

## 32) 25 August 2024 – AzureLoadBalancer

1) Very big use case – Say like, I have done end to end automation, any customer who comes for onboarding, so we have automated making management group, subscription, resource group etc.

2) Offboarding – If any needs to be decommissioned, so we will remove tfvars then it will be decommissioned.

3) Subscription never gets deleted, it actually gets cancelled. Make another management group of decommissioning management group in which make prod and non prod management group.

i) Onboarding prod management group

ii) Offboarding non prod management group

4) After cancelling subscription we can again activate it like for 90 days, but after 90 days subscription gets auto deleted.

5) For 7 days we can recover everything in subscription and after 7 days only we can forever cancel subscription from management group

6) AD GROUP – Now if we want to give access on subscription, then make AD group and there is a different provider for it eg – azuread

7) Global administrator role – biggest role in azure

i) group creator role – group creation and group deletion role

8) Intra ID and Azure AD are the same things

+++++

10 Yesterday we removed public ip and put bastion

11) Under vnet 2 subnets are created.

12) First vm should be created in 1<sup>st</sup> subnet and second vm should be created in 2<sup>nd</sup> subnet.

```
providers.tf ...\azurerm_Virtual_Machine
main.tf ...\azurerm_Virtual_Machine
main.tf ...\azurerm_Virtual_Netw
terraform.tfvars ...

Modules > azurerm_Virtual_Machine > main.tf > resource "azurerm_linux_virtual_machine" "vm" > name
1 #####datablock for subnet
2 data "azurerm_subnet" "subnets" {
3   for_each = var.vms_map
4   name     = each.value.subnet_name
5   virtual_network_name = each.value.virtual_network_name
6   resource_group_name = each.value.resource_group_name
7 }
8
9 #####nic
10 resource "azurerm_network_interface" "nic" {
11   for_each = var.vms_map
12   name     = each.value.nic_name
13   location = each.value.location
14   resource_group_name = each.value.resource_group_name
15 }
16
17 ip_configuration {
18   name = "nic_ip1"
19   subnet_id = data.azurerm_subnet.subnets[each.key].id
20   private_ip_address_allocation = "Dynamic"
21 }
22
23 #####Virtualmachine
24 resource "azurerm_linux_virtual_machine" "vm" {
25   for_each = var.vms_map
26   name     = each.value.vm_name
27   resource_group_name = each.value.resource_group_name
28   location = each.value.location
29   size     = each.value.size
30   admin_username = data.azurerm_key_vault_secret.kvsecret_use
31   admin_password = data.azurerm_key_vault_secret.kvsecret_pas
32   disable_password_authentication = false
33 }

Modules > azurerm_Virtual_Machine > terraform.tfvars > vms_map
1 vms_map = {
2   vm1 = {
3     subnet_name = "subnet_frontend_dev1"
4     virtual_network_name = "vnetdev1"
5     resource_group_name = "rgdev1"
6     nic_name = "Virtual_nic_1"
7     location = "centralindia"
8     vm_name = "Virtual-machine_1"
9     size = "Standard_F2"
10   }
11
12   vm2 = {
13     subnet_name = "subnet_backend_dev2"
14     virtual_network_name = "vnetdev1"
15     resource_group_name = "rgdev1"
16     nic_name = "Virtual_nic_2"
17     location = "centralindia"
18     vm_name = "Virtual-machine_2"
19     size = "Standard_F2"
20   }
21 }
22
23
```

```

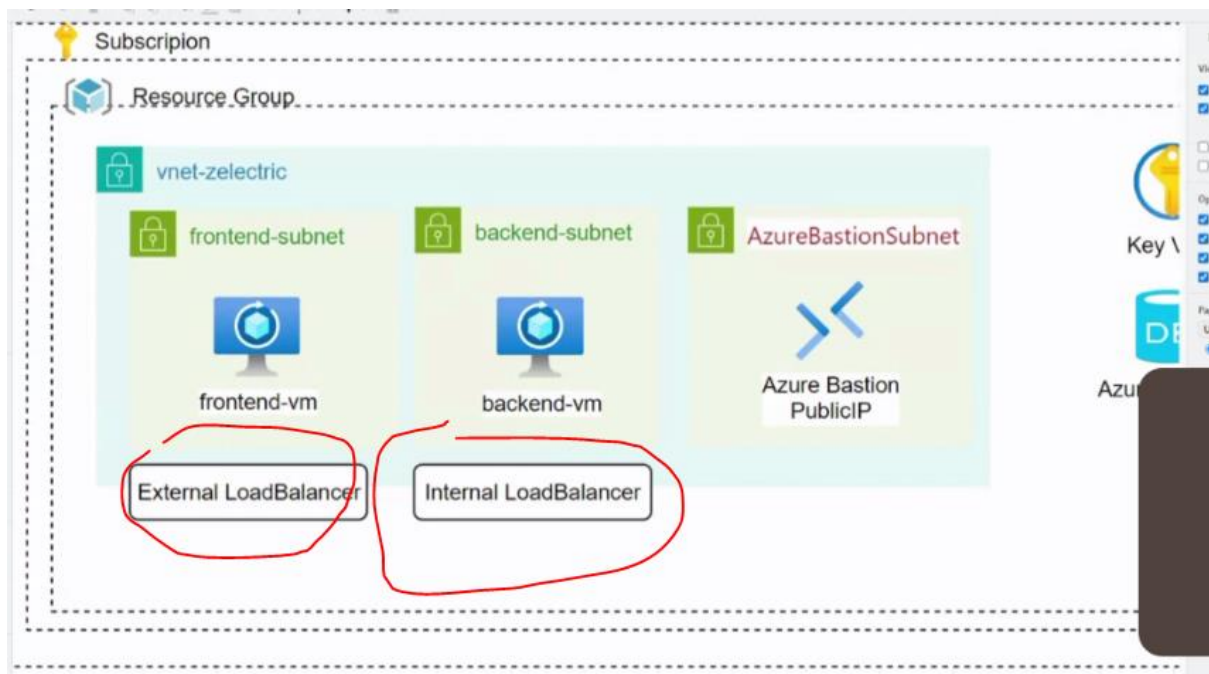
Modules > azurem_virtual_machine > main.tf > resource "azurerm_linux_virtual_machine" "vm" > name
10 resource "azurerm_network_interface" "nic" {
16   ip_configuration {
19     private_ip_address_allocation = "Dynamic"
20   }
21 }
22
23 #####Virtualmachine
24 resource "azurerm_linux_virtual_machine" "vm" {
25   for_each = var.vms_map
26   name     = each.value.vm_name
27   resource_group_name = each.value.resource_group_name
28   location = each.value.location
29   size     = each.value.size
30   admin_username = data.azurem_key_vault_secret.kvsecret_username
31   admin_password = data.azurem_key_vault_secret.kvsecret_password
32   disable_password_authentication = false
33   network_interface_ids = [azurerm_network_interface.nic[each.key].id]
34 }
35
36 os_disk {
37   caching = "ReadWrite"
38   storage_account_type = "Standard_LRS"
39 }
40
41 source_image_reference {
42   publisher = "Canonical"
43   offer     = "0001-com-ubuntu-server-jammy"
44   sku      = "22_04-lts"
45   version  = "latest"
46 }
47 }

Modules > azurem_virtual_machine > terraform.tfvars > vms_map
1 vms_map = {
2   vm1 = {
3     subnet_name = "subnet_frontend_dev1"
4     virtual_network_name = "vnetdev1"
5     resource_group_name = "rgdev1"
6     nic_name = "Virtual_nic_1"
7     location = "centralindia"
8     vm_name = "Virtual-machine_1"
9     size = "Standard_F2"
10   }
11   vm2 = {
12     subnet_name = "subnet_backend_dev2"
13     virtual_network_name = "vnetdev1"
14     resource_group_name = "rgdev1"
15     nic_name = "Virtual_nic_2"
16     location = "centralindia"
17     vm_name = "Virtual-machine_2"
18     size = "Standard_F2"
19   }
20 }
21
22
23

