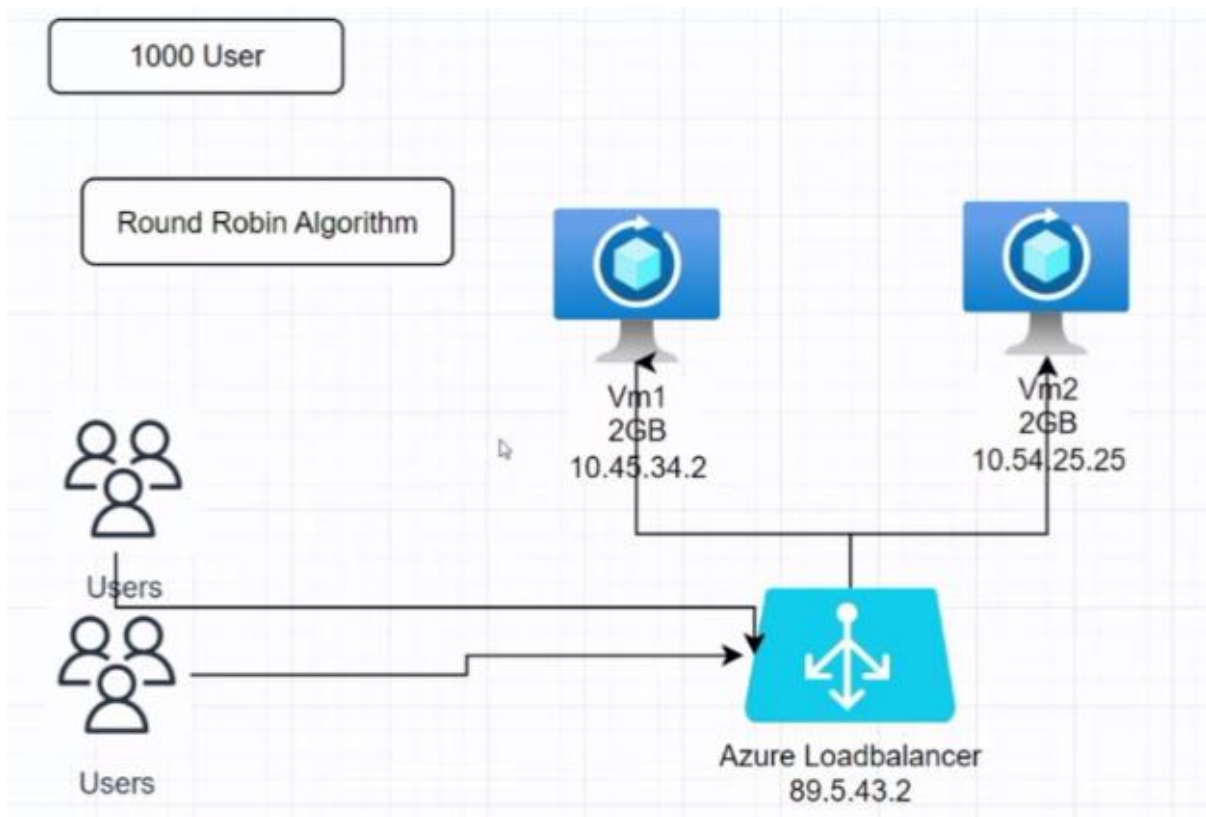
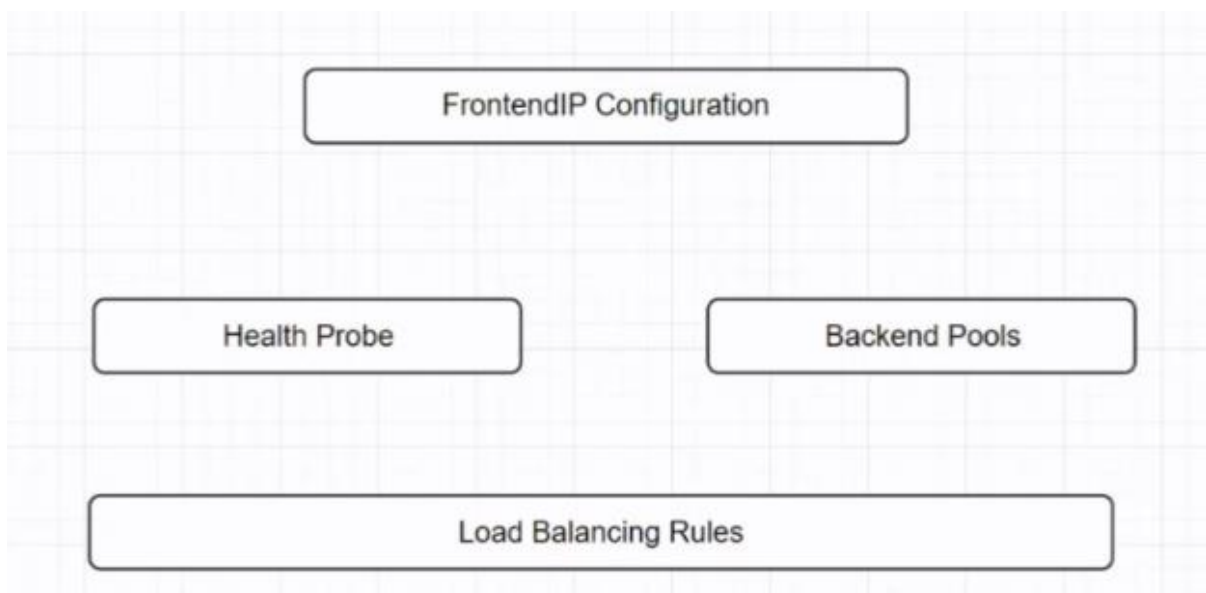


