Contents

[AZURE - NETWORKING 2](#_Toc122206408)

[IP ADDESSS 2](#_Toc122206409)

[IPv4 ADDRESSING 2](#_Toc122206410)

[BINARY TO DECIMAL 2](#_Toc122206411)

[CLASSES IN IP ADDRESSING 2](#_Toc122206412)

[NETWORK ID IN IP ADDRESSING 3](#_Toc122206413)

[PUBLIC IP ADDRESS 3](#_Toc122206414)

[PRIVATE IP ADDRESS 4](#_Toc122206415)

[DIFFERENCE BETWEEN PUBLIC VERSUS PRIVATE IP 4](#_Toc122206416)

[SUBNETING 5](#_Toc122206417)

[SUBNET, SUBNET MASK AND CIDR NOTATION 6](#_Toc122206418)

[VIRTUAL NETWORK 6](#_Toc122206419)

[CREATING A VNET 7](#_Toc122206420)

[DEPLOYING A VM IN VIRTUAL NETWORKS 10](#_Toc122206421)

[STATIC IP ADDRESS 11](#_Toc122206422)

[NETWORK INTERFACE 11](#_Toc122206423)

[NETWORK SECURITY GROUP(NSG) 14](#_Toc122206424)

[COMMUNCATION WITH VIRTUAL NETWORKS 19](#_Toc122206425)

[LOAD BALANCER 27](#_Toc122206426)

[LOAD BALANCER CONCEPT 28](#_Toc122206427)

[COMPONENTS OF LOAD BALANCER 28](#_Toc122206428)

[TYPES OF LOAD BALANCER 28](#_Toc122206429)

[LOAD BALANCER SKUs 28](#_Toc122206430)

[BASIC LOAD BALANCER 29](#_Toc122206431)

[STANDARD LOAD BALANCER 34](#_Toc122206432)

[AZURE BASTION SERVICE 34](#_Toc122206433)

[SETTING UP BASTION SERVICE 35](#_Toc122206434)

[AZURE TRAFFIC MANAGER 35](#_Toc122206435)

[AZURE TRAFFIC MANAGER SET UP 35](#_Toc122206436)

[AZURE TRAFFIC MANAGER ROUTING METHODS 35](#_Toc122206437)

[PRIORITY ROUTING METHOD 36](#_Toc122206438)

[PERFORMANCE ROUTING METHOD 38](#_Toc122206439)

[GEOGRAPIC ROUTING METHOD 38](#_Toc122206440)

[SUBNET ROUTING METHOD 39](#_Toc122206441)

[MULTIVALUED ROUTING METHOD 39](#_Toc122206442)

[WEIGHTED ROUTING METHOD 40](#_Toc122206443)

# AZURE - NETWORKING

## IP ADDESSS

* IP is the unique identifier of a device in a network.



|  |  |
| --- | --- |
|  | * The IP address has 2 parts – The Network part and a Host part. * Each host in a network will have same Network Address |

### IPv4 ADDRESSING

|  |  |
| --- | --- |
| * It’s a 32-bit logical address * It consists of 4 octet – and each octet ranges from 0 -255 * IP address has parts ***Network ID and Host ID*** |  |

### BINARY TO DECIMAL

|  |  |
| --- | --- |
| * When the IP is represented in Binary the decimal representation can be done using the ***power to 2*** | * Hence the equivalent decimal will be 192.168.100.1 (128+64).(128 + 32 +8).(64+32+4).(1) |

## CLASSES IN IP ADDRESSING



* Ranges 127.x.x.x are reserved for the [loopback or localhost](https://www.computerhope.com/jargon/l/locahost.htm), for example, 127.0.0.1 is the loopback address.
* Range 255.255.255.255 [broadcasts](https://www.computerhope.com/jargon/b/broadcas.htm) to all hosts on the local network.

HOW TO DECIDE THE CLASS OF IP ADDRESS?

To decide the class of IP address we consider the first octet for example – **132**.20.10.192 – This IP belong to CLASS B.

### NETWORK ID IN IP ADDRESSING

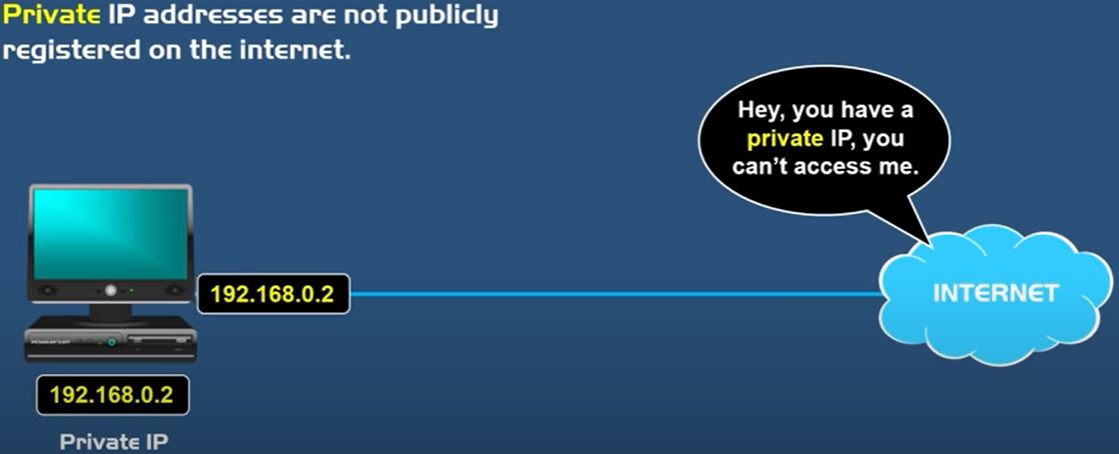
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CLASS | IP RANGE |  | NUMBER OF NETWORK | NUMBER OF HOST |
| Class A |  | * *1st 8 bits are Network Id* * *Remaining 24 bits are for Host* | **126** | **224 = 16777216**  **16777216 – Network Id – BroadCast id =16777216** |
| Class B |  | * *1st 16 bits are Network Id* * *Remaining 16 bits are for Host* | **27 = 16,384** | **216 = 65536-2 = 65534** |
| Class C |  | * *1st 24 bits are Network Id* * *Remaining 8 bits are for Host* | **221 = 2,097,152** | **28 = 256-2 =254** |

|  |  |
| --- | --- |
|  | For the network part of the IP   1. For Class A - the 1st bit always “0” 2. For Class B - the 1st 2 bits always “10” 3. For Class C - the 1st 3 bits always “110”   CALCULATION OF NETWORK ID (EXAMPLE CLASS B)  Since the 1st 2 bits are always 1and 0 hence number of variable bits are 214 = 16384 |
|  | In the Host Part of IP. One Id is has been given to network ID and last Id will be broadcast ID |

### PUBLIC IP ADDRESS

|  |  |
| --- | --- |
|  | **WHAT IS A PUBLIC IP ADDRESS?**   * A public IP address is an IP address that can be accessed directly over the internet and is assigned to your network router by the internet service provider (ISP). |

### PRIVATE IP ADDRESS



**WHAT IS A PRIVATE IP ADDRESS?**

* A private IP address is the address your network router assigns to your device. **Each device within the same network is assigned a unique private IP address** (sometimes called a private network address) — this is how devices on the same internal network talk to each other.
* Private IP addresses let devices connected to the same network communicate with one another without connecting to internet. This makes difficult for an external host to establish a connection, hence **private IPs help bolster security within a specific network**.
* **DHCP is a service used in the routers to assign private to the devices.**
* For all those devices in Home / office network- can access the internet only via public IP. The router assigns the private address of the device to public IP which we have been given by ISP(i.e. Router public IP address). ***The Service that does this translation of private to public and vice versa is called NAT (Network address translation). This servics are built-in the Router.***
* Reference: <https://www.youtube.com/watch?v=po8ZFG0Xc4Q>

**PRIVATE IP RANGE**



### DIFFERENCE BETWEEN PUBLIC VERSUS PRIVATE IP

|  |  |
| --- | --- |
| **PUBLIC IP ADDRESS** | **PRIVATE IP ADDRESS** |
| External (global) reach | Internal (local) reach |
| Used for communicating outside your private network, over the internet | Used for communicating within your private network, with other devices in your home or office |
| A unique numeric code never reused by other devices | A non-unique numeric code that may be reused by other devices in other private networks |
| Found by Googling: "What is my IP address?" | Found via your device’s internal settings |
| Assigned and controlled by your internet service provider | Assigned to your specific device within a private network |
| Not free | Free |
| Any number not included in the reserved private IP address range  Example: 8.8.8.8. | **IP RANGE**  **10.0.0.0 — 10.255.255.255; 172.16.0.0 — 172.31.255.255;  192.168.0.0 — 192.168.255.255**  **Example: 10.11.12.13** |

### SUBNETING

* Subnet is a network withing a network.
* Subnetting is a concept of using the IP address effieciently

#### SUBNET MASK

|  |  |
| --- | --- |
|  | CALCULATION OF SUBNET MASK(CLASS A) |

#### CIDR NOTATION (CLASSLESS INTER DOMAIN ROUTING)

* CIDR gives the number of bit given to the network bit. For example, **192.168.100.1/24** This represent that the first 24 bits are used for network id. It is also called network pre-fix . Hence in a 32 bit Ip address , 24 bit has been occupied by Network

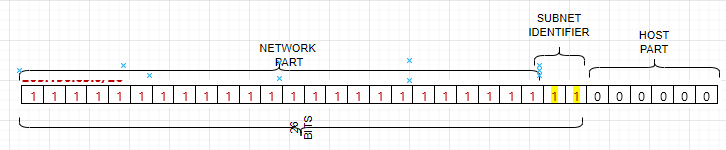
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

* **205.150.65.0/26**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

Conclusion from CIDR notation

1. The IP belongs to Class C
2. By default, in Class C - 24 bits are assigned to network and remaining 8 bits host. Hence the CIDR notation for Class C address in “24“
3. But in above example – since 2 more bits has been taken from Host. Hence the CIDR notation will be “26”



WHY WE DO THIS? – This help in subnetting.

#### SUBNETING CALCULATONS

Question: If Network Address Is – 205.150.65.0/26.

