Contents

[LOAD BALANCER 2](#_Toc149392522)

[LOAD BALANCER CONCEPT 3](#_Toc149392523)

[COMPONENTS OF LOAD BALANCER 3](#_Toc149392524)

[TYPES OF LOAD BALANCER 3](#_Toc149392525)

[LOAD BALANCER SKUs 4](#_Toc149392526)

[BASIC LOAD BALANCER (AVAILABILITY SET) 4](#_Toc149392527)

[BASIC LOAD BALANCER (SCALE SET) 12](#_Toc149392528)

[STANDARD LOAD BALANCER 18](#_Toc149392529)

[AZURE APPLICATION GATEWAY 25](#_Toc149392530)

[APPLICATION GATEWAY – PATH BASED ROUTING 25](#_Toc149392531)

[APPLICATION GATEWAY COMPONENTS 26](#_Toc149392532)

[SETTING UP APPLICATION GATEWAY – URL ROUTING 27](#_Toc149392533)

[USER DEFINED ROUTES 31](#_Toc149392534)

[USECASE - SETTING UP USER DEFINED ROUTE 31](#_Toc149392535)

[NETWORK WATCHER 40](#_Toc149392536)

[AZURE BASTION SERVICE 40](#_Toc149392537)

[SETTING UP BASTION SERVICE 41](#_Toc149392538)

[AZURE TRAFFIC MANAGER 46](#_Toc149392539)

[GEOGRAPIC ROUTING METHOD 47](#_Toc149392540)

[AZURE TRAFFIC MANAGER SET UP 47](#_Toc149392541)

[AZURE TRAFFIC MANAGER ROUTING METHODS 47](#_Toc149392542)

[PRIORITY ROUTING METHOD 48](#_Toc149392543)

[PERFORMANCE ROUTING METHOD 50](#_Toc149392544)

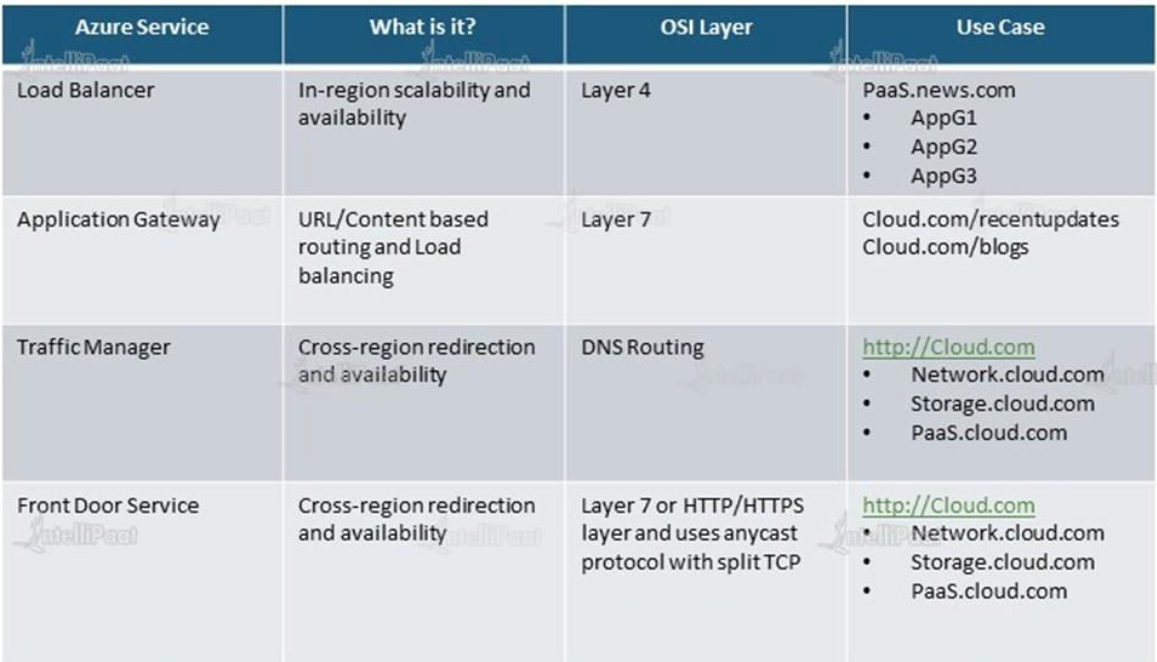
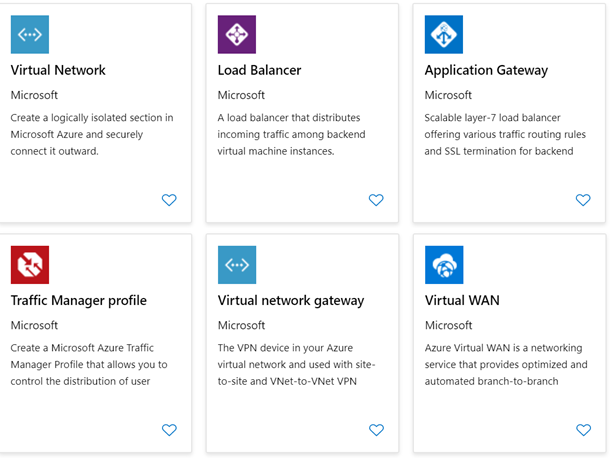
[SUBNET ROUTING METHOD 50](#_Toc149392545)

[MULTIVALUED ROUTING METHOD 51](#_Toc149392546)

[WEIGHTED ROUTING METHOD 51](#_Toc149392547)

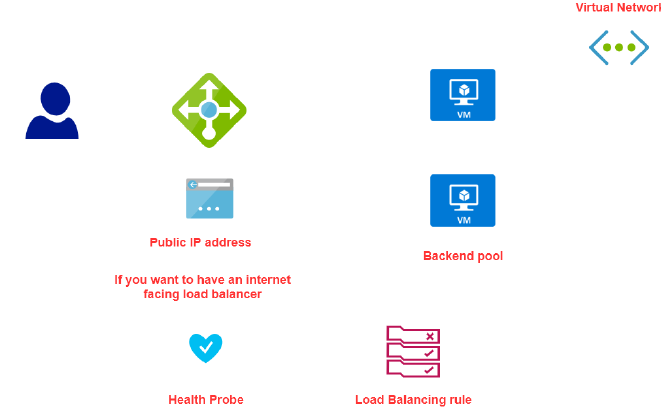
## LOAD BALANCER

Azure network services offer a range of components with functionalities and capabilities, as shown in the following image:



|  |  |
| --- | --- |
|  | * The Load Balancer Service, distribute the user requests across the pool of resources (e.g., Azure virtual machines) to maintain availability. * *The azure load balancer works on Layer 4(Transport Layer) and Application Gateway (Application Layer) of the OSO Model* * The Azure load balancer comes in 2 pricing model (SKU)   + BASIC LOAD BALANCER   + STANDARD LOAD BALANCER |

### LOAD BALANCER CONCEPT



### COMPONENTS OF LOAD BALANCER

|  |  |
| --- | --- |
| **BACKEND POOL** | * When we create a load balancer, we specific something called backend pool, which is a set of VMs to which the LB will route the traffic to. |
| **FRONTEND IP** | * The LB has a public IP address (called Frontend IP address). * The user can be able to access the VM via public IP of the load balancer. * The redirect of request from LB to VM are done via private IP of VM |
| **HEALTH PROBE** | * This helps the LB monitor the health of VMs in the backend pool |
| **LOAD BALANCING RULES** | * These rules drive how request can be distributed across the VM (which are part of backend pool) |

### TYPES OF LOAD BALANCER

There are 2 types of load balancer

1. INTERNAL LOAD BALANCER
2. PUBLIC LOAD BALANCER

#### INTERNAL LOAD BALANCER

|  |  |
| --- | --- |
|  | * The internal load is used to direct traffic only between Azure’s internal resources i.e., the resources managed by Azure infrastructure or resources connected to Azure infrastructure using secure VPN * Internal load balancer can be used when want to divide the traffic to coming from other Azure Resources * It can also use internal load balancer or the traffic coming from on-premises network that is connected to an Azure resource via s secure VPN connection. |

##### SETTING UP INTERNAL LOAD BALANCER

#### PUBLIC LOAD BALANCER

* Public load balancer is used to handle the traffic between public facing IP address of incoming traffic to private IP address of Azure resources.

### LOAD BALANCER SKUs

|  |  |
| --- | --- |
| **BASIC LOAD BALANCER** | **STANDARD LOAD BALANCER** |
| *Free* | *Charges on Per hour basis* |
| *The VM in the backend pool must be part of availability set or scale set* | *VM can be an independent machine that are part of a VNET* |
| *Health Probes support TCP and HTTP only* | *Health Probes support TCP and HTTP and HTTPS* |
| *No support for availability zone* | *Support for availability zone* |
| *NO SLA* | *SLA of 99.99%* |

### BASIC LOAD BALANCER(VMs in AVAILABILITY SET)

|  |  |
| --- | --- |
|  | * **The VM has to part of Scale set or availibility set** |

#### SETTING UP BASIC LOAD BALANCER (VM IN AVALIBILITY SET)

To illustrate the basic load balancer services set up. Let’s follow the following steps.

* ***STEP 1*:**
  + *We will create VMs, which will be a part of an* ***Availability Set (Make sure the VMs are part of same VNET)***
  + *The VM will have no public IP address as the communication between LB and VMs will be done via private IPs*
  + *Availability Sets can be used in conjunction with Azure Load Balancer to distribute traffic across multiple VMs within the set, providing load balancing and scaling capabilities.*
* ***STEP 2:*** *Install IIS with a Default.html page.*
* ***STEP 3:*** 
  + *Create a Public IP address (Note public IP address is a separate resource in Azure).*
  + *The Public IP will be assigned to the Load Balancer. Also known as* ***Front End public IP address***
* ***STEP 5:*** *Create a NSG which will be attached to at the subnet level of which VM is part of. Note – the NSG created with the VMs must be detached.*
* ***STEP 6:*** *Create a NSG which will be attached to at the subnet level of which VM is part of. Note – the NSG created with the VMs must be detached.*
* ***STEP 7:*** *Create and configure a Load Balancer resource. As part of configuration -we need to set up a Backend pool of the VMs created in Step 1*
* ***STEP 8:*** *Configure the health probe to check the health of VM in the backend pool*
* ***STEP 9:***  *Create Load balancing Rules – which will give the routing of request to the VM in backend pool*
* ***STEP 10:*** *Now we can be able send the request to load balancer via its public IP address, which will then redirect the traffic to VM in the backend pool.*

#### SETTING UP BASIC LOAD BALANCER (VM IN SCALE SET)

STEP 1: *We will create VMs, which will be a part of an* ***Availability Set***

**In this step**

1. **Port 80 has been enabled**
2. **ISS has been installed on the VM**
3. **After validation of IIS installation – Disassociate the Public IP address of VM**

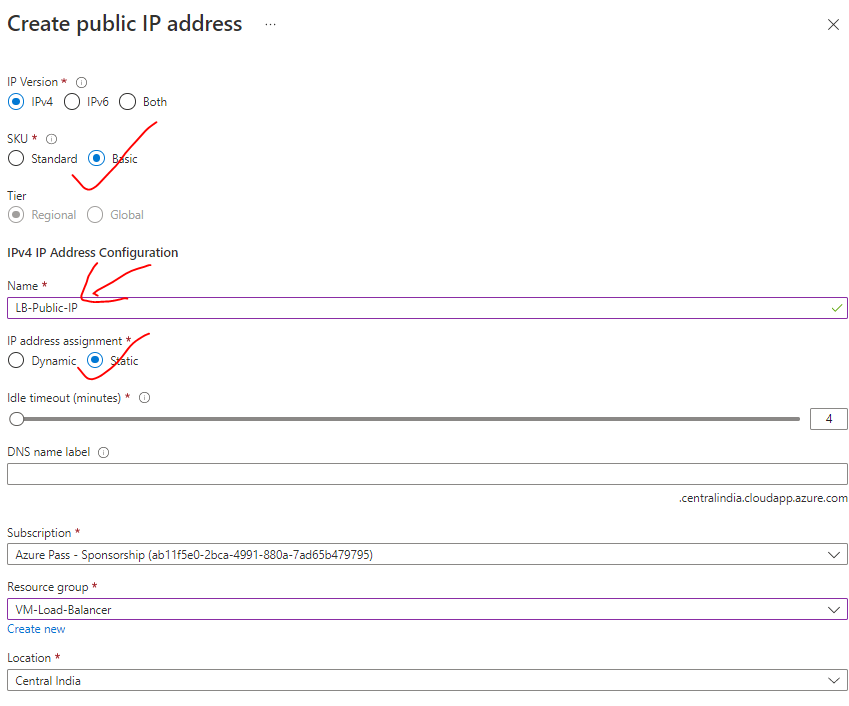


|  |  |
| --- | --- |
|  | * The communication between the Load balancer and the VMs in the backend pool – happens using private IP address. * For now – let’s keep the public IP address so that we can deploy IIS and validate it. * After that we can disassociate the public IP address from the VM as the communication between LB and VM are through private IPs |

**DISASSOCIATE PUBLIC IP FROM VM**

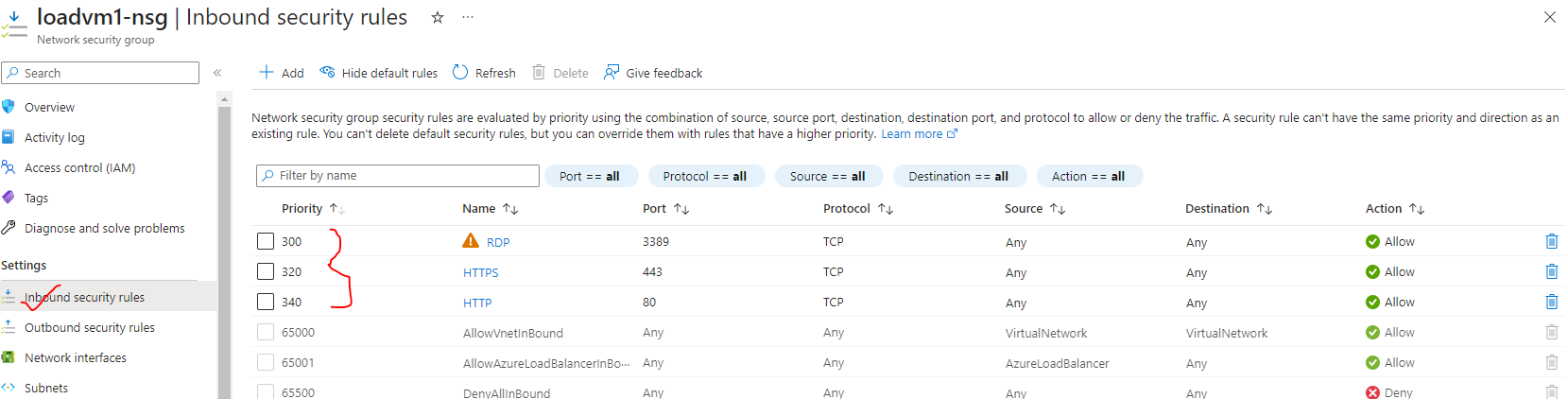
|  |  |
| --- | --- |
|  | 1. Go to VM 🡪 Networking 🡪Network Interface 🡪 IP Configuration 2. Disassociate the Public IP and make the provide IP as a static Ip 🡪 Save |

***STEP 3:*** *Create a Public IP address (Note public IP address is a separate resource in Azure). The Public IP will be assigned to the Load Balancer. Also known as Front End public IP address.* ***For basic load balancer the IP must be of type Basic SKU***