### 33) 31 August 2024 –AzureLoadBalancer

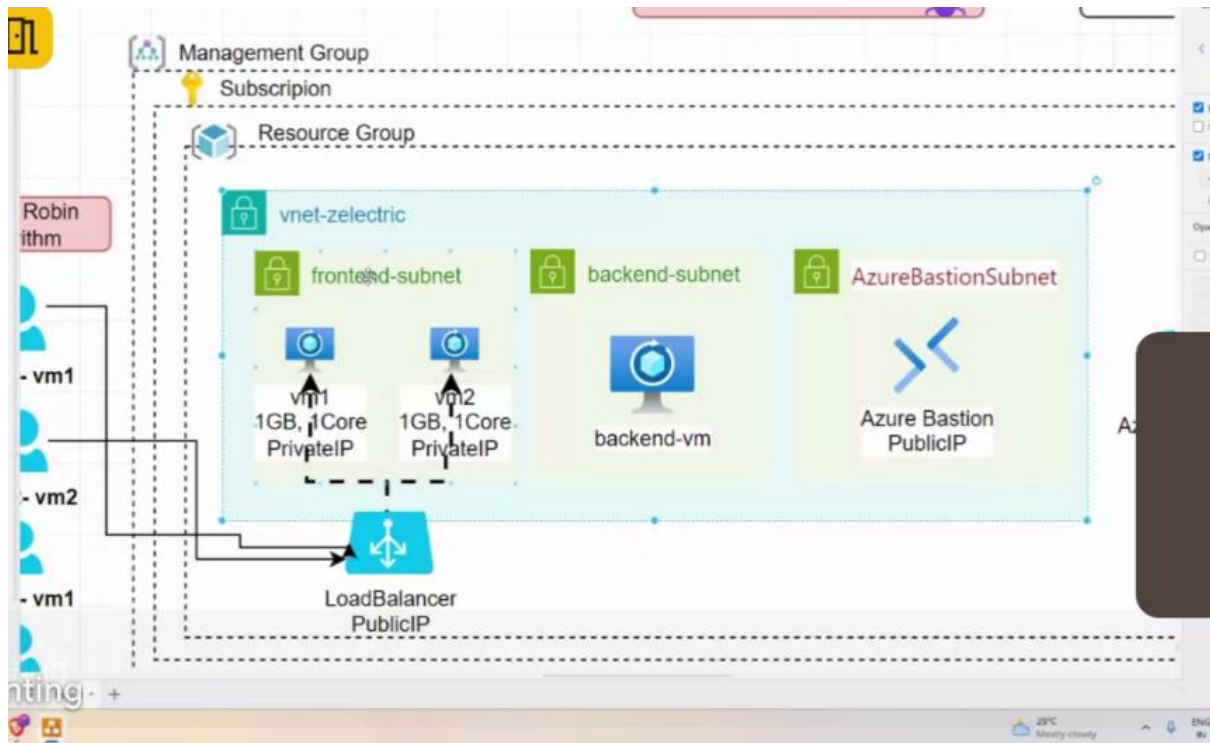
1) What is horizontal and vertical scaling?



2) For configuring loadbalancer, what things are required



3) Prerequisites required to configure load balancer are



- i) 1 Rg
  - ii) 1 vnet
  - iii) 2 subnet – frontend and backend
  - iv) Azure Bastion subnet having azure bastion – public ip
  - v) 2 VMs in frontend subnet – private ips
  - vi) 1LB – public ip
- 4) To configure bastion in code, make subnet 3 in terraform.tfvars file

```

28 }
29
30 #####subnet
31 snet_details = {
32   snet1 = {
33     name           = "subnet_frontend_dev1"
34     resource_group_name = "rgdev1"
35     virtual_network_name = "vnetdev1"
36     address_prefixes   = ["10.0.1.0/24"]
37   }
38
39   snet2 = {
40     name           = "subnet_backend_dev2"
41     resource_group_name = "rgdev1"
42     virtual_network_name = "vnetdev1"
43     address_prefixes   = ["10.0.2.0/24"]
44   }
45   snet3 = {
46     name           = "AzureBastionSubnet"
47     resource_group_name = "rgdev1"
48     virtual_network_name = "vnetdev1"
49     address_prefixes   = ["10.0.3.0/24"]
50   }
51 }

```

5) For now keyvault not required so commit code

```

1  #####datablock for keyvault
2  # data: "azurerm_key_vault", "kv" {
3  #   name = "Keyvaultdev1" #Note - Use same name as we use in manually created kv
4  #   resource_group_name = "rgdev1"
5  # }
6
7
8  #####datablock for keyvaultsecretusername
9  # data: "azurerm_key_vault_secret", "kvsecret_username" {
10 #   name = "vmusername" #Note - Use same name as we use in manually created kv in portal
11 #   key_vault_id = data.azurerm_key_vault.kv.id
12 # }
13
14 #####datablock for keyvaultsecretpassword
15 # data: "azurerm_key_vault_secret", "kvsecret_password" {
16 #   name = "vmpassword" #Note - Use same name as we use in manually created kv in portal
17 #   key_vault_id = data.azurerm_key_vault.kv.id
18 # }