1. **SUBNET MASK** 
   1. Since it a class C IP. The default subnet mask is 255.255.255.0. But since we borrowed 2 more bits from the host bit hence the new subnet mast will be – 255.255.255.192 (*2^7 + 2 ^6 = 192*)
2. **NETWORK ID**
   * It’s a logical add operation of Network address and Subnet mask – **205.150.65.0**
3. **NUMBER OF POSSIBLE SUBNETS**
   1. 2**(number of subnet identifier)** = 22 = 4 Subnets
4. **NUMBER OF POSSIBLE USABLE HOST (IN EACH SUBNET)**
   1. 2**(number of host bits)** -2 = 26 -2 = 64-2 = 62 Hosts
   2. 2 is subtracted because 1st IP is reserved for Network and last IP for broadcast.
5. **BROADCAST IPs (IN EACH SUBNET)**
   1. This is the last IP in SUBNET. So, it will be
   2. Broadcast Id for Subnet 0 - 205.150.65.63
   3. Broadcast Id for Subnet 1 - 205.150.65.128
   4. Broadcast Id for Subnet 2 - 205.150.65.192
   5. Broadcast Id for Subnet 3 - 205.150.65.256

Question: If Network Address Is – 205.150.65.0/24. Create 10 Subnet from this network address

* This is a Class C IP address. If we have to create 10 subnets – Then we have to first decide how many bit we will be borrowing from host bits. To create minimum 10 subnets, it will be 24 = 16 (4 bits we need to borrow to accommodate 10 subnets)

|  |  |  |  |
| --- | --- | --- | --- |
| SUBNET MASK | | 255.255.255.240 | |
| NETWORK ID | | 205.150.65.0 | |
| NO OF SUBNET | | 24 = 16 Subnets | |
| NO OF HOST (PER SUBNET) | | 24-2 = 14 Hosts per subnet | |
| BROADCAST ID | | 205.150.65.15 | |

### SUBNET, SUBNET MASK AND CIDR NOTATION

* Note the network id is represent by “1” and host is represented by “2”.

#### CALCULATING SUBNET MASK

* Example : Calculate the Subnet mask of : **115.10.10.20 ? Ans – 255.0.0.0**

##### WHAT IS THE JOB OF SUBNET MASK?

* Since the IP is consist of 2 parts - Network and Host part. Network Part in a network will be same for all host.
* Host decide the network Id with the help of Subnet Mask.
* Subnet mask is also of 32 bits -which has a mapping with the IP address. The 1s represent the network portion and 0s are the host portion. In the above example – When an IP is given to a device – then the subnet mask is also configured. The bits represented with “1” is the network id in the IP – when compared from left to right.

#### BROADCAST ID

* Broadcast IP is used to broadcast to all the host in the network.
* Find the class, network id, broadcast id and usable IP of the following IP

|  |  |
| --- | --- |
| 150.10.20.30 | * The IP belongs to Class B * Network Id: 150.10.0.0 * Broadcast ID: To calculate the broadcast id, set the host part of Network to 255. Hence the broadcast Id - 150.10.255.255 * Usable Host IP = Total Number of IP address – (Network IP + Broadcast IP) i.e.   + For a network there will be on Network IP and one Broadcast IP.   + Usable Host IP = Total IP -2= 216 – 2 = 65536 – 2= **65534** |

## VIRTUAL NETWORK

* Azure virtual network allow us to create an isolated network on cloud.

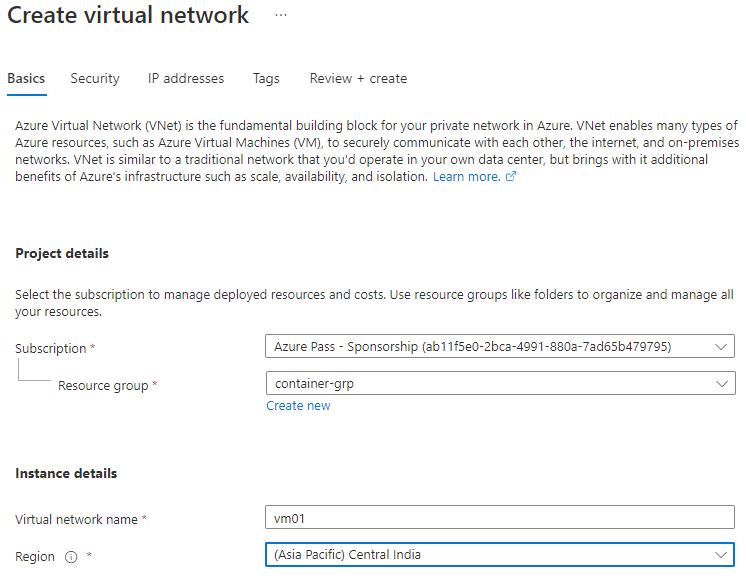
|  |  |
| --- | --- |
|  | * The resources like VMs can be made part of Virtual network. * Once the network becomes part of VNET – The resources (VMs) get a private IP * The Private IP of the machine are taken from the address range of the subnet which is part of VNET |

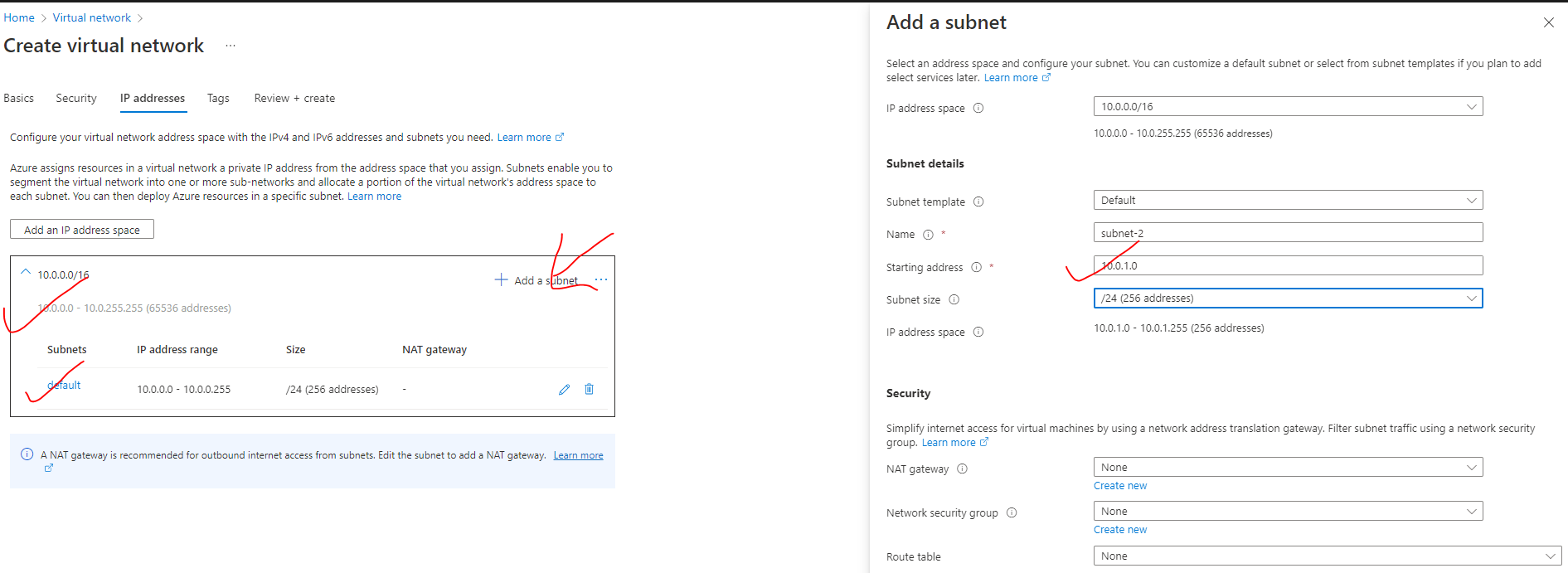
### CREATING A VNET

* One of the building blocks of network in Azure. VNET creates a LAN – where all the associated VMs get a private IP.
* Creating such private network help in managing the in and out traffic.
* When a VNET is created in Azure – by default it will have internet address- Hence all the associated VMs will have internet access as well , whether the VM has public IP or not.

To create a VM the inputs that must be provided are

|  |  |  |
| --- | --- | --- |
| **REGION** | From region standpoint if a VNET has been created a region, only those VM can be a part of the V-NET if it belongs to the same region | |
| **SUBSCRIPTION** | Azure Subscription | |
| **ADDRESS POOL** | This is the range of Private IP address that will be given to the associated VMs. This is a private IP because VNET creates a LAN |  |
| **SUBNET** | Network inside the VNET | |

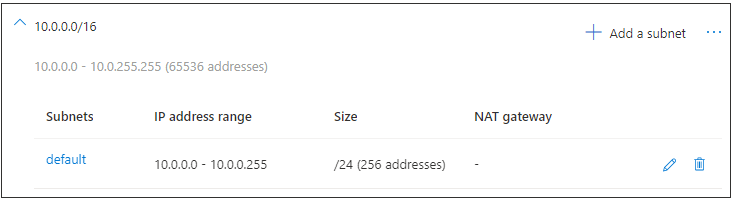


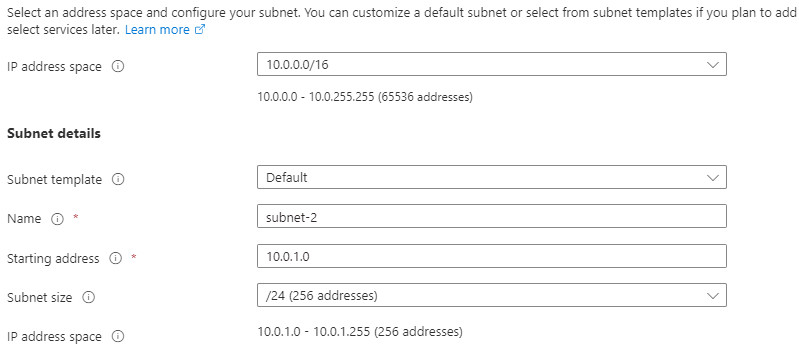


**IP ADDRESS**

* As per the above configuration (default configuration)- The VNET has 65536 addresses (Private IP)
* By default, it creates a subnet of 10.0.0.0/24 – of 256 addresses.

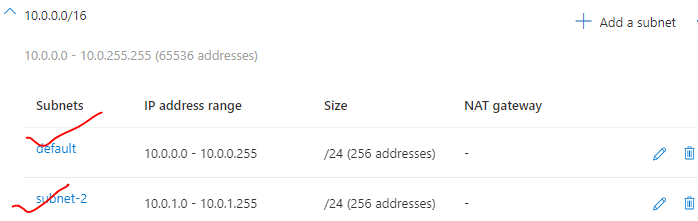
**ADDING A NEW SUBNET**

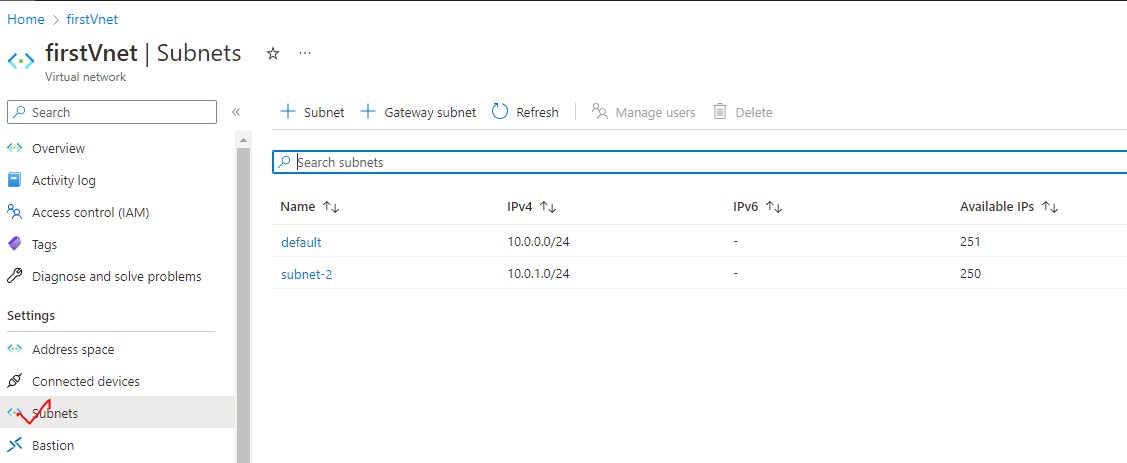
* [https://www.davidc.net/sites/default/subnets/subnets.html?network=10.0.0.0&mask=20&division=23.f42331](https://www.davidc.net/sites/default/subnets/subnets.html?network=10.0.0.0&mask=20&division=23.f42331) When the subnet is created starting 5 addresses are reserved for
* **1 FOR DEFAULT GATEWAY**
* **2 FOR DNS**
* **1 FOR NETWORK ID (10.0.0.0)**
* **1 FOR BROADCAST (10.0.0.255)**
* Hence if the IP range of a subnet start from 10.0.0.0 , then the 1st VM will get the IP of 10.0.0.4