***STEP 4:*** *Disassociate the NSG associated with the VM*

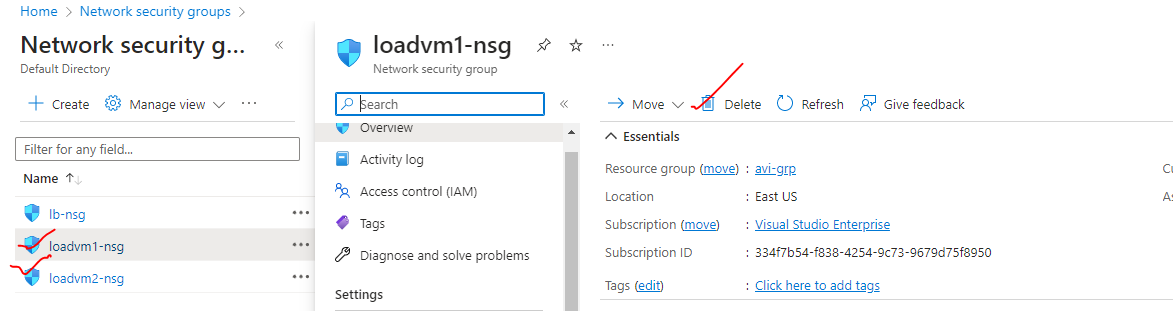
1. Delete the NSG Rule



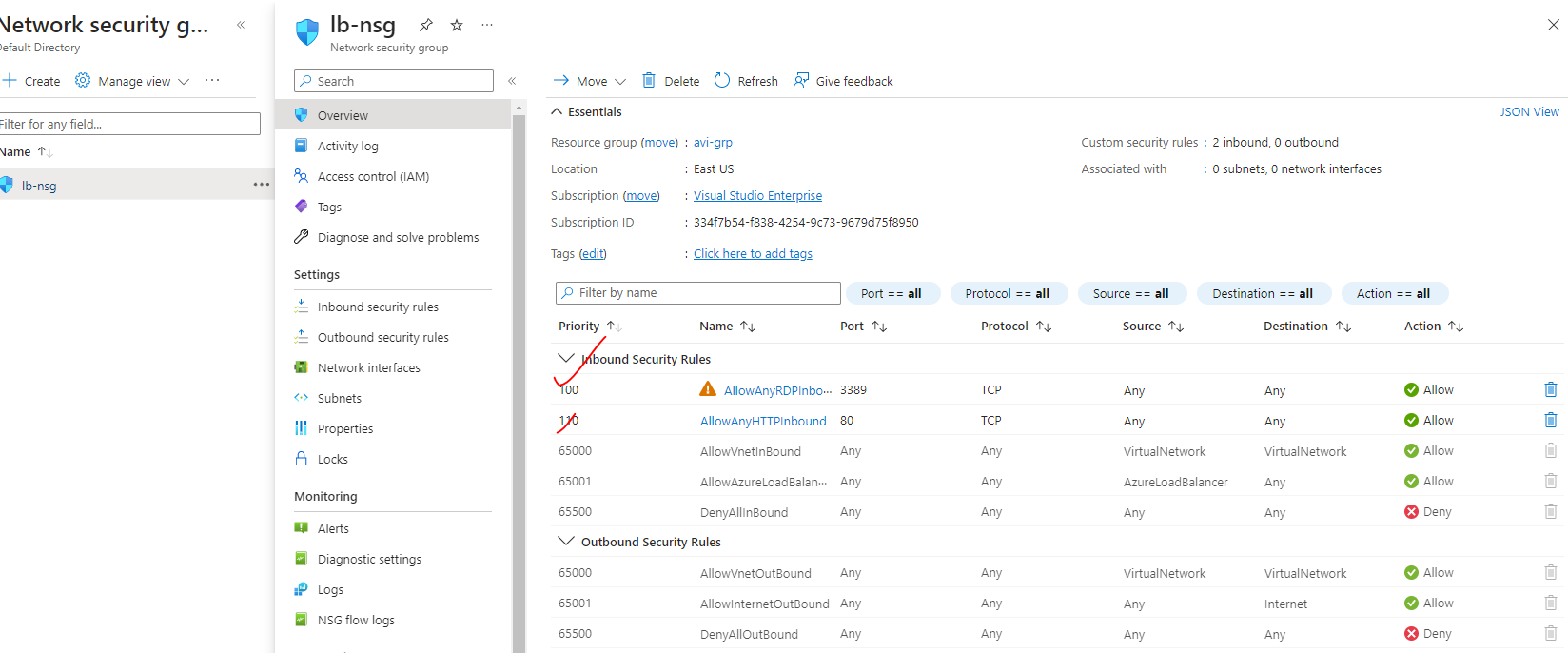
1. Disassociate the NSG from the NIC

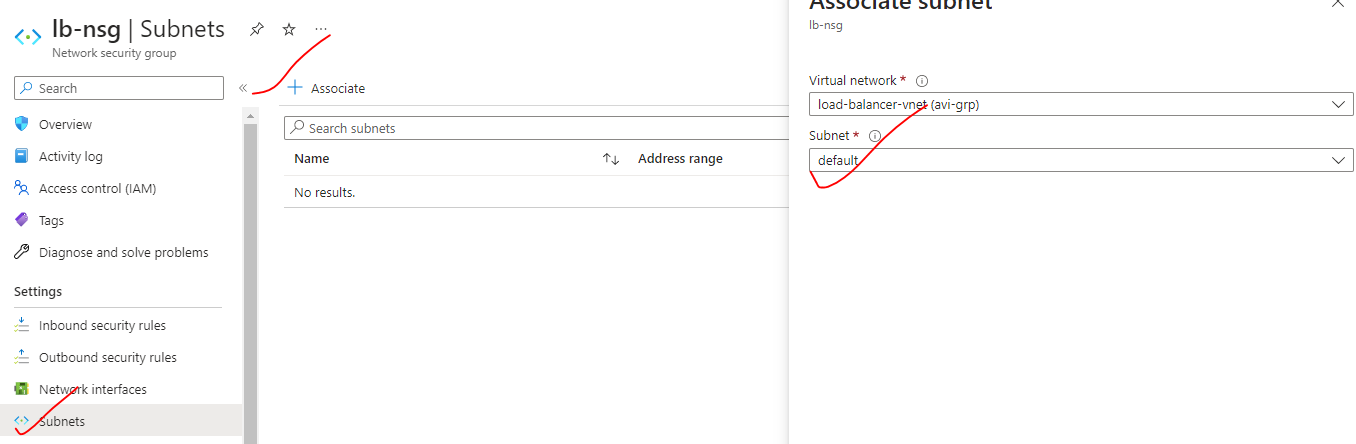


1. Delete the NSG



***STEP 5:*** *Create NSG and attach it to the subnet level of the VM.(Associate the subnet with the NSG)*





***STEP 7:*** *Create and configure a Basic Load Balancer resource. As part of configuration -we need to set up a Backend pool of the VMs created in Step 1*

|  |  |
| --- | --- |
| BASIC | FRONT END CONFIGURATION |
|  | Create the Load balancer  Note – Backend pool configuration can be done while configuring the LB  **TIER CONFIGURATION**  REGIONAL – VMs is a region will be load balanced  **GLOBAL –** VMs across multiple regions can be load balanced. |

##### LOAD BALANCER CONFIGURATIONS

###### CONFIGURING BACKEND POOL

|  |  |
| --- | --- |
|  | Add the VMs in the backend pool |

###### CONFIGURING HEALTH PROBE

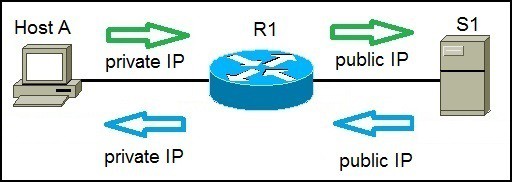
|  |  |
| --- | --- |
|  | * After the interval of every 5 second – Load balancer will initiate a TCP handshake. if the VM acknowledge the TCP request – then the LB will know that VM is healthy |

###### CONFIGURING LOAD BALACING RULES

|  |  |
| --- | --- |
|  | * PORT- 80 - This setting is the port associated with the frontend IP that you want traffic to be distributed based on this load-balancing rule. * **BACKEND PORT** – It’s the Port in the VM which is the PORT of the IIS- This means – If the request comes to port number 80 of the load balancer (normal HTTP web request default port is 80) – then it will be distributing the traffic to backend pool port.   Load balancer rule reference diagram |

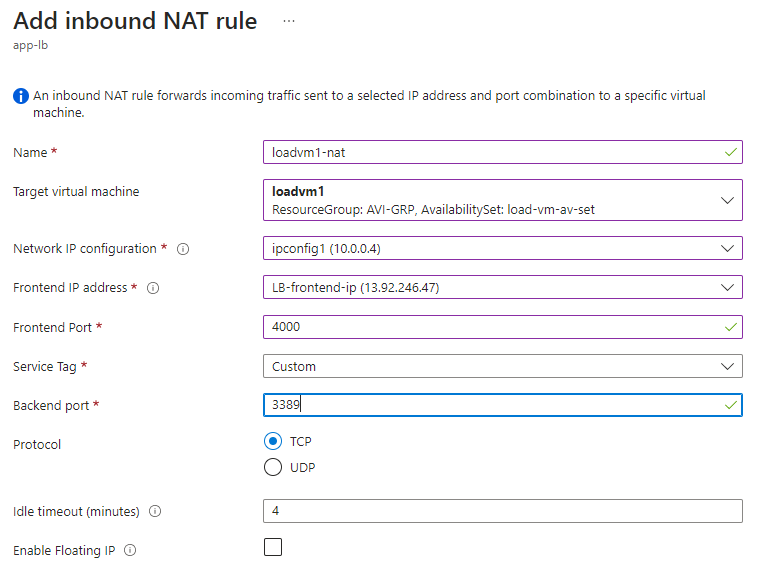
#### LOAD BALANCER NAT RULES

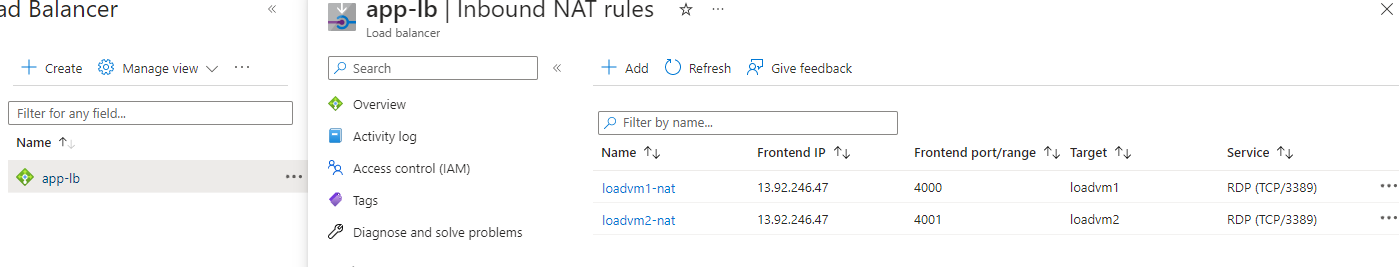
NAT stands for Network Address Translation. It is a technique used in networking to translate IP addresses between different networks. NAT allows devices on a private network with non-routable private IP addresses to communicate with devices on public networks, such as the internet.



|  |  |
| --- | --- |
|  | * As the backend VMs don’t have public IP addresses, we can’t be able to connect these VM to internet or RDP onto these machines. * For such use cases, Load balancer provide option to connect to those VMs using its own public IP address, this is done with the help of ***NAT (Network Address Translation)*** * Go to the load balancer 🡪 Inbound NAT rules 🡪 Add * **Note- We need to add the NAT rules for all the connected VMs** |

* The Frontend port is a random port given for each connected VM. The will be used to connect (RDP) with the VM
* Backend port is the default port number for RPD

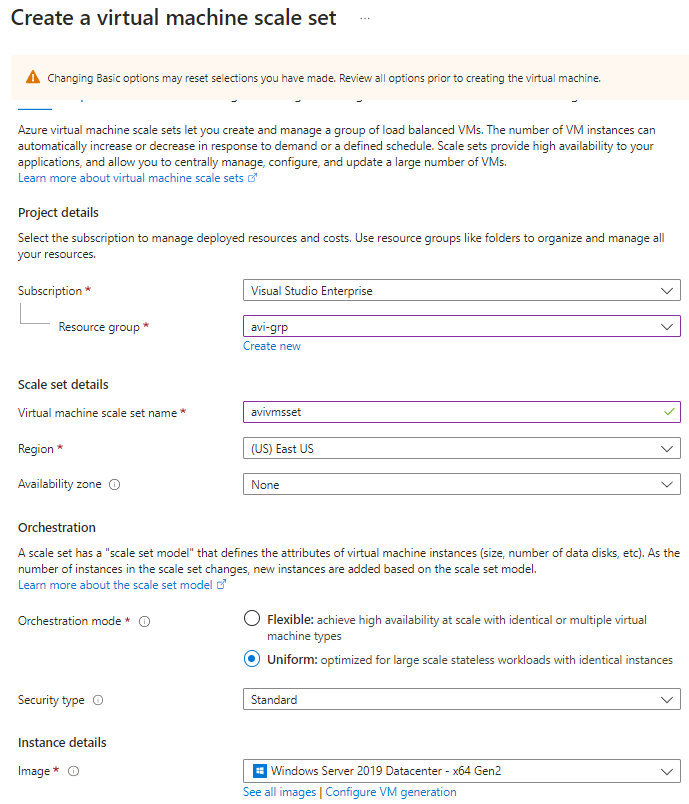




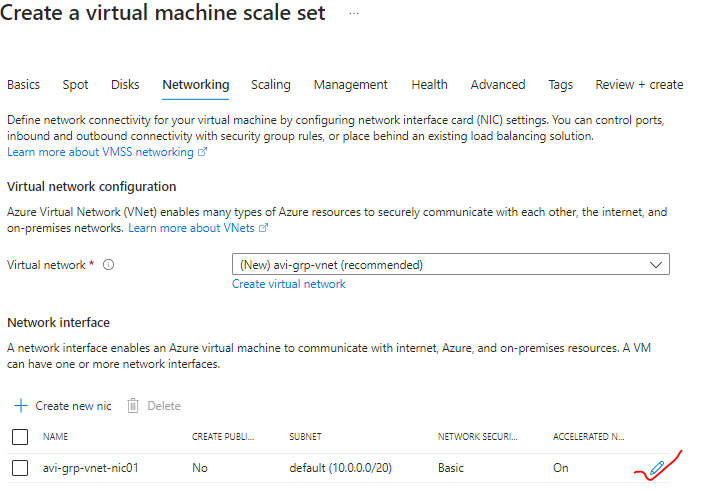
|  |  |
| --- | --- |
|  | * To connect with the respective VM we use   **<LOAD\_BALANCE\_FRONTEND\_IP>:<FRONT END PORT CONFIGURED FOR THE VM>** |

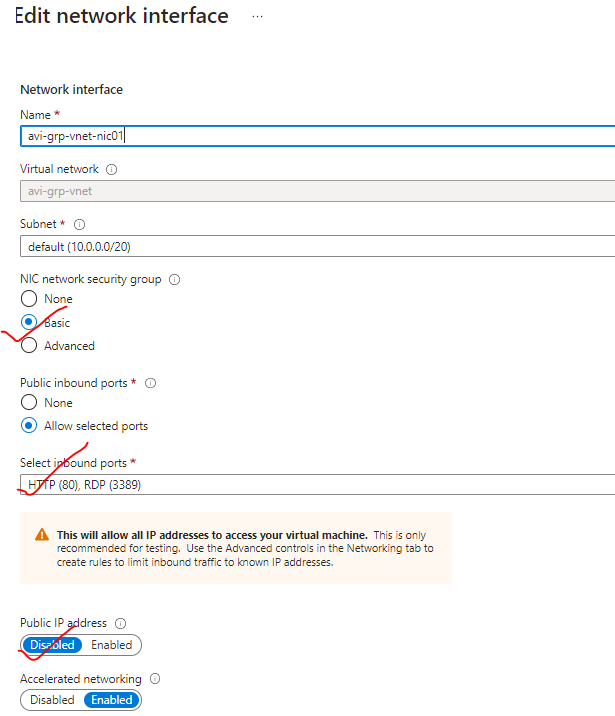
### BASIC LOAD BALANCER (SCALE SET)

#### STEP 1: CREATE A VM SCALE SET



**NETWORKING**: Allow the HTTP & RDP ports





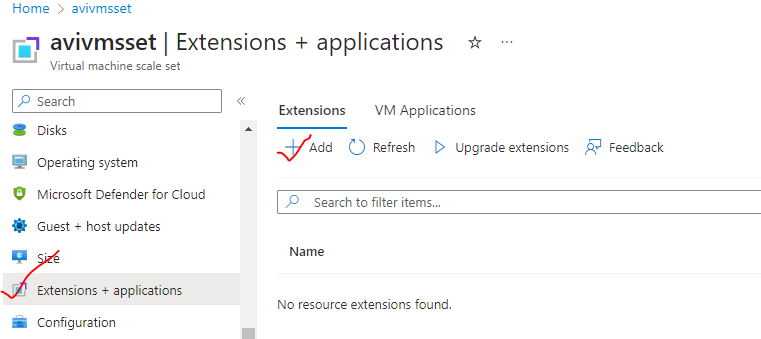
|  |  |
| --- | --- |
|  | * Let’s keep the initial instance count as :1 |
|  |  |

