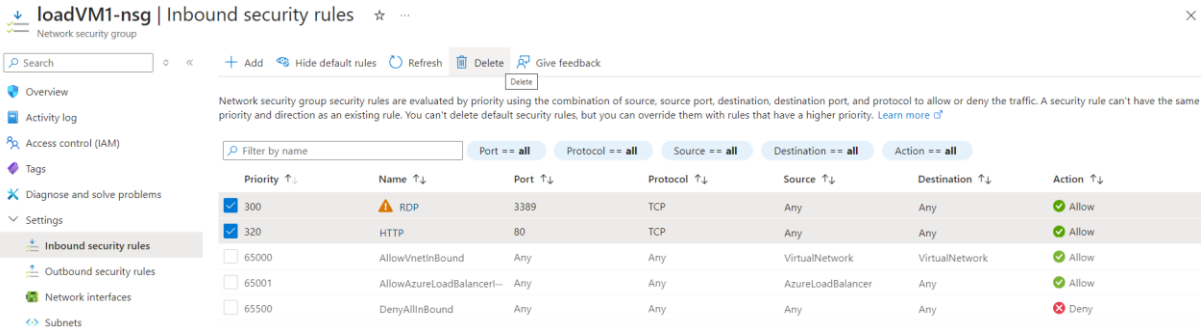
8. Now we are going to create a new Network Security Group (NSG), for that in the marketplace search for it.
9. You just have to choose your resource group and create your NSG.



10. Now in a new tab you have to open all resources and filter out the NSGs. Then open them both in a new tab separately.

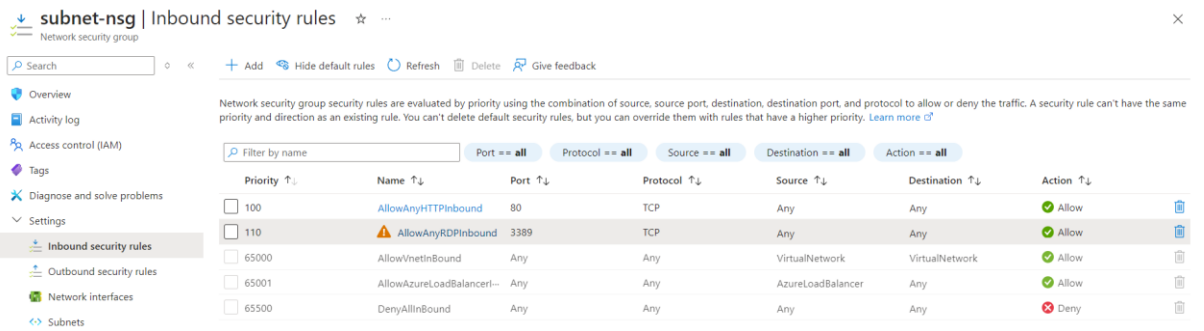


11. Then go to inbound rules and remove both of them.

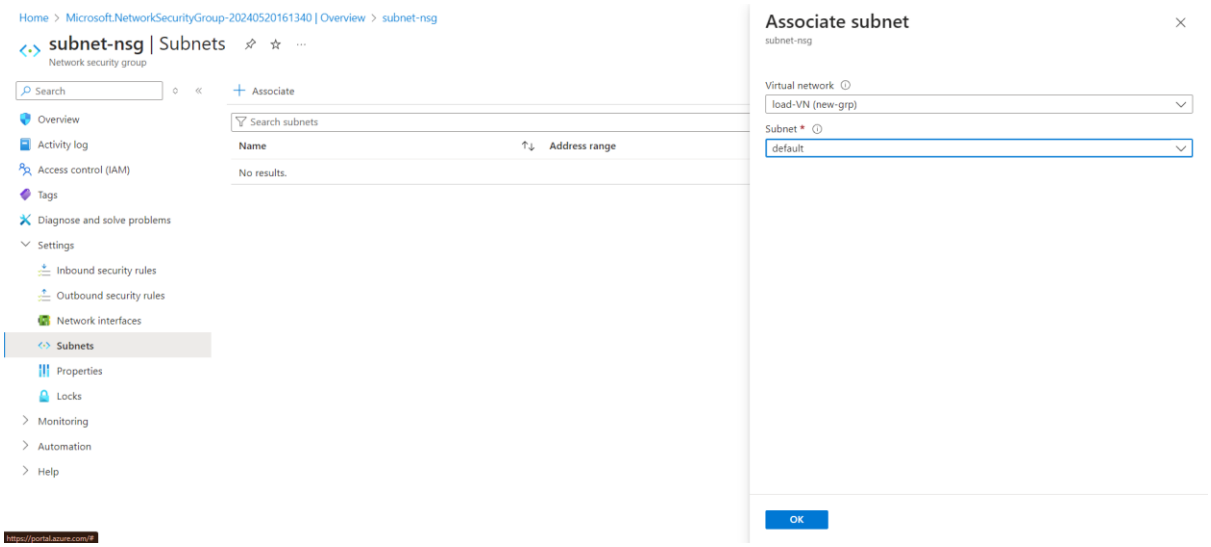


12. Then go to network interfaces and dissociate it. Then move to overview and delete your NSG.

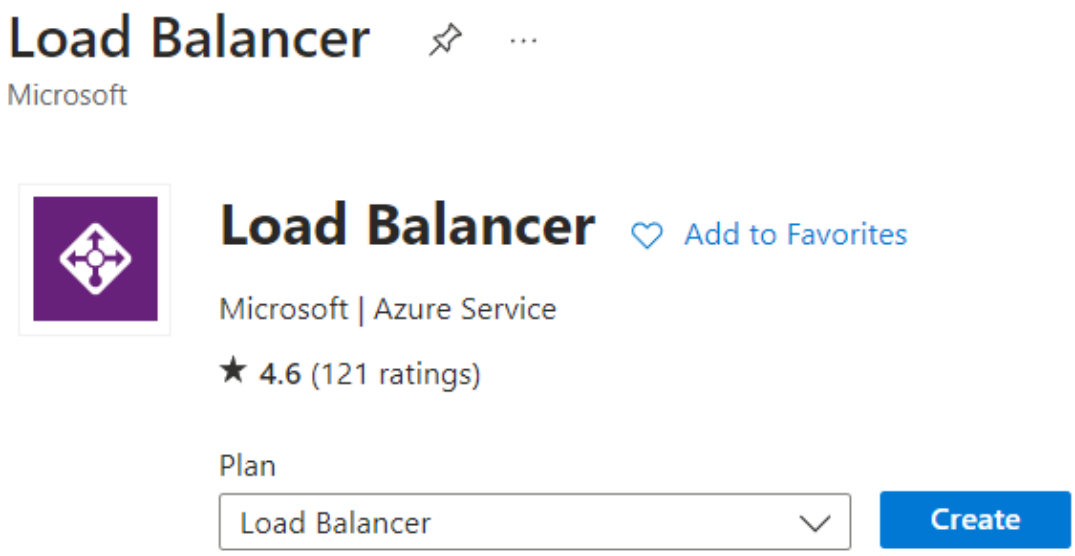
13. You have to do the same for other VM also.
14. After that you have to open the New NSG and add two inbound rules in it one for HTTP and other one for RDP.