**SUBNET ADDED**

* As the IP is a Class A IP – hence

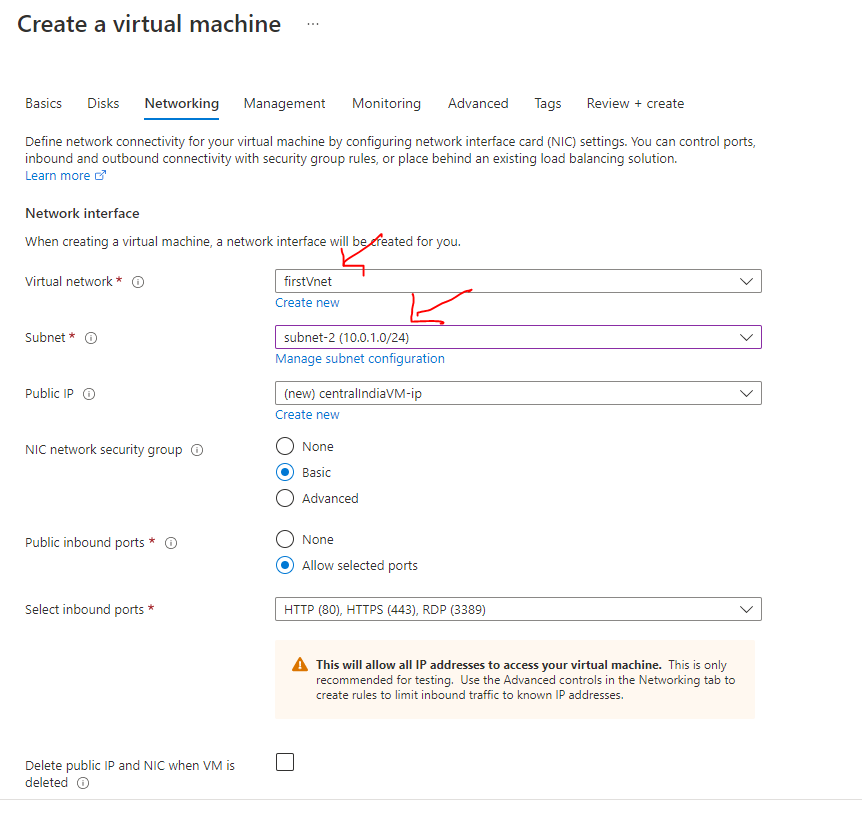


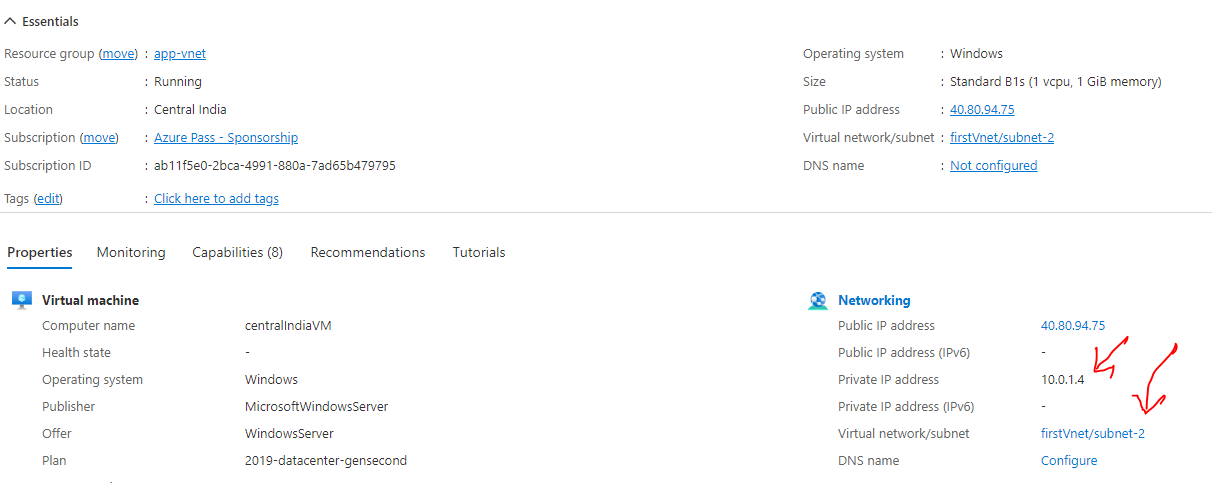


|  |  |
| --- | --- |
|  | * When we want to communicate between 2 different networks (the network having 2 different network id are considered as different network) * The VM associated with different subnets within the VNET can be able to communicate with each other via their private IP. There is no need of public IP address for the communication with the VNET. * This happens because azure enables the routing by default between the subnets |

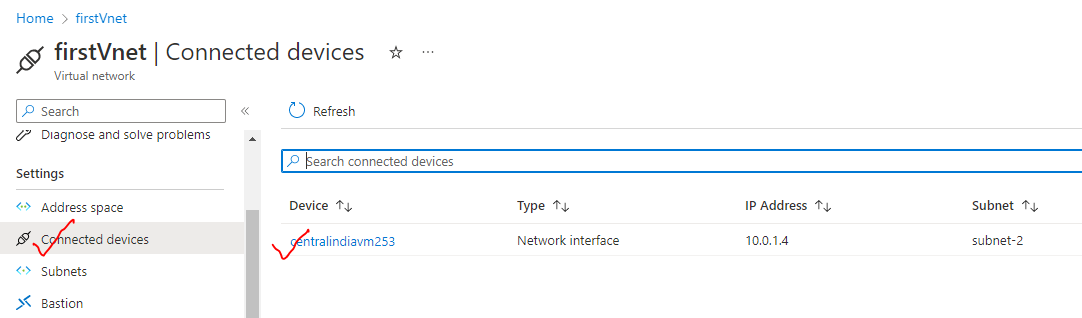
### DEPLOYING A VM IN VIRTUAL NETWORKS

* To associate a VM to a given virtual network. The virtual machine must be created in in the same region.
* In the below example –while creating the VM, we are selecting a subnet – 10.0.1.0/24. The first VM will get the private IP of 10.0.0.4 (first 5 Ips are reserved for default gateway -1, network id -1, broadcast id -1, DNS- 2)





* The VM will show up as a Connected device in the Virtual network.
* Note – Until a VM is connected to a subnet – the subnet cannot be deleted from the VNET.



### STATIC IP ADDRESS

|  |  |
| --- | --- |
|  | * When we create an associate VM,we get a public IP address and private IP address. All of these are dynamic IPs. * When we stop and the start the VM, it will get allocated with a new public IP address * This can be an issue when, let's say that we have a Web application hosted on the VM. To make it accessible to the user using some DNS (e.g., **cloudportalhub.com**), the external DNS provider map the domain name with the public IP address of the VM. * If the VM is stopped and started again, the IP address the changes and if the IP address changes, this link is broken so users will not be able to reach the application. * To mitigate this issue, we can allocate a static IP address to both the public IP and private IP address. |

### NETWORK INTERFACE

|  |  |
| --- | --- |
|  | * The network interface is a physical device is used for the routing of traffic. In Azure, the VMs are the cloud, they are known as a ***virtual network interface***. * *Any device (Physical or Virtual) which need to connect to internet needs a Network Interface Card (It can be wired or wireless).* ***Every VM in Azure has virtual network interface that is attached to it by default****. It gets assigned to VM while creating a VM itself.* * **The NIC are then attached to VNET/Subnet** * A VM can have multiple VNIC – the number of VNIC that can be attached to it depends on the size of the VM itself |

THE VIRTUAL NETWORK INTERFACE CARD GETS BOTH **PRIVATE AND PUBLIC IP ADDRESS**

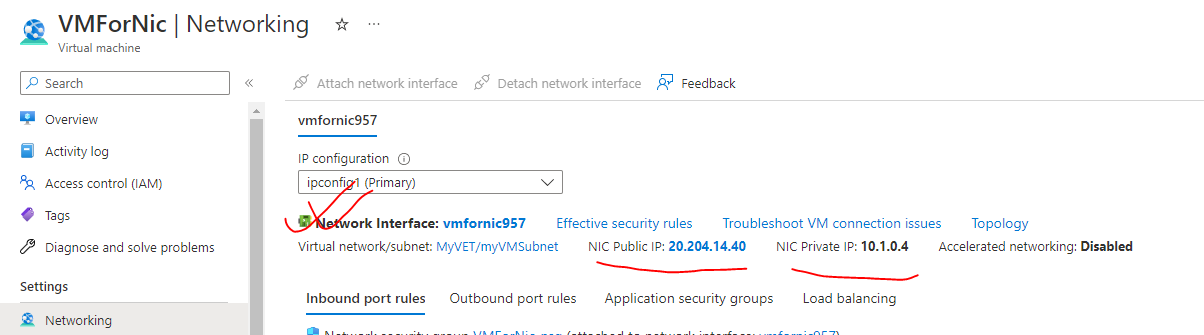
|  |  |
| --- | --- |
| PRIVATE IP ADDRESS | PUBLIC IP ADDRESS |
| * The private IP of VNI is used for the communication within the VNET. * The private IP lies in the range of Subnet of which the VM is part of | * The public IP is used for the communication on internet * This is assigned by Azure itself |

#### VIRTUAL NETWORK INTERFACE

|  |  |
| --- | --- |
|  | * The VNI has something called IP Config which has a private IP address. * For public IP address – we need to create one more resource – which then get attached to the IP config of the VNI |

#### DEFAULT VIRTUAL NETWORK INTERFACE OF VM

* A default VNIC get attached to a VM during its creation
* The VNIC has both private and public IP