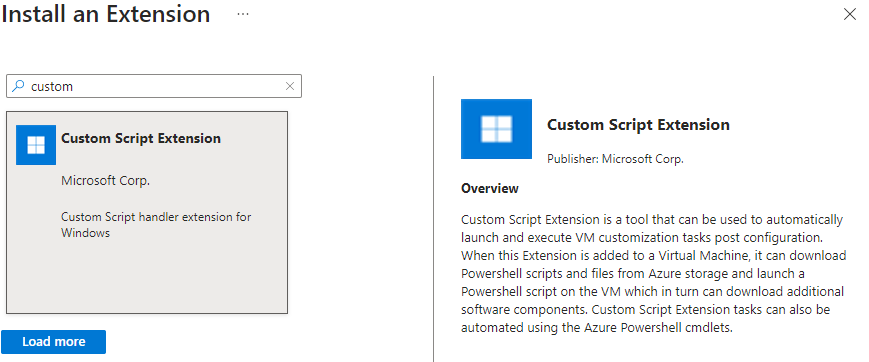
#### STEP 2: UPLOAD THE SCRIPT IN THE STORAGE ACCOUNT

* CREATE STORAGE ACCOUNT
* UPLOAD THE SCRIPT IN THE EXTENSION
* To install IIS on VM scale set – we can make use of ***custom script extension***. For that we need to **Create Storage Account and upload a script to install IIS**

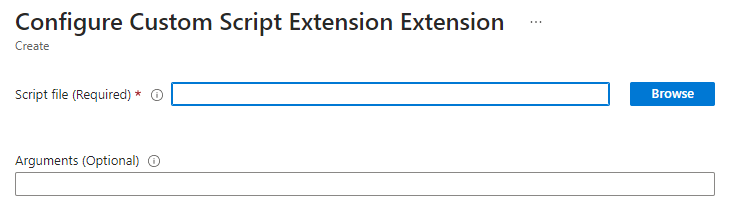
|  |  |
| --- | --- |
| WINDOWS VM (installing IIS)  NEED TO CREATE POWERSHELL FILE – e.g – iis.ps1 | LINUX VM(install nginx)  Need to create Bash file : install\_web.sh |
| import-module servermanager  add-windowsfeature web-server -includeallsubfeature  set-content -path "C:\inetpub\wwwroot\Default.html" -Value "<h1>Sever name: $($env:computername)</h1>" | apt-get update -y && apt-get upgrade -y  apt-get install -y nginx |

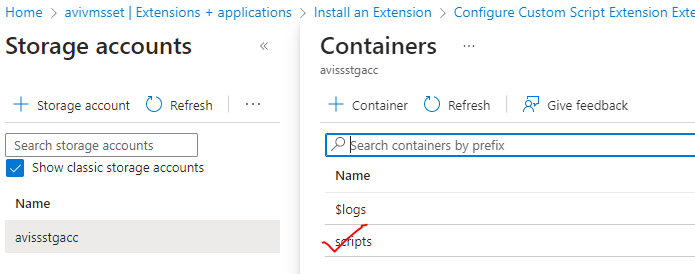
#### STEP 3: INSTALL CUSTOM SCRIPT EXTENSION AND RUN



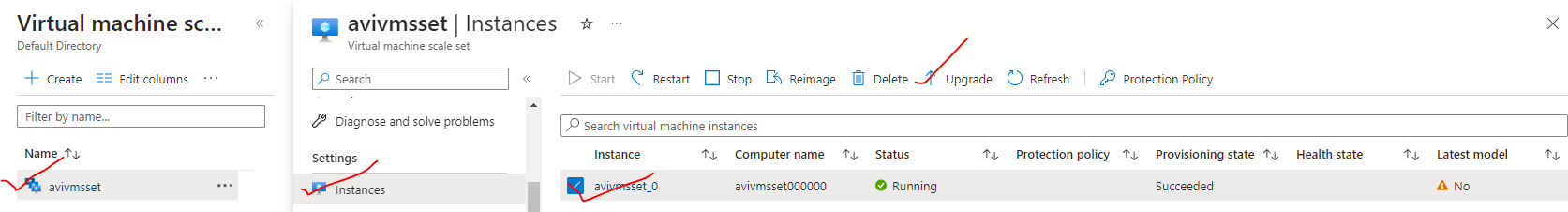


1. Browse the Storage account container (scripts)
2. Then select the script 🡪 Create

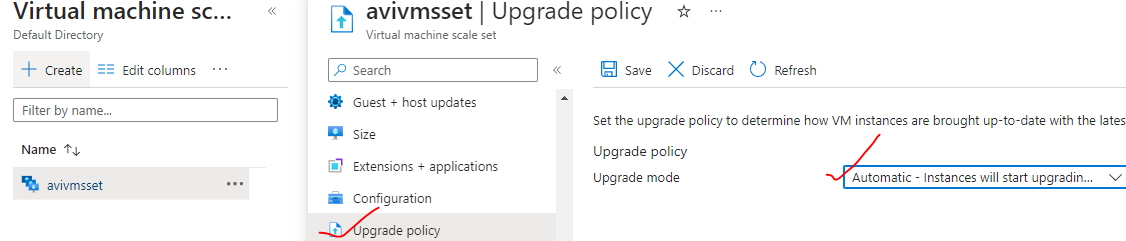




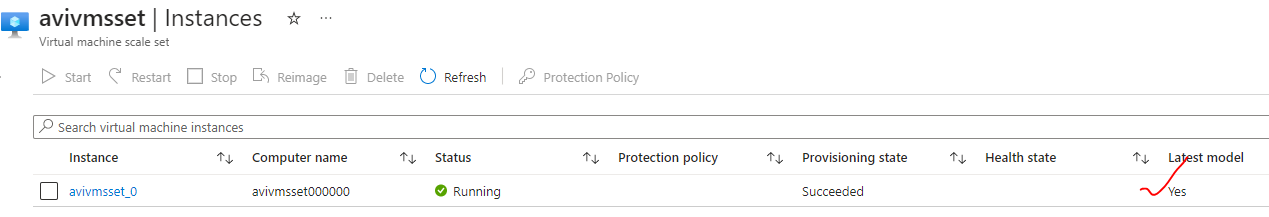
#### STEP 4: UPGRADE THE SCALE SET TO APPLY CUSTOM SCRIPT CHANGES



* **SET THE UPGRADE POLICY TO BE AUTOMATIC**



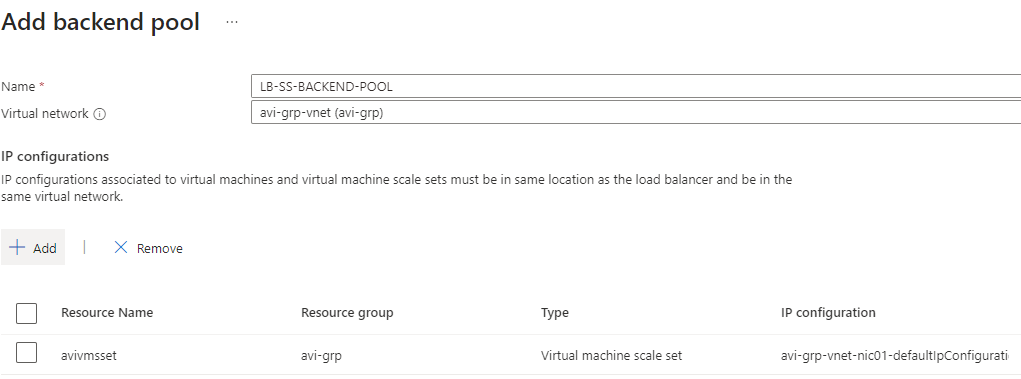
Since we setup the upgrade policy to Automatic – the scale set is showing Latest model as “Yes”



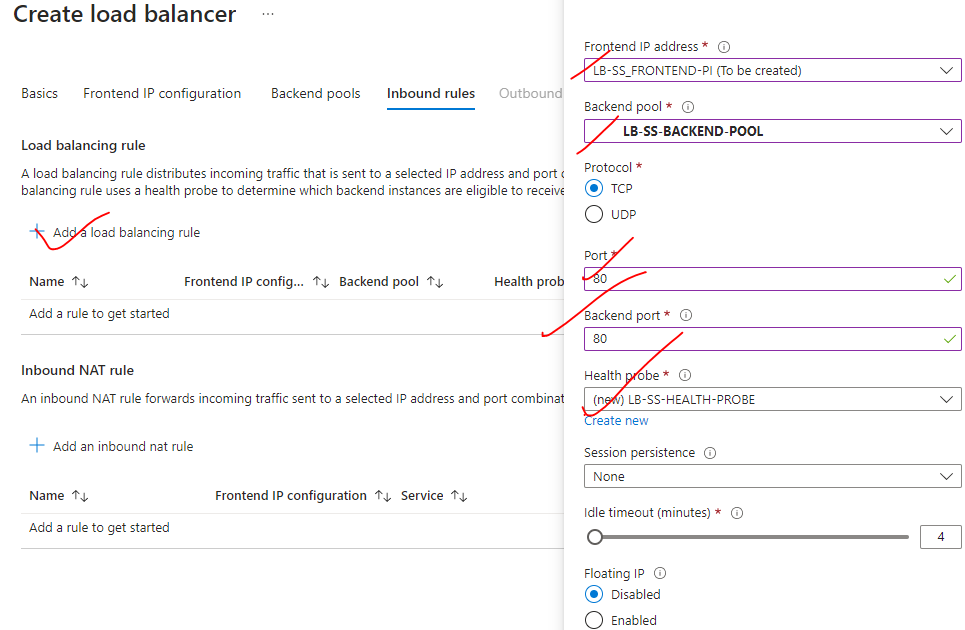
#### STEP 5: SET UP THE LOAD BALANCER

Create a basic load balancer. Configure the

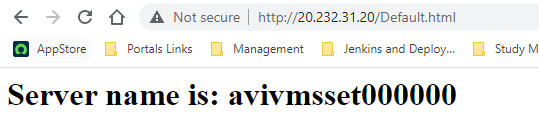
1. Frontend IP
2. Public IP
3. Backend pool as VM scale set



1. Set up the load balancing rule



#### STEP 6: ACCESS THE VM IN THE SCALE SET VIA LB FRONT END IP

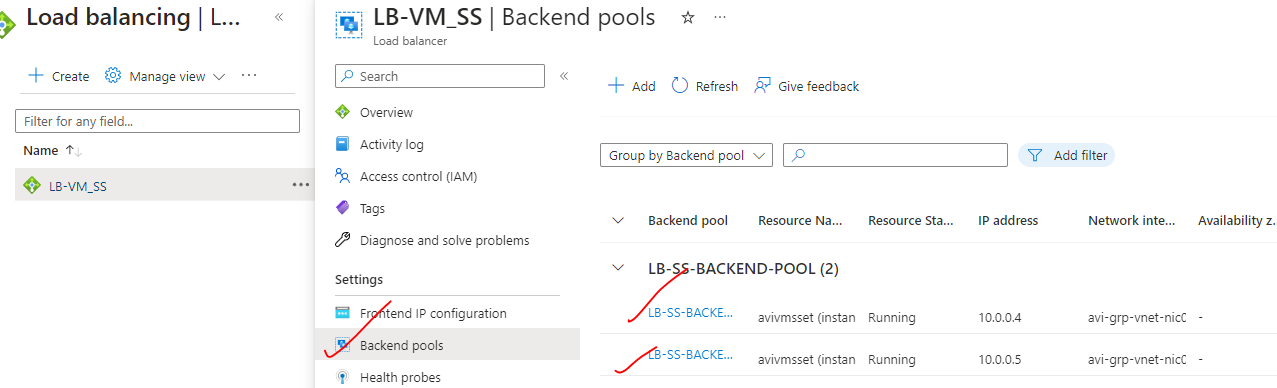


#### STEP 7: ADDING INSTANCES TO VM SCALE SET (MANUAL SCALING)

* While setting up the scale set we added just one instance to the scale set.
* To validate – whether the load balancer is load balancing the traffic withing the VMs of the Scale set – we can more instances.
* The new instance will be added on to the virtual machine scale set for that new instance. The custom script extension, installing internet information services and having the Default.html page in place at the same time.
* Along with that - the instance will be also reachable from the Azure load balancer.
* In the below example – lets increase the instance count = 2

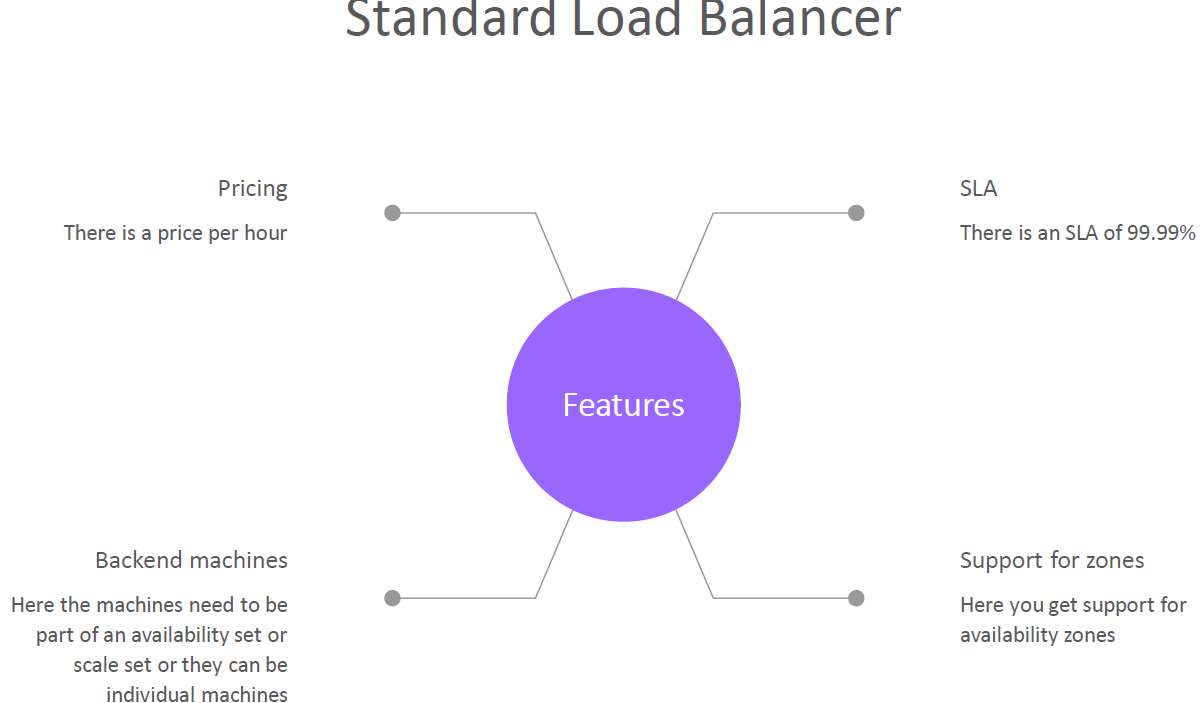


The instance will automatically get added to the backend pool of the load balancer



### STANDARD LOAD BALANCER

* Comparison between Basic and Standard load balancer: <https://learn.microsoft.com/en-us/azure/load-balancer/skus#skus>
* For high availability – we should choose Standard load balancer.



|  |  |
| --- | --- |
| BASIC LOAD BALANCER | STANDARD LOAD BALANCER |
| The VM must be part of **VM Scale set or Availability set** | ***Any virtual machines or virtual machine scale sets in a single virtual network***. They need not to be part of VMSS or Availability Set |