```

6) Now edit username and password

```

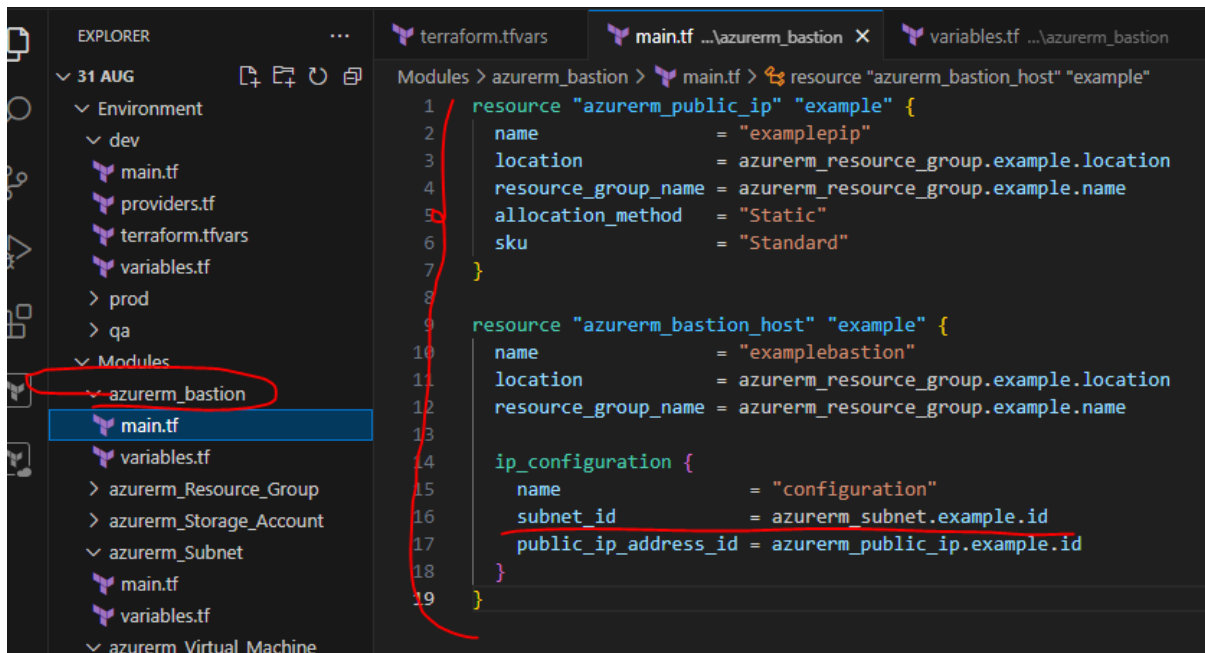
admin_username = data.azurerm_key_vault_secret.kvsecret_username.value
admin_password = data.azurerm_key_vault_secret.kvsecret_password.value

admin_username = "azureuser"
admin_password = "Mommy7Daddy!"
disable_password_authentication = false

```

7) Set vm code in module form

8) Make new folder "azurerm\_bastion"



9) Now for subnet id, we need data block to fetch it. Actually we need subnet id to create bastion

10) SEARCH - azurerm subnet data source

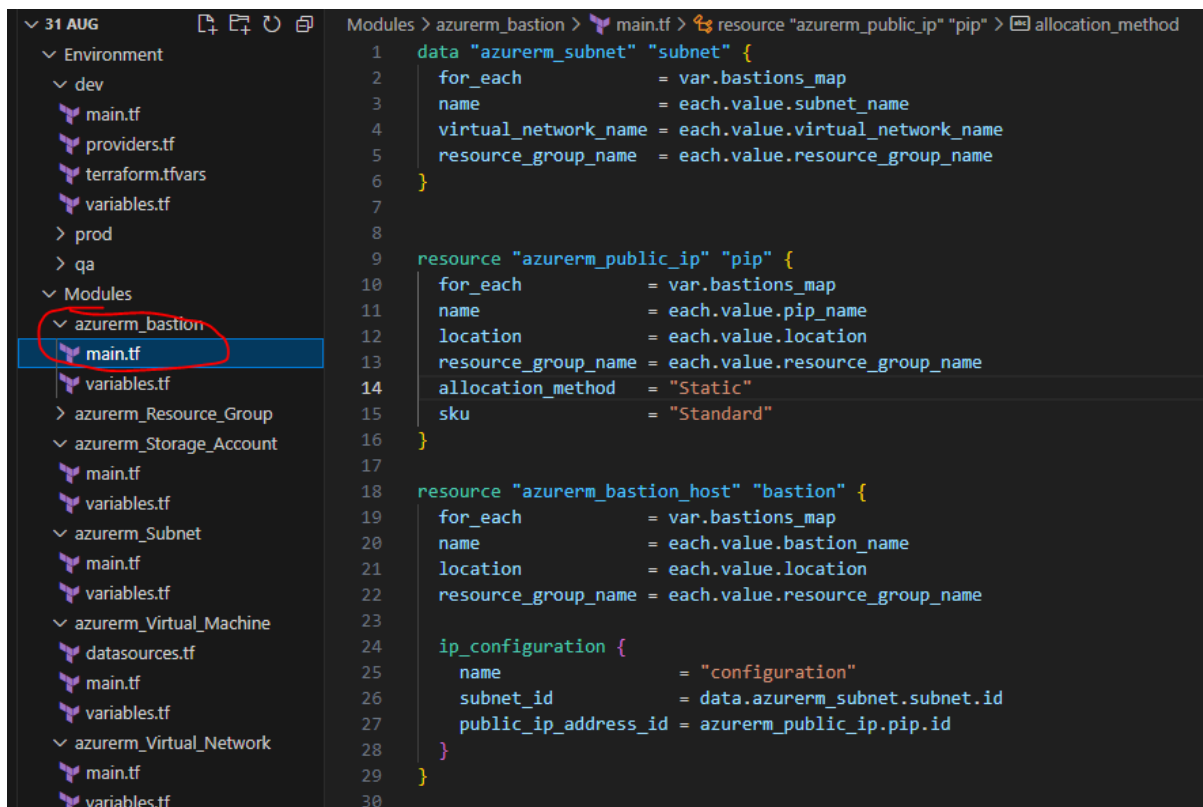
## Data Source: azurerm\_subnet

Use this data source to access information about an existing Subnet in a Virtual Network.