VNIC PROPERTIES



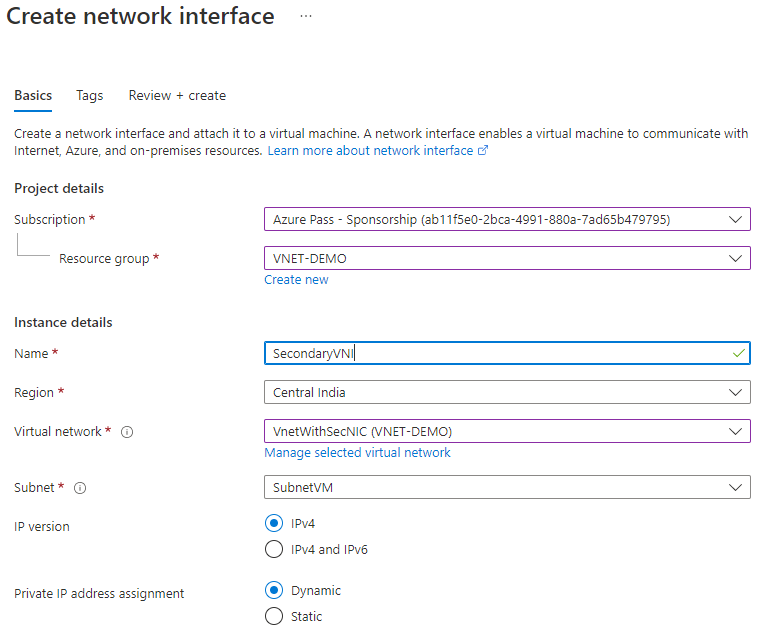
#### SECONDARY VIRTUAL NETWORK INTERFACE

* By default, the VM has a Virtual Network Interface attached to it – But we can attach a secondary VNI to a VM

##### CREATING A SECONDARY VIRTUAL NETWORK INTERFACE

ADDING A SECONDARY V-NIC

|  |  |
| --- | --- |
|  | * Search for “Network Interface” service.   WHY WE NEED A SECONDARY VNI? |



ATTACH VNIC TO VM

|  |  |
| --- | --- |
|  | * To attach a Network interface to a VM – we need to first STOP the VM(Deallocated) * **We can connect network interfaces in the same VM to different subnets within a virtual network. However, the network interfaces must all be connected to the same virtual network.** |
|  |

### NETWORK SECURITY GROUP(NSG)

|  |  |
| --- | --- |
|  | * The purpose of network security groups is to manage (Stop/ Allow) the traffic that's coming in and going out of the virtual machine * As we know – A VM has a virtual network interface attached to it (by default), * The VM can have a public and private IP address. The private IP address is taken from the subnet range. * **When we create a VM - it creates a network security group that is attached to the virtual network interface.** |

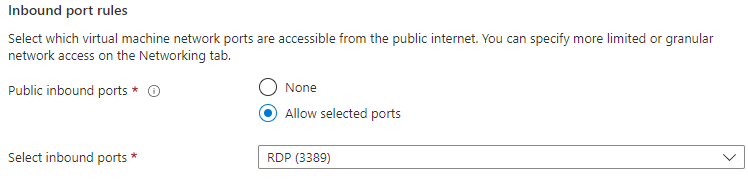
#### NETWORK SECURITY GROUP OVERVIEW

* NSG is separate resource in Azure
* It has rules for inbound (coming into the VM) and outbound (going out of the VM) traffic. Note -By default, there is no inbound rule to allow traffic from the internet.
* We can have multiple NSG rules, which has the priority associated with it. The rules are evaluated in the order of the priority.

|  |  |
| --- | --- |
| NSG-processing | NOTE – the NSG can be applied on 2 level   * **AT THE SUBNET LEVEL OF WHICH THE VM IS PART OF – (This takes high priority)** * **AT THE NETWORK INTERFACE CARD LEVEL OF THE VM** |

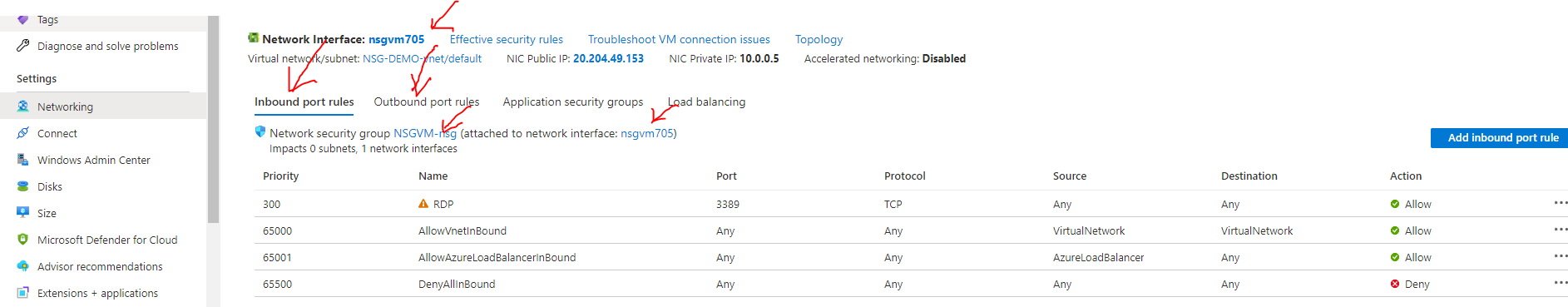
#### SETTING UP NSG FOR A VM ON NIC LEVEL

* **STEP 1: CREATE A VM** – While creating a VM (in Basic and Networking tab we have setting available for NSG )– we set some “inbound port rules”- the actually creates a inbound NSG rule for the given VM



The NSG is a separate resource in Azure VM. In the below image – we can see that

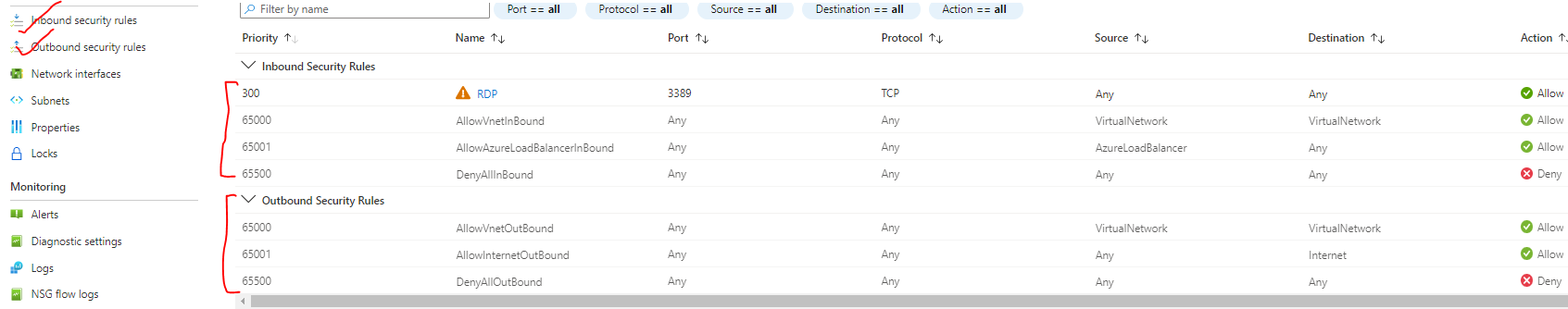
* The NSG can be viewed in Networking section of the VM
* The NSG has both Inbound and outbound rules for the traffic
* The NSG are attached to the NIC (Network interface). Note – we can also attach the NSG to the subnet (of which VM is part of) as well



##### NSG RULES

DEFAULT NSG RULES

* By default, NSG has some default inbound and outbound NSG rules. We cannot change the default NSG rukes



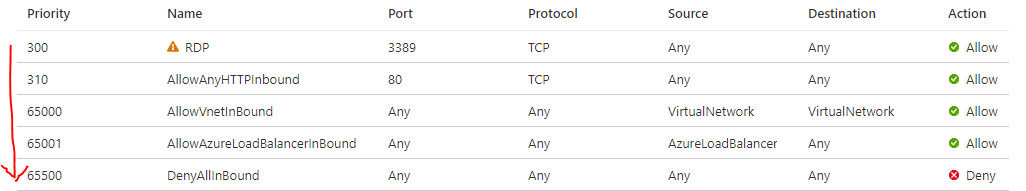
|  |  |
| --- | --- |
| DEFAULT RULE (INBOUND) | DESCRIPTION |
| AllowVnetInBound | Allow all traffic within the virtual network |
| AllowAzureLoadBalancerInBound | If the VM is placed behind a Load balancer by default, all the traffic from the load balancer to the VM is allowed |
| DenyAllInBound | Any other traffic mentioned explicitly in the NSG rules are denied . |

|  |  |
| --- | --- |
|  | USE CASE FOR NSG –   * Let’s say we have IIS installed on VM – which we want to allow the access from any machine. * For this we need to add a NSG rule (inbound – as it’s an inbound traffic) |

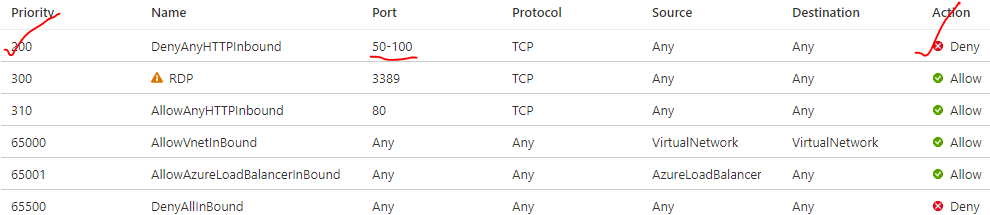
##### PRIORIRY SETTINGS IN NSG RULES

| PROPERTY | EXPLANATION |
| --- | --- |
| **NAME** | A unique name within the network security group. |
| **PRIORITY** | * A number between **100 and 4096.** Rules are processed in priority order, with lower numbers processed before higher numbers, because lower numbers have higher priority. * Once traffic matches a rule, processing stops. As a result, any rules that exist with lower priorities (higher numbers) that have the same attributes as rules with higher priorities aren't processed. |
| **SOURCE OR DESTINATION** |  |
| **PROTOCOL** | TCP, UDP, ICMP, ESP, AH, or Any. The ESP and AH protocols aren't currently available via the Azure portal but can be used via ARM templates. |
| **DIRECTION** | Whether the rule applies to inbound, or outbound traffic. |
| **PORT RANGE** | * We can specify an individual or range of ports. |
| **ACTION** | Allow or deny |

* Lower the number higher is the priority.
* The rules are executed from to bottom
* **If the condition is meet for a configured rule – No further rules are evaluated**



* *Example - In the below rules the Inbound to PORT 80 is denied. As the port “80” comes in the range of 50-100. Now once the rules in matched – no further evaluation of rules will be processed*



#### SETTING UP NSG FOR A VM ON SUBNET LEVEL

* As the NSG rules can be applied to NIC and Subnet level (Subnet of which the VM is part of). In this case the NSG rule at subnet level will be evaluated first for NIC rules.