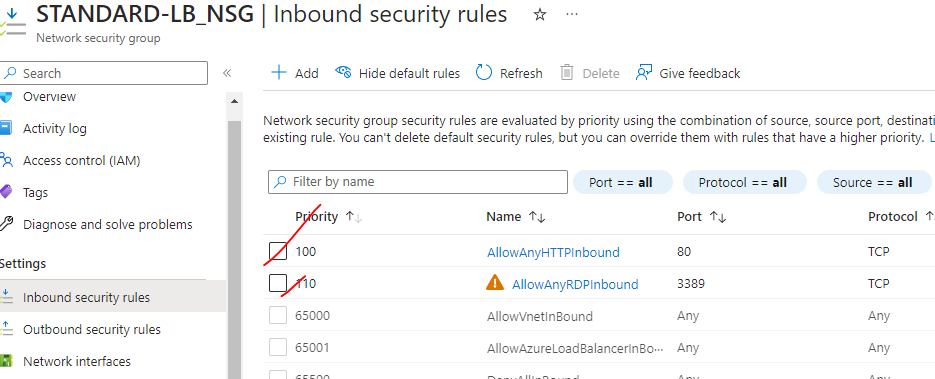
#### SETTING UP STANDARD LOAD BALANCER

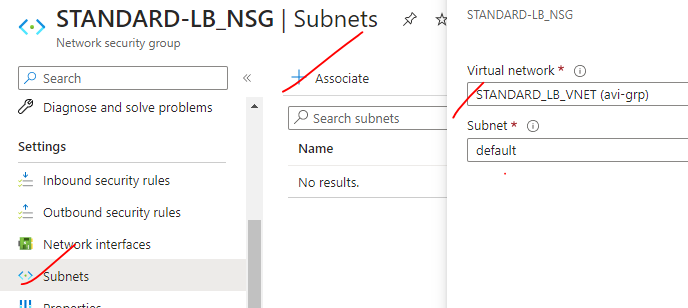
1. **Step 1: Create a Storage account and upload the custom script to install IIS to all VMs**

|  |  |
| --- | --- |
|  | **Step 2: Create 2 VMs**   1. The VMs will have no public IP address. 2. The VMs will have no NSG attached – we will create NSG as separate resource and attach it at subnet level. |
|  | **Step 3:**   1. Install the custom script extension. 2. Browse to the custom script in the storage account container |

**Step 4: Create the NSG**

1. **Add the inbound rule for HTTP and RDP traffic.**
2. **Attach it to the Subnet**

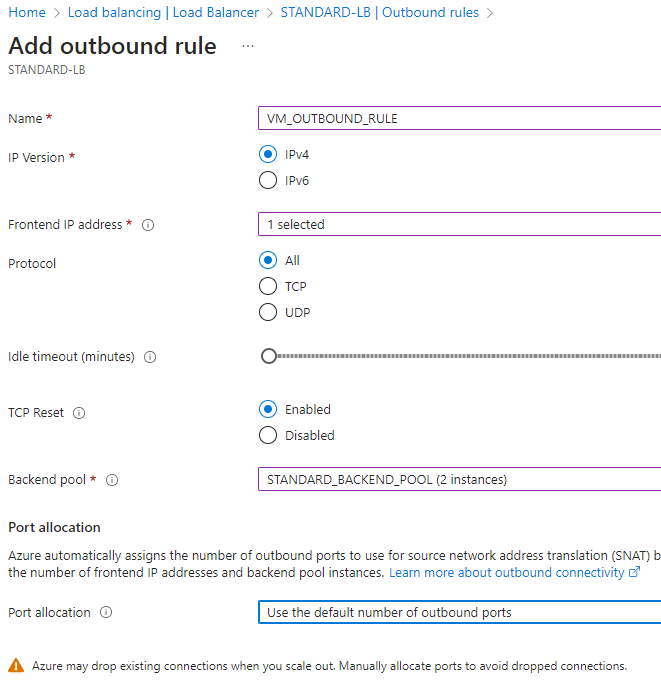




|  |  |  |
| --- | --- | --- |
|  | **STEP 5: CREATE THE LOAD BALANCER**  **Note – The Standard load balancer supports HTTPS health probe too.**  ***What's the difference between the TCP protocol and HTTP protocol in the standard load balancer?***  In HTTP health probe - The probe is going to make a proper HTTP request onto the path configured in the path. If it gets a response back, then it will consider backend instance has healthy. | |
|  | For the Standard load balancer:   * The public IP should also be of Standard SKU * The standard SKU creates a “***static public IP address***” | |
| **SET UP THE LOAD BALANCING RULE** |  | |
|  | | **INBOUND NAT RULES**   * To access the VMs using RDP – we need to configure the NAT rules. * We need to use the frontend IP of the Load balance for RDP   **<Public\_ip>:<frontendport of vm>**   * By default, the outbound connection from the standard LB to internet will be disabled – which must be enabled explicitly |
|  | |  |

#### OUTBOUND RULE

* When we access the VM via LB’s public IP. The outgoing internet connection from the VM are disabled.
* To enable - ***Go to the Outbound rules of the LB*** 🡪 configure the following.

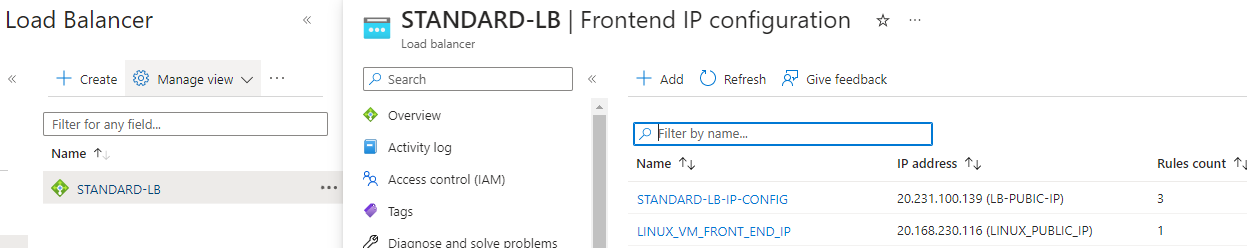


#### MULTIPLE BACKEND POOL

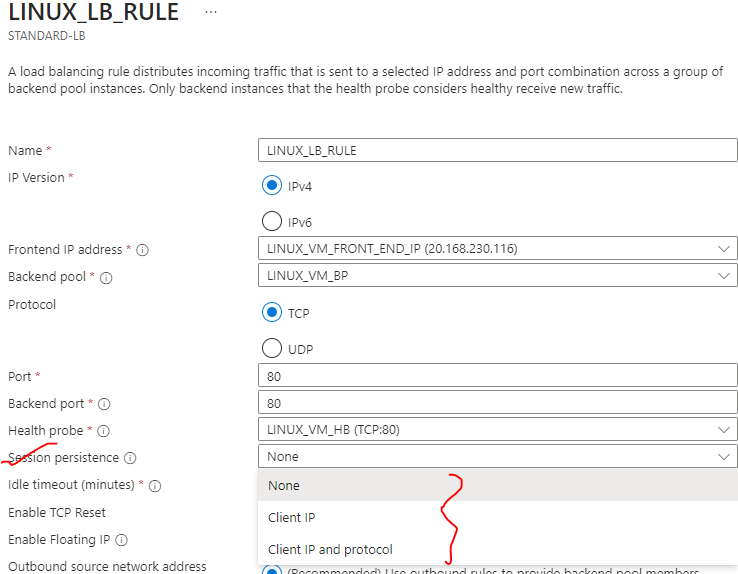
|  |  |
| --- | --- |
|  | * **The standard load balancer can have multiple backend pool.** * **To route the traffic to respective backend pool we can have dedicated frontend IP address.** |

To configure the Multiple backend pool

|  |  |
| --- | --- |
|  | Create a VM that can be added to the backend pool. |
|  | CREATE NEW FRONTEND IP FOR NEW BACKEND POOL |
|  | CREATE A NEW HEALTH PROBE FOR NEW BACKEND POOL |
|  | CREATE A NEW LOAD BALANCING RULE FOR NEW BACKEND POOL |



#### LOAD BALANCER SESSION PERSISTENCE

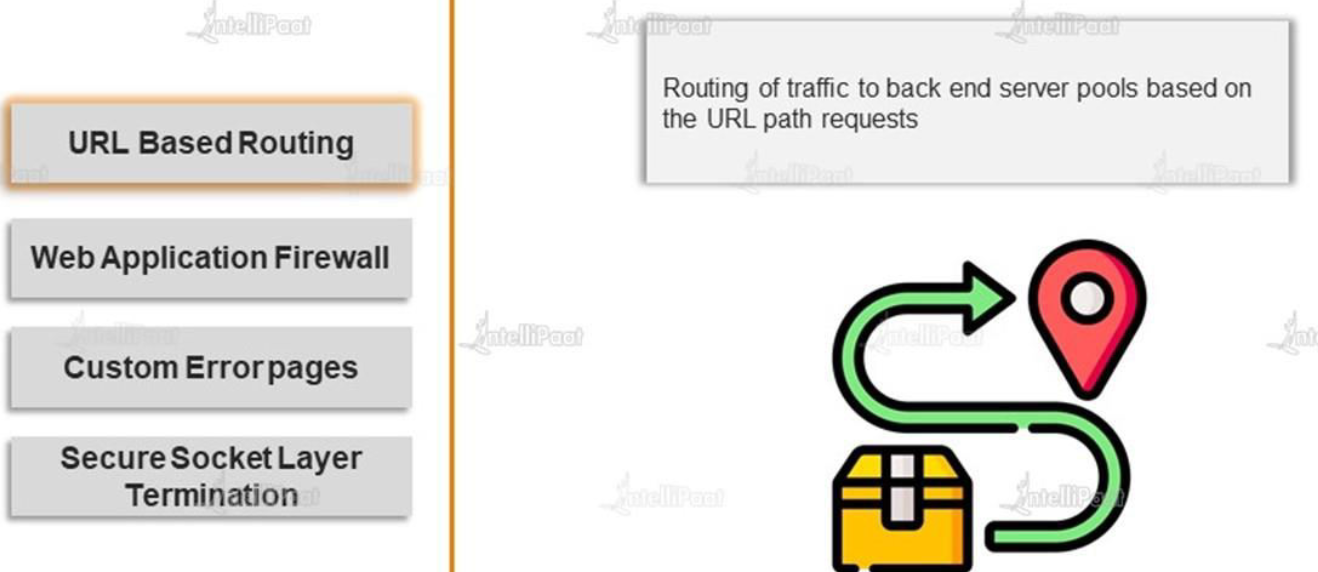


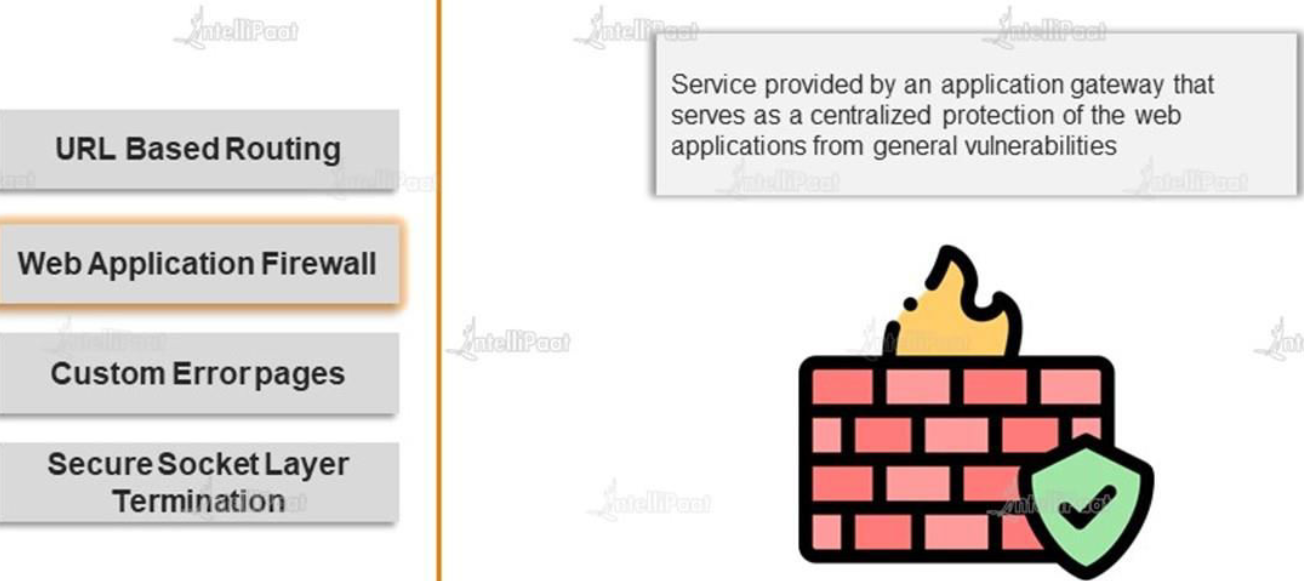
* Let's say that we have a user that has made a request via the load balancer on its frontend public IP address and the user has been directed on to load VM-1.
* If we **enable session persistence**, the load balancer will kind of create an affinity to the session between the load balancer and the client. Hence, this particular user for that particular session will always be directed onto load vm-1
* **ADVANTAGE**
  + Helps in better performance for the sessions itself. - if we don't want in a particular session the request to be distributed across the VMs
* **DISADVANTAGE**
  + But the disadvantage is if we have long sessions in those sessions – all the requests are directed towards loadvm-1 causing more load in loadvm-1) but load Vm2 may not be having that much load

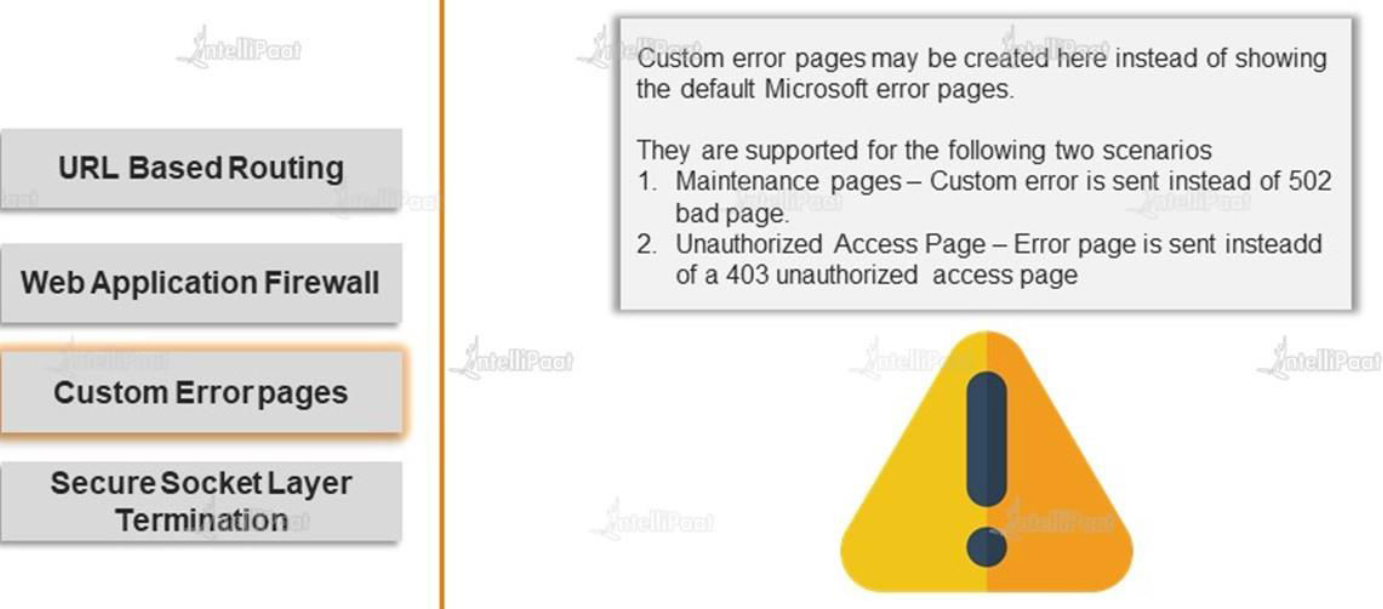
## AZURE APPLICATION GATEWAY

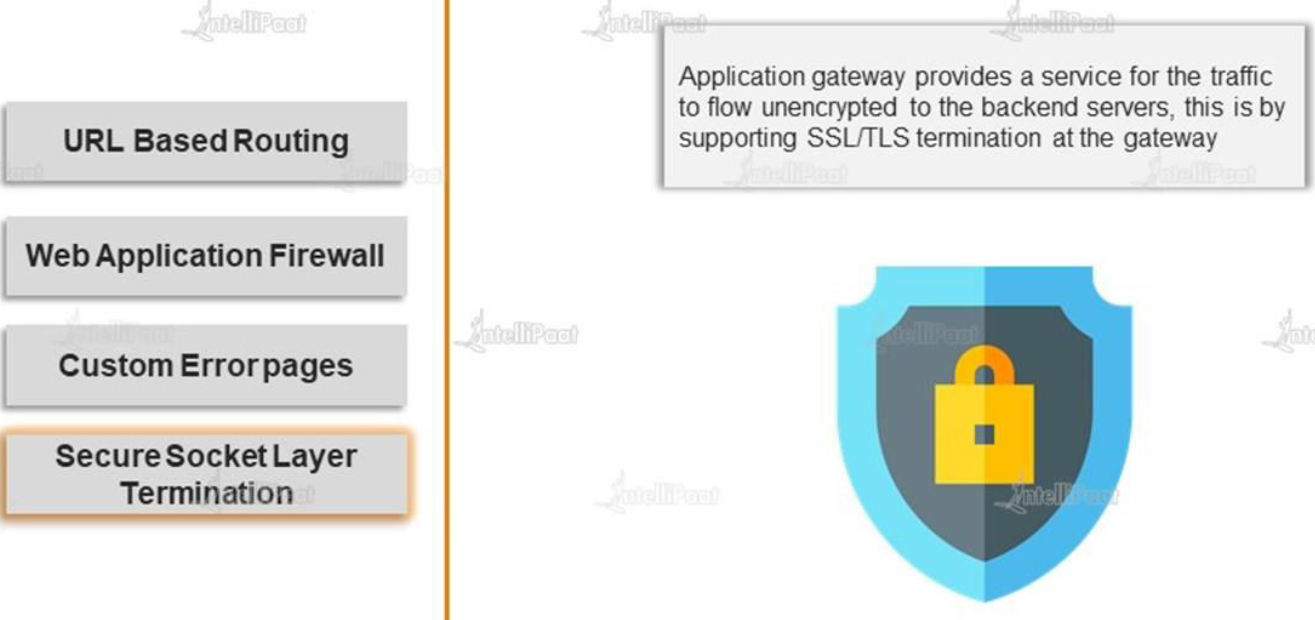
* Application gateway is another load balancing solution on the Azure platform.
* It is a web traffic load balancer that can be used for managing web traffic for the applications.
* ***This is a layer -7 load balancer***.
* We can make routing decisions based on the **HTTP attributes.**