### Example Usage

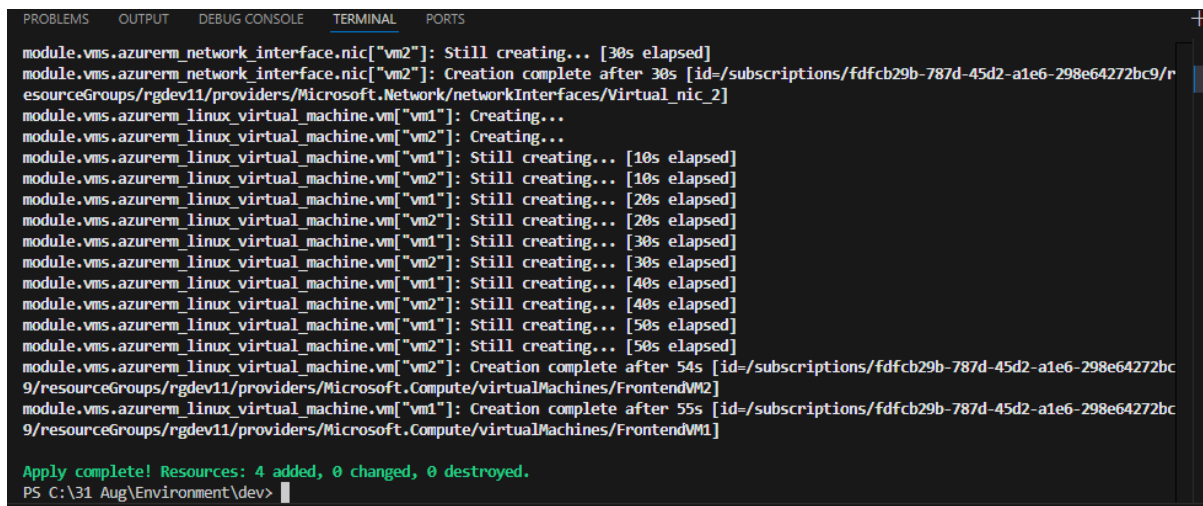
```
data "azurerm_subnet" "example" {
  name           = "backend"
  virtual_network_name = "production"
  resource_group_name = "networking"
```

11) Put data source code of subnet in code



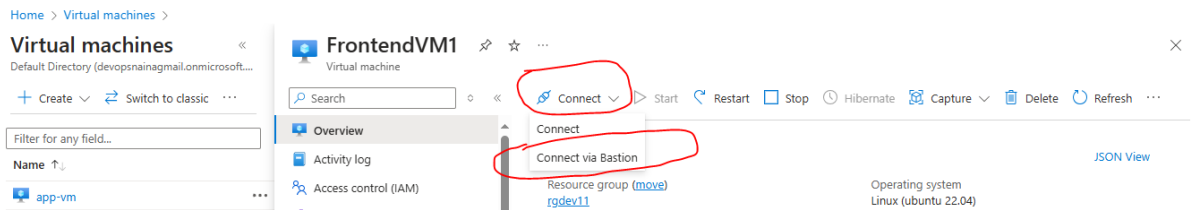
12) Make bastion module and then give values in terraform.tfvars

13) Run terraform init and other commands

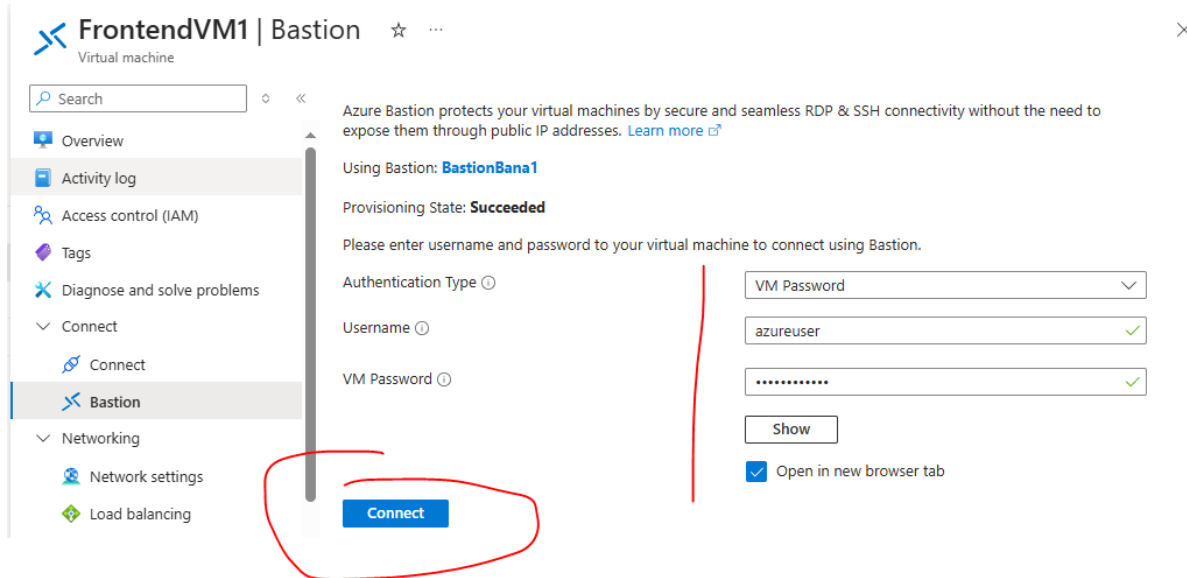


## AGENDA – After creating 2 vms, install nginx on them

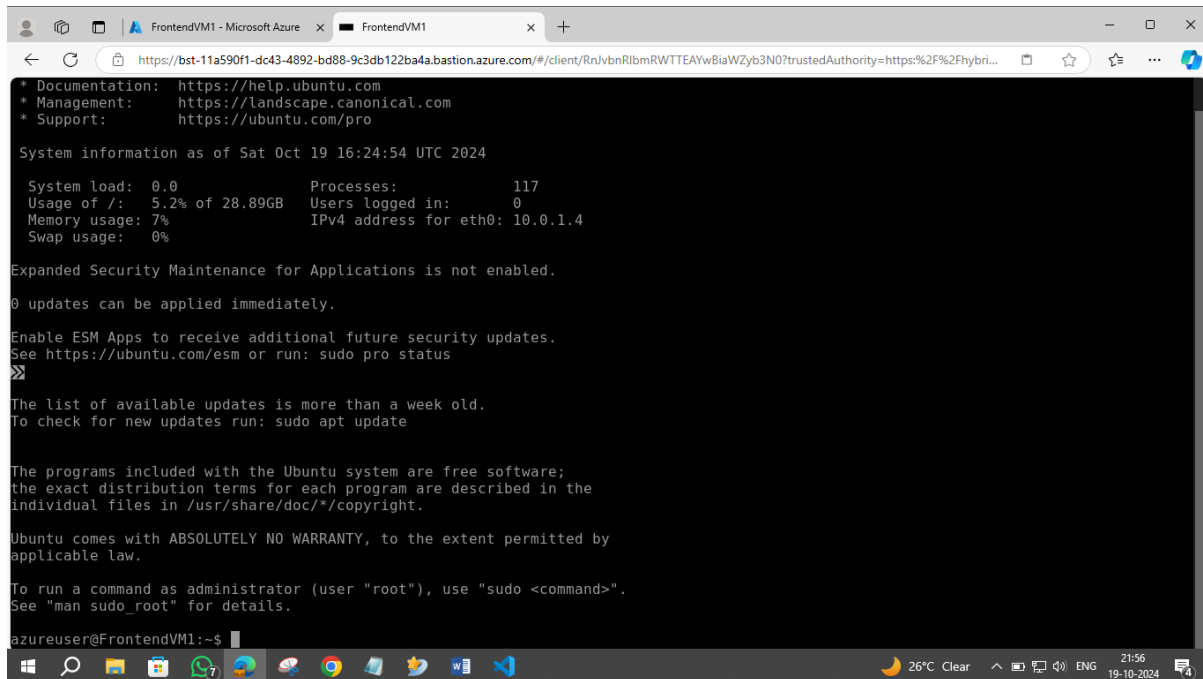
1) In "FrontendVM1", "connect via bastion".



