Load Balancer

Objective

To deploy and verify both an External Load Balancer (ELB) and an Internal Load Balancer (ILB) using the Azure Portal GUI. The setup includes two backend VMs hosting a simple web server (`nginx`) and demonstrates how to use Azure's load balancing services through the portal.

Components Used

- **Resource Group:** A logical container that groups related Azure resources like VMs, VNets, and load balancers. It enables unified management, monitoring, access control, and billing. Resources in a group can be managed collectively and can be deployed or deleted together.

- **Virtual Network (VNet):** A foundational building block in Azure networking. It provides an isolated and secure environment where you can launch Azure resources. VNets can span multiple subnets and support IP address assignment, traffic routing, and security controls.

- **Subnet:** A segment of a virtual network that logically separates resources within a VNet. Subnets help implement micro-segmentation, manage routing policies, and apply fine-grained Network Security Groups (NSGs) to control traffic flow between tiers or services.

- **Virtual Machines (VMs):** Compute resources used as backend servers in this setup. Each VM is an isolated instance of an operating system that runs applications—in this case, a basic web server like NGINX. VMs are used to simulate real-world backend workloads.

- **Load Balancer (Standard SKU):** A Layer 4 (TCP/UDP) load balancer that distributes incoming network traffic across healthy backend VMs to ensure reliability and scalability.
    - **External Load Balancer (ELB)** uses a static public IP to distribute traffic from the internet.
    - **Internal Load Balancer (ILB)** uses a private IP to manage traffic within the VNet between services.

- **Backend Pool:** A group of network interfaces (NICs) from VMs that the load balancer routes traffic to. Each NIC in the pool must be associated with the same VNet as the load balancer and is continuously health-checked for availability.

- **Health Probe:** Used by the load balancer to determine the health status of VMs in the backend pool. It periodically checks a specified port and protocol (HTTP, TCP) to ensure the VM is responsive. If a probe fails, the VM is temporarily removed from the pool.

- **Load Balancing Rule:** Defines how and where the traffic is directed. It connects the frontend IP and port (e.g., port 80) to the backend pool and port, with an associated health probe to monitor availability. Rules can specify session persistence and idle timeouts for optimized traffic flow.

Implementation on Azure

1. Create Resource Group

- Navigate to **Azure Portal** → **Resource groups** → **+ Create**
- Name: *LoadBalancerRG*
- Region: East US (or any supported region)
- Click **Review + Create**, then **Create**

2. Create Virtual Network

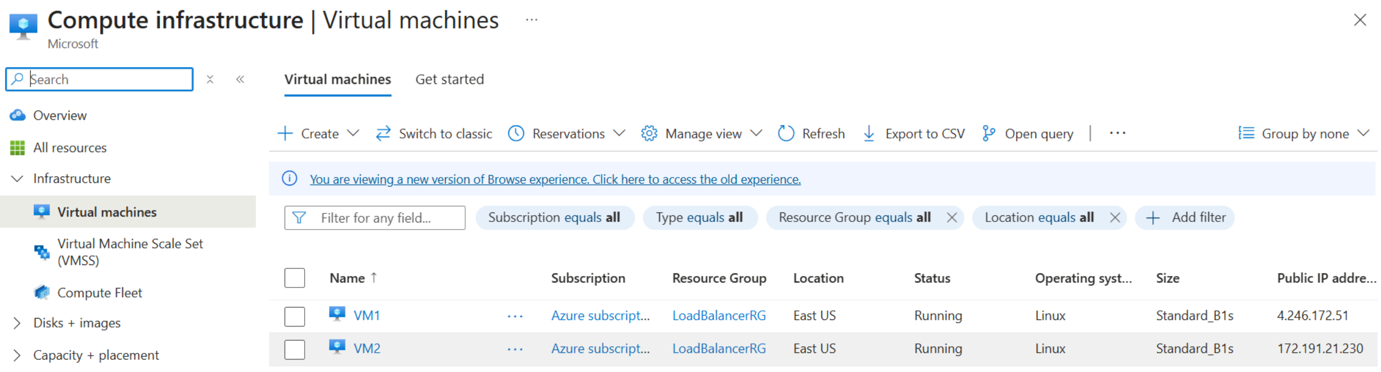
- Go to **Virtual Networks** → **+ Create**
- Name: *LBVNet*
- Address space: Default (10.0.0.0/16)
- Subnet:
- Name: *LBSubnet*
- Address range: 10.0.0.0/24
- Resource Group: *LoadBalancerRG*
- Click **Review + Create**, then **Create**