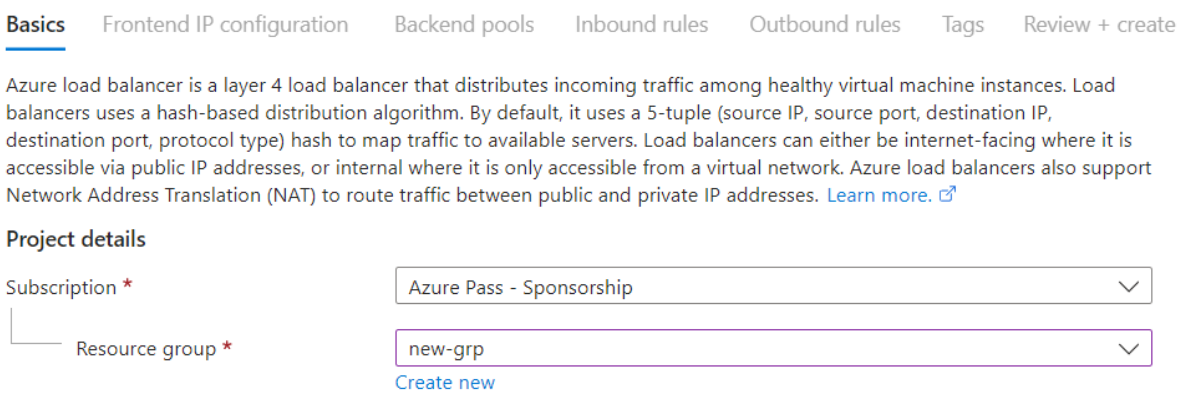
15. Then go to Subnets and associate your subnet with it.



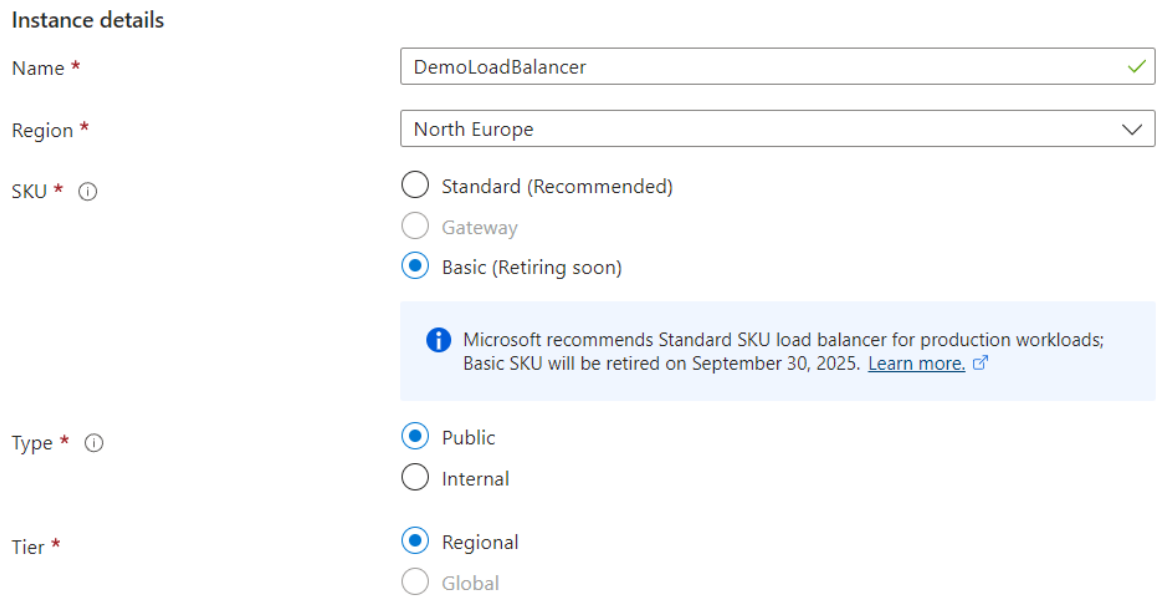
16. Now we are going to deploy our Load Balancer. For that in the marketplace search for the load balancer and choose this service accordingly.



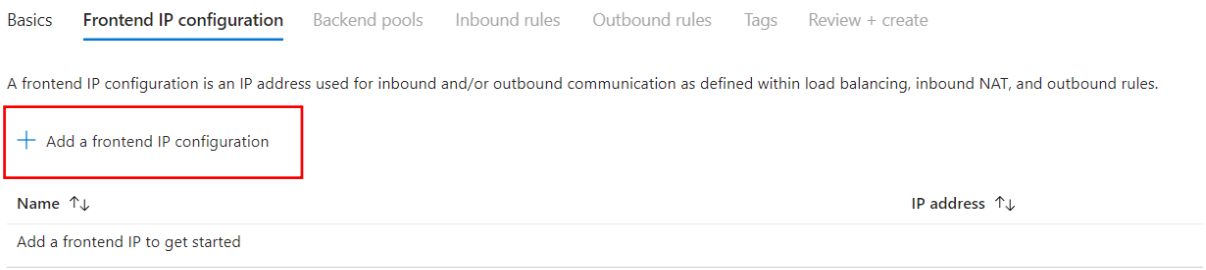
17. First we need to choose our Resource Group.