```

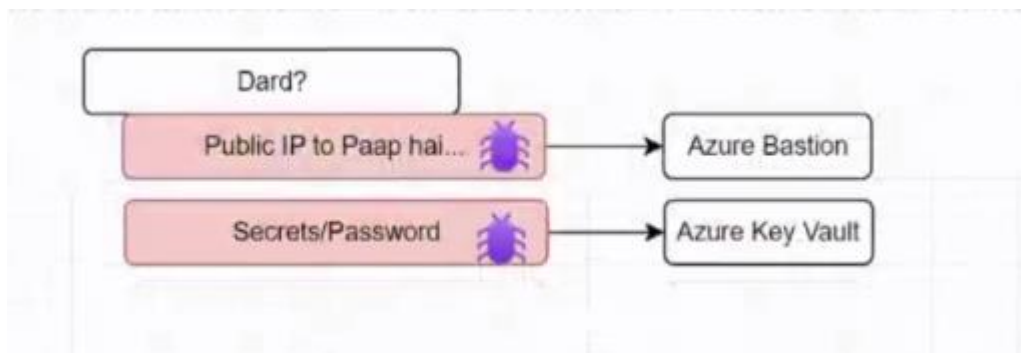
## AGENDA – How to distribute load?

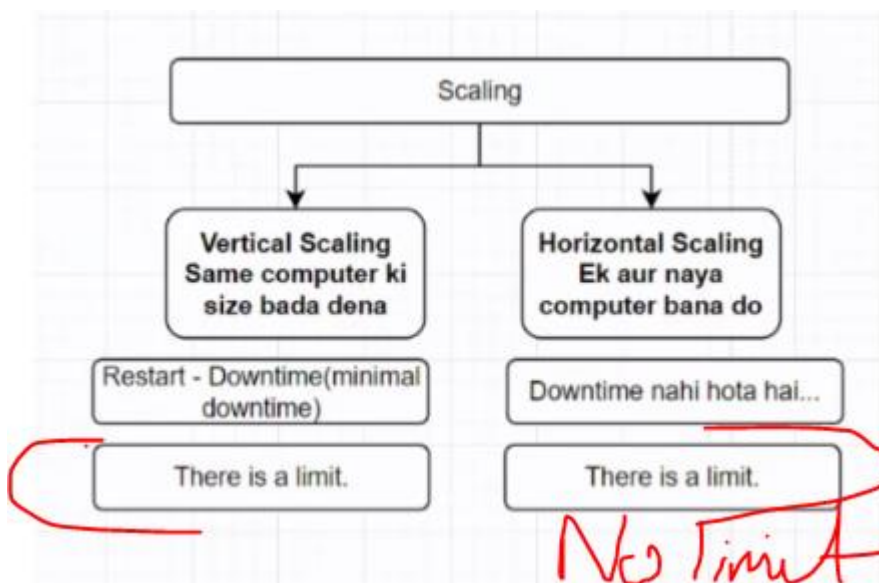
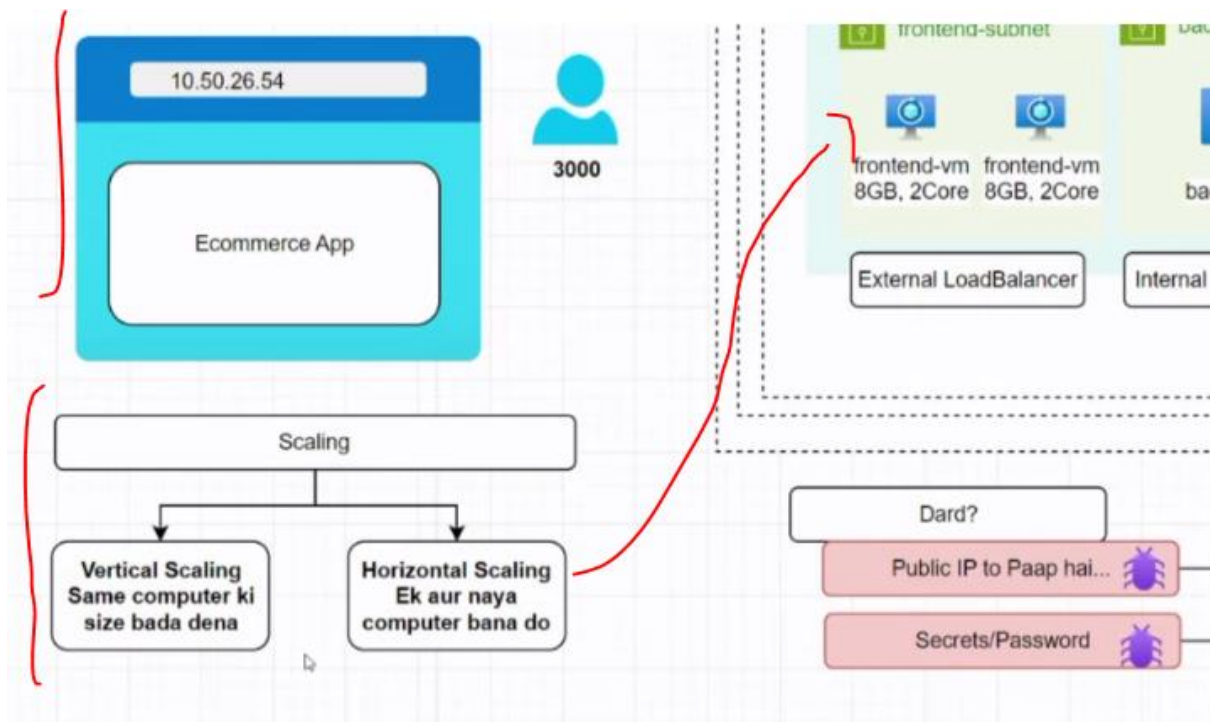
- 1) VMSS – To automatically increase and decrease load VMs
- 2) For frontend VM we use load balancer that has public ip
- 3) For backend VM, we use internal load balancer that has no public ip