3. Create Two Virtual Machines (Repeat Steps for Each)

- Go to **Virtual Machines** → **+ Create**
- Name: *VM1* (Repeat as *VM2* for second)
- Image: **Ubuntu LTS**
- Size: B1s or B2s
- Authentication: Password or SSH
- Select VNet: *LBVNet*, Subnet: *LBSubnet*
- Public IP: Yes (needed for SSH; optional otherwise)
- NIC NSG: Create new or allow HTTP (port 80) and SSH (port 22)
- Click **Review + Create**, then **Create**

Install Web Server (SSH into both VMs)

```
sudo apt update && sudo apt install -y nginx
echo "This is VM1" | sudo tee /var/www/html/index.html  # Use VM2 on second
```



You are viewing a new version of Browse experience. Click here to access the old experience.

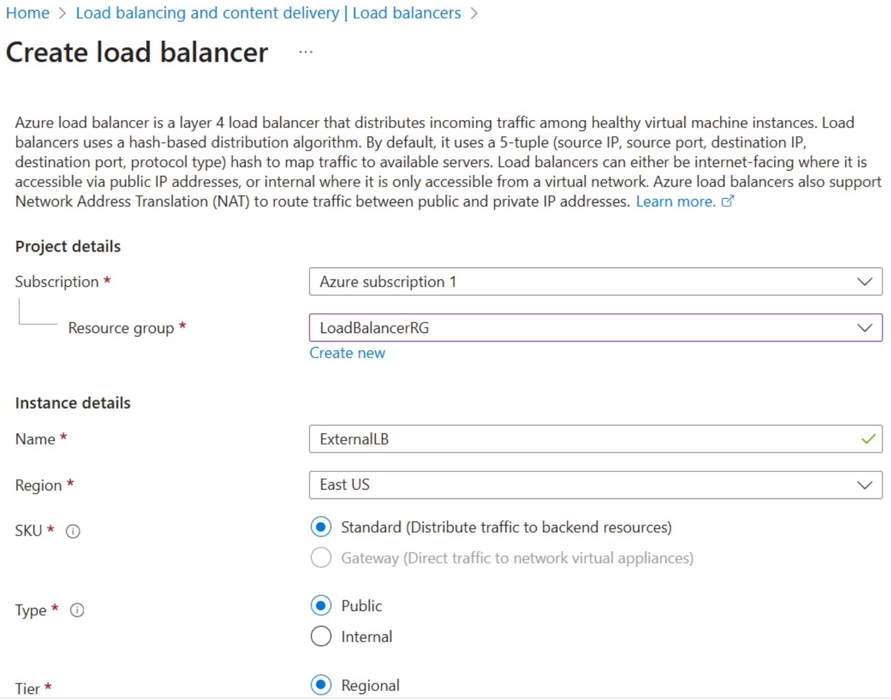
Subscription equals **all** | Type equals **all** | Resource Group equals **all** | Location equals **all**

Name ↑	Subscription	Resource Group	Location	Status	Operating syst...	Size	Public IP addre...
VM1	Azure subscript...	LoadBalancerRG	East US	Running	Linux	Standard_B1s	4.246.172.51
VM2	Azure subscript...	LoadBalancerRG	East US	Running	Linux	Standard_B1s	172.191.21.230