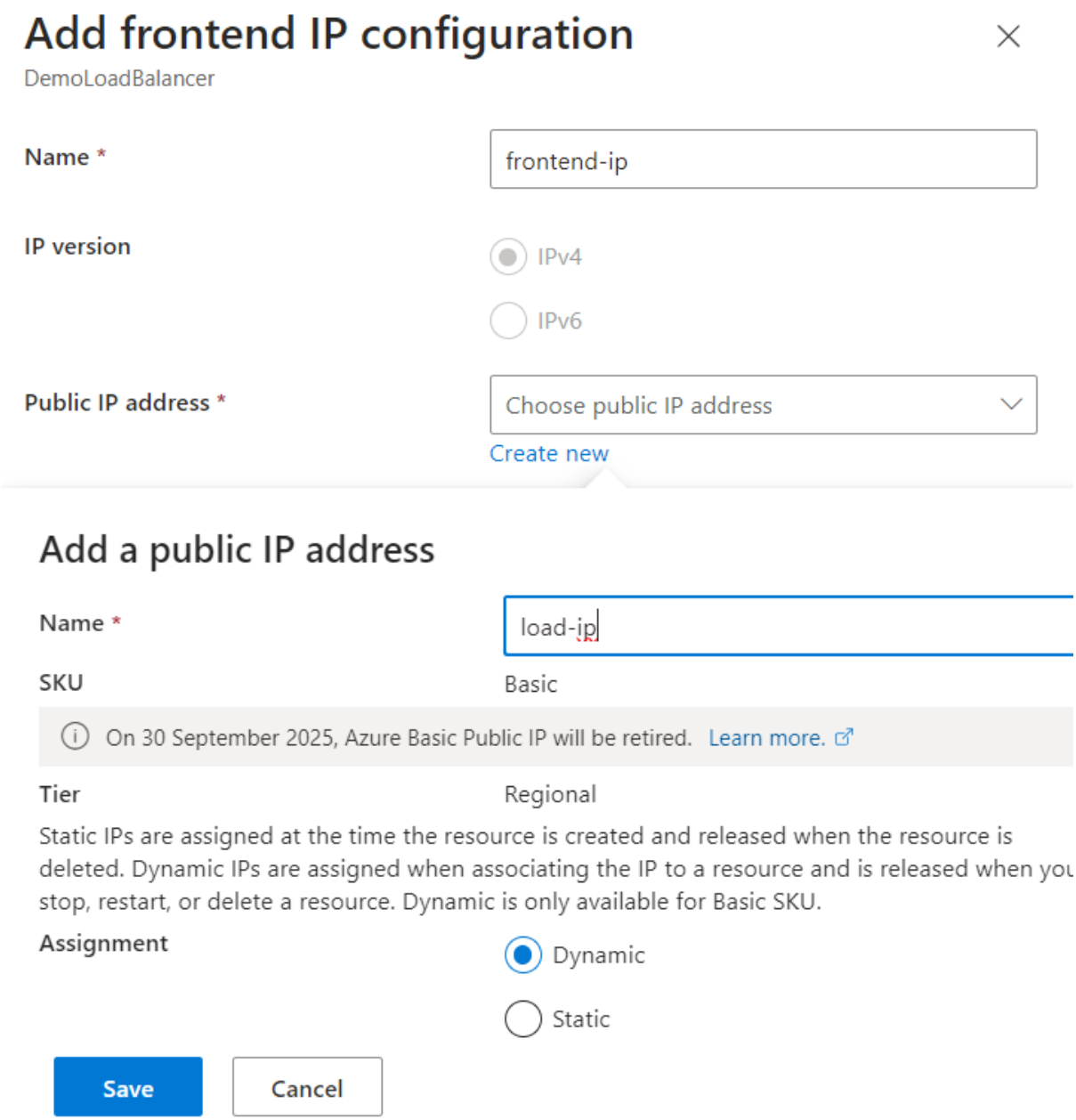
18. Then we need to give a name to it and choose our region then we need to choose our SKU which will be basic for the time being. And the type should be Public. Then move to next option.



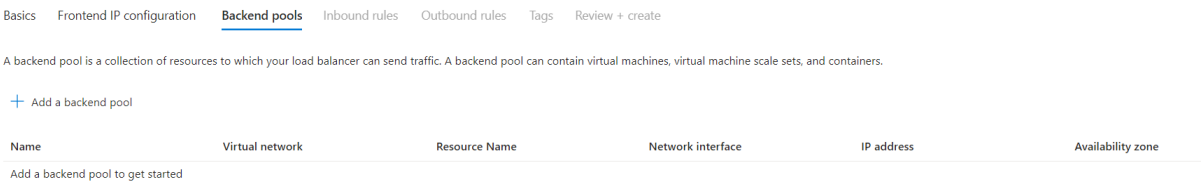
19. In front-end IP configuration you have to click on Add.



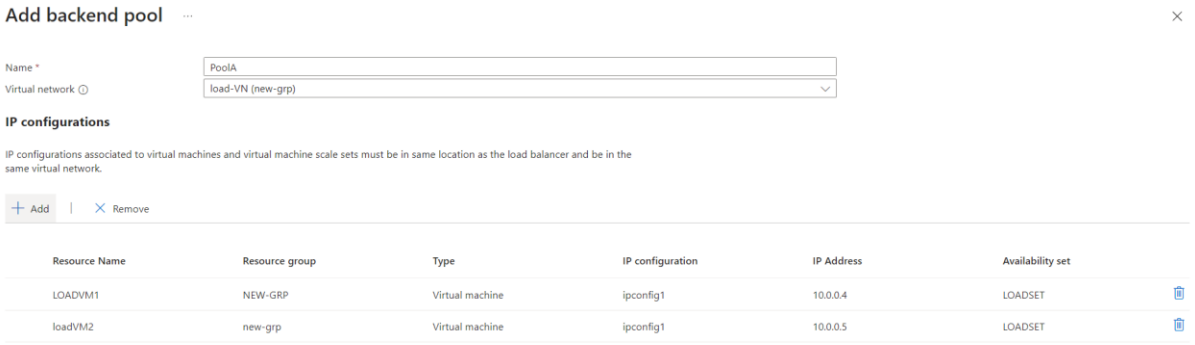
20. After that you need to give it a name then for public IP click on Create new.



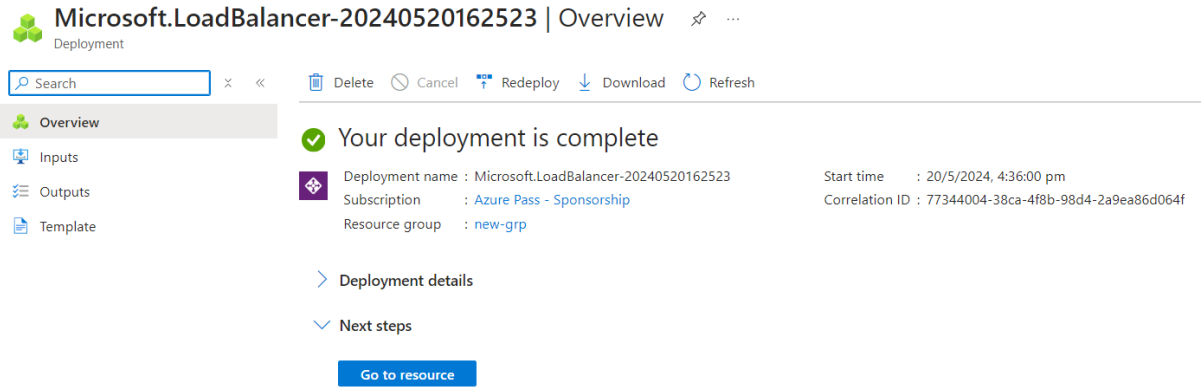
21. Now we are going to add the backend pool or in simple words our VMs here. For that click on add.



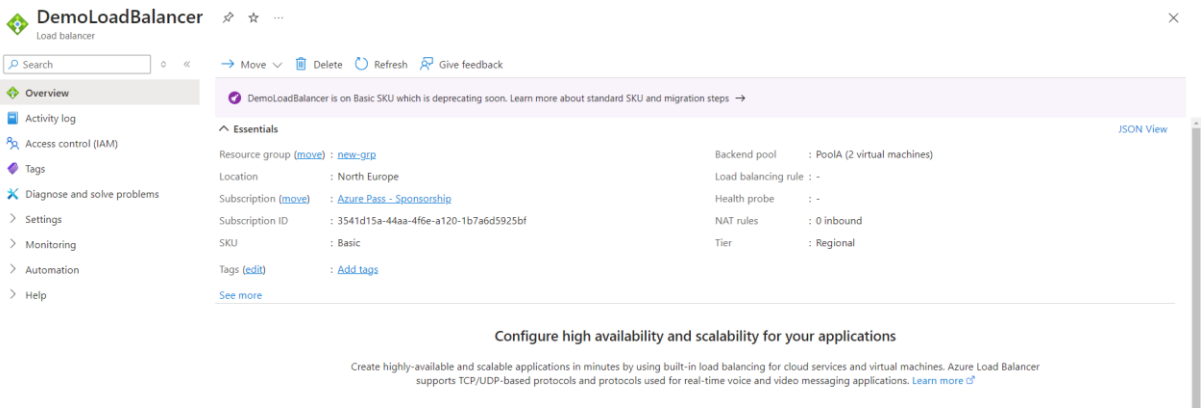
22. Then you have to give it a name then choose the virtual network after in the IP configuration you have to click on Add and choose both of the load balancers. Then just click on save.