### APPLICATION GATEWAY FEATURES



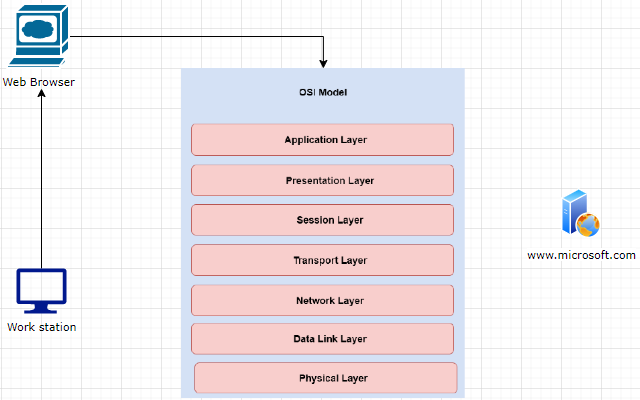


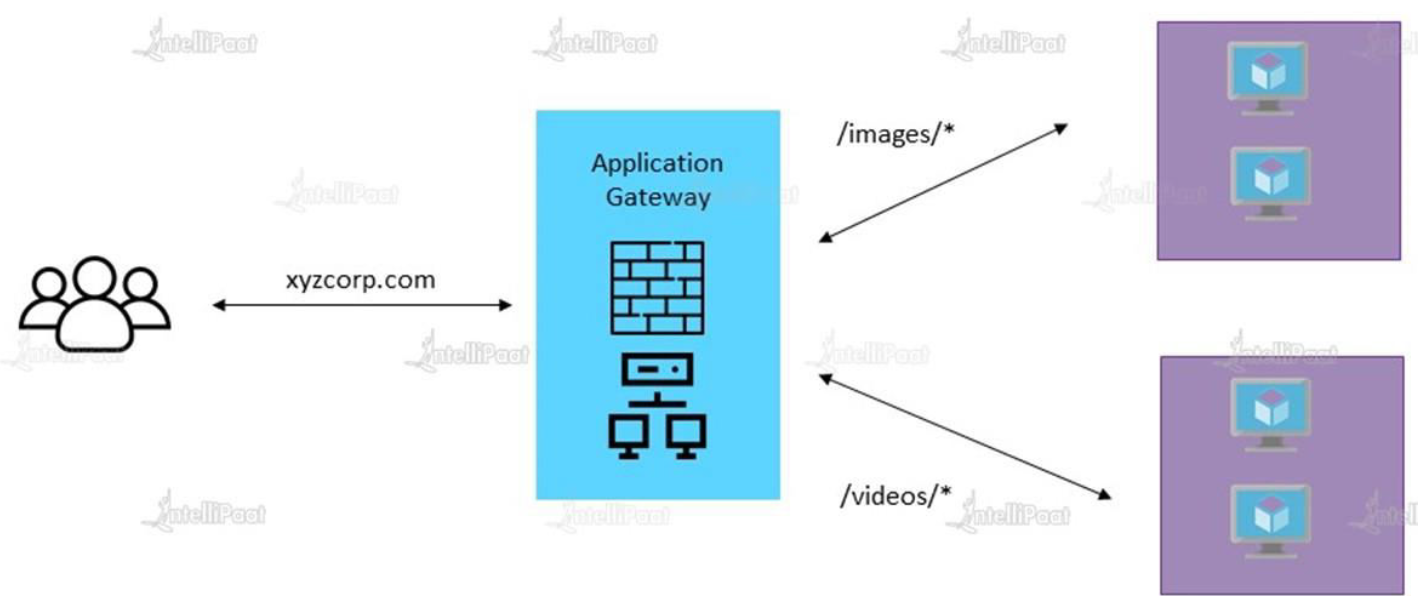




### APPLICATION GATEWAY – PATH BASED ROUTING

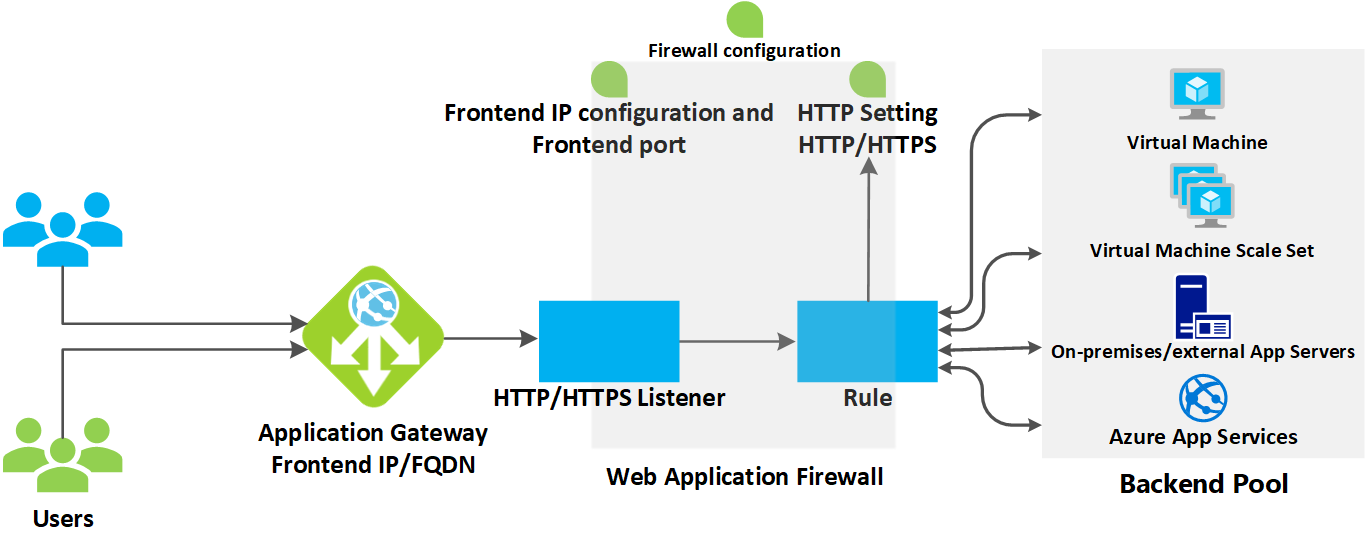
* As App gateway is a layer 7 load-balancer. It will all different HTTP request attributes. Hence this information helps in routing the traffic based on path





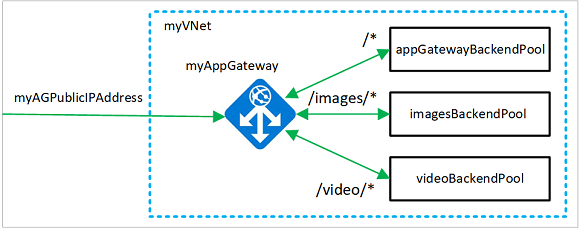
* **In the Azure Application Gateway – we have another feature which we can enable known as the Web Application firewall to protect the web applications from Internet based attacks.**

### APPLICATION GATEWAY COMPONENTS



|  |  |
| --- | --- |
| **BACKEND POOL** | In the backend pool we can have   1. **VM & VM SCALE SETS** 2. **WEB APP** 3. **ON PREM SERVERS** |
| **FRONTEND-IP** | * Public IP of load balancer |
| **HTTP/HTTPS LISTENER** | * The application gateway is going to listen for requests on the front end IP address |
| **RULES** | * These rules decide how the traffic is an routed to backend pool. |
| **HTTP/HTTPS SETTINGS** |  |

### SETTING UP APPLICATION GATEWAY – URL ROUTING



#### STEP 1: SET THE VMS

* Create 2 VMs. Let’s call it as imagesVm and videos Vm
* Install IIS in both VMs .
* Create a images folder in imagevm and place Default.html
* Create a videos folder in videovm and place Default.html

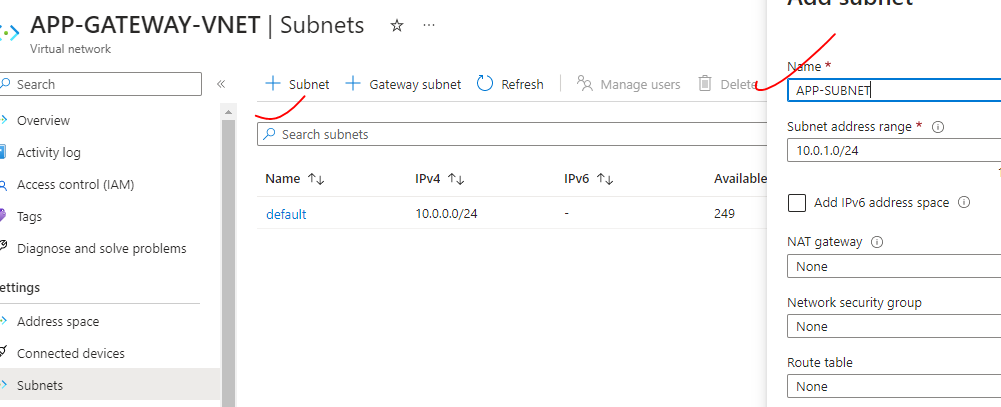
***With this set up – request via Application gateway frontend ip to the path “/images” will be routed to “imagevm” and “/videos” will be routes to videovm***

#### STEP 2: CREATE EMPTY SUBNET FOR APPLICATION GATEWAY

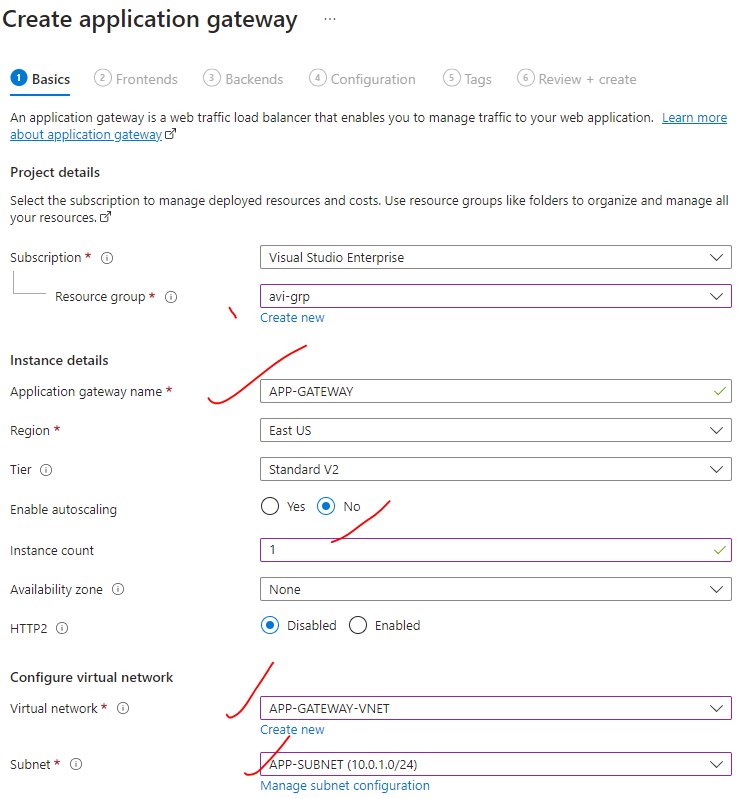
* Before deploying Application Gateway - we have to add a new subnet (empty subnet)in the VNET, (Note - VMS are deployed in default subnet)

NOTE:

* The Application Gateway is going to deploy its own compute machines, its own resources on to this empty subnet. These resources are required for the routing of traffic to the back-pool.
* Since App gateway is layer -7 load balancer - that means it needs to process the request, and then do the routing accordingly. That’s why it need an infrastructure (managed by Azure itself)

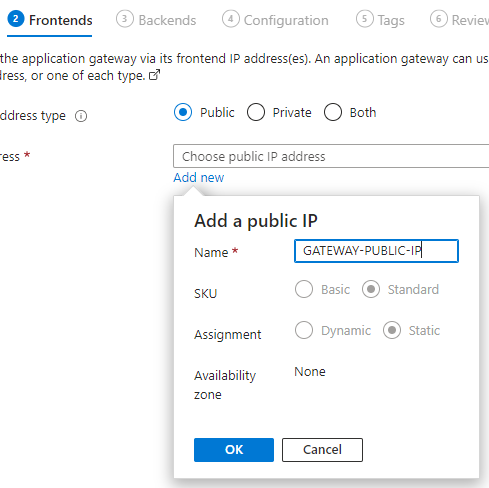


#### STEP 3: DEPLOY APPLICATION GATEWAY

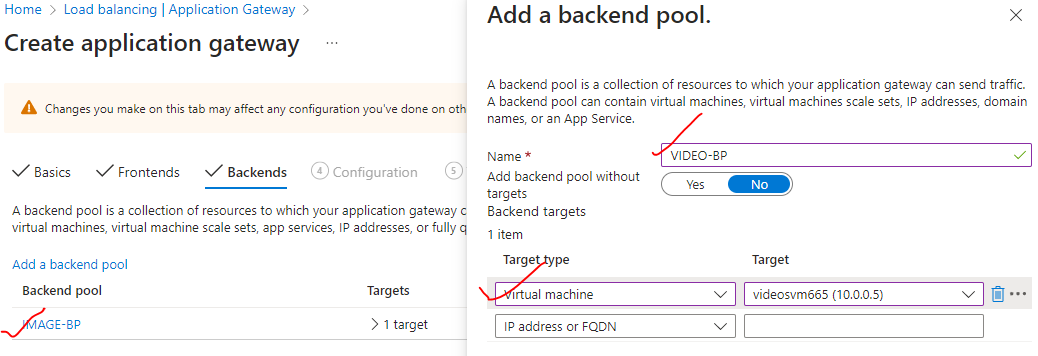


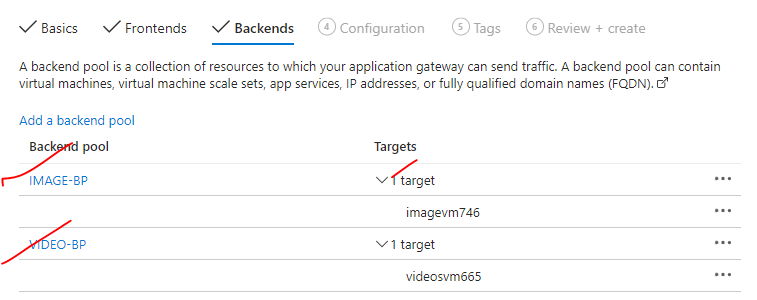
|  |  |
| --- | --- |
| AUTOSCALING | * The Application Gateway is going to deploy its own compute machine as part of subnet. * Hence – as part of autoscaling - if the demand increases-i.e. if the number of requests on the Azure Application Gateway service increases, then it can go ahead and automatically scale up   the amount of compute resources. |