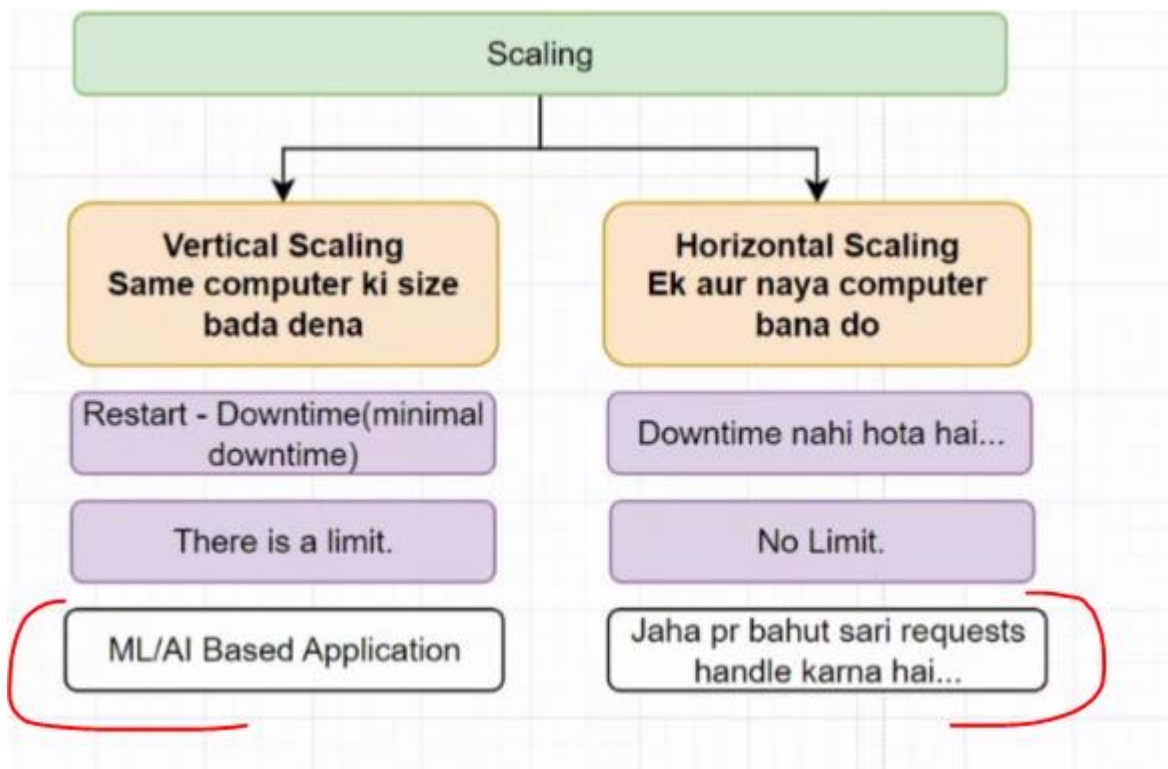


- 4) For big environments loadbalancer is not used, it is only used internally in landing zone
- 5) So instead of load balancer azure provides “application gateway” for big services. But it is a regional service not global so for global we use “traffic manager” and “frontdoor”.
- 6) “Frontdoor” is global regional service which are very costly services.