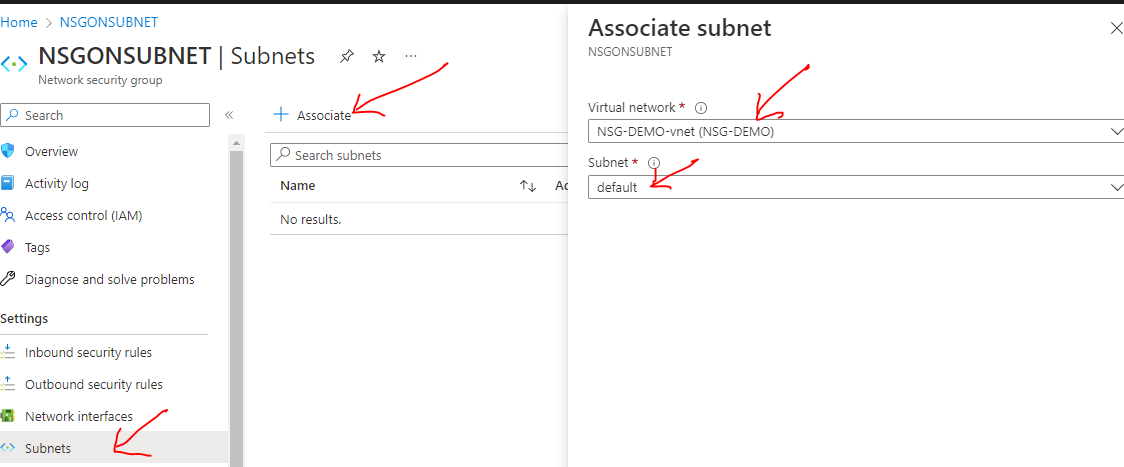
**TO CONFIGURE NSG ON SUBNET LEVEL**

Step 1: CREATE AN NSG –

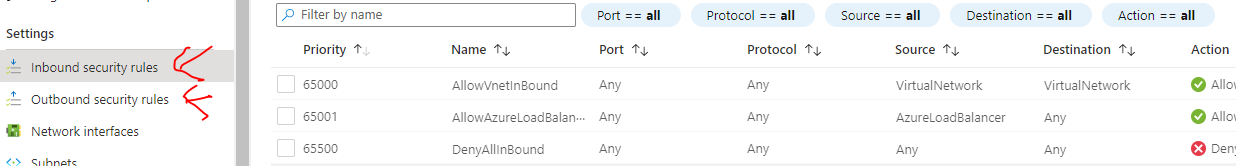
* NSG is a separate resource in Azure. The NSG has to be created in the same region of Virtual Network of the Virtual Machine.

|  |  |
| --- | --- |
| VNET OF THE VM |  |

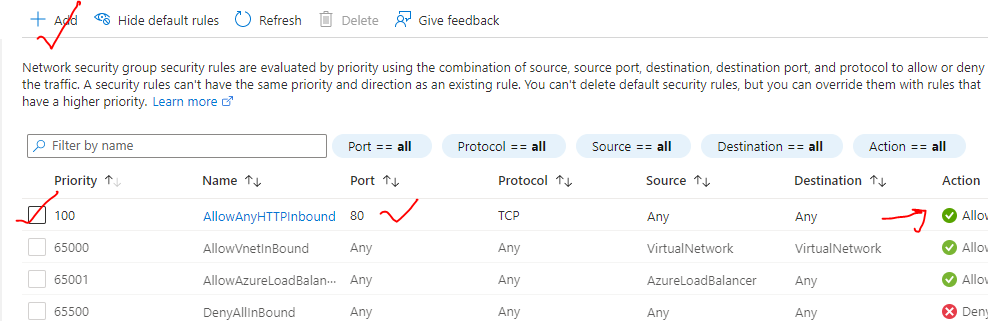
Step 2: NAVIGATE TO THE NSG AND ASSOCIATE THE SUBNET

Step 3: RULES ON SUBNET LEVEL

* After the newly created rules is attached to the VM’s subnet . This gets evaluated first
* In the below rules – there is no rule for “80” PORT hence the request for the service running on “80” will be denied



Step 4: ADDING A RULE ON SUBNET LEVEL



#### IMPORTANT POINTS ON NSG

1. NO NSG

|  |  |
| --- | --- |
|  | 1. When a VM is created without any NSG – means it is open to all type of requests (e.g., RDP, HTTP etc.) 2. To create such VM🡪 Networking Tab 🡪 Select NSG as none |

1. HOW TO DIS- ASSOCIATE NSG FROM VNIC?



|  |  |
| --- | --- |
|  | * Networking 🡪 Network Interface 🡪 Disassociate. This will disassociate the NSG from the VNIC * Once it is disassociated – The VM will be open for all request – as there is no NSG rule associated to it. |

1. ATTACHING AND DETACHING NSG

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | * As the NSG is a separate resource – it can attached and detached form Network interface / Subnet * On the same note, being a separate resource – If we have NSG attached to NIC – we can detach it from NIC and attach the same on Subnet and vice versa * As shown below |
|  | | 1. Step 1 - NSG ATTACHED TO NIC | |
|  | STEP 2: DISASSOCIATE THE NSG FROM NIC | | |
|  | | STEP 3: ATTACH TO THE SAME NSG TO SUBNET | |
|  | | | |

### COMMUNCATION WITH VIRTUAL NETWORKS

* The virtual network is isolated network in the cloud, hence by default there is no communication can happen between the VNETS
* The communication between VNETS can be done using ***VNET PEERING*** – where the VNETS can be either with a same region or different region
* The traffic between virtual machines in peered virtual networks uses the **MICROSOFT BACKBONE INFRASTRUCTURE** not via internet
* In a peered VNET – the VMs can communicate using their private IP – even If they belong to a different region. They don’t need any public IP
* We can also peer virtual networks that are located across different subscriptions.
* The virtual networks can't have overlapping CIDR blocks

#### VIRTUAL NETWORK PEERING

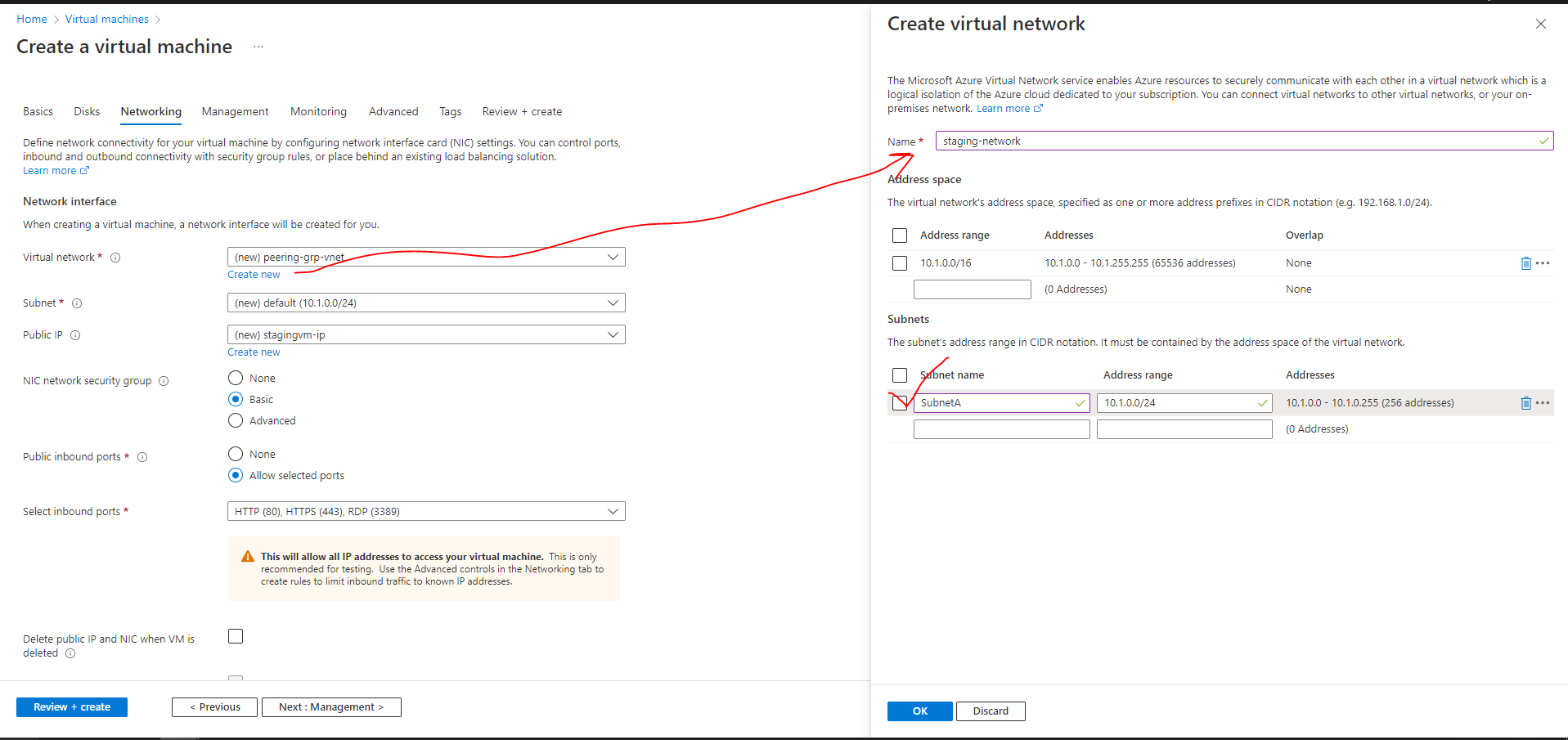
Azure supports the following types of peering:

1. **VIRTUAL NETWORK PEERING**: CONNECTING VIRTUAL NETWORKS WITHIN THE SAME AZURE REGION.
2. **GLOBAL VIRTUAL NETWORK PEERING**: CONNECTING VIRTUAL NETWORKS ACROSS AZURE REGIONS.

|  |  |
| --- | --- |
|  |  |
| **VNET PEERING NOT TRANSITIVE** | * If the VNET peering has been done between Region 1 🡪Region 2 and between Region 2🡪 Region 3 . Then doesn’t mean that a VM from Region 1 can able to communicate to a VM in Region 3 * A separate peering has to be done for this communication between Region 1 and Region 3 |

##### VIRTUAL NETWORK PEERING LAB

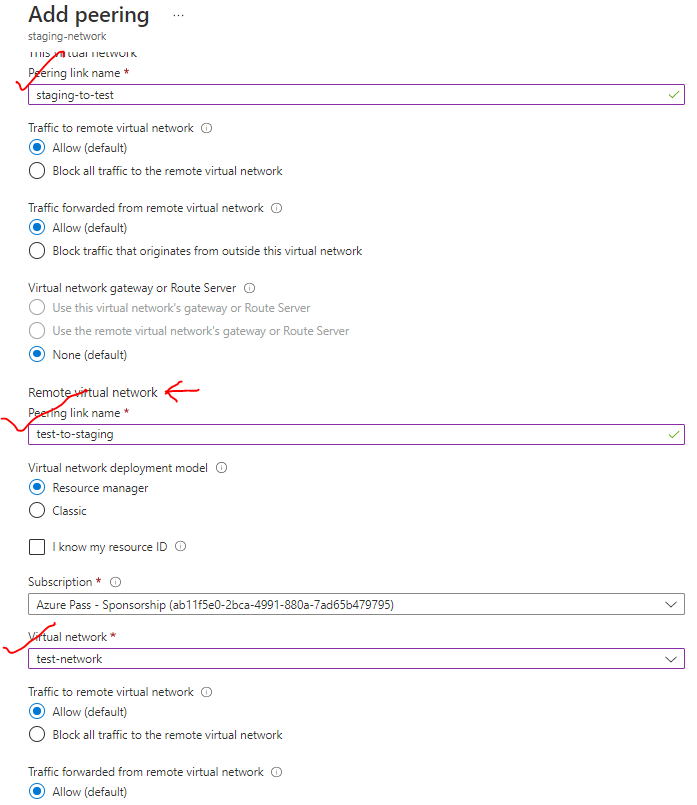
|  |  |
| --- | --- |
|  | Step 1 – Create 2 VMs which will be part of 2 different VNET.  Note – To validate the communication between VM sin different VNET   1. Install IIS on the 1st VM(stagingvm) 2. After peering - login to testVm (using RDP) and access the ISS of stageVm via its private IP |



###### PEERING THE VNETS

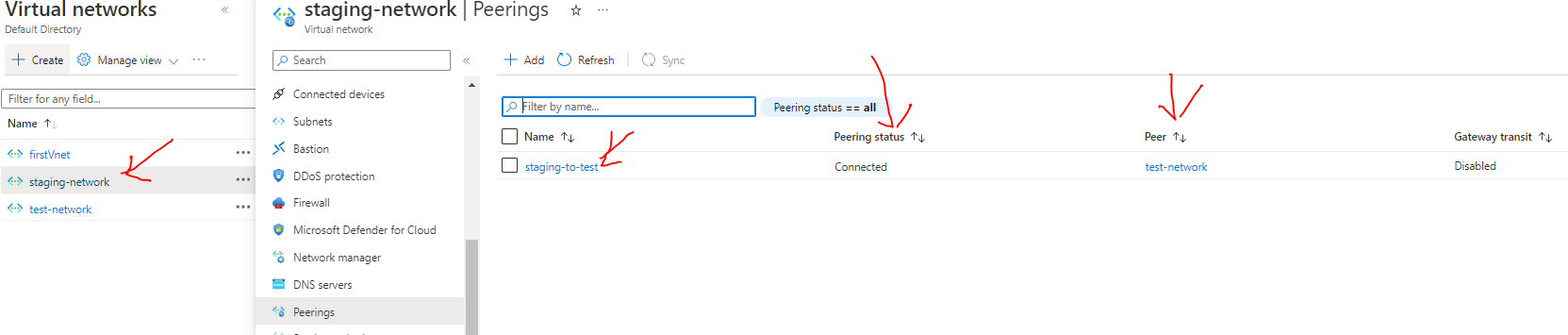
* In peering configuration – we need create the link from both sides. This will allow cross communication between the virtual machines and both of the networks.
* Go to either of the Vnet 🡪 Peering 🡪 Add (This configuration will establish cross peering between VNets )

|  |  |
| --- | --- |
| Peering link name | We need to provide 2 peering link name one from each VNET |
| Virtual Network | Since we are on Staging network – hence the VNET we want to establish the peering has been provided i.e., “test-network” |

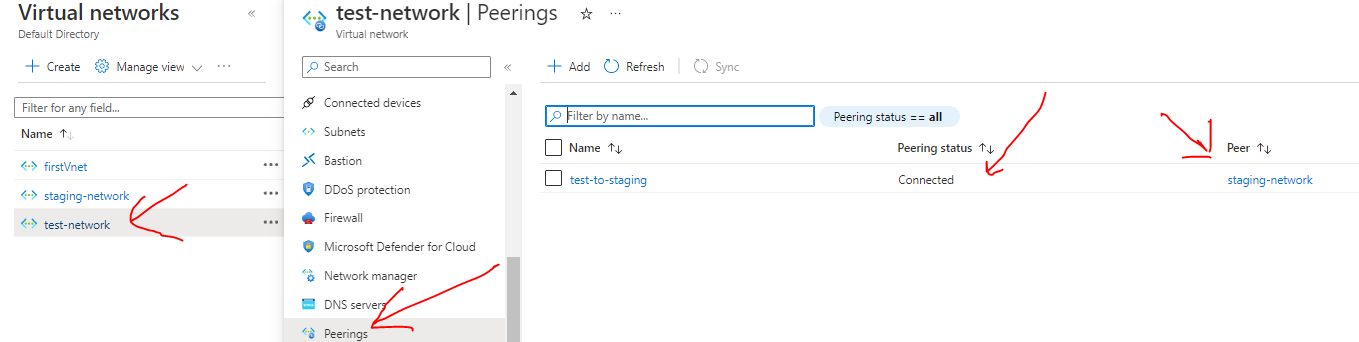


PEERING ESTABLISHED

STAGE TO TEST



TEST TO STAGE

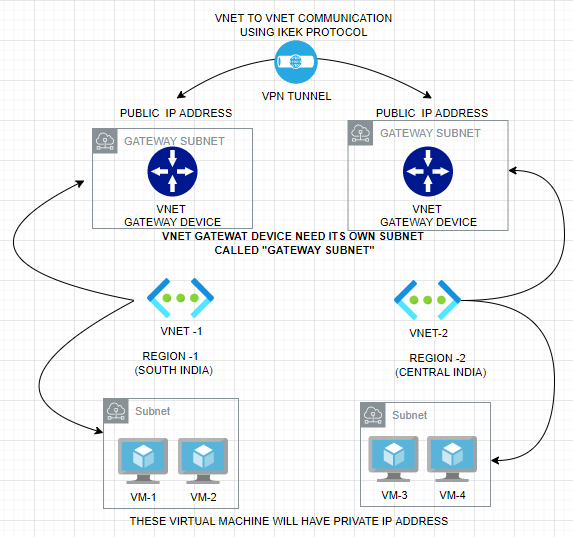


* **After peering RDP to testvm 🡪 Access the IIS using the private IP of stagevm**

#### VPN (VIRTUAL PRIVATE NETWORK)

|  |  |
| --- | --- |
| How Does a VPN Work Illustration | * VPN are used to create a private network. The data is transmitted through an encrypted tunnel over the internet. * The VPN client software on our computer encrypts the data traffic and sends it to the VPN server through a secure connection. The data goes through the ISP, but it’s been so scrambled because of the encryption, they can no longer decipher it. * The encrypted data from our computer is decrypted by the VPN server. The data is then sent to the internet and receives a reply that’s meant for us. The traffic is then encrypted again by the VPN server and is sent back to us. * The VPN client on your device will decrypt the data so you can understand and use it. |

##### VNET TO VNET CONNECTION USING GATEWAY DEVICE



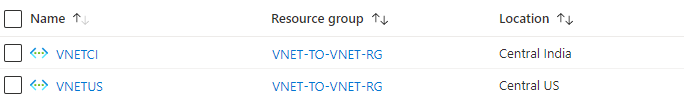
* In VNET-to-VNET connection using gateway device. We need to deploy a VNET gateway device on both VNETs
* The VNET requires, its own a dedicated Subnet – which we refer as **Gateway Subnet**
* When we deploy a VNET gateway device it will have a Public IP address (static IP address) associated with it, by default
* Between these Virtual Gateway device – we establish a VPN Tunnel. The protocol used for this communication is **IKEK**
* Now the VMs in VNET-1 and VNET-2 can be able to communicate with each other via VPN connection(tunnel) using their private IP address.

###### SETTING UP GATEWAY COMMUNICATION

1. STEP 1: Create 2 V-NETS -Let’s say in two regions (Central India and Central US). While creating the VNET – a default subnet is also created where we will deploy the VMs.
2. STEP 2: While creating the V-NET make sure we select different IP address range for both the VNETs

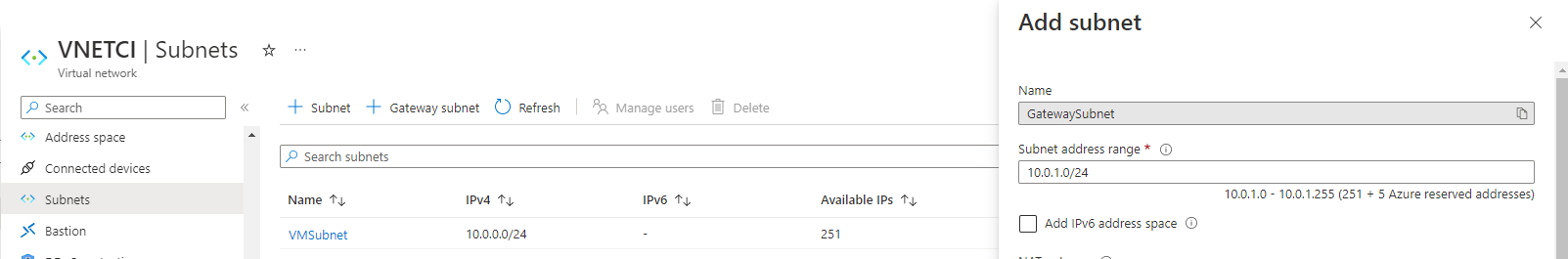
|  |  |
| --- | --- |
| **VNET -1**  **(CENTRAL INDIA)** |  |
| **VNET -2**  **(CENTRAL US)** |  |

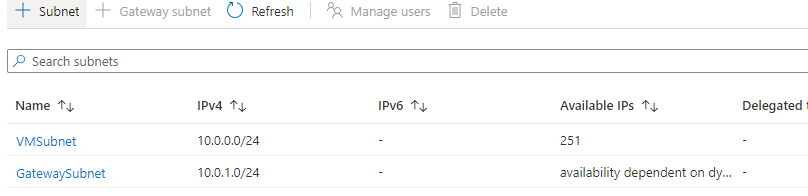
1. STEP 3: VNET CREATED



1. STEP 4: DEPLOYING A GATEWAY SUBNET

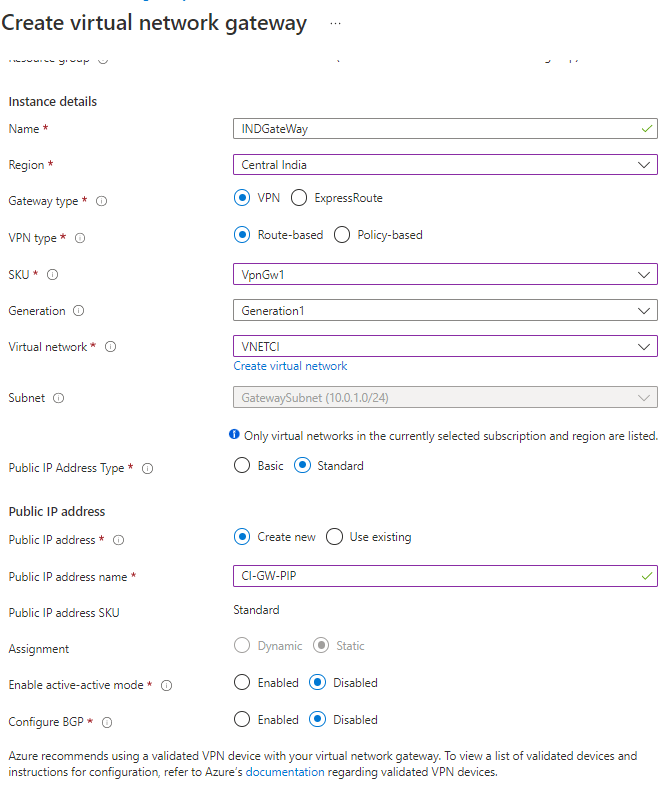
* To deploy the Gateway subnet 🡪 Navigate to Corresponding subnet

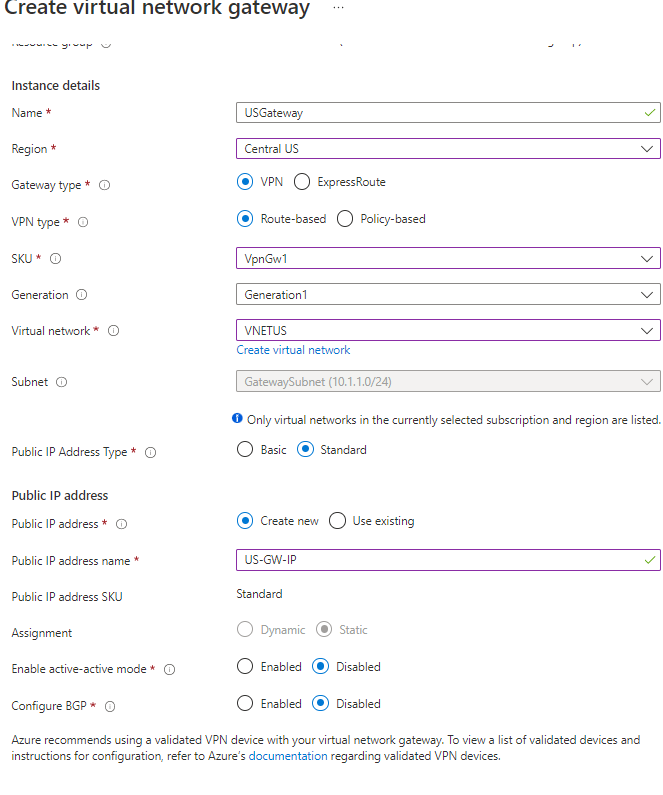




1. STEP 5: DEPLOYING VITUAL NETWORK GATEWAY DEVICE

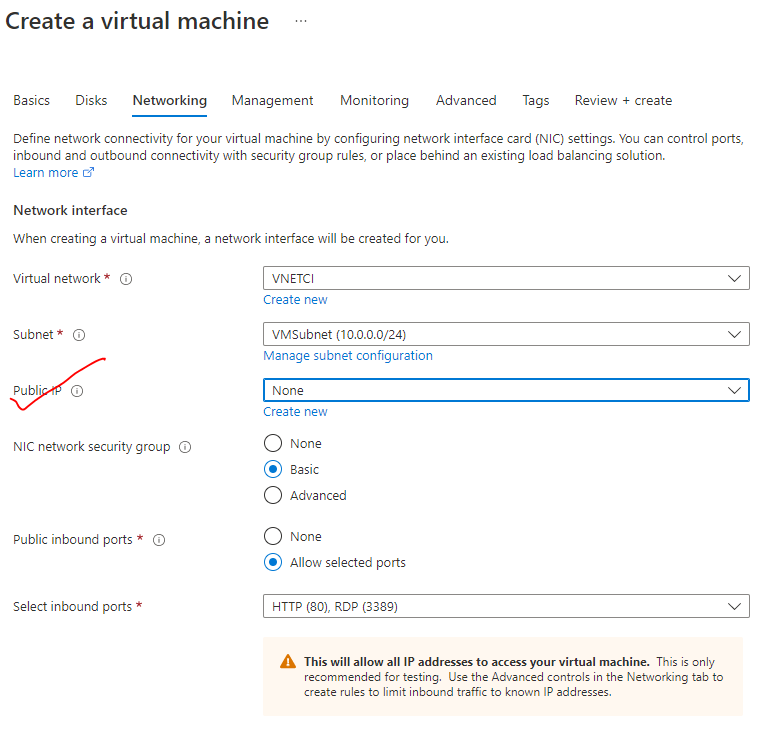
We need to deploy gateway device in both the Virtual networks. As shown below





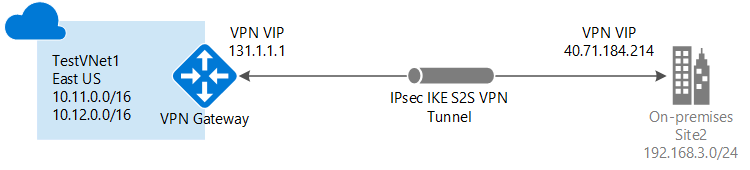
1. STEP 6: DEPLOYING VM IN THE VNET

* ***We create VM in a subnet – as the communication will happen using private IP -hence we don’t need have public address***



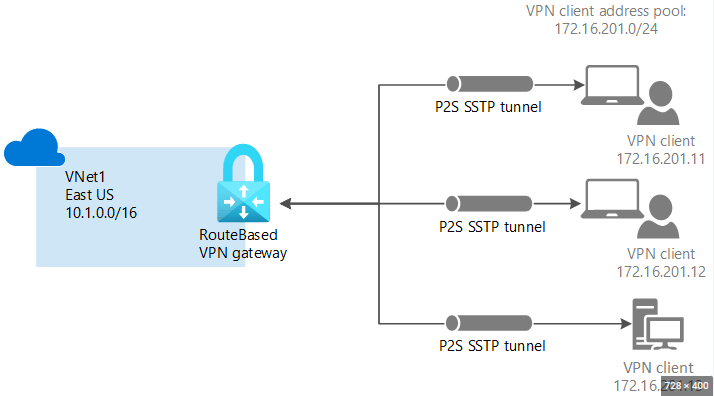
#### TYPES OF VPN CONNECTION

##### SITE TO SITE VPN CONNECTION



* It’s a VPN connection between two different site i.e S2S connections are used when we want to connect an entire on premises network onto as the virtual network.
* As it’s a VPN connection , it’s is a private communication(using private IPs of the resources) between the sites via a virtual tunnel usinf IKEK protocol.

##### POINT TO SITE VPN CONNECTION



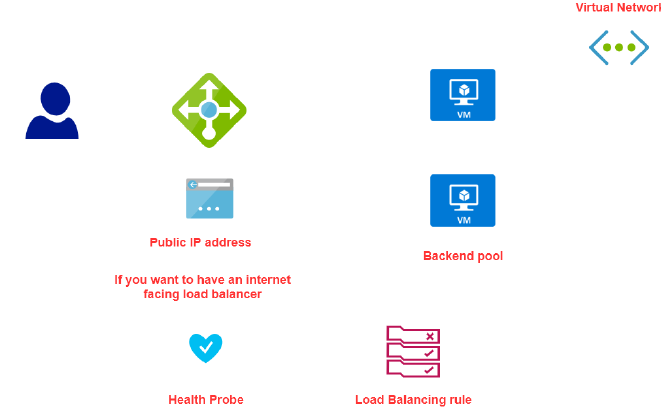
* **A Point-to-Site (P2S) VPN gateway connection lets you create a secure connection to the virtual network from an individual client computer. A P2S connection is established by starting it from the client computer.**
* This solution is useful for telecommuters who want to connect to Azure V-Nets from a remote location, such as from home or a conference. P2S VPN is also a useful solution when we have only a few clients that need to connect to a V-Net.
* So- a secure connection is first made from the client machine onto a virtual network. And then via that VPN connection, we can then use the private IP address of the Azure virtual machine for the communication. Hence we don’t need a public IP address of the VM

|  |  |
| --- | --- |
| **POINT TO SITE CONNECTION** | **SITE TO SITE CONNECTION** |
| 1. P2S connection can be used when small number of clinet machines has to connect with the VNET. | 1. S2S connections are used when we want to connect an entire on premises network onto as the virtual network. |

## LOAD BALANCER

|  |  |
| --- | --- |
|  | * The Load Balancer Service, distribute the user requests across the pool of resources (e.g., Azure virtual machines) to maintain availability. * There are many distributions-based software available, the basic one is Azure Load Balancer. * *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

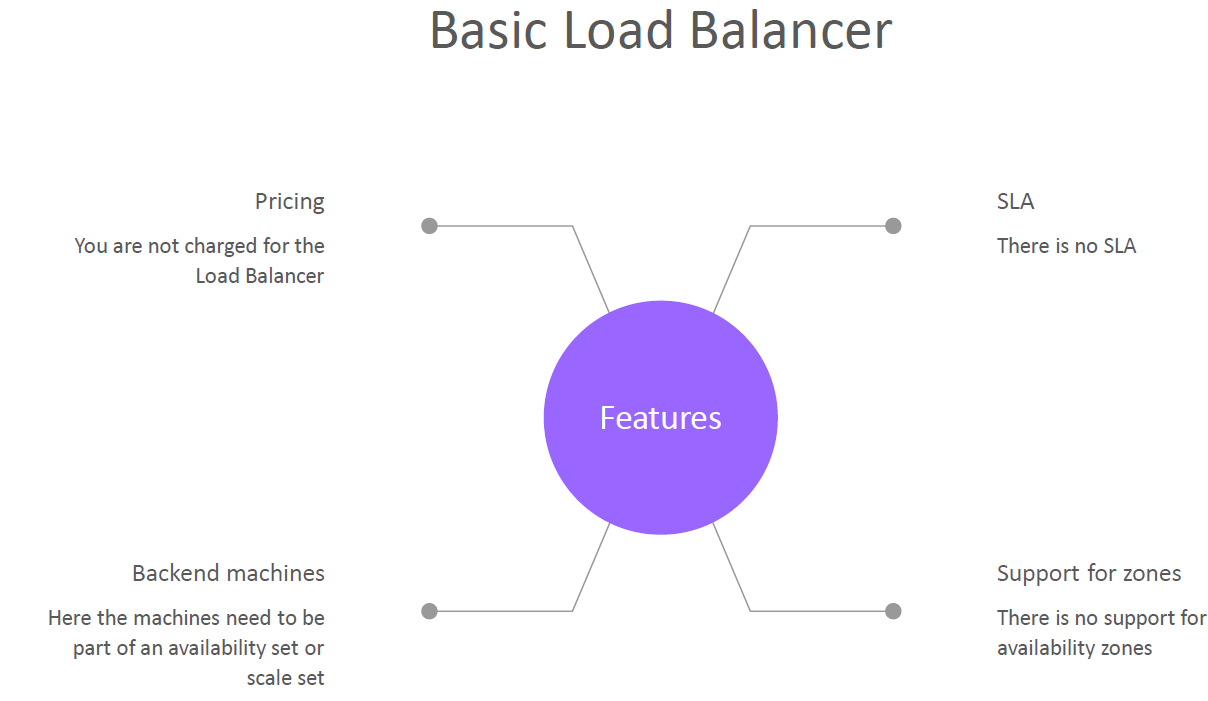
#### PUBLIC LOAD BALANCER

* 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



#### 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)***
* ***STEP 2:*** *Install IIS with a Default.html page*
* ***STEP 3:*** *Create a Public IP address (Note public IP address is a resource in Azure). The Public IP will be assigned to the Load Balancer. Also known as Front End public IP address*
* ***STEP 4:*** *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 5:*** *Configure the health probe to check the health of VM in the backend pool*
* ***STEP 6:***  *Create Load balancing Rules – which will give the routing of request to the VM in backend pool*
* ***STEP7:*** *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**