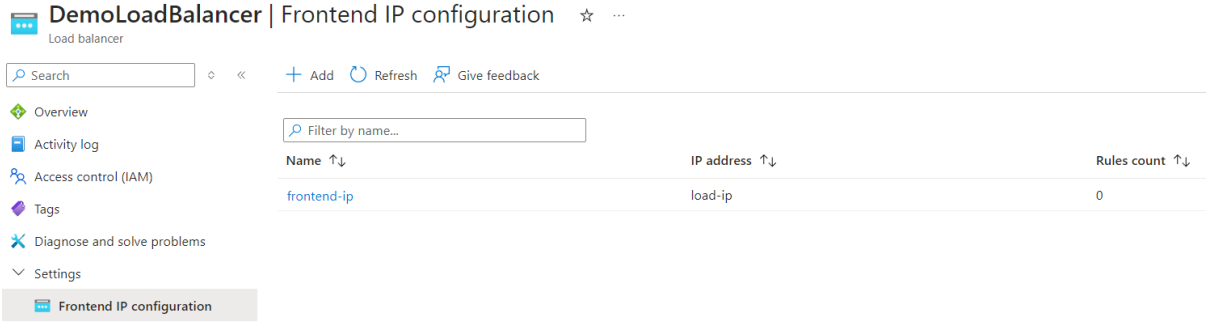
23. After that move to the review page and create your load balancer. Once the deployment is complete click on go to resources.



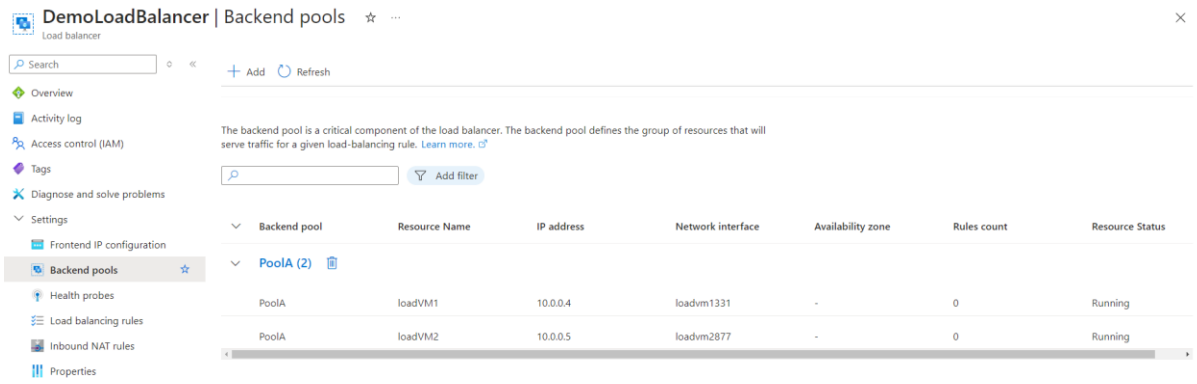
24. This is the dashboard of the load balancer you can explore it if you want.



25. In the front-end IP configuration, we can see that we have our public Ip address.



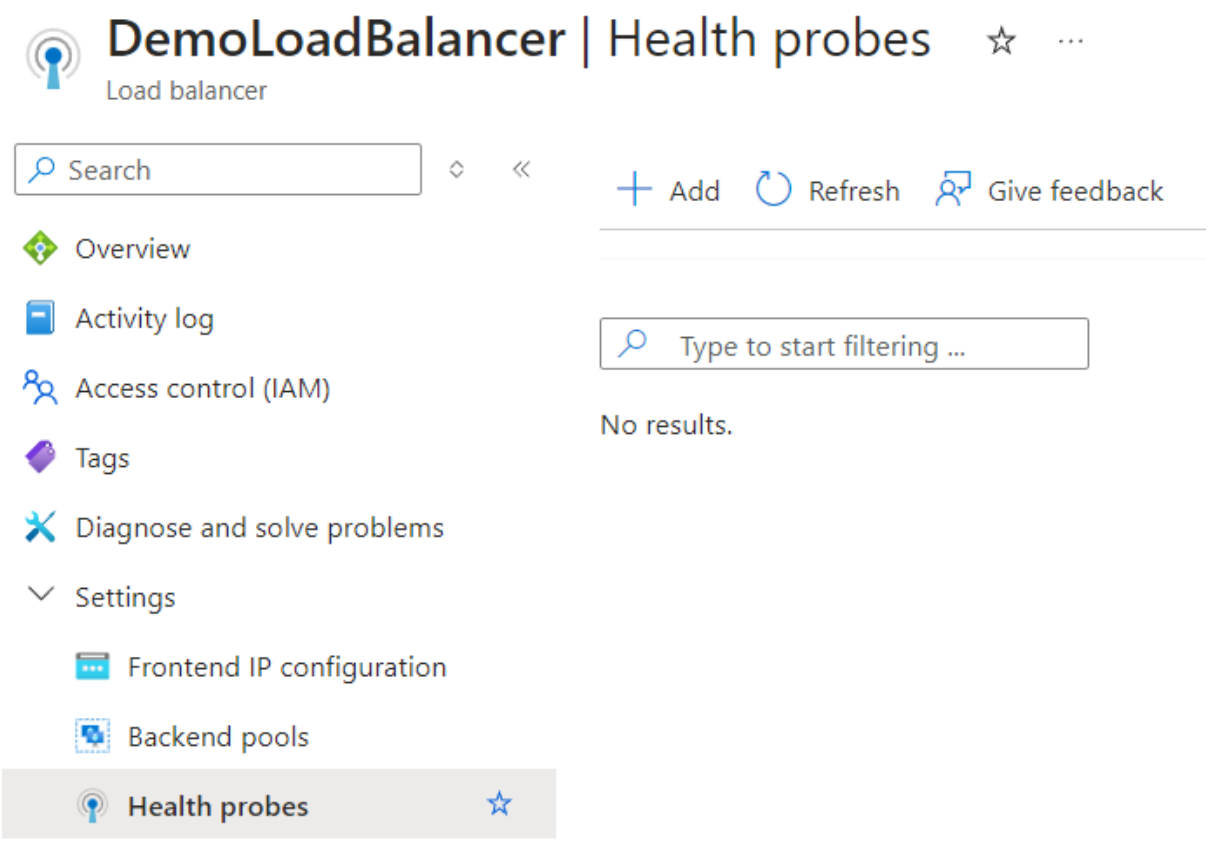
26. And in the back-end pools we have our VMs running.