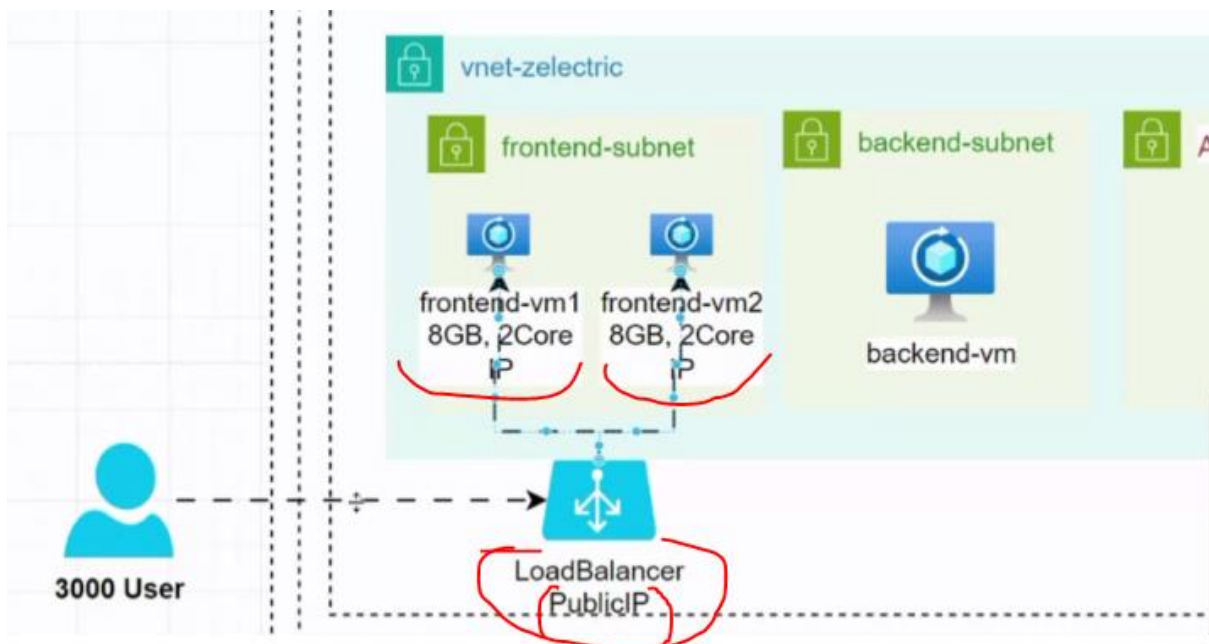




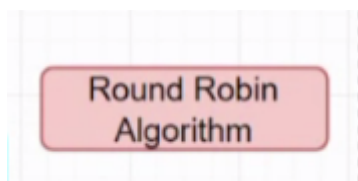
7) Types of scaling



#### 8) Working of load balancer



#### 9) Load balancer uses round robin



10) Backend pool – A place where load goes

i) frontend ip configuration – It communicates or handles load before load balancer

ii) frontend ip configuration – fitted with public ip – called as external load balancer

iii) frontend ip configuration – fitted with private ip – called as internal load balancer

11) Load balancing rules –

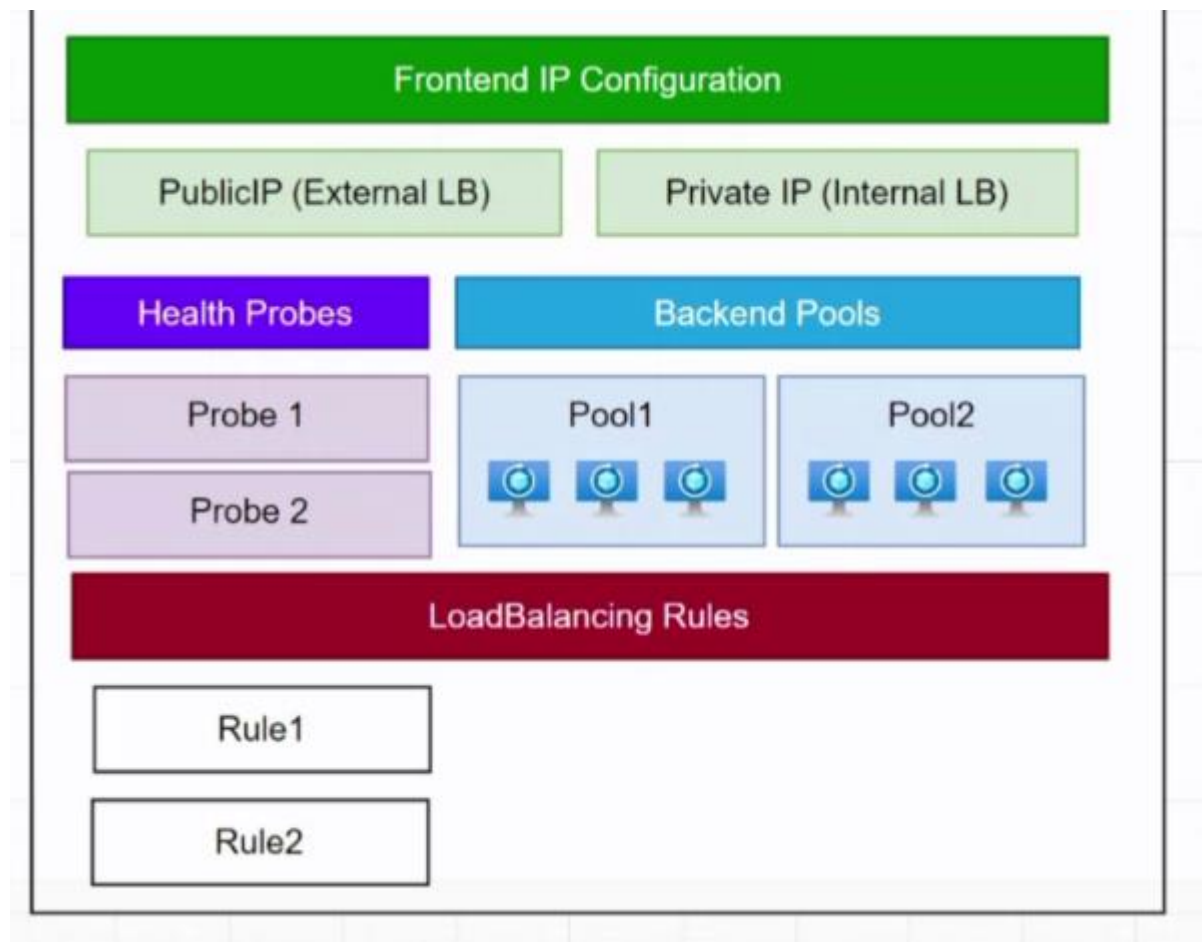
12) 4 components if LB

i) frontend ip configuration = comes before load balancer

ii) backend pool – whole load will be distributed here

iii) Load balancing rules – will happen by some rules

iv) Health probe – Before putting load we will check its health



+++++

**AGENDA – Create 2 vms in frontend subnet and put nginx in both of them and will put frontend website. Also will put load balancer**

1) We will create 1 vnet, 1 subnet

2) create 1 vm



3) 1 computer can have how many ports/ doors

65536

4) Remove public ip

Virtual machines >

## Create a virtual machine

Help me create a low cost VM | Help me create a VM optimized for high availability | Help me choose the right VM size for my workload

Password \*

Confirm password \*

### Inbound port rules

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

Public inbound ports \* ☐ None ☒ Allow selected ports

Select inbound ports \*

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

## Create a virtual machine

Help me create a low cost VM | Help me create a VM optimized for high availability | Help me choose the right VM size for my workload

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

Configure monitoring options for your VM.