## 2) Put credentials and connect



## 3) Now linux will be opened in new tab of browser



## 4) Now run

**sudo apt update**

**sudo apt install nginx**

## sudo systemctl status nginx

```
azureuser@FrontendVM1:~$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
  Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
  Active: active (running) since Sat 2024-10-19 16:29:26 UTC; 2min 2s ago
    Docs: man:nginx(8)
  Process: 2426 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
  Process: 2427 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
  Main PID: 2522 (nginx)
    Tasks: 3 (limit: 4676)
  Memory: 5.4M
    CPU: 38ms
  CGroup: /system.slice/nginx.service
          └─2522 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
            └─2524 "nginx: worker process"
              └─2525 "nginx: worker process"

Oct 19 16:29:26 FrontendVM1 systemd[1]: Starting A high performance web server and a reverse proxy server...
Oct 19 16:29:26 FrontendVM1 systemd[1]: Started A high performance web server and a reverse proxy server.
azureuser@FrontendVM1:~$
```

5) Similarly, In "FrontendVM2", "connect via bastion". And run nginx

## sudo apt update

## sudo apt install nginx

## sudo systemctl status nginx

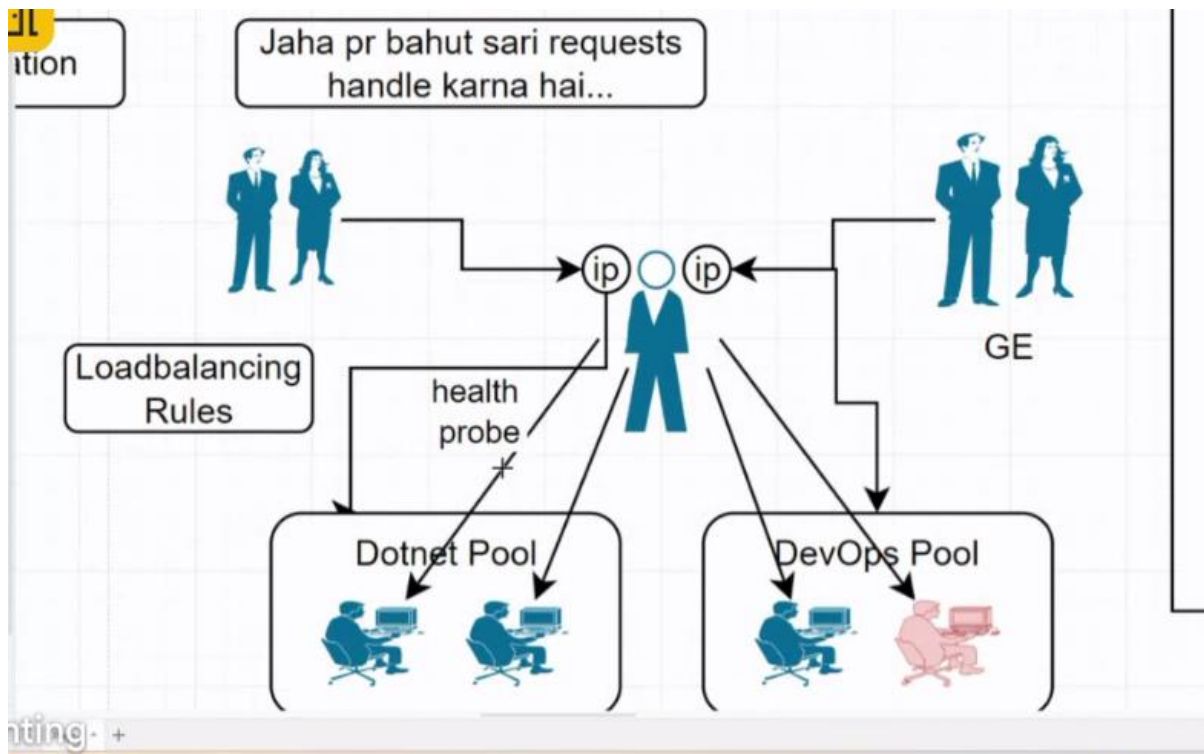
```
● nginx.service - A high performance web server and a reverse proxy server
  Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
  Active: active (running) since Sat 2024-10-19 16:36:42 UTC; 2min 24s ago
    Docs: man:nginx(8)
  Process: 2391 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
  Process: 2392 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
  Main PID: 2487 (nginx)
    Tasks: 3 (limit: 4676)
  Memory: 5.4M
    CPU: 41ms
  CGroup: /system.slice/nginx.service
          └─2487 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
            └─2489 "nginx: worker process"
              └─2490 "nginx: worker process"

Oct 19 16:36:42 FrontendVM2 systemd[1]: Starting A high performance web server and a reverse proxy server...
Oct 19 16:36:42 FrontendVM2 systemd[1]: Started A high performance web server and a reverse proxy server.
azureuser@FrontendVM2:~$
```