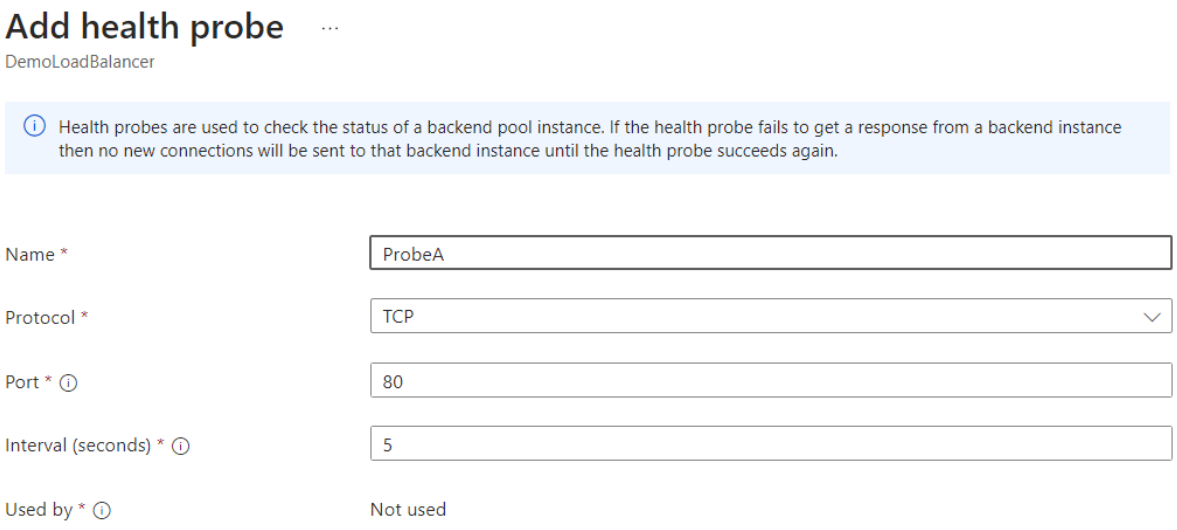
😊 Basic Load Balancer Configuration

In this process, we're configuring an Azure Load Balancer to enhance application reliability and optimize resource usage. We're adding health probes to ensure traffic is directed only to healthy backend instances, defining load balancing rules to distribute incoming requests efficiently, and setting up inbound NAT rules for remote access to individual virtual machines. The end goal is to establish a robust load-balancing setup that improves application availability and facilitates remote management of backend resources.

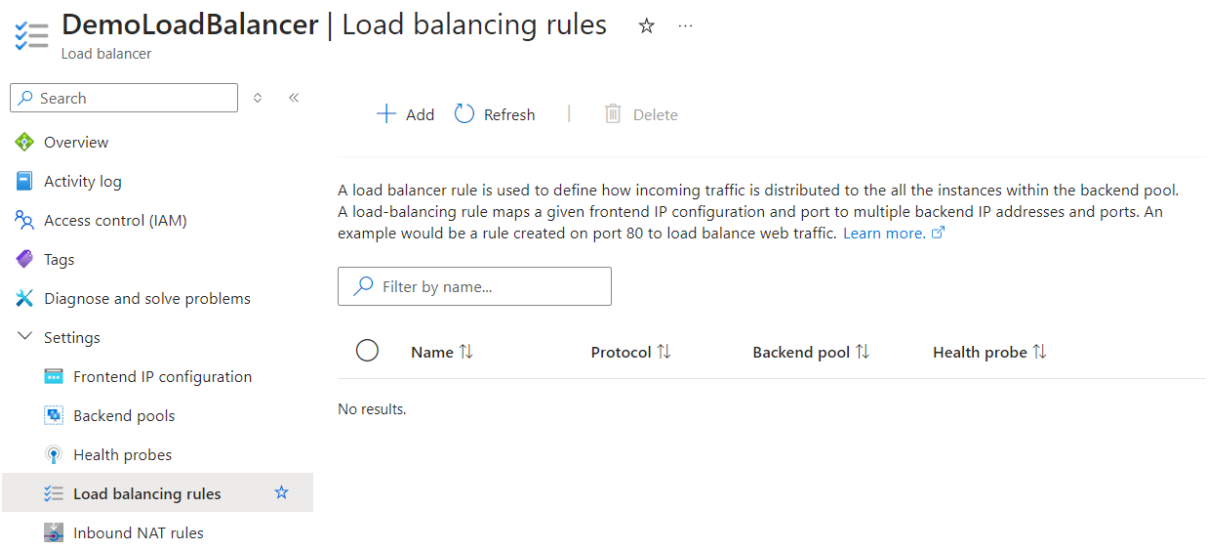
1. Now in this we will add some configurations in our Load balancer.
2. Inside of your load balancer click on heath probes and click on add.



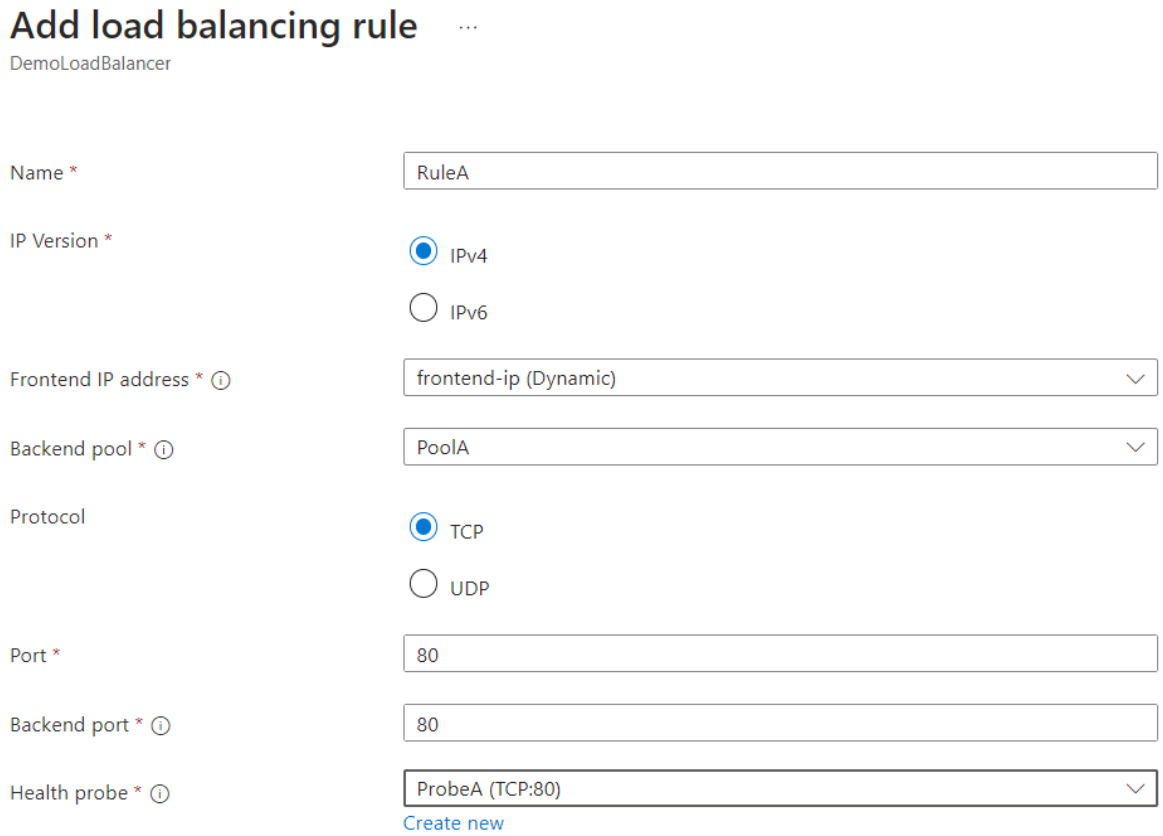
3. Then in there you need to give it a name and choose the same configuration as shown below.