4. Create External Load Balancer (ELB)

- **Step 1: Create Public IP**
  - Go to **Public IP addresses → + Create**
  - Name: *PublicLBIP*
  - SKU: **Standard**
  - Assignment: Static
  - Resource Group: *LoadBalancerRG*
  - Click **Create**

- **Step 2: Create Load Balancer**
  - Go to **Load Balancers → + Create**
  - Name: *ExternalLB*
  - SKU: Standard
  - Type: **Public**
  - Frontend IP: Select *PublicLBIP*
  - VNet: *LBVNet*
  - Resource Group: *LoadBalancerRG*
  - Click **Review + Create**, then **Create**



Home > Load balancing and content delivery | Load balancers >

Create load balancer ...

Azure load balancer is a layer 4 load balancer that distributes incoming traffic among healthy virtual machine instances. Load balancers uses a hash-based distribution algorithm. By default, it uses a 5-tuple (source IP, source port, destination IP, destination port, protocol type) hash to map traffic to available servers. Load balancers can either be internet-facing where it is accessible via public IP addresses, or internal where it is only accessible from a virtual network. Azure load balancers also support Network Address Translation (NAT) to route traffic between public and private IP addresses. Learn more.

Project details

Subscription * : Azure subscription 1

Resource group * : LoadBalancerRG

Create new

Instance details

Name * : ExternalLB ✓

Region * : East US

SKU * ⓘ

● Standard (Distribute traffic to backend resources)

○ Gateway (Direct traffic to network virtual appliances)

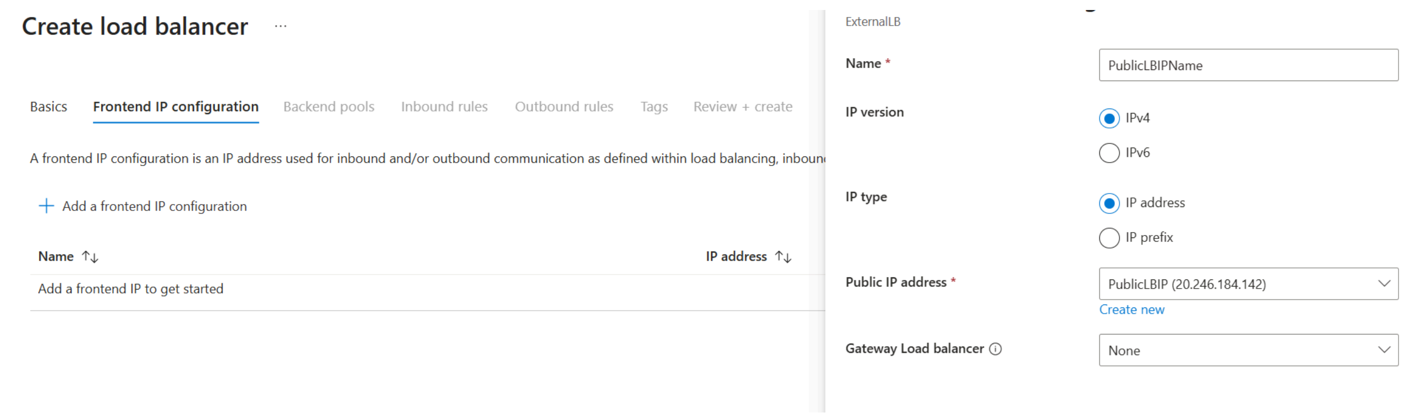
Type * ⓘ

● Public

○ Internal

Tier *

● Regional



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, inboun

+ Add a frontend IP configuration

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

ExternalLB

Name * : PublicLBIPName

IP version

● IPv4

○ IPv6

IP type

● IP address

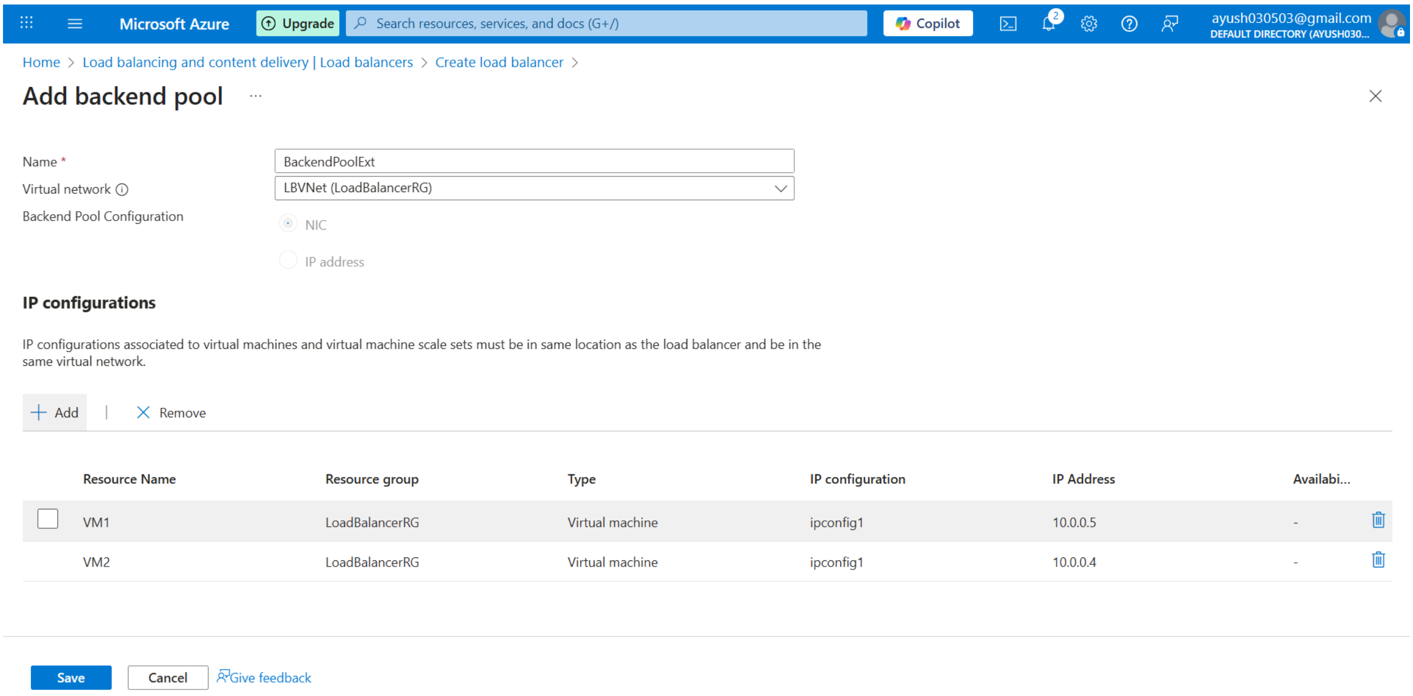
○ IP prefix

Public IP address * : PublicLBIP (20.246.184.142)

Create new

Gateway Load balancer ⓘ : None

- **Step 3: Add Backend Pool**
    - Navigate to *ExternalLB*
    - Select **Backend pools → + Add**
    - Name: *BackendPoolExt*
    - VNet: *LBVNet*
    - Add NICs of *VM1* and *VM2*
    - Save


Microsoft Azure
Upgrade
Search resources, services, and docs (G+/)
Copilot
ayush030503@gmail.com
DEFAULT DIRECTORY (AYUSH030...

Home > Load balancing and content delivery | Load balancers > Create load balancer >

Add backend pool

Name *: BackendPoolExt

Virtual network ⓘ: LBVNet (LoadBalancerRG)

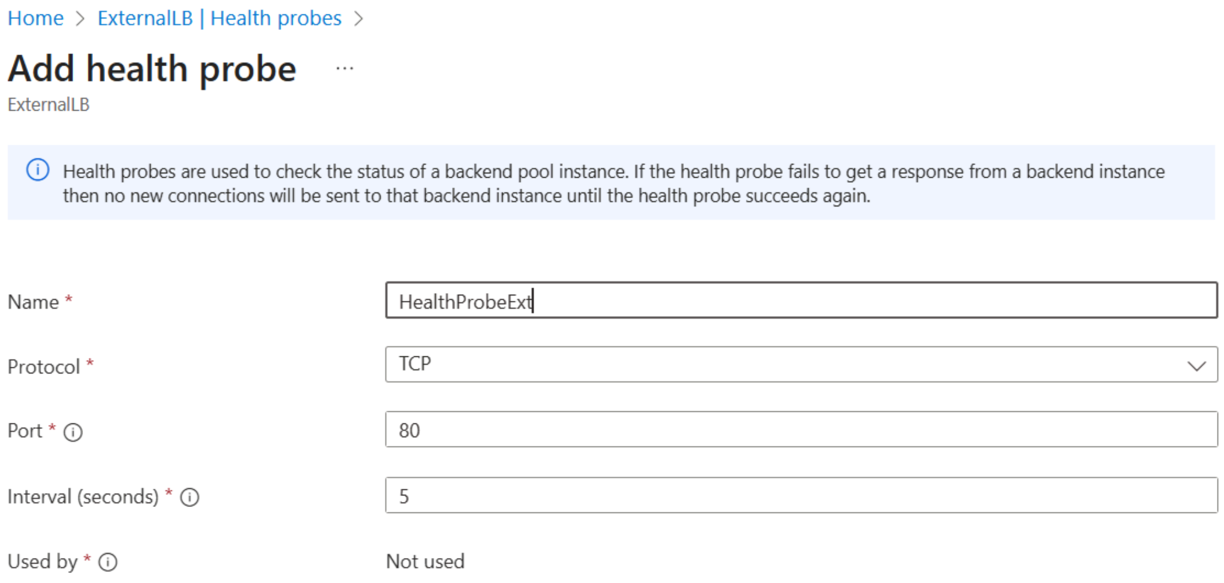
Backend Pool Configuration: ● NIC ○ IP address

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.

Resource Name	Resource group	Type	IP configuration	IP Address	Availabi...
VM1	LoadBalancerRG	Virtual machine	ipconfig1	10.0.0.5	-
VM2	LoadBalancerRG	Virtual machine	ipconfig1	10.0.0.4	-

- **Step 4: Add Health Probe**
    - Go to **Health Probes → + Add**
    - Name: *HealthProbeExt*
    - Protocol: TCP
    - Port: 80
    - Interval & threshold: default
    - Save



Home > ExternalLB | Health probes >

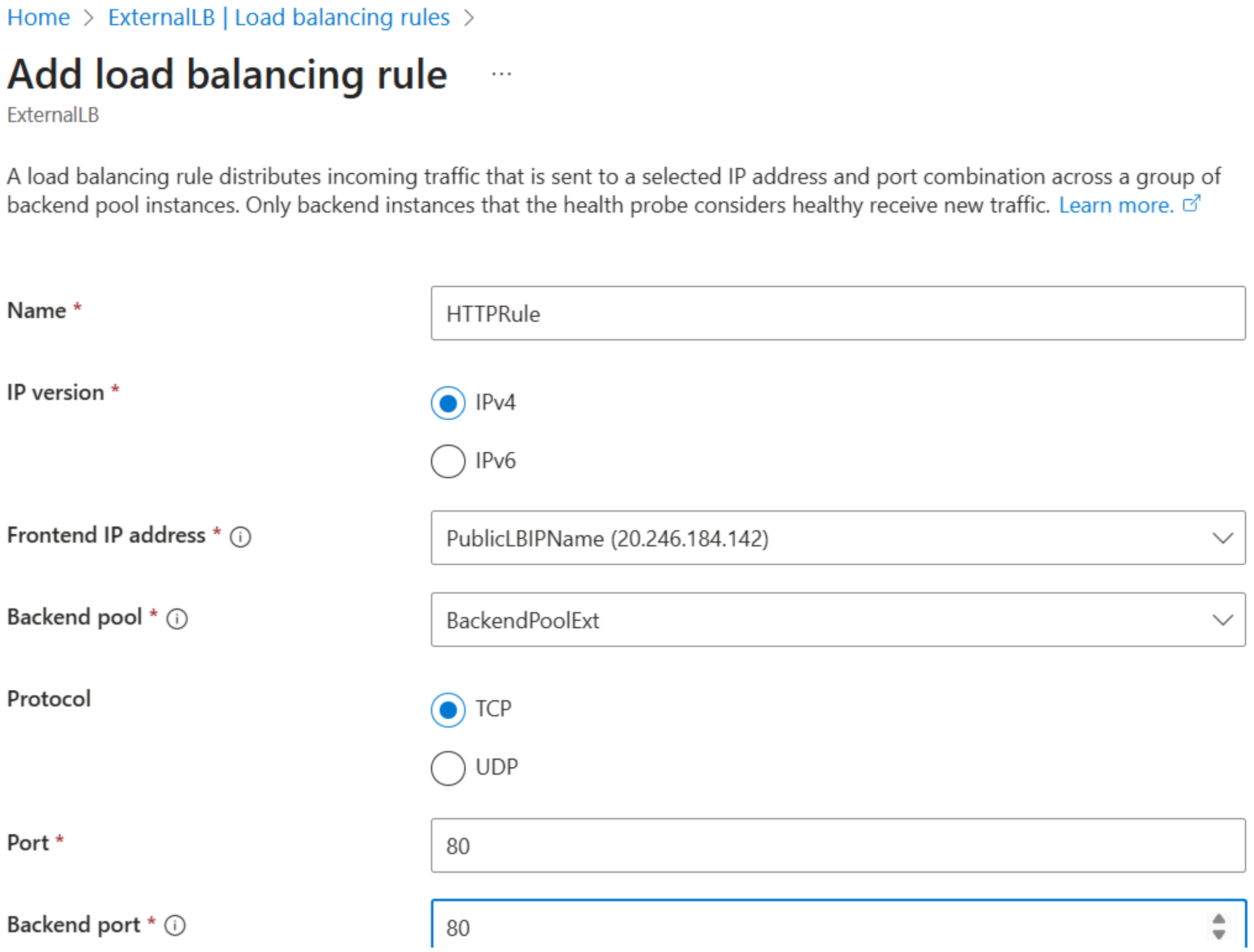
Add health probe

ExternalLB

ⓘ Health probes are used to check the status of a backend pool instance. If the health probe fails to get a response from a backend instance then no new connections will be sent to that backend instance until the health probe succeeds again.

Name *	HealthProbeExt
Protocol *	TCP
Port * ⓘ	80
Interval (seconds) * ⓘ	5
Used by * ⓘ	Not used

- **Step 5: Add Load Balancing Rule**
  - Go to **Load balancing rules → + Add**
  - Name: *HTTPRule*
  - Frontend IP: *PublicFrontend*
  - Port: 80 (Frontend and Backend)
  - Backend pool: *BackendPoolExt*
  - Health probe: *HealthProbeExt*
  - Save



Home > ExternalLB | Load balancing rules >

Add load balancing rule ...

ExternalLB

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. Only backend instances that the health probe considers healthy receive new traffic. Learn more.

Name *	HTTPRule
IP version *	● IPv4 ○ IPv6
Frontend IP address * ⓘ	PublicLBIPName (20.246.184.142)
Backend pool * ⓘ	BackendPoolExt
Protocol	● TCP ○ UDP
Port *	80
Backend port * ⓘ	80

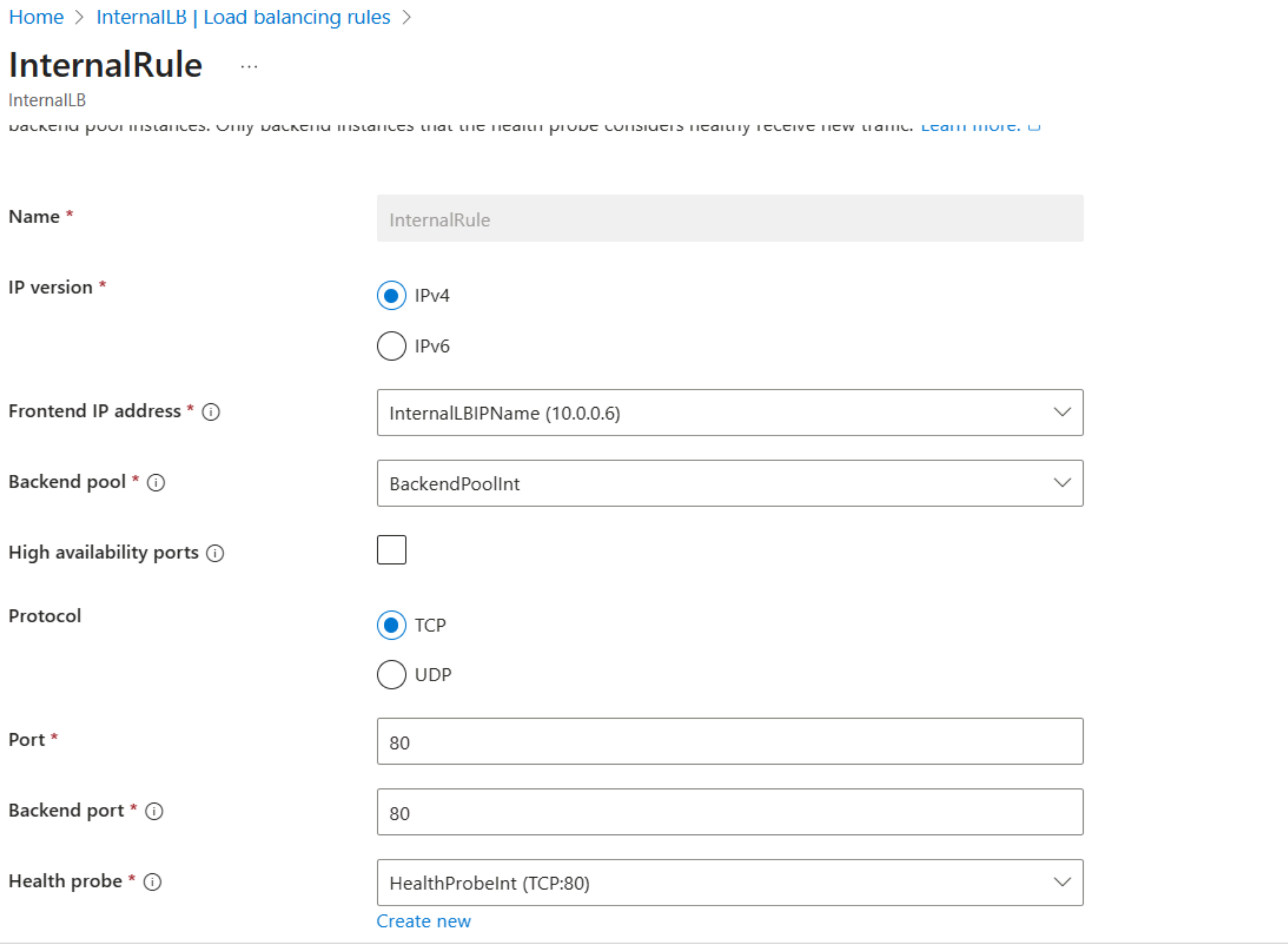
Create Internal Load Balancer (ILB)

- **Step 1: Create Load Balancer**
  - Go to **Load Balancers → + Create**
  - Name: *InternalLB*
  - SKU: Standard
  - Type: **Internal**
  - VNet: *LBVNet*, Subnet: *LBSubnet*
  - Private IP: Static (e.g., *10.0.0.100*)
  - Resource Group: *LoadBalancerRG*
  - Click **Review + Create**, then **Create**

- **Step 2: Add Backend Pool**
  - Same as ELB. Use the same VMs (*VM1*, *VM2*).

- **Step 3: Add Health Probe**
  - Name: *HealthProbeInt*
  - Protocol: TCP
  - Port: 80
  - Save

- **Step 4: Add Load Balancing Rule**
  - Name: *ILBRule*
  - Frontend Port: 80, Backend Port: 80
  - Frontend IP: Internal Frontend
  - Backend Pool: *BackendPoolInt*
  - Health Probe: *HealthProbeInt*
  - Save



Home > InternalLB | Load balancing rules >

InternalRule ···

InternalLB

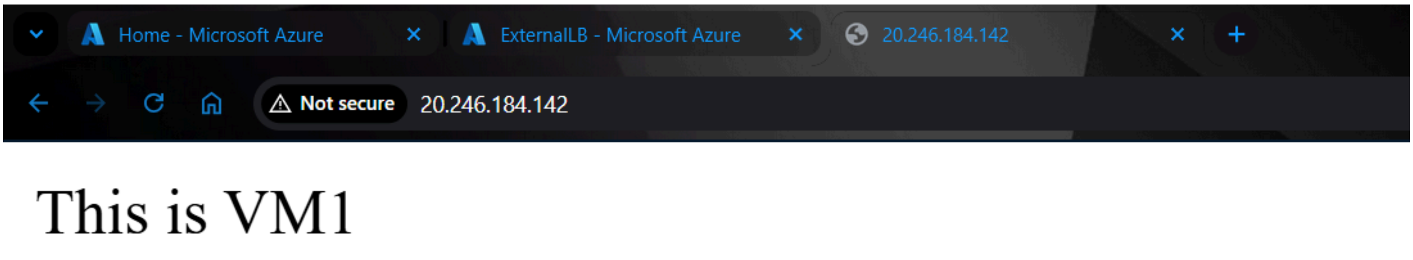
backend pool instances. Only backend instances that the health probe considers healthy receive new traffic. Learn more.

Name *	InternalRule
IP version *	● IPv4 ○ IPv6
Frontend IP address * ⓘ	InternalLBIPName (10.0.0.6)
Backend pool * ⓘ	BackendPoolInt
High availability ports ⓘ	□
Protocol	● TCP ○ UDP
Port *	80
Backend port * ⓘ	80
Health probe * ⓘ	HealthProbeInt (TCP:80) Create new

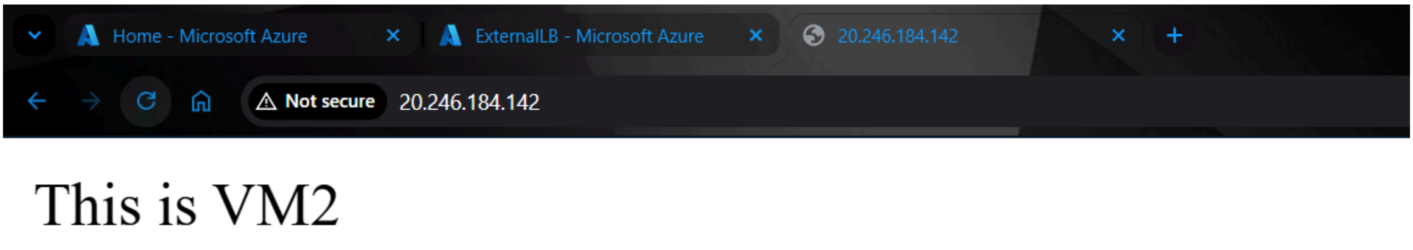
Verification

1. External Load Balancer

- Go to **PublicLBIP** and copy its IP
- Open in browser
- Refresh several times → Should alternate responses from *VM1* and *VM2*