+++++

## AGENDA – Now creating Azure load balancer

1) SEARCH – Load balancer in portal



## 1) SEARCH – Loadbalancer in portal

[Home](#) > [Load balancing](#) | [Load balancer](#) >

### Create load balancer ...

Network Address Translation (NAT) to route traffic between public and private IP addresses. [Learn more.](#)

#### Project details

Subscription \*

Free Trial

Resource group \*

rgdev11

[Create new](#)

#### Instance details

Name \*

LB10

Region \*

Germany West Central

SKU \* ⓘ

☒ Standard (Recommended)

☐ Gateway

☐ Basic (Retiring soon)

Type \* ⓘ

☒ Public

☐ Internal

Tier \*

☒ Regional

☐ Global

[Review + create](#)

[< Previous](#)

[Next : Frontend IP configuration >](#)

[Download a template for automation](#) [Give feedback](#)

## 2) Set Frontend IP configuration



Microsoft Azure

Upgrade

Search resources, services, and docs (G+)

Copilot

gaurav.singh@devopsn...  
DEFAULT DIRECTORY (DEVOPSN...

Home > Load balancing | Load Balancer >

Create load balancer ...

BasicsFrontend IP configurationBackend poolsInbound rulesOutbound rulesTagsReview + 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 NAT.

+ Add a frontend IP configuration

| Name ↑↓                          | IP address ↑↓ |
|----------------------------------|---------------|
| Add a frontend IP to get started |               |

Review + create

< Previous

Next : Backend pools >

Download a template for automation

Give feedback

Add frontend IP configuration

LB10

Name \*

IP1

IP version

IPv4

IPv6

IP type

IP address

IP prefix

Public IP address \*

(new) IP1

Create new

Gateway Load balancer ⓘ

None

Save

Cancel

Give feedback

### 3) Create one more ip config

Microsoft Azure

Upgrade

Search resources, services, and docs (G+)

Copilot

gaurav.singh@devopsn...  
DEFAULT DIRECTORY (DEVOPSN...

Home > Load balancing | Load Balancer >

Create load balancer ...

BasicsFrontend IP configurationBackend poolsInbound rulesOutbound rulesTagsReview + 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 NAT.

+ Add a frontend IP configuration

| Name ↑↓ | IP address ↑↓             |
|---------|---------------------------|
| IP1     | (new) IP1 (To be created) |

Review + create

< Previous

Next : Backend pools >

Download a template for automation

Give feedback

Add frontend IP configuration

LB10

Name \*

IP2

IP version

IPv4

IPv6

IP type

IP address

IP prefix

Public IP address \*

(new) IP2

Create new

Gateway Load balancer ⓘ

None

Save

Cancel

Give feedback

Home > Load balancing | Load Balancer >

## Create load balancer ...

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 ↑↓

IP1

(new) IP1 (To be created)

IP2

(new) IP2 (To be created)

Review + create

< Previous

Next : Backend pools >

Download a template for automation Give feedback

https://portal.azure.com/#view/Microsoft\_Azure\_Network/LoadBalancingHubMen...

### 4) Now add backend pools and add corresponding vms

Microsoft Azure Upgrade Search resources, services, and docs (G+) Copilot gaurav.singh@devopsn... DEFAULT DIRECTORY (DEVOPSN...

Home > Load balancing | Load Balancer > Create load balancer

### Add backend pool ...

Name \* Backendpool1

Virtual network vnetdev10 (rgdev10)

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

Filter by name...

Location : germanywestcentral Virtual network : vnetdev10 Add filter

Show resources that are not available for selection

|                                     | Resourc... | Resourc... | Type        | IP confi... | IP Addr... | Availabi... | Tags |
|-------------------------------------|------------|------------|-------------|-------------|------------|-------------|------|
| Virtual machine (2)                 |            |            |             |             |            |             |      |
| <input checked="" type="checkbox"/> | Fronten... | rgdev11    | Virtual ... | nic_ip1     | 10.0.1.4   | -           | 0    |
| <input checked="" type="checkbox"/> | Fronten... | rgdev11    | Virtual ... | nic_ip1     | 10.0.1.5   | -           | 0    |

Save Cancel Give feedback

Add Cancel

Home > Load balancing | Load Balancer > Create load balancer >

## Add backend pool

Name \*

Virtual network

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 |
|---------------|----------------|-----------------|------------------|------------|
| FrontendVM1   | rgdev11        | Virtual machine | nic_ip1          | 10.0.1.4   |
| FrontendVM2   | rgdev11        | Virtual machine | nic_ip1          | 10.0.1.5   |

[Give feedback](#)

5) Now every load balancer can have multiple ip configurations and Backend pools and under ip configurations their corresponding vm will be added

Home > Load balancing | Load Balancer >

## Create load balancer

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

A backend pool is a collection of resources to which your load balancer can send traffic. A backend pool can contain virtual machines, virtual machine scale sets, and containers.

+ Add a backend pool

| Name         | Virtual network | Resource Name | Network interface | IP address | Availability zone | Admin state |
|--------------|-----------------|---------------|-------------------|------------|-------------------|-------------|
| Backendpool1 | vnetdev10       | FrontendVM1   | virtual_nic_1     | 10.0.1.4   | -                 | None        |
| Backendpool1 | vnetdev10       | FrontendVM2   | virtual_nic_2     | 10.0.1.5   | -                 | None        |

6) Now add load balancing rule

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

Home > Load balancing | Load Balancer >

## Create load balancer

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

### Add 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 pool instances. The load balancing rule can be configured to use either a fixed IP address or a floating IP address.