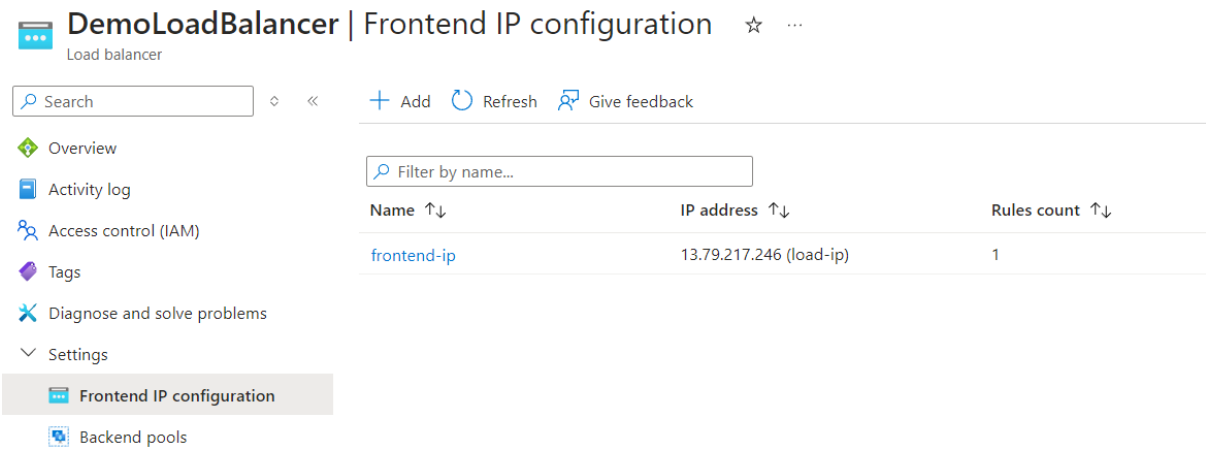
4. Now we have to define load balancing rules. For that click on add. Moreover, the load balancing rule will understand how to direct requests onto the backend machines.



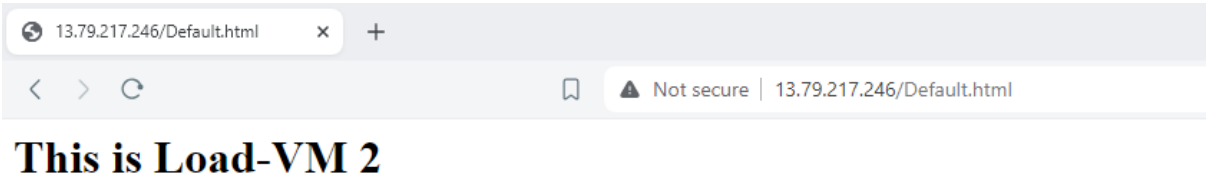
5. Now here you have to give it a name first then choose your frontend IP address and backend pool. Then the port and backend port are both 80. After that choose your health probe and click on save.



6. Now you have to go to frontend IP configuration and here you will see that the Public IP address has been assigned. Now copy this IP and paste it in a new tab or browser.



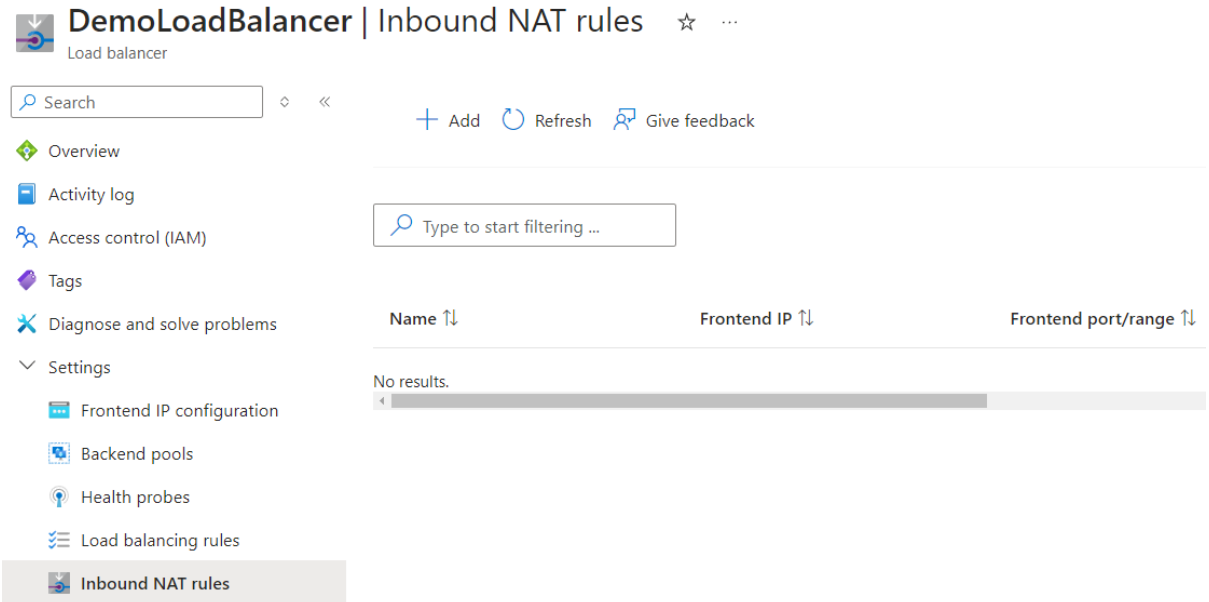
7. Now it has directed the request to Load VM2.
8. So, now the load balancer is directing requests from, let's say my laptop onto loadvm2. If another user tries to again go onto the public IP address of the load balancer, the load balancer might direct the request. Let's say onto loadmvm1.