### Alerts

Enable recommended alert rules ☐

### Diagnostics

Boot diagnostics ☐ Enable with managed storage account (recommended) ☐ Enable with custom storage account ☒ Disable

Enable OS guest diagnostics ☐

### Health

+++++

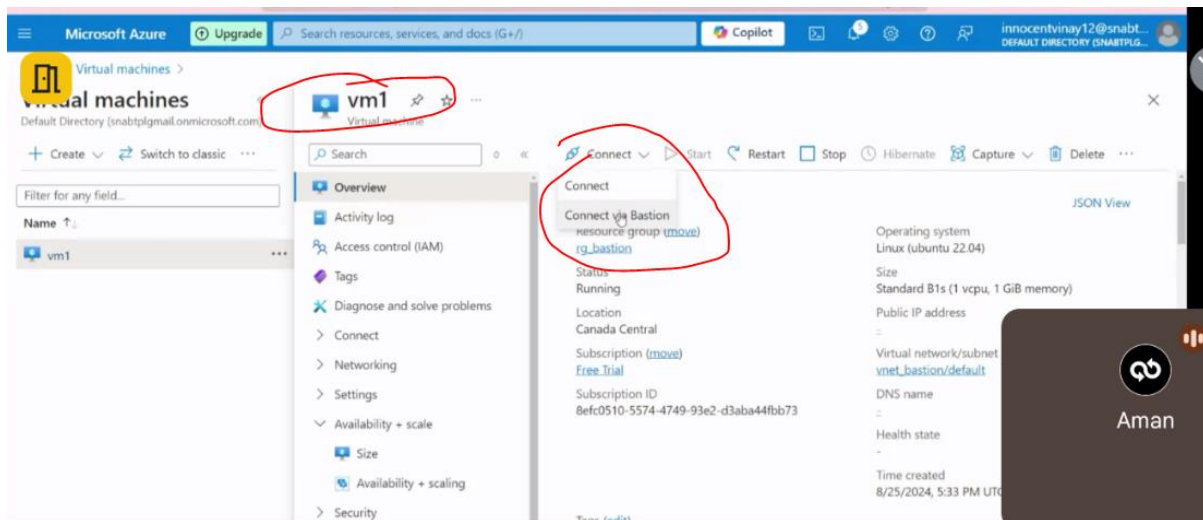
1) Create another VM 2

2) Now both vms created

+++++

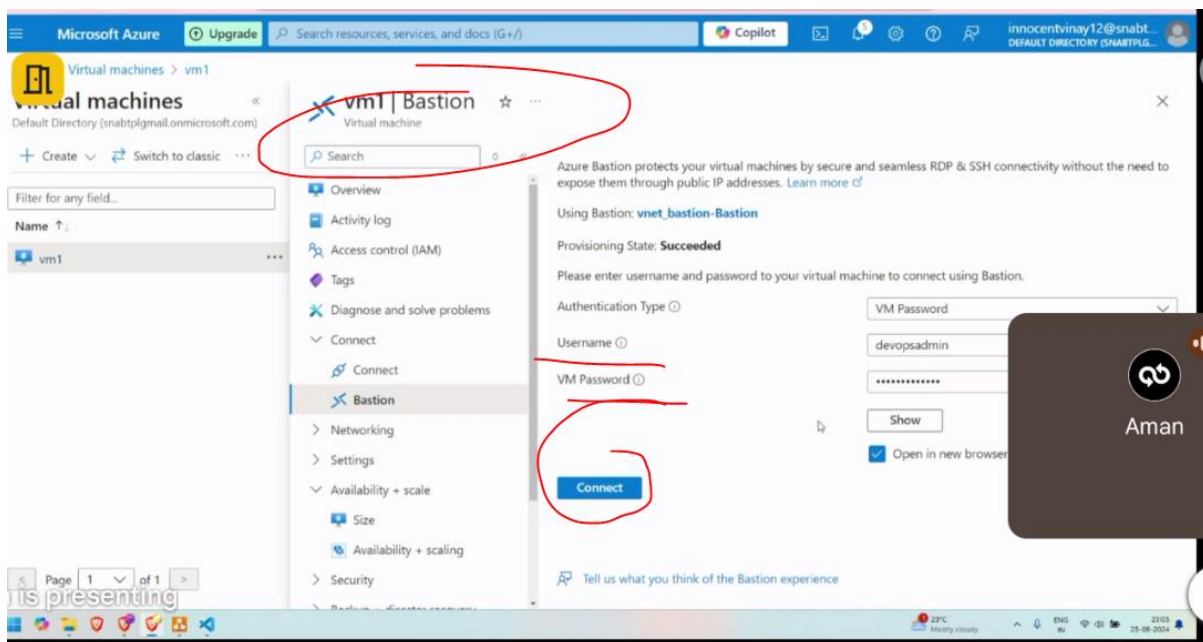
1) Now in VM1 and VM2, we have to install nginx

2) For this, go to VM1 and select “connect via bastion”



shant srivastava left

3) Then it will ask for vm id and password and then connect which will open linux machine in next tab