+ 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

| Name                      | Frontend IP configuration | Service | Target |
|---------------------------|---------------------------|---------|--------|
| Add a rule to get started |                           |         |        |

[Download a template for automation](#) [Give feedback](#)

### Add load balancing rule

Frontend IP address \*

Backend pool \*

Protocol

☒ TCP

☐ UDP

Port \*

Backend port \*

Health probe \*

[Create new](#)

Session persistence

Idle timeout (minutes) \*

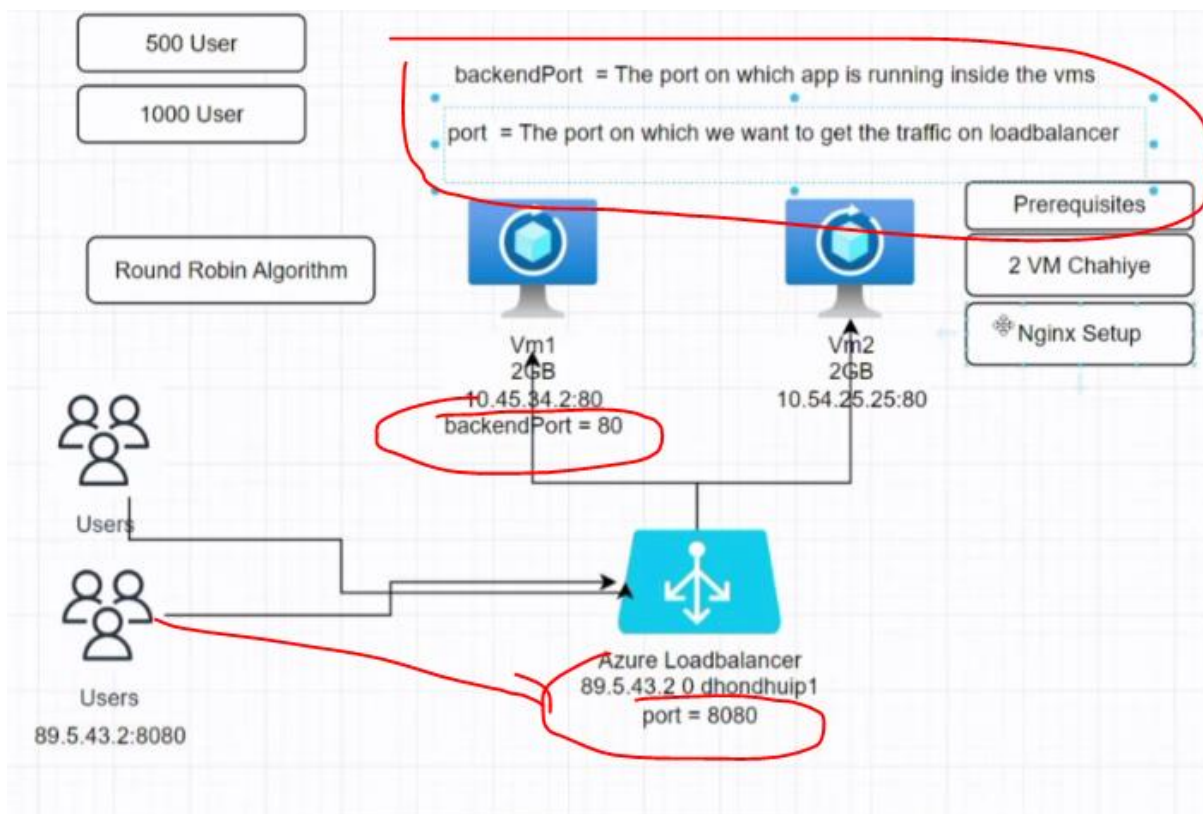
Enable TCP Reset ☐

Enable Floating IP ☐

Outbound source network address ☒ (Recommended) Use outbound rules to

[Give feedback](#)

7) Now users will hit on ip of LB

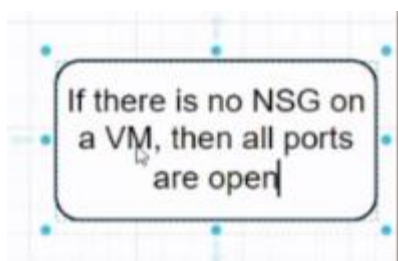


8) Now go to LB

The screenshot shows the 'Frontend IP configuration' page for 'LB10' in the Azure portal. The page title is 'LB10 | Frontend IP configuration'. A sidebar on the left contains navigation links: Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings, Frontend IP configuration (selected), Backend pools, Health probes, and Load balancing rules. The main content area explains that the frontend IP address configuration serves as the entry point for incoming traffic. Below this, a table lists the IP addresses and the number of rules associated with each.

| Name | IP address           | Rules count |
|------|----------------------|-------------|
| IP1  | 72.144.232.175 (IP1) | 1           |
| IP2  | 72.144.232.198 (IP2) | 0           |

9) Now we have to open backend port 80 on both vms



10) To add nsg click below in network settings of both vms

FrontendVM1 | Network settings

Virtual machine

Search

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Connect

Networking

Network settings

Load balancing

Application security groups

Network manager

Settings

Availability + scale

Security

Backup + disaster recovery

Operations

This is a new experience. [Please provide feedback](#)

vnetdev10 / subnet\_frontend\_dev10

0 (Configure)

Public IP address

Network security group

- (Configure)

-

Private IP address

Accelerated networking

10.0.1.4


Disabled

Admin security rules

Effective security rules

0 (Configure)

0




No port rules to display

Network security group is an essential component of Azure's network security architecture, providing inbound and outbound traffic filtering for network resources. Customers can connect to their virtual machines securely by adding the appropriate port rules, allowing granular control over ~~inbound traffic~~ and ensuring authorized access to specific ports and protocols.