##### SET UP PUBLIC IP ADDRESS

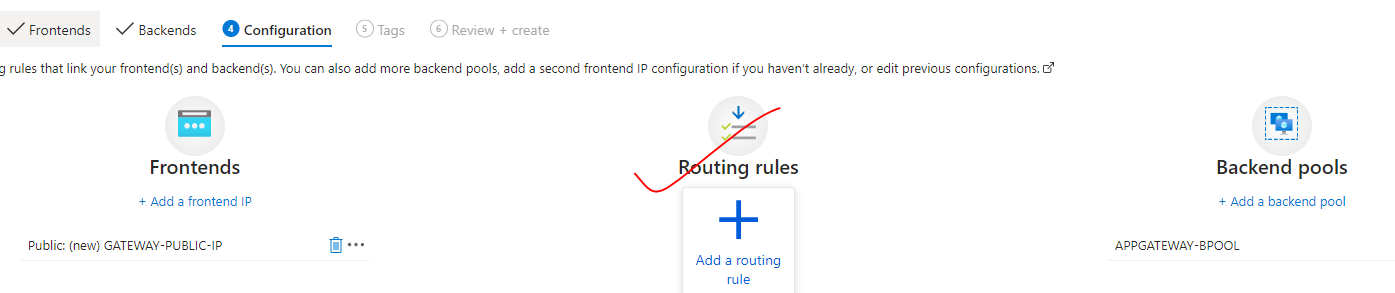


##### BACKEND POOL





##### SET UP THE ROUTING RULES



|  |  |
| --- | --- |
|  | * So here we have two types of listeners.   + BASIC LISTENER: Listens to just one domain name   + MULTISITE: Listens on multiple domain names.   Path Configuration |
|  | |

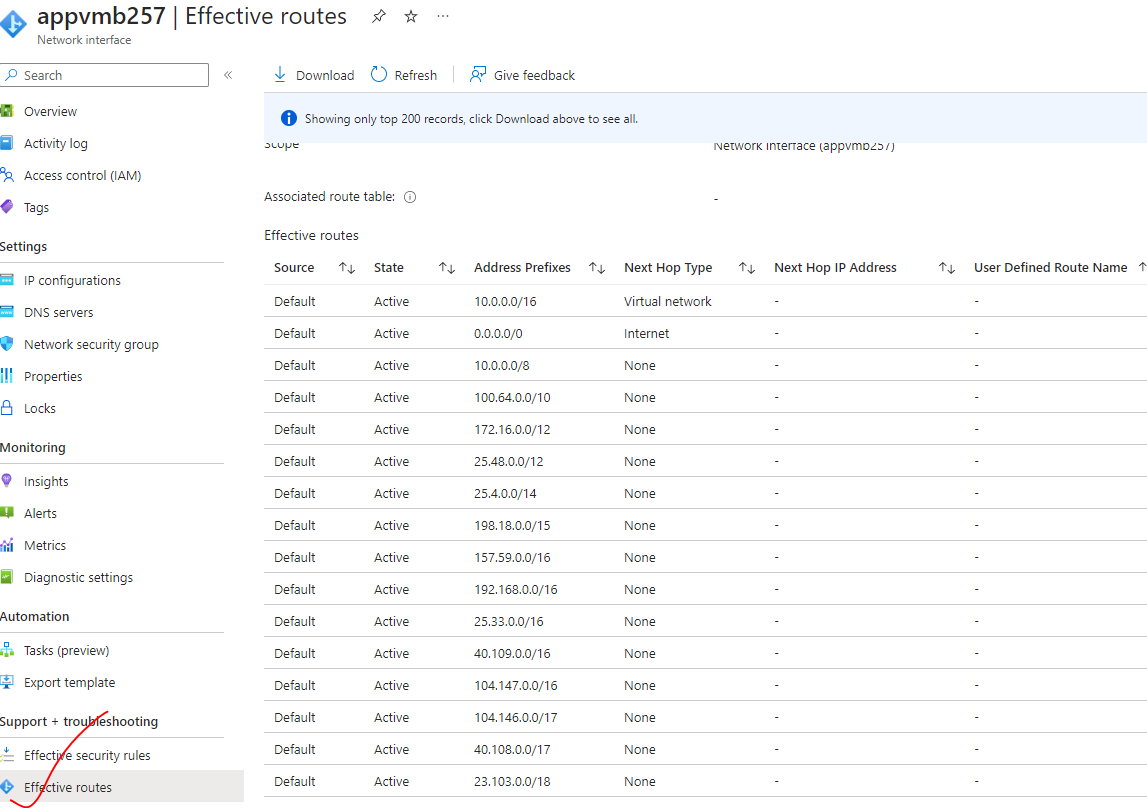
## APPLICATION GATEWAY – MULTIPLE SITES

## USER DEFINED ROUTES

* In a virtual network- which has multiple subnets – we can have VMs located in each subnet. By default, they can communicate with each other(using private IP).
* In the VNET - this is a default system route that tells the packets of information how to be routed onto or how

to be routed from the source on to the destination virtual machine within the same virtual network itself.

* These default system routes are configured at the Network interface of the VMs. Similar to below we can create our own routes.



### USECASE - SETTING UP USER DEFINED ROUTE

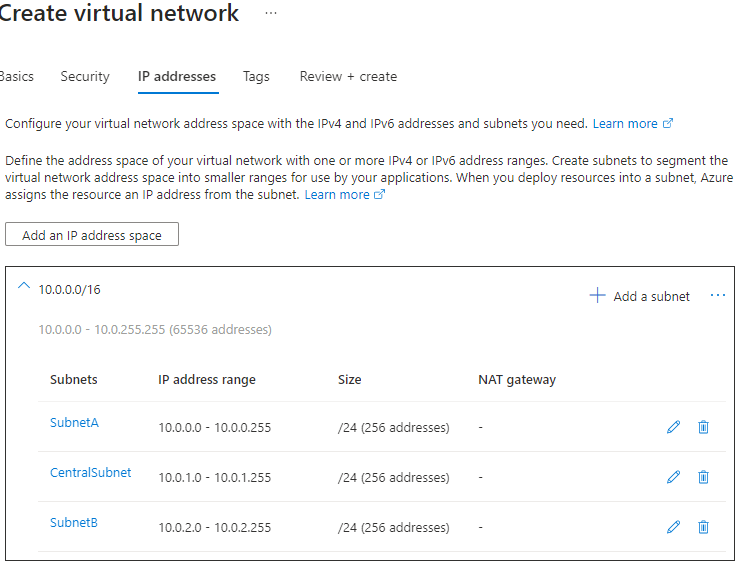
|  |  |
| --- | --- |
|  | * **The objective of setting up the user defined route is that traffic from appvmA to appVmB has to go via centralVm** * The can be accomplished using User defined route * **Note – All the VM within a VNET can communicate with each other via their private IP address** * We don’t have to enable the NSG for PORT “80” as within the the VNET . This VNET rule is need when the VM are getting accessed from internet. |
| * The “AllowVnetBound” is the default NSG rule which allow the communication with VNET (even with private IP address) | |

#### STEP TO SETTING UP USER DEFINED ROUTE

1. CREATE A VNET (APP-VNET)
2. CREATE SUBNET AND ATACH A VM TO EACH SUBNET

|  |  |  |
| --- | --- | --- |
| VMs | SUBNETs | PUBLIC IP ON VM? |
| appVmA | SubnetA | No (private ip - 10.0.0.4) |
| appVMB | SubnetB | No( private ip - 10.0.2.4) |
| CentralVM | CentralSubnet | Yes + INSTALL IIS(private ip - 10.0.1.4) |

#### STEP 2: CREATE THE VNET



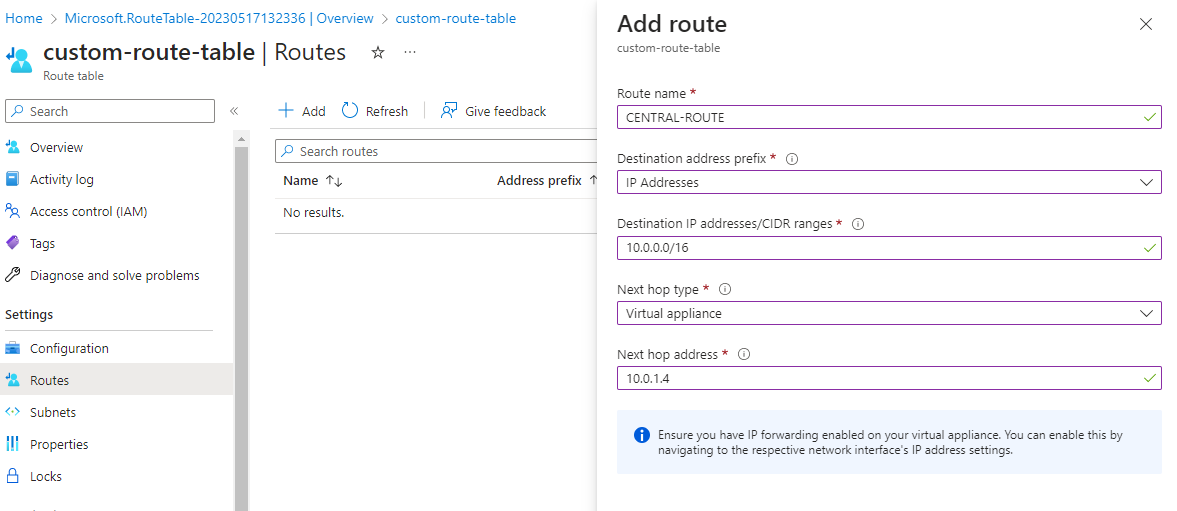
#### STEP 3: CREATE THE ROUTE TABLE

* Create a “Route table” resource.
* Defined the route in the route table
* Associate the rout with the subnet

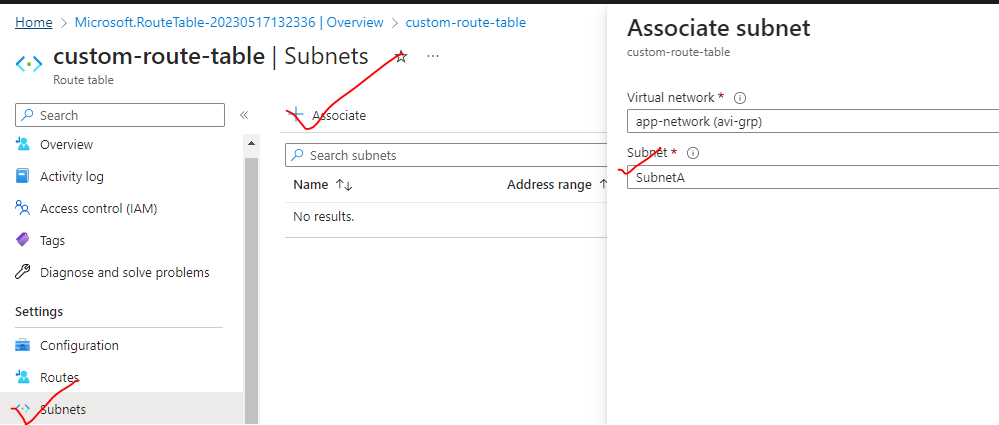
CREATE A ROUTE TABLE



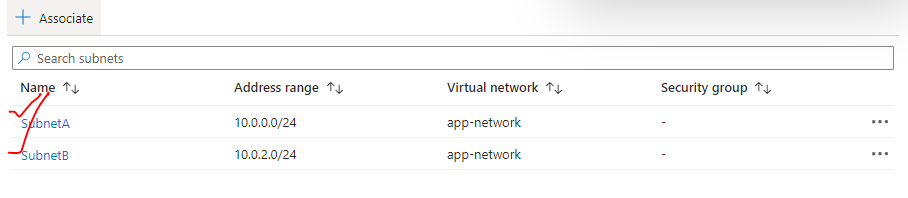
ADDING A ROUTE



ASSOCIATE A SUBNET WITH THE ROUTE



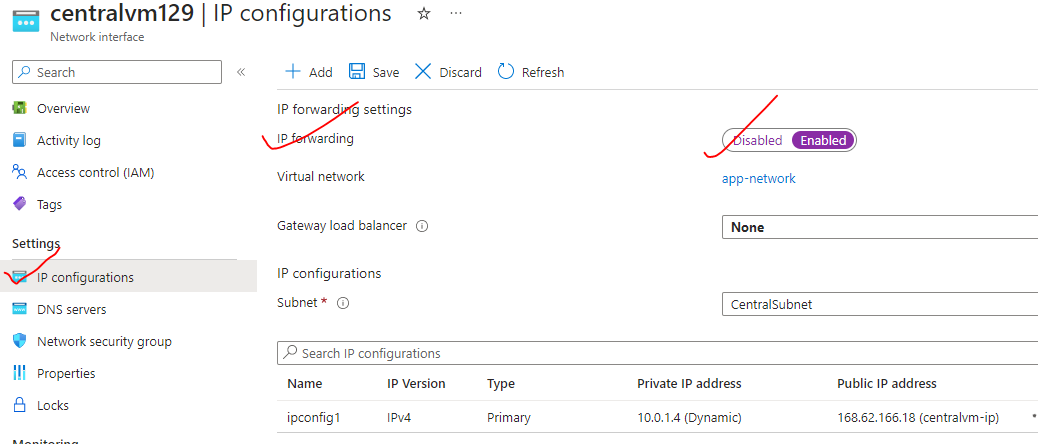
* Associate both SubnetA and SubnetB with the route



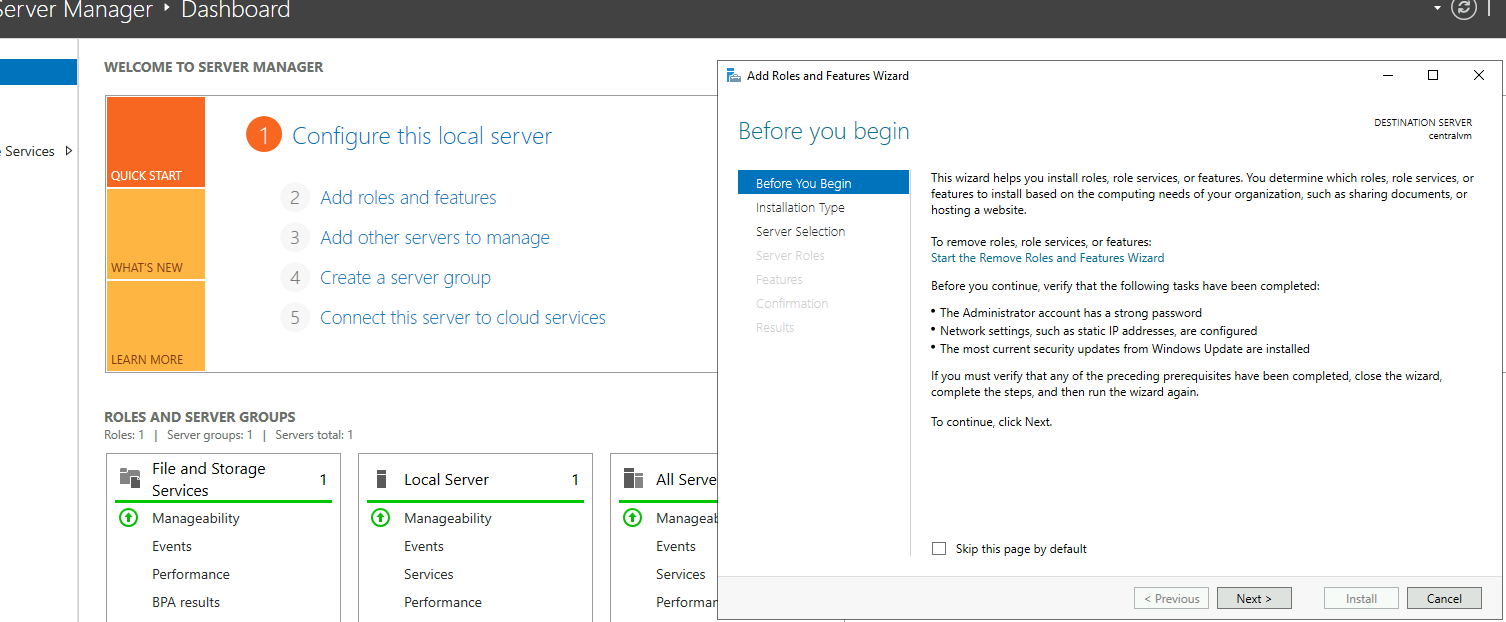
#### STEP 4: ENABLE FORWARDING

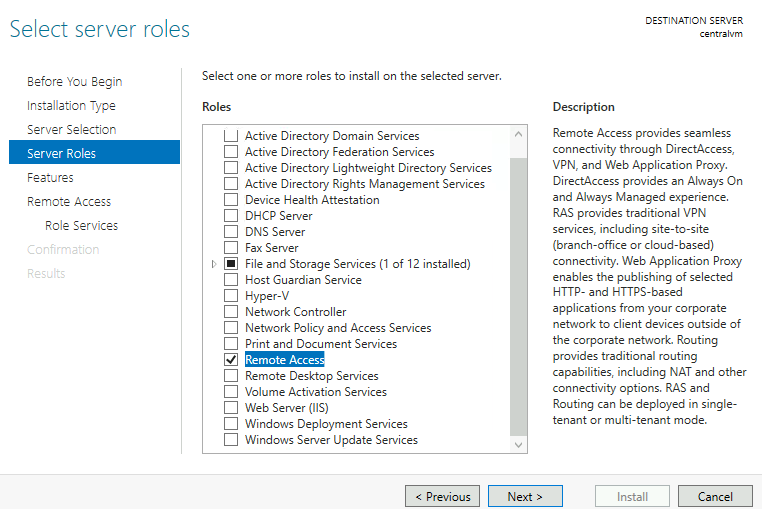
* In this step – we need to forward the request from central VM to appvmB(destination)