This is Load-VM 2

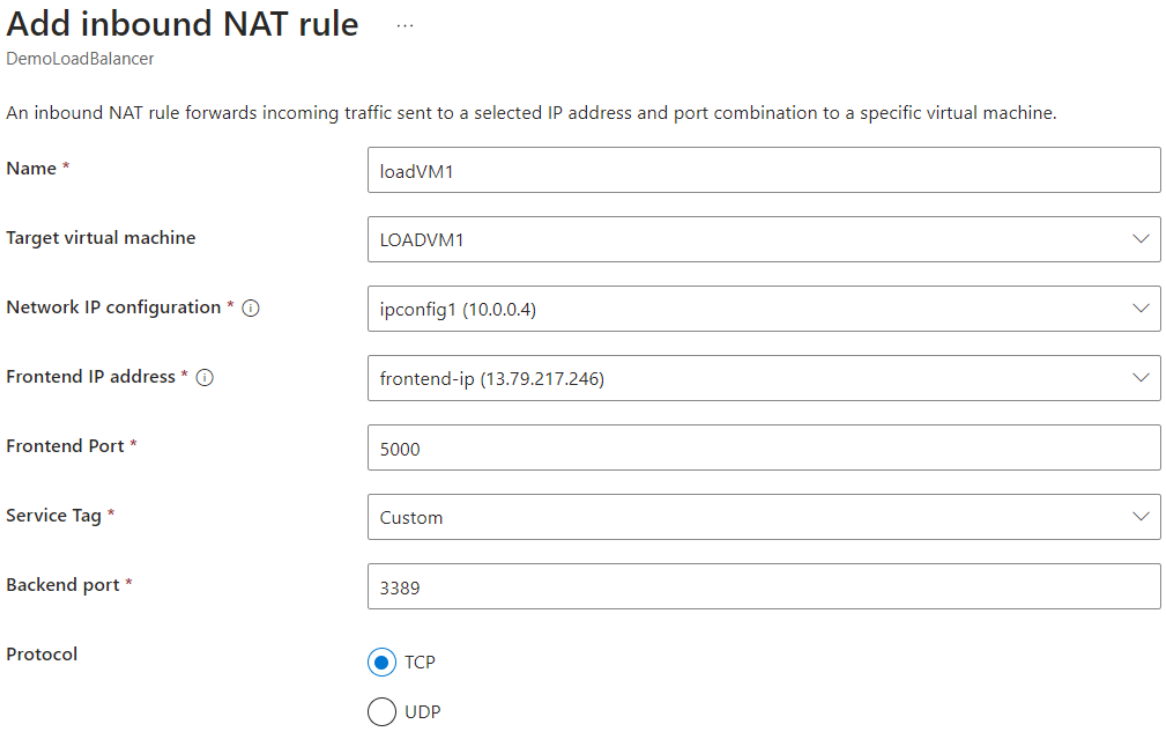
😄 NAT Rules

1. Now you have to go to Inbound NAT rules and click on Add.

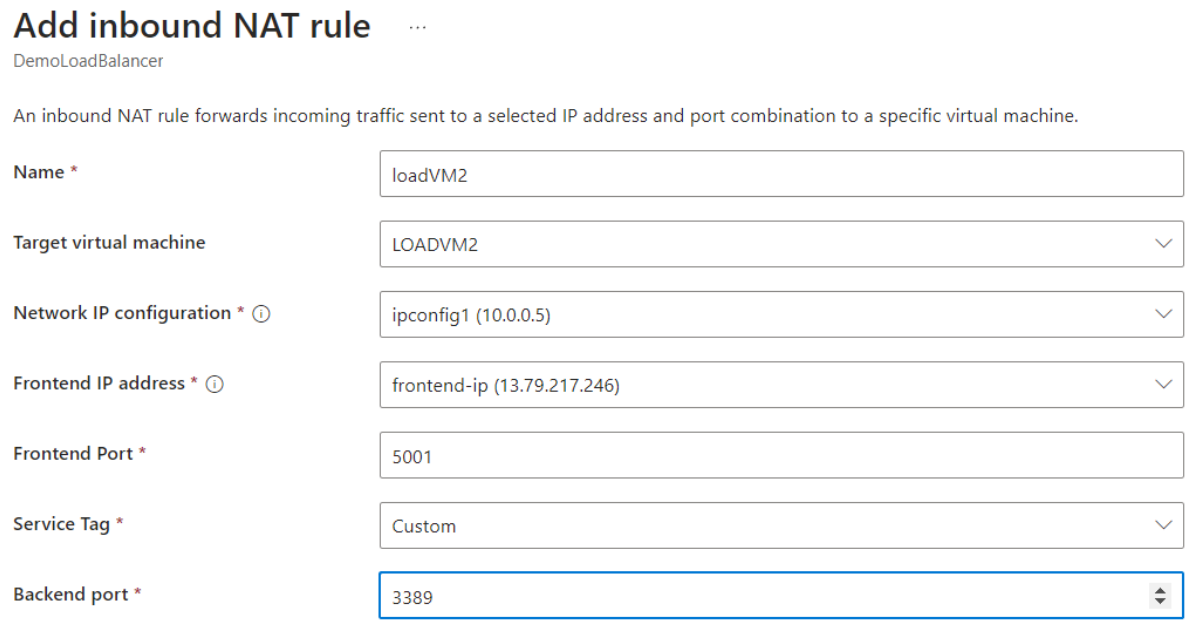


2. Then you have to give it a name and choose the target virtual machine which is load VM1 and then we have to choose network IP and frontend IP. Then the frontend port

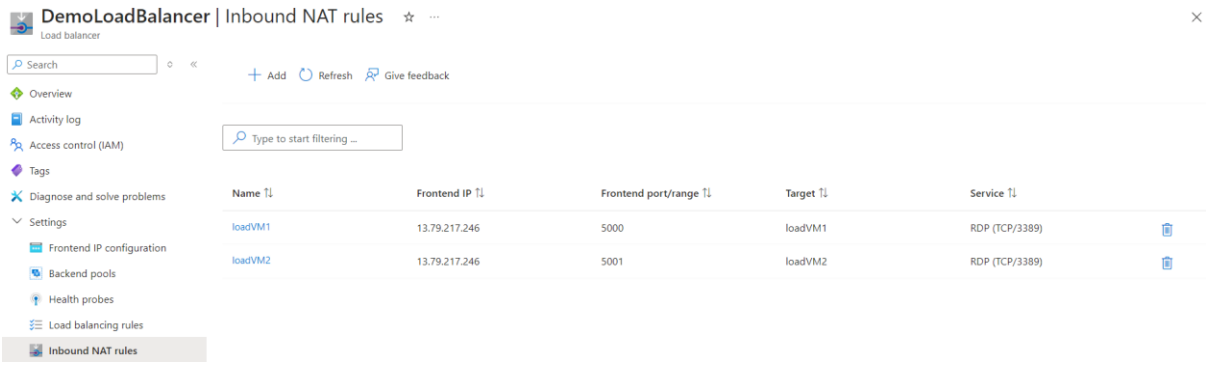
is random which we wrote is 5000 and we want to direct our request to the backend port which is 3389. After that just click on save.