4) Run below commands in it

i) **sudo apt update**

ii) **sudo apt install nginx**



```

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The following packages are included with the Ubuntu system are free software;
the distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>"
See "man sudo root" for details.

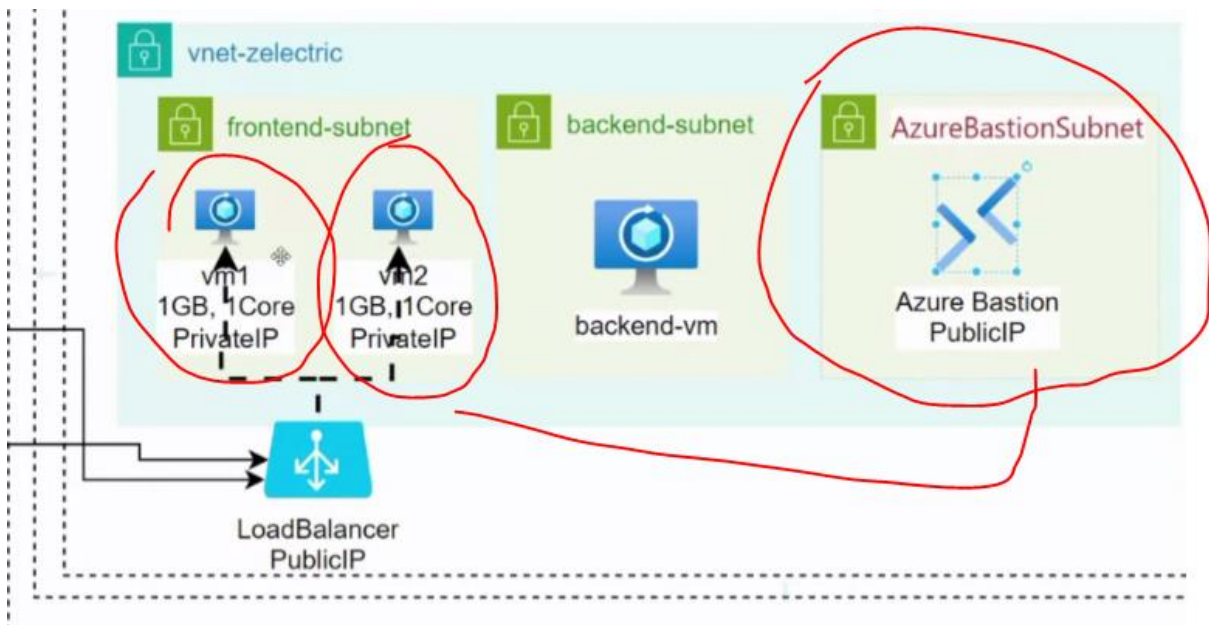
devopsadmin@vm1:~$ ls
devopsadmin@vm1:~$ sudo apt update
Hit:1 http://azure.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://azure.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:3 http://azure.archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Get:4 http://azure.archive.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:5 http://azure.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:6 http://azure.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5052 kB]
Get:7 http://azure.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:8 http://azure.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:9 http://azure.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:10 http://azure.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]
Get:11 http://azure.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1987 kB]
Get:12 http://azure.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [349 kB]
Get:13 http://azure.archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [17.8 kB]
Get:14 http://azure.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [2384 kB]
Get:15 http://azure.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [410 kB]
Get:16 http://azure.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1112 kB]
Get:17 http://azure.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [260 kB]
Get:18 http://azure.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [26.0 kB]
Get:19 http://azure.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [43.3 kB]
Get:20 http://azure.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [10.8 kB]
Get:21 http://azure.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 c-n-f Metadata [444 B]
Get:22 http://azure.archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [92.4 kB]
Get:23 http://azure.archive.ubuntu.com/ubuntu jammy-backports/main Translation-en [11.1 kB]
Get:24 http://azure.archive.ubuntu.com/ubuntu jammy-backports/main amd64 c-n-f Metadata [388 B]
Get:25 http://azure.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 c-n-f Metadata [116 B]
Get:26 http://azure.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [31.4 kB]
Get:27 http://azure.archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en [16.5 kB]
Get:28 http://azure.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 c-n-f Metadata [672 B]
Get:29 http://azure.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 c-n-f Metadata [116 B]
Get:30 http://azure.archive.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1769 kB]
Get:31 http://azure.archive.ubuntu.com/ubuntu jammy-security/main Translation-en [290 kB]
Get:32 http://azure.archive.ubuntu.com/ubuntu jammy-security/main amd64 c-n-f Metadata [13.2 kB]
Get:33 http://azure.archive.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [2388 kB]

```

5) Similarly do it for vm2

6) So now we have connected both machines vm1 and vm2 through bastion

7)