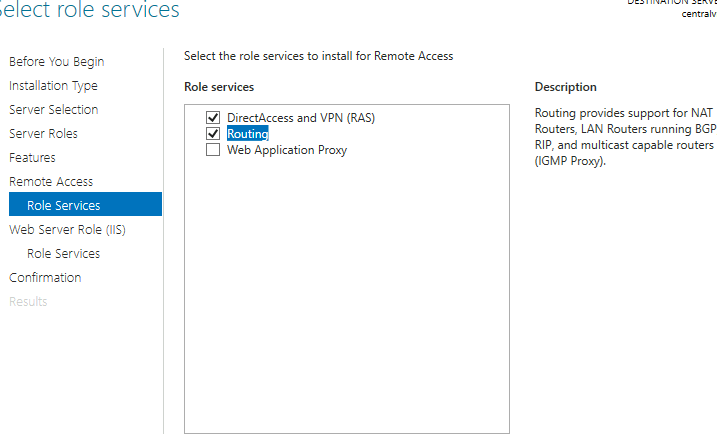
ENABLE EVENT FORWARDING ON CENTRAL VM

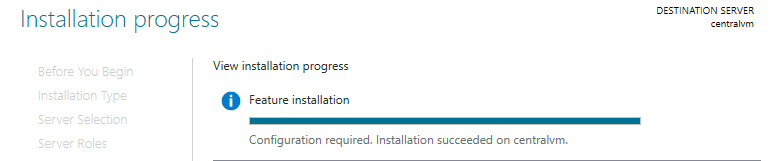


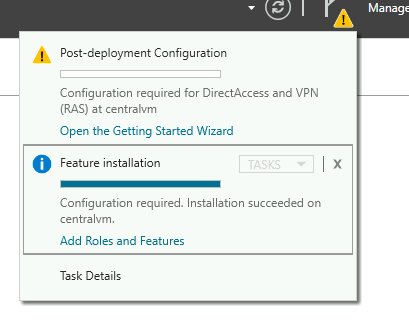
ENABLE EVENT FORWARDING ON CENTRAL VM

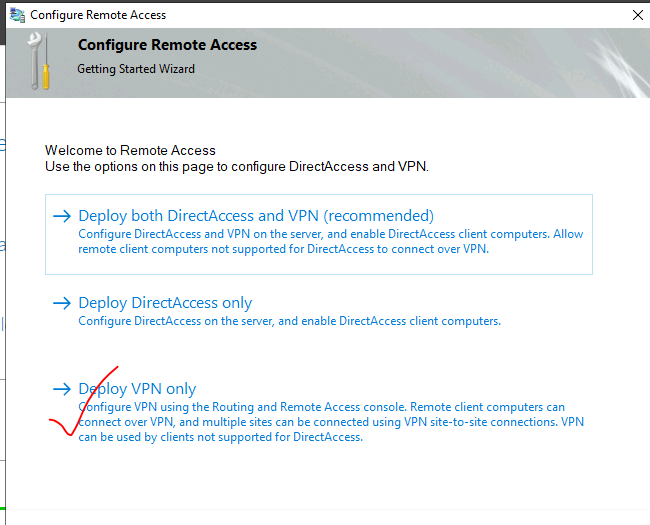


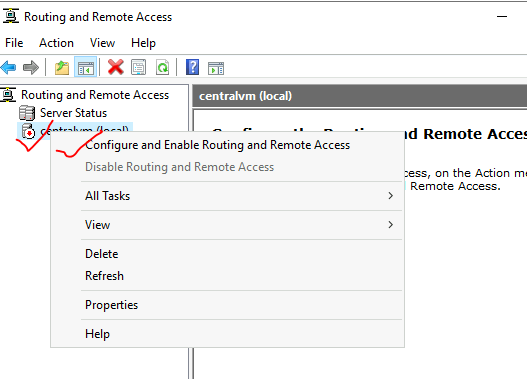


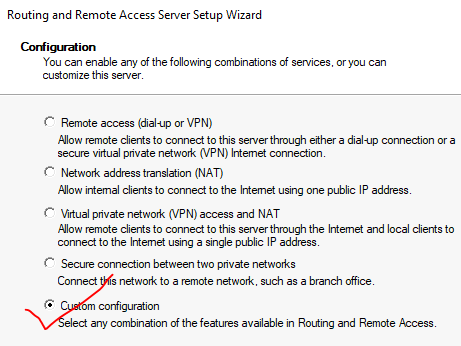


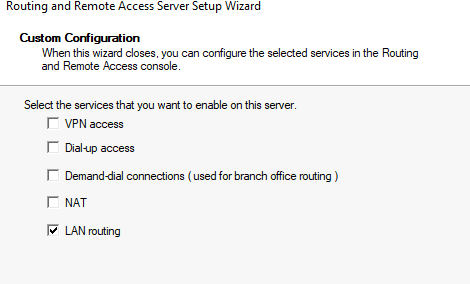


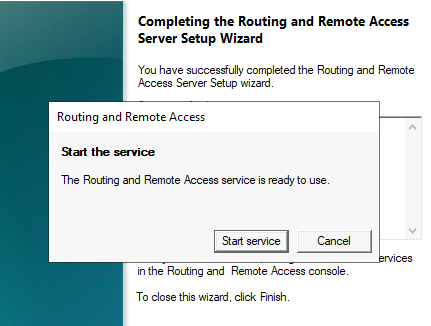


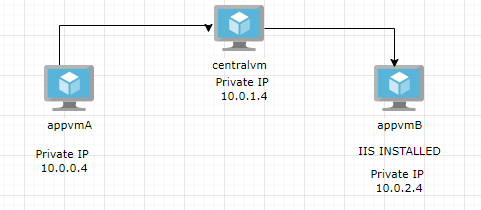


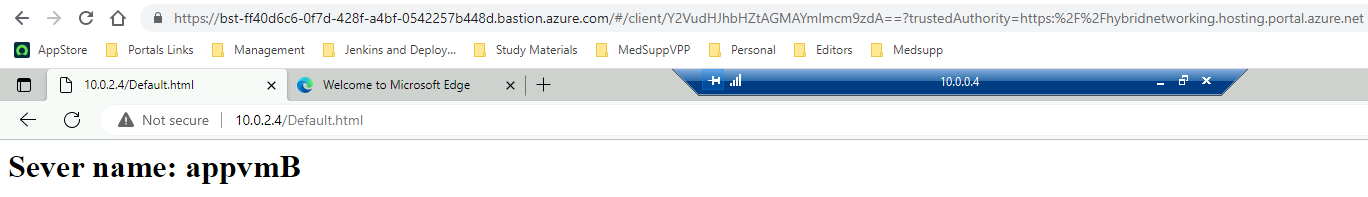












## NETWORK WATCHER

## AZURE BASTION SERVICE



* In a virtual network, if we have VMs that are part of different subnets. To connect with them we need to allocate a public IP address on each VM. But in certain use cases, where we might not prefer to assign public IP to VMs while running internal workloads – which we do not want to expose to the public internet. **The bastion helps us to connect with the VM without assigning a public IP to the VMs**

|  |
| --- |
| *Traditionally – in such use case we create a VM in an different subnet (Subnet C). We then login to the VM using its public IP address. Then we can connect to other VMs in (Subnet A & B) using its private IP.(VM can do a communication within with a Vnet using their private Ips*  *In Azure Bastion service – the subnet (Subnet C) is managed by Azure . The subnet is named as AzureBastionService* |

* Azure Bastion is a service, lets us connect to a virtual machine using browser and the Azure portal, or via the native

SSH or RDP client already installed on the local computer.

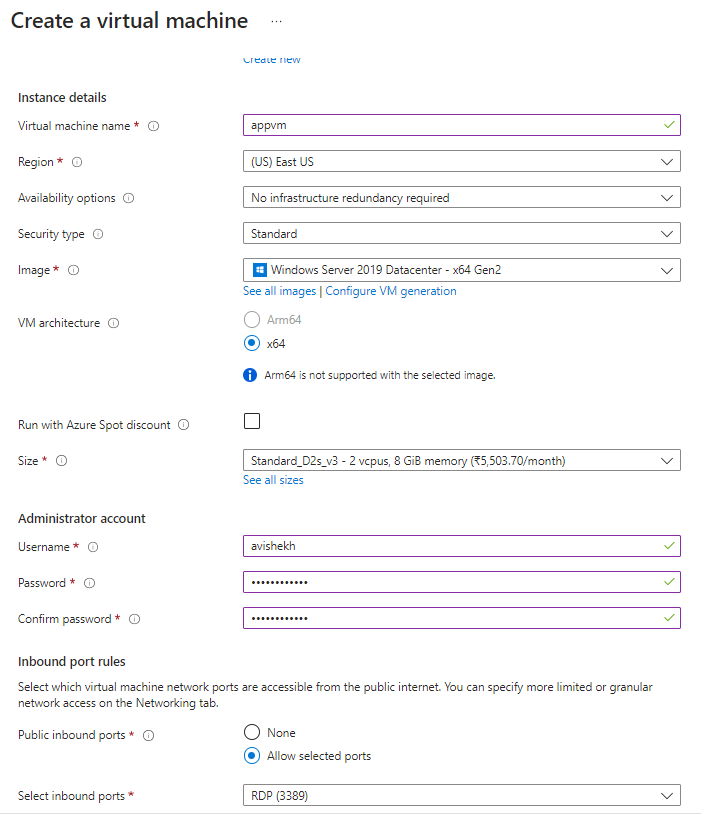
* The Azure Bastion service is a fully platform-managed **PaaS** service that we provision inside the virtual network. It provides secure and seamless **RDP/SSH connectivity to the virtual machines directly from the Azure portal over TLS(Transfer Layer Security)**.
* *When we connect via Azure Bastion, the virtual machines don't need a public IP address*
* **Using Azure Bastion protects the virtual machines from exposing RDP/SSH ports to the outside world, while still providing secure access using RDP/SSH.**
* When we use this service, it's going to creates its own compute machines that are going to be responsible

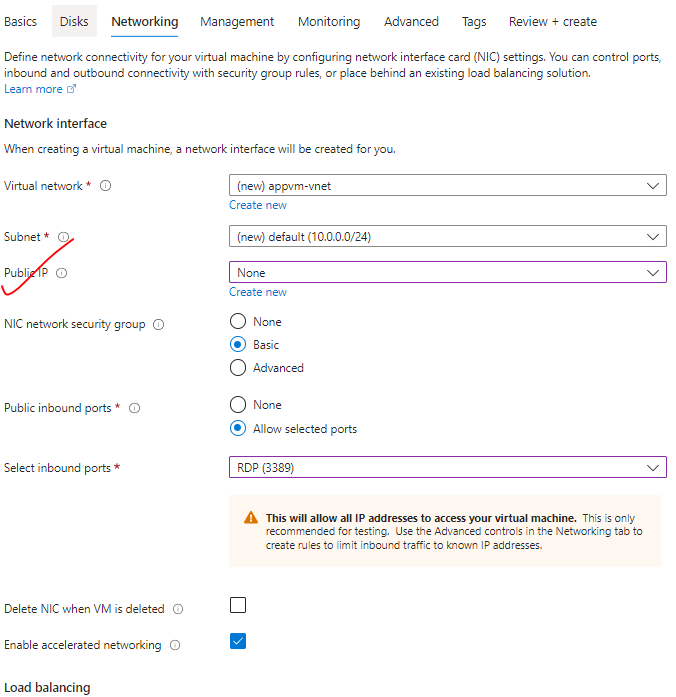
for establishing the connectivity. All the compute machines are managed by Azure itself.

* The compute must be part of a separate subnet in our virtual network. The name of the subnet named as “***AzureBastionSubnet***”