Add network security group

11) open port 80 on both vms

 **Add inbound security rule** ×

Virtual\_nic\_2-nsg

Source ⓘ

Any

Source port ranges \* ⓘ

\*

Destination ⓘ

Any

Service ⓘ

HTTP

Destination port ranges ⓘ

80

Protocol

☐ Any

☒ TCP

☐ UDP

☐ ICMPv4


Action

☒ Allow

☐ Deny

Add

Cancel

 Give feedback

12) Now open ssh port 22 also on both vms

## Add inbound security rule

Virtual\_nic\_2-nsg

Source ⓘ  
Any

Source port ranges \* ⓘ  
\*

Destination ⓘ  
Any

Service ⓘ  
SSH

Destination port ranges ⓘ  
22

Protocol  
☐ Any  
☒ TCP  
☐ UDP  
☐ ICMPv4

Action  
☒ Allow  
☐ Deny

**Add** Cancel

[Give feedback](#)

### Virtual machines

Default Directory (devopsnain@gmail.onmicrosoft.com)

+ Create - Switch to classic

Filter for any field...

Name ↑

- app-vm
- FrontendVM1
- FrontendVM2
- web-vm

### FrontendVM1 | Network settings

Virtual machine

Search

- Overview
- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems
- Connect
- Networking
  - Network settings**
  - Load balancing
  - Application security groups
  - Network manager
- Settings
- Availability + scale

### Network security group Virtual\_nic\_1-nsg

(attached to networkInterface: Virtual\_nic\_1)  
Impacts 0 subnets, 1 network interfaces

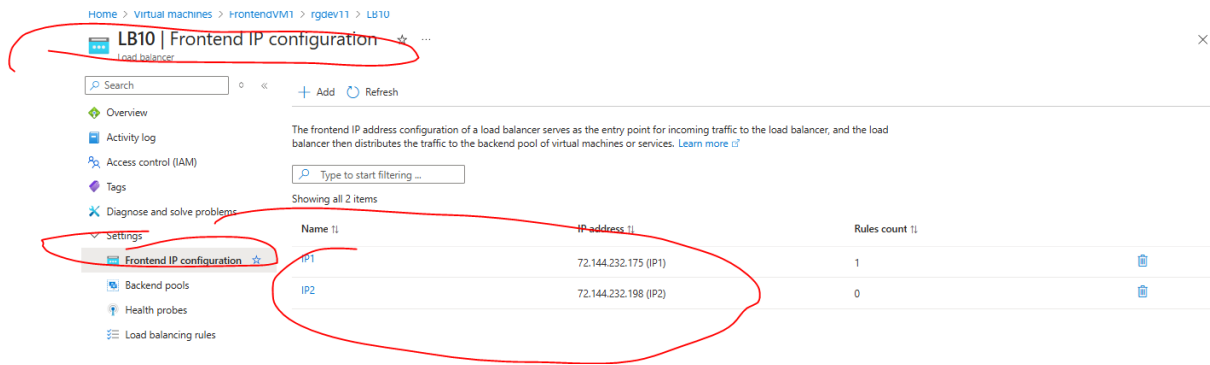
+ Create port rule

Search rules

Source == all Destination == all Protocol == all Action == all

| Priority ↑              | Name                            | Port | Protocol | Source            | Destination    |
|-------------------------|---------------------------------|------|----------|-------------------|----------------|
| Inbound port rules (5)  |                                 |      |          |                   |                |
| 100                     | AllowAnyHTTPInbound             | 80   | TCP      | Any               | Any            |
| 110                     | AllowAnySSHInbound              | 22   | TCP      | Any               | Any            |
| 65000                   | AllowVnetInbound ⓘ              | Any  | Any      | VirtualNetwork    | VirtualNetwork |
| 65001                   | AllowAzureLoadBalancerInBound ⓘ | Any  | Any      | AzureLoadBalancer | Any            |
| 65500                   | DenyAllInBound ⓘ                | Any  | Any      | Any               | Any            |
| Outbound port rules (3) |                                 |      |          |                   |                |

13) Go to LB



14) Now run in browser ip of FrontendVM1 and port 9090 - 72.144.232.175:9090

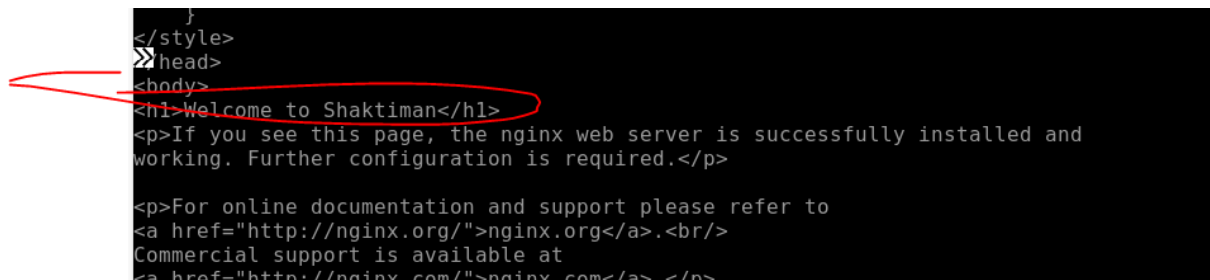
15) Now run in frontend vm linux machine

```
cd /var/www/html
```

```
ls
```

```
index.nginx-debian.html
```

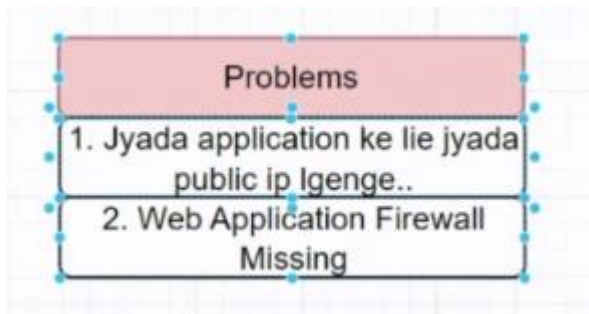
```
sudo nano index.nginx-debian.html *
```



17) Similarly run “Welcome to Rajnikant” for FrontendVM2 by doing ssh through bastion

18) Now problems with Loadbalancer are





iii) Also its layer 4

iv) Only regional applicable

19) Loadbalancer runs on layer 4 that has no SSL termination feature.

20) So, to resolve issue of loadbalancer we will use application gateway. But this is also regional service so to resolve it we will use front door which is globally available but it has also issue of DNS. So to resolve that traffic manager will be used which has both DNS as well as globally available feature.