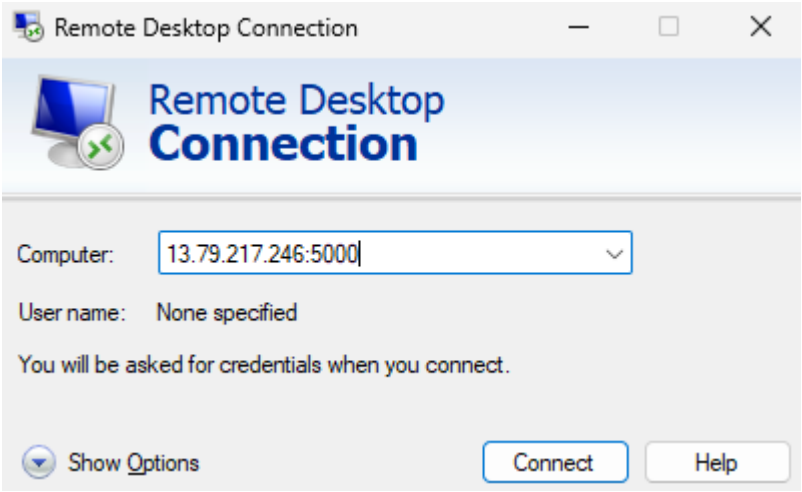
3. After that we do the same for virtual machine 2 and you can copy the configuration from the snapshot below.



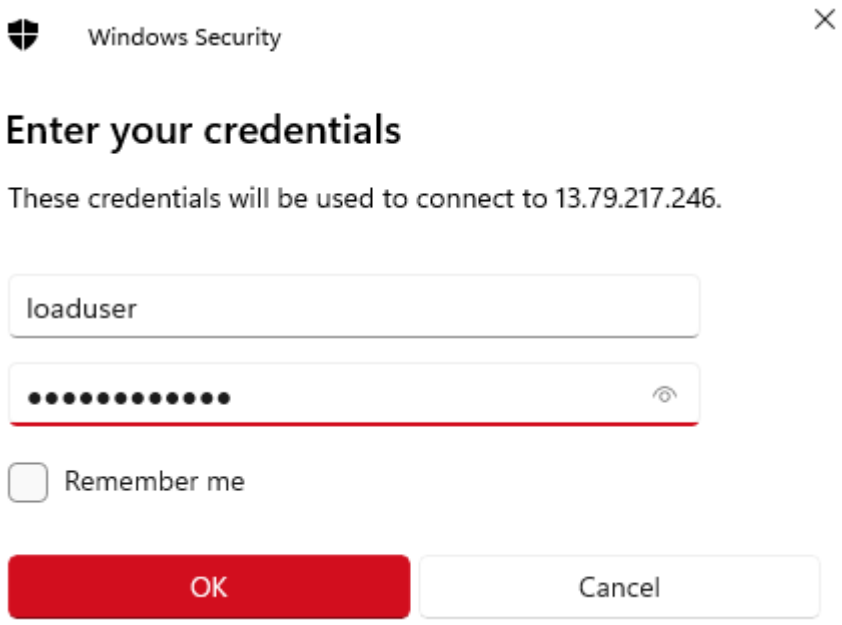
4. Below you can see that both the VMs are added.



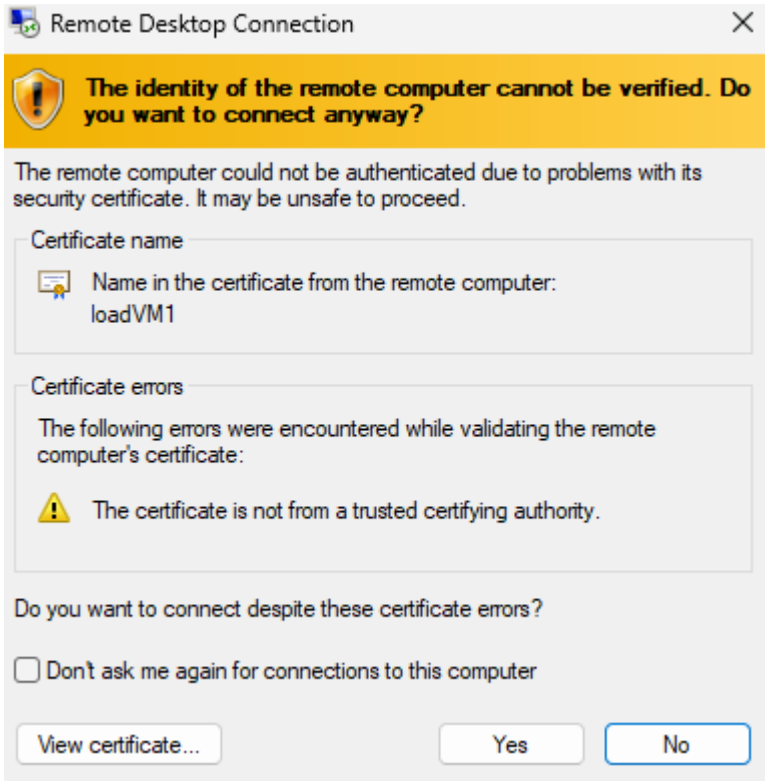
5. Now you can do the RDP into your machine using the Public IP address from the Frontend IP.
6. Paste the IP address and append it with your front-end port which is 5000 for load VM1.



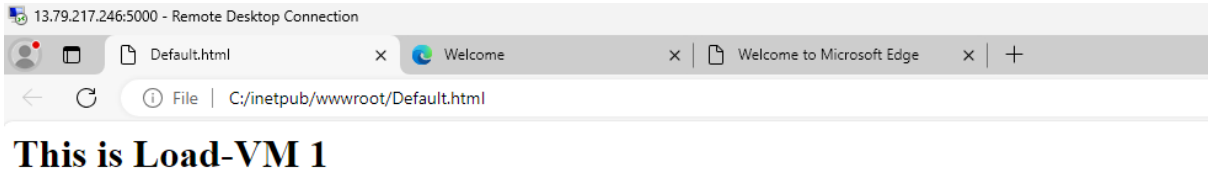
7. Now you can see that it is asking to enter the VM credentials.



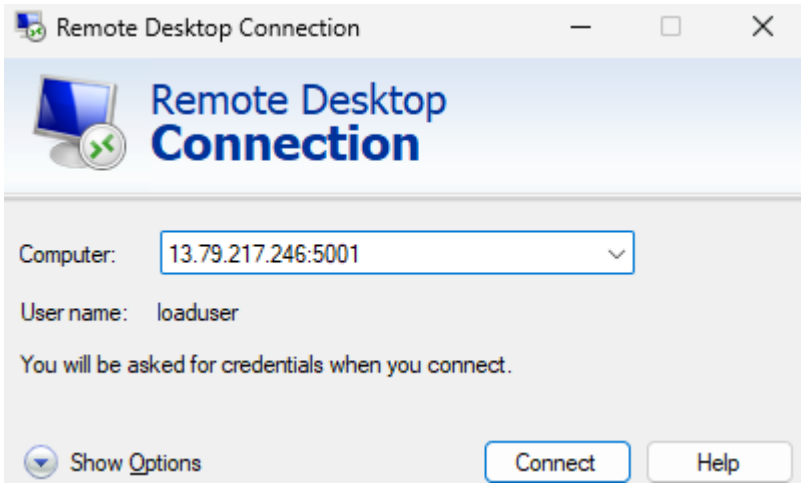
8. Below you will see that the connection with loadVM1 is going to establish.



9. And you can see that we are in Load VM1.



10. Now close it and open load VM2 by following the same steps.
11. This time just change the IP with 5001 and click on connect.



12. Below you can see that we are going to connect with Load VM2.

13. Once you are done with all this just delete all of your resources.