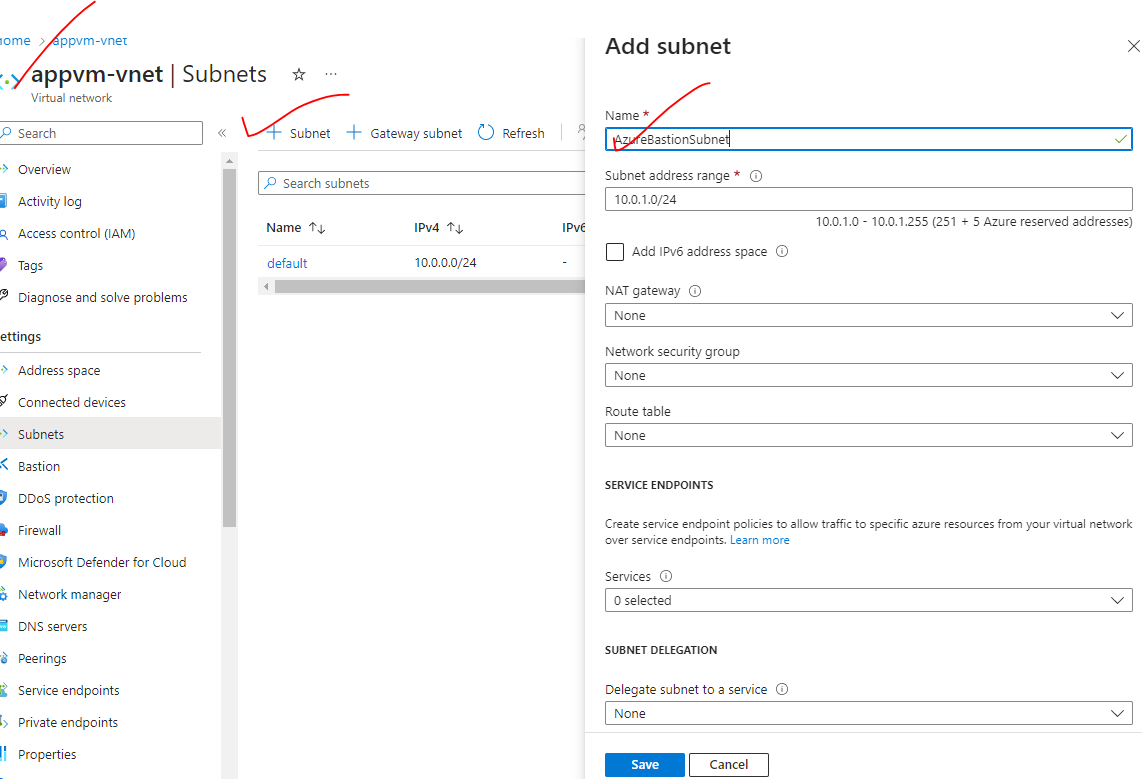
### SETTING UP BASTION SERVICE

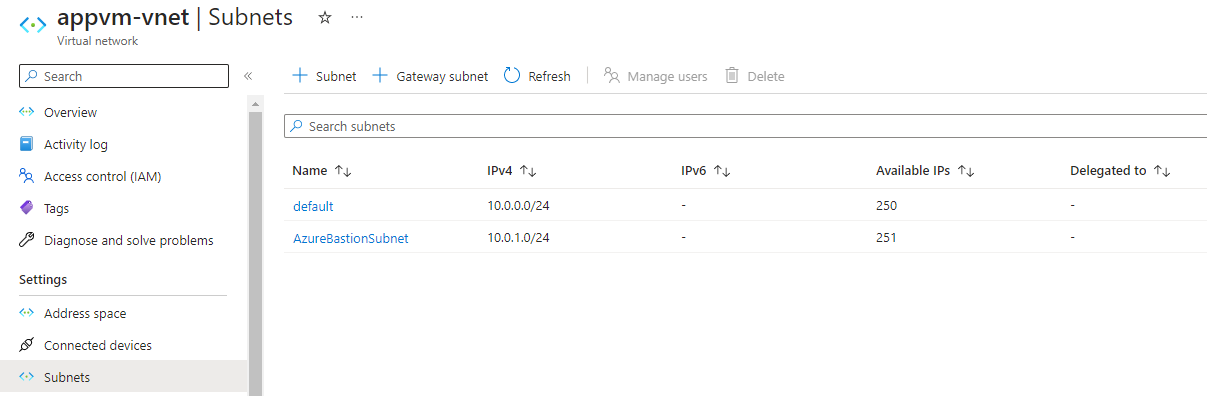
**CREATING A VM WITHOUT PUBLIC IP**



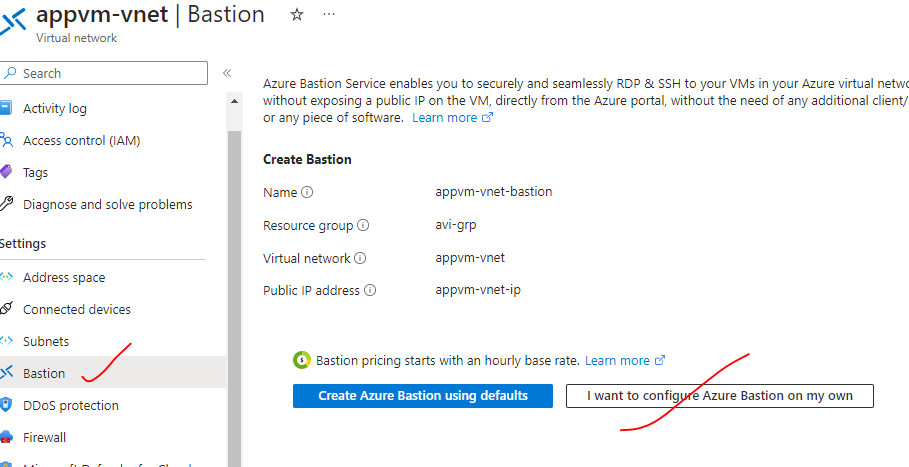


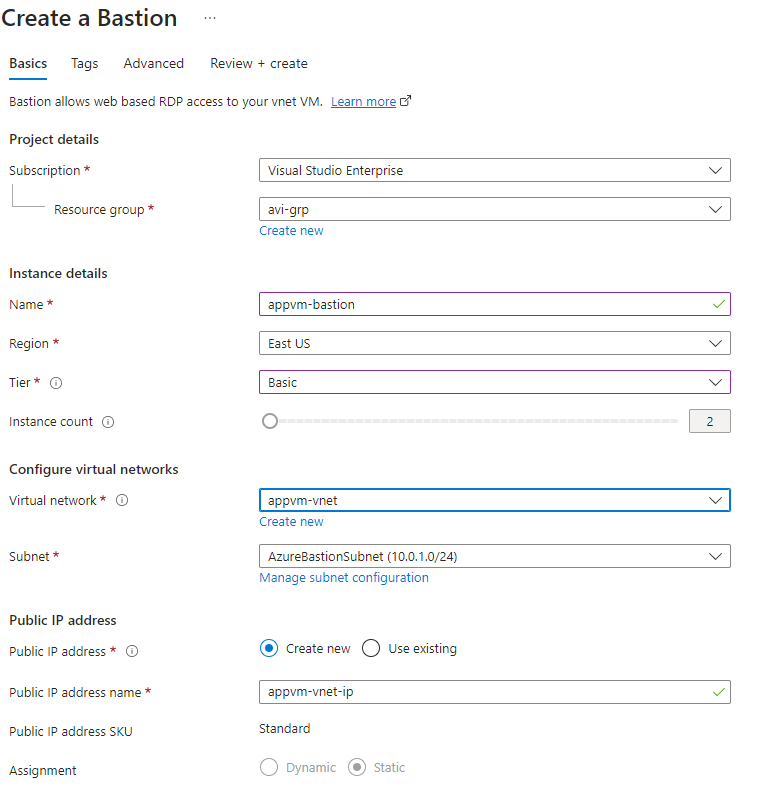
**CREATING SUBNET WITH NAME “AzureBastionSubnet”**



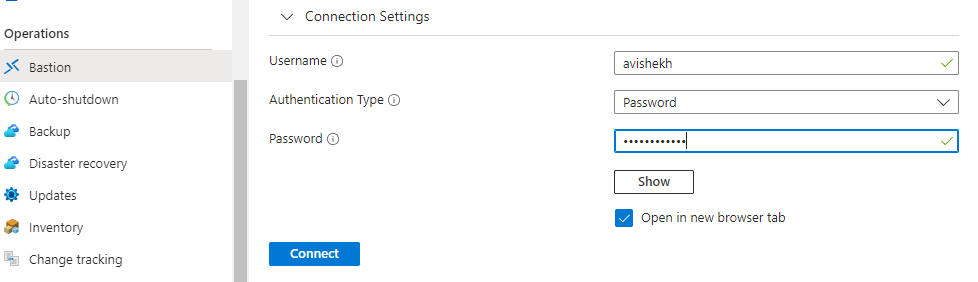


**CONFUGURE THE BASTION SERVICE**





LOGIN INTO VM USING BASTION



## AZURE TRAFFIC MANAGER

* **Traffic Manager a DNS based traffic load balancer.**
* The Azure load balancer and Application gateways are regional (i.e.it can manage the resources in a given region) , but a traffic manager is a global service. Hence when we deploy a traffic manager profile- we don’t to specify a region for the deployment of the resource.
* The traffic manager allows us to direct the client requests to an appropriate service endpoint (end point can be VMs , Web App or application hosted on-premise datacenter) based on routing method. The endpoint needs to be public endpoint that can be hosted inside or outside Azure
* The Traffic manager monitors the health monitoring of the endpoints.

### GEOGRAPIC ROUTING METHOD

1. ***In Geographic routing method the end point depending upon the location the DNS query originates from (it depends upon user’s location).***
2. This is different from the performance routing method. In the performance routing method, the traffic manager looks for a resource which takes less time in terms of latency. That is the endpoint that will be returned to the user.

#### SETTING GEOGRAPIC ROUTING METHOD

Step 1: Create 2 VMs in two different Region

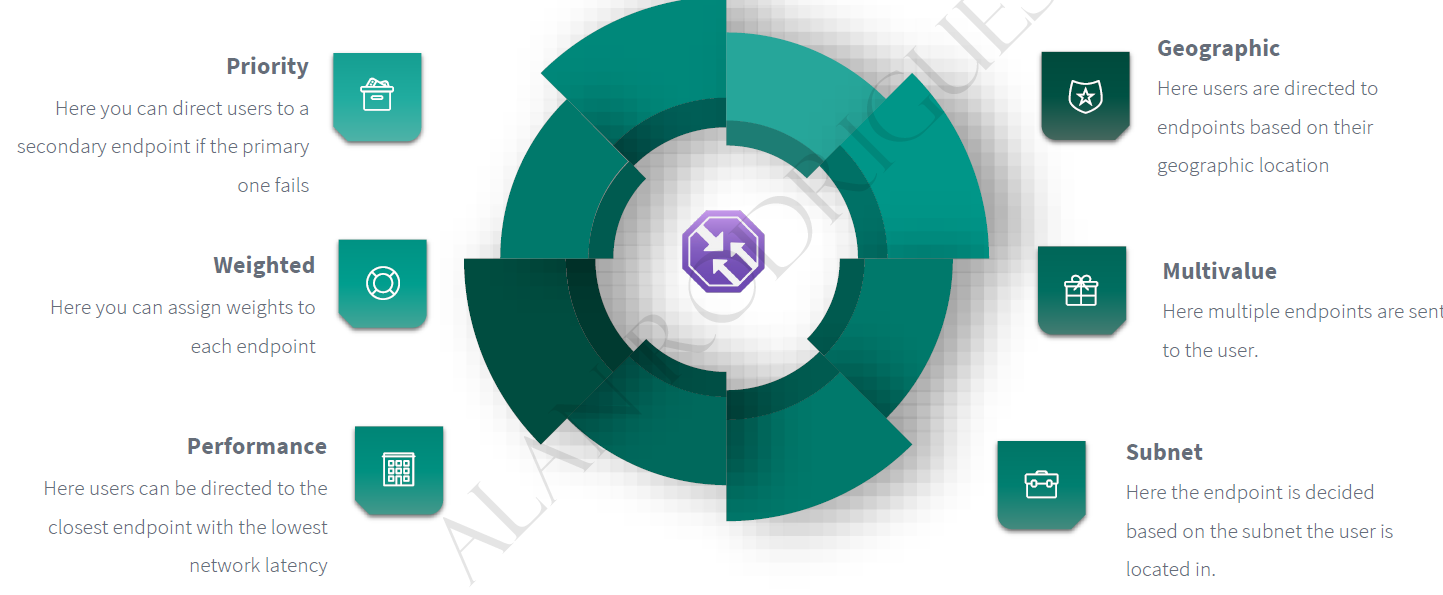
Step 2: Configure the traffic manager endpoint 🡪 Select the region which will serve the request which is coming from the same region

|  |  |
| --- | --- |
|  |  |

### AZURE TRAFFIC MANAGER SET UP

### AZURE TRAFFIC MANAGER ROUTING METHODS

Azure Traffic Manager supports **six traffic-routing methods to determine how to route network traffic to the various service endpoints.** For any profile, Traffic Manager applies the traffic-routing method associated to it to each DNS query it receives. The traffic-routing method determines which endpoint is returned in the DNS response.



### PRIORITY ROUTING METHOD

|  |  |
| --- | --- |
|  | * This routing method help in direct traffic to secondary endpoint of primary endpoint is not available. |

#### SETTING UP PRIORITY ROUTING METHOD

STEP 1*: SET UP VMs IN DIFFRENT REGION (it can be in same region too. Since TM is a global service, it can manage resources in any region)*

STEP 2:

* + INSTALL IIS on both the VM – These are the VM where the traffic will be directed
  + Add a DNS names to the VMs – as traffic manager does DNS based routing.

|  |  |
| --- | --- |
|  | * STEP 3: Search for **Traffic Manager Profile 🡪 Create** * *As it is a global service, hence we don’t need to mention the region of the Traffic Manager Profile (unlike Load Balancer and App Gateway).* |
|  | * Add a DNS names to the VMs * Go to VM Overview 🡪 Click on Public IP address🡪 DNS name label |

STEP 4: Add the VMs to the traffic Manager endpoint

|  |  |  |
| --- | --- | --- |
|  | **The priority will define – which public IP(VM public IP) will be a Primary End point** | |
| * When the priority is higher means when users access the traffic manager end point, all of the users will be directed to this VM. Only when the primary VM goes down that user will be directed the secondary VM the IBM or the endpoint, this is like a fallback mechanism. * After adding the endpoint – make sure Monitor Status is “Online” | |

STEP 4: Now the VMs can be accessed via Traffic Manager DNS name.



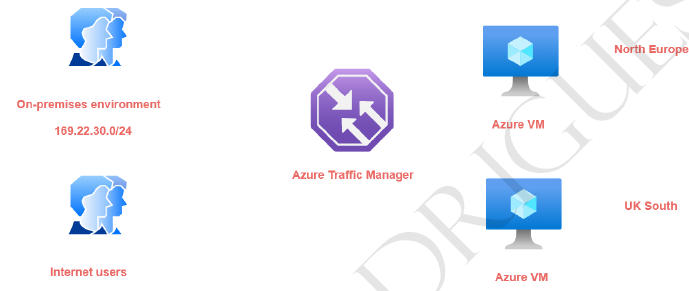
|  |  |
| --- | --- |
|  | * Note – In the above example – if we stop the primary VM – the request will be then route to secondary endpoint * PROBING INTERVAL (In sec.) – The time interval in which probing will happen * TOLERATED NUMBER OF FAILURES – After this TM will consider it as unhealthy resource * PROBE TIMEOUT - |
|  | |

|  |  |
| --- | --- |
|  | **TO UPDATE THE EXISTING ROUTING METHOD**   1. Delete the existing endpoints 2. Go to configuration of TM 🡪 Routing Method 🡪 Save 3. Add the endpoints again |

### PERFORMANCE ROUTING METHOD

1. In performance routing method – it routes the traffic to the location which is closest to use location

### SUBNET ROUTING METHOD



|  |  |
| --- | --- |
|  | * In this routing method – we map a specific set of end-user IP address range to specific endpoint. For example, if we want to direct the traffic from on premises environment- having a particular IP address range to an endpoint and all other internet users onto another endpoint. * The subnet routing methods - looks at the IP address range of the clients that are connecting to the traffic manager and then they be directed onto the end point. * If we want to create a default endpoint that is created- so that if the other end points don't * match the subnet range then the default end point will serve the request. **For default endpoint the subnet routing settings is kept empty.** * Hence – let’s say if the user hit the TM DNS and if the user’s IP does not fall in the range of Subnet routing setting – it will fallback to the default route |

### MULTIVALUED ROUTING METHOD

### WEIGHTED ROUTING METHOD

|  |  |
| --- | --- |
|  | * In weighted routing method we weight each endpoint in the traffic manager profile * Lets say if we 2 VM (endpoints)- which are weighted **50.**  The traffic to TM will be distibuted across endoint independently evenly. |

#### USE CASE OF WEIGHTED ROUTING METHOD

* So in our case, we have two AC vents in place
* Now we can assign leads on to the end points that are mapped on to our traffic margin profile.
* So, for example, I can't assign a bit of go on to the end point for the as the IBM in the North,
* their application.
* And then I can assign a weight of 50 here by the end point that is marked onto the UK South location.
* So this is like actually load balancing now request across these virtual machines.
* So when half of the users are connecting onto the air traffic manager profile, half of them will be
* directed onto the beam in the Noctua application and the other half will be directed onto the VM in
* the UK South location.
* So here the requests are actually distribute based on the bits.
* Now, one of the most useful scenarios of using the reedit routing method is when it comes onto something
* known as blue green deployments.
* So normally what happens is that initially all your users.
* So you could be having an endpoint with a bit of full 100 percent when an after users are being directed
* onto one end point.
* Now let's say you are deploying a newer version of the application on new end point, so it could be
* an A-Z IBM.
* It could be a ACR bebop.
* It could be another endpoint.
* So now you want us to form some of the users to be detected on to the newer version of your application.
* So you might maybe change the reteach from 80 percent here and 20 here.
* So 20 percent of the users are now directed onto the newer version of the application.
* Once you know you know that there are no issues with the newer version of your application, then you
* can slowly switch up.
* You know, you could make this 80 percent and you can make this 20 percent.
* And in the end, you could do a full cut over that, and this could be 100 percent and you can just
* go and disable this endpoint.
* So this is good for actually distributing the traffic across multiple endpoints.
* Again, this is very easy to achieve.
* Let's see how we can implement this.
* So again, first of all, I'll go onto my traffic manager profile.
* I'll go on to my end points and let's delete and points.
* And I'll go back to my profile.
* Let's change the configuration.
* So from the multi value onto the Beated.
* Let me click on Save.
* Go on to the profile.
* Let me go on to and points out and then point.
* You can put an end point one there, I can choose the cloud service, the public IP address, you can
* put traffic via one.
* And here in the weeds, you can put 50 and you can go out and click on Add.
* Take a look at the other end point.
* Give it a name.
* Yeah, public IP address to the public IP does a stop at the M2 and put the webpage has again 15.
* And click on Add.
* You have no unequal distribution when it comes onto the beach on two end points.
* And if I go ahead and just take the the innocent point.
* In a new type, if I go on to be followed, Estima, so I'm being directed onto the Not Europe location.
* So with the help of the state, you can actually go in and disrupt traffic across the end points when
* it comes as air traffic manager.

## AZURE FRONTDOOR