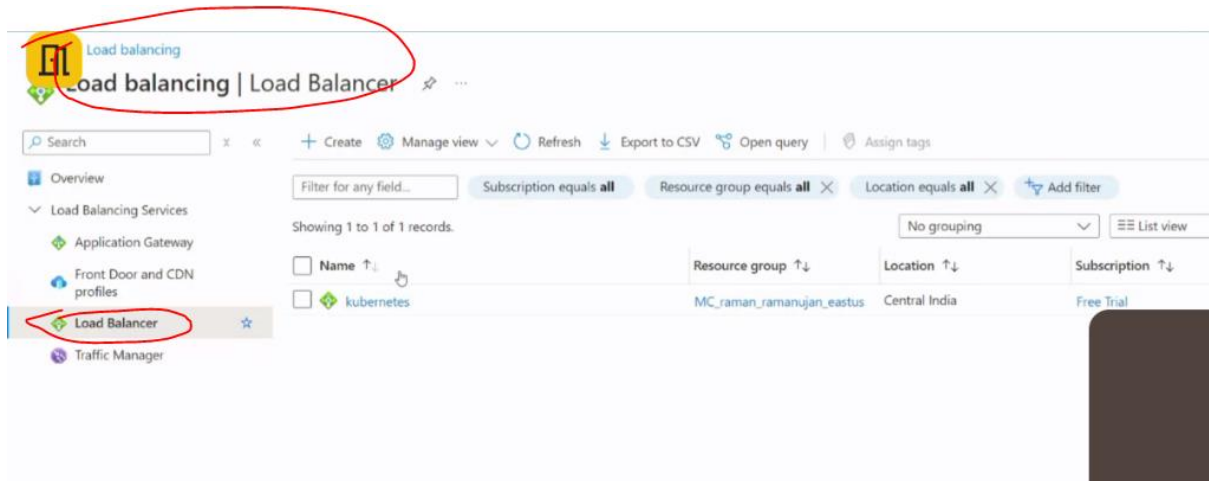


8) Now we will fit laodbalancer on both these machines

+++++

## AGENDA – Create Loadbalancer

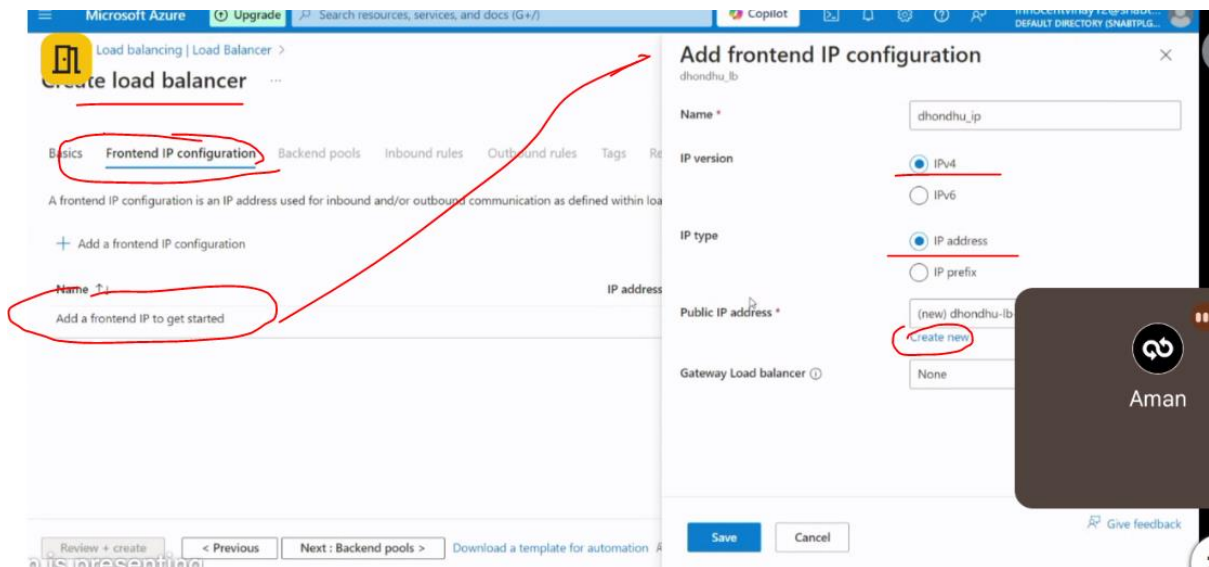
1)



2) So we know LB has 4 components

- i) frontend ip configuration = comes before load balancer
- ii) backend pool – whole load will be distributed here
- iii) Load balancing rules – will happen by some rules
- iv) Health probe – Before putting load we will check its health

3) Add ip configuration



4) Adding one more ip configuration as similar as above

Basics **Frontend IP configuration** Backend pools Inbound rules Outbound rules Tags Review + create

A frontend IP configuration is an IP address used for inbound and/or outbound communication as defined within load balancing, inbound NAT, and outbound rules.

+ Add a frontend IP configuration

Name ↑↓	IP address ↑↓
dhondhu_ip	(new) dhondhu-lb-ip (To be created)
dhondhu_ip1	vnet_bastion-ip (52.237.13.229)

## 5) Add backend pool

**Add backend pool**

Name \* dhondhu\_pool

Virtual network vnet\_bastion (rg\_bastion)

Backend Pool Configuration

☒ NIC ☐ IP address

**IP configurations**

IP configurations associated to virtual machines and virtual machine scale sets must be in same location as the load balancer and be in the same virtual network.

+ Add | - Remove

Resource Name	Resource group	Type	IP configuration	IP Address
---------------	----------------	------	------------------	------------

Save Cancel Give feedback

Microsoft Azure Upgrade Search resources, services, and docs (G+I) Copilot

**Add IP configurations to backend pool**

IP configurations associated to virtual machines and virtual machine scale sets must be in same location as the load balancer and be in the same virtual network.

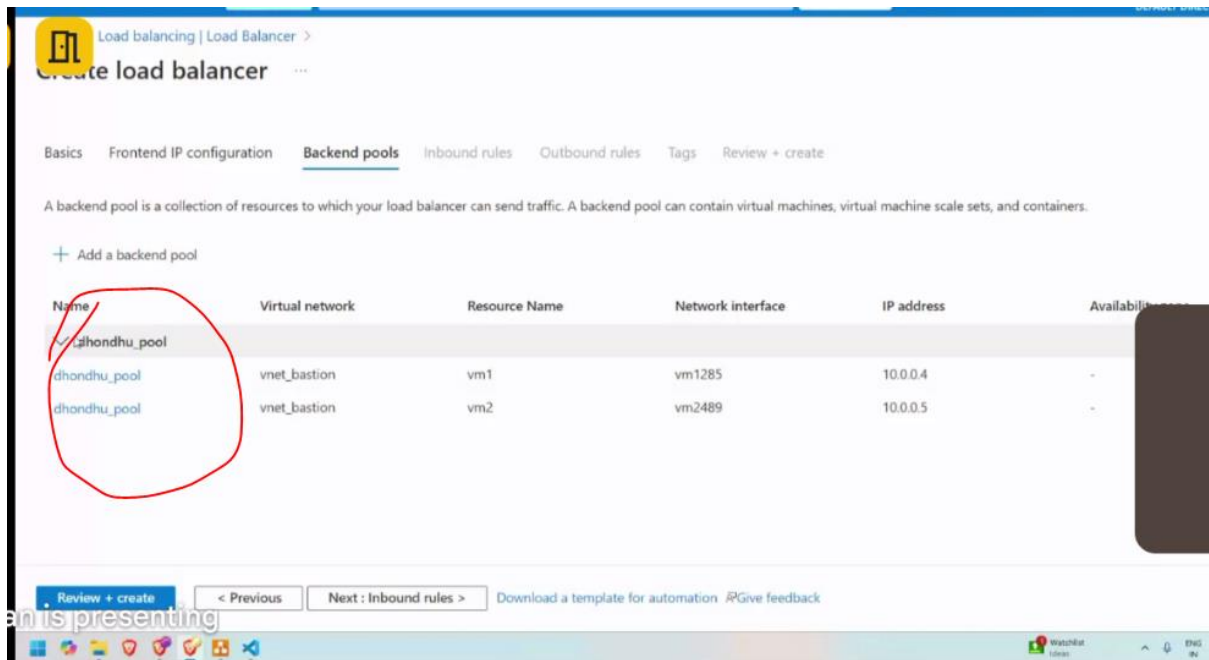
Filter by name... Location: canadacentral Virtual network: vnet\_bastion Add filter