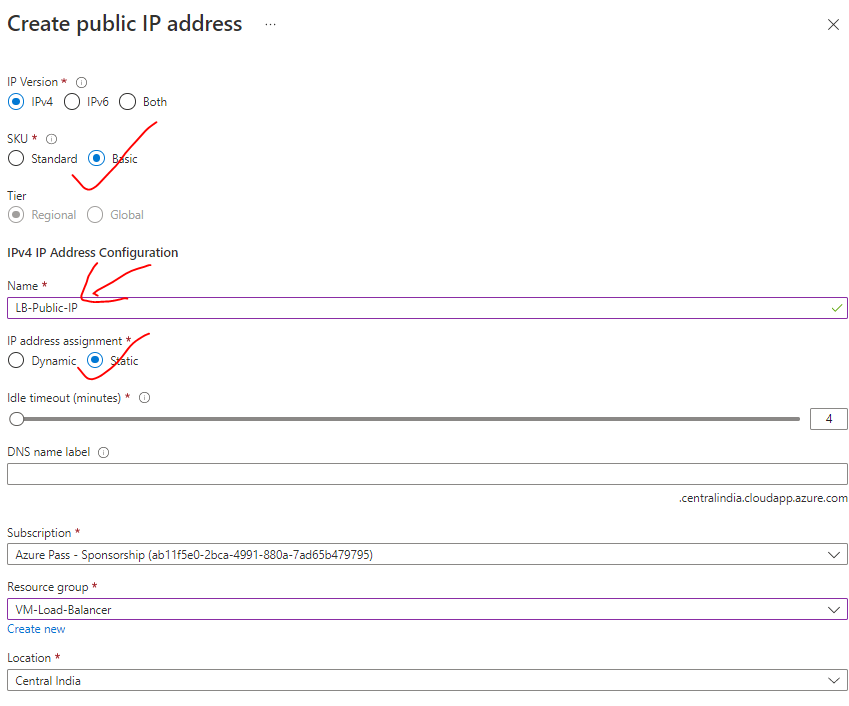


|  |  |
| --- | --- |
|  | * The communication between the Load balancer and the VMs in the backend pool – happens using private IP address * For now – let’s keep the public IP address so that we can deploy IIS and validate it * After that we can disassociate the public IP address from the VM as the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



***STEP 4:*** *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 |

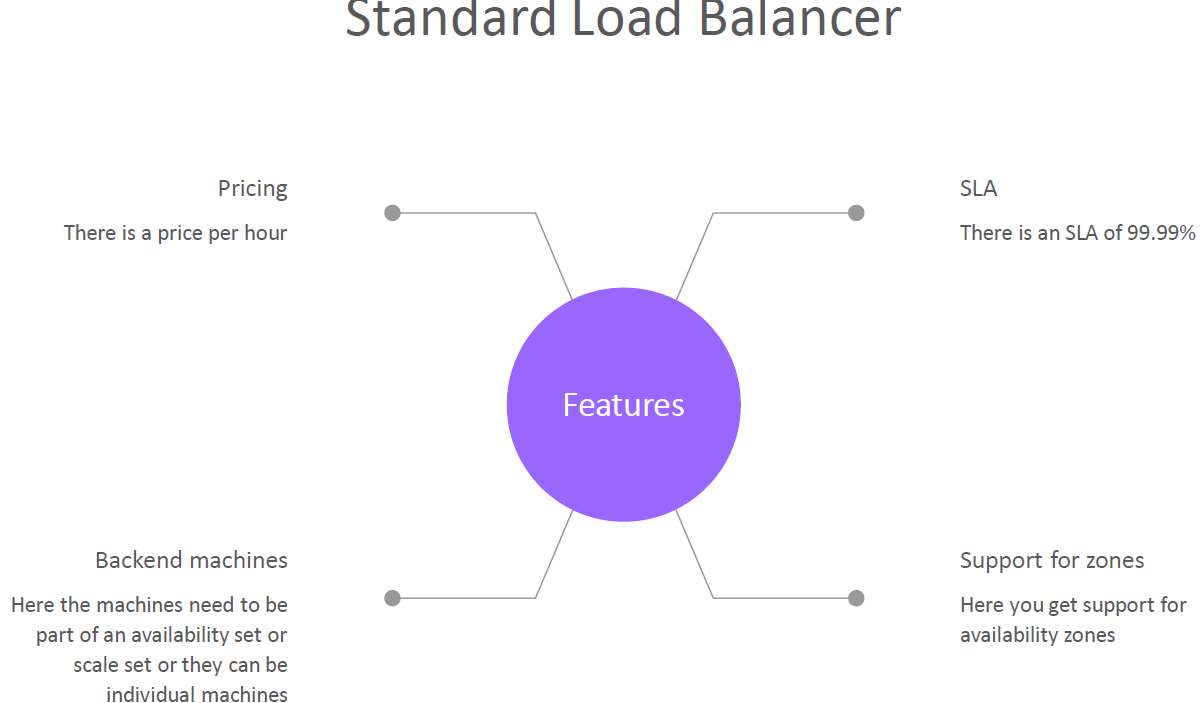
CONFIGURING LOAD BALACING RULES

|  |  |
| --- | --- |
|  | 1. PORT- 80 2. 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 NAT RULES

* For the load balancer, we have our backend virtual machines, which are part availability set.
* Since, these VMs don’t have public IP addresses, we can’t be able to connect these VM to internet or RDP onto these machines from our workstations
* For such use cases, Load balancer provide option to connect using its own public IP address - onto a port or use a particular service on the virtual machine- this is done with the help of ***NAT (Network Address Translation)***
* Using the virtual machine scale self-service - we were able to spin up multiple machines based on certain conditions. For example - let’s say the CPU percentage of the machines and the skill set was going beyond a particular threshold, then we would increase the number of machines that were part of the skill set.
* Scale Set give a capability on the platform to scale
* Azure Load Balancer service ensures that all the user requests are evenly distributed across these machines so that the load is uniformly dispersed across the machines that are part of your scale set.
* This service is used to distribute the incoming network traffic across a group of backend resources of servers
* You can define two types of load balancers –***Public or Private Load Balancers***
* You have 2 SKUs for the Load Balancer –**Standard and Basic Load Balancer**

### STANDARD LOAD BALANCER



## 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
* 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 you provision inside the virtual network. It provides secure and seamless **RDP/SSH connectivity to the virtual machines directly from the Azure portal over TLS**.
* *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 has to be part of a separate subnet in our virtual network. The name of the subnet named as “***AzureBastionSubnet***”

### SETTING UP BASTION SERVICE

STEP 1: CREATE A VM

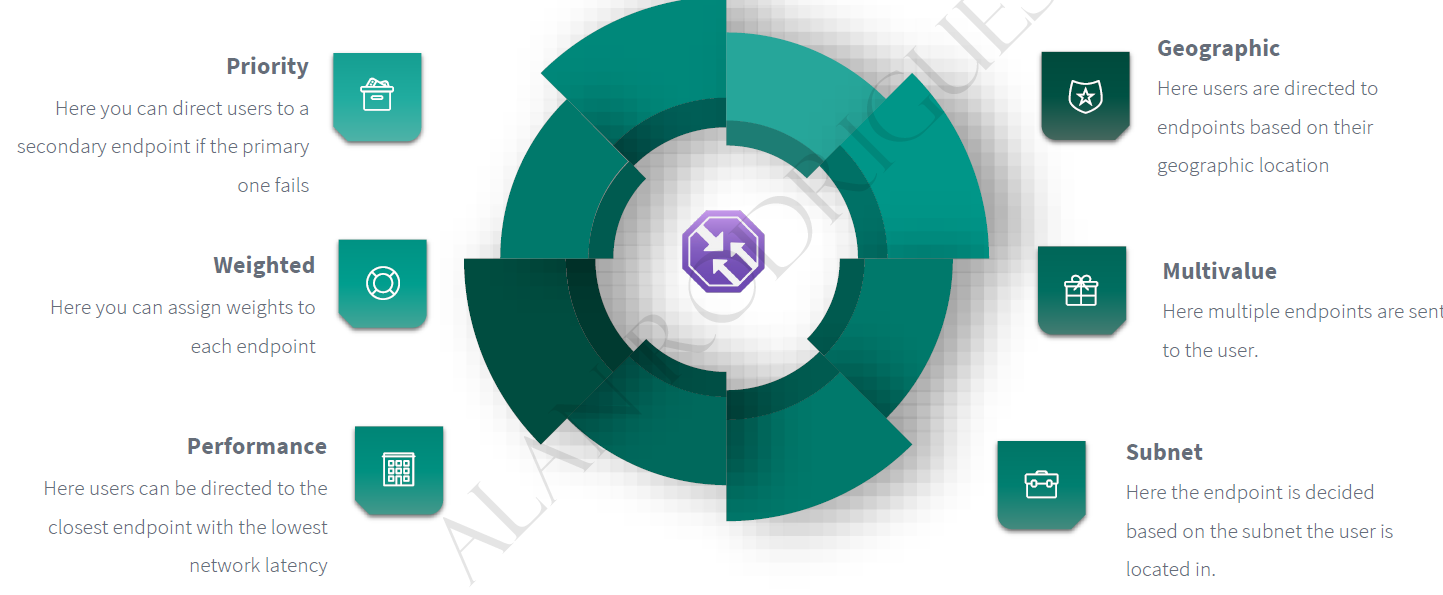
## AZURE TRAFFIC MANAGER

* **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 datascenter) 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.

### 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

### 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.

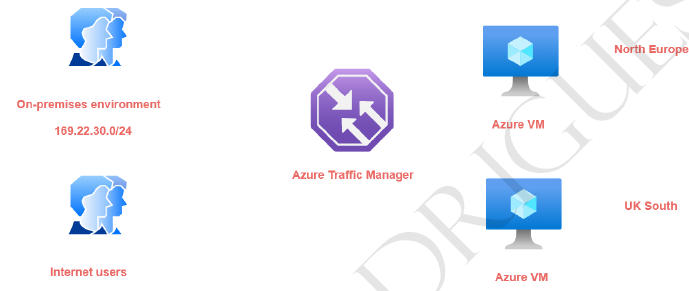
#### SETTING GEOGRAPIC ROUTING METHOD

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

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

### SUBNET ROUTING METHOD



|  |  |
| --- | --- |
|  | 1. 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. 2. 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. 3. 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.**   1. 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.