☐ Show resources that are not available for selection

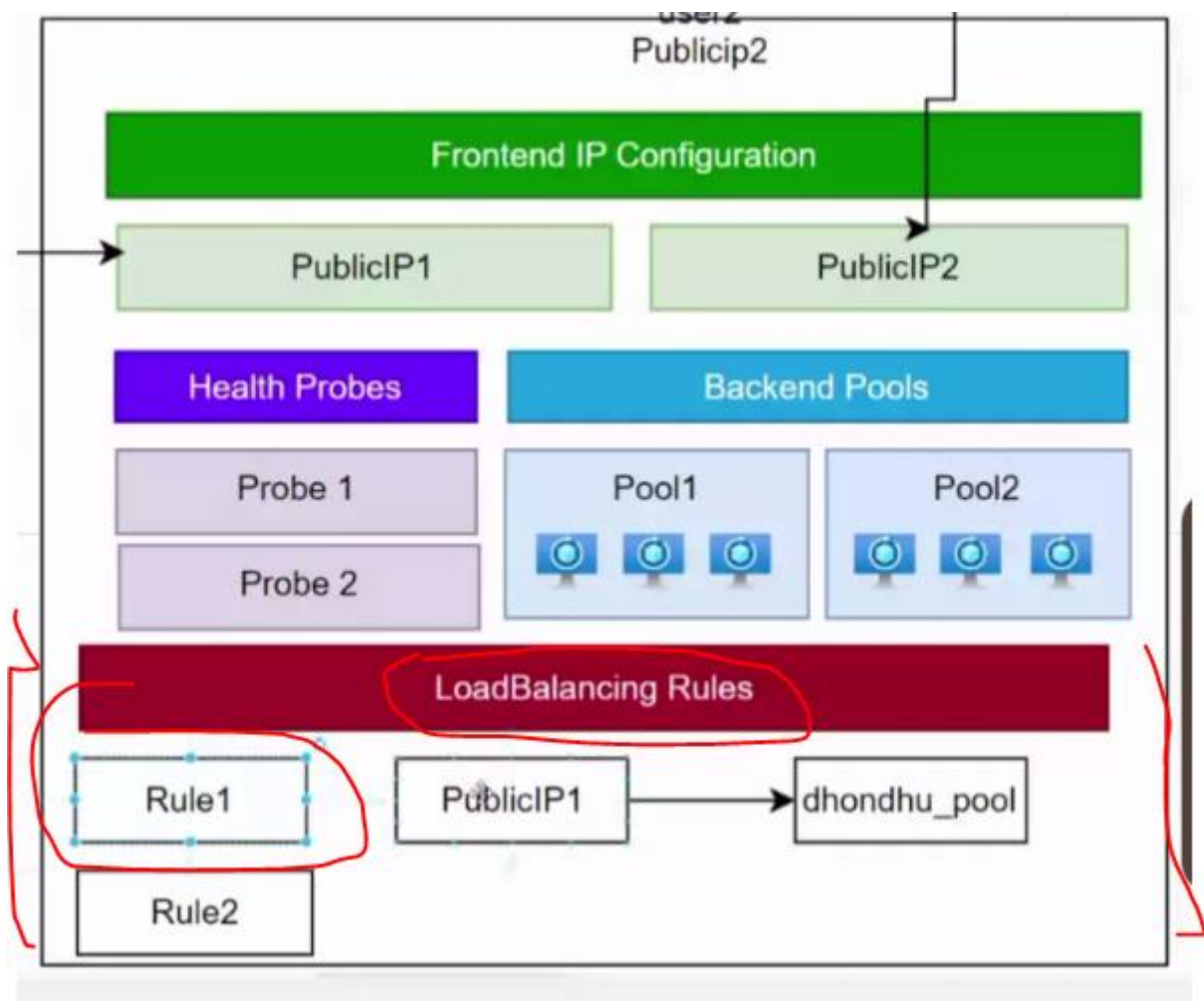
Resource Name	Resource group	Type	IP configuration	IP Address	Availability	Tags
Virtual machine (2)						
<input checked="" type="checkbox"/> vm1	rg_bastion	Virtual ...	ipconfig1	10.0.0.4	-	-
<input checked="" type="checkbox"/> vm2	rg_bastion	Virtual ...	ipconfig1	10.0.0.5	-	-

+ Add | - Remove

Save Cancel Add



6) Load balancing rule – It sets on which backend pool our traffic will go



7) Creating Backend pool / Loadbalancing rule

Microsoft Azure | Upgrade | Search resources, services, and docs (G+/I) | Copilot | innocenty12@snabl... | DEFAULT DIRECTORY (SNABTLG...)

**Load balancing | Load Balancer >**

**Create load balancer**

Basics Frontend IP configuration Backend pools **Inbound rules** Outbound rules Tags Re

**Load balancing rule**

A load balancing rule distributes incoming traffic that is sent to a selected IP address and port combination across a group of backend instances that are eligible to receive traffic.

**+ Add a load balancing rule**

Name ↑↓ Frontend IP configuration ↑↓ Backend pool ↑↓ Health probe ↑↓

Add a rule to get started

**Inbound NAT rule**

An inbound NAT rule forwards incoming traffic sent to a selected IP address and port combination to a specific virtual machine.

**+ Add an inbound nat rule**

**Add load balancing rule**

dhondhu\_lb

A load balancing rule distributes incoming traffic that is sent to a selected IP address and port combination across a group of backend instances that are eligible to receive traffic. Learn more.

Name \* Load Balancing Rule Name

IP Version \* ☒ IPv4 ☐ IPv6

Frontend IP address \* Select an existing

Backend pool \* Select an existing

Protocol ☒ TCP ☐ UDP

Port \*

Save Cancel

Give feedback

an is presenting

Microsoft Azure | Upgrade | Search resources, services, and docs (G+/I) | Copilot | innocenty12@snabl... | DEFAULT DIRECTORY (SNABTLG...)

**Load balancing | Load Balancer >**

**Create load balancer**

Basics Frontend IP configuration Backend pools **Inbound rules** Outbound rules Tags Re

**Load balancing rule**

A load balancing rule distributes incoming traffic that is sent to a selected IP address and port combination across a group of backend instances that are eligible to receive traffic.

**+ Add a load balancing rule**

Name ↑↓ Frontend IP configuration ↑↓ Backend pool ↑↓ Health probe ↑↓

Add a rule to get started

**Inbound NAT rule**

An inbound NAT rule forwards incoming traffic sent to a selected IP address and port combination to a specific virtual machine.

**+ Add an inbound nat rule**

**Add load balancing rule**

dhondhu\_lb

IP Version \* ☒ IPv4 ☐ IPv6

Frontend IP address \* dhondhu\_ip (To be created)

Backend pool \* dhondhu\_pool

Protocol ☒ TCP ☐ UDP

Port \* 80

Backend port \* 80

Health probe \* No existing probe  
Create new

Session persistence ☐ None

Save Cancel

Give feedback

an is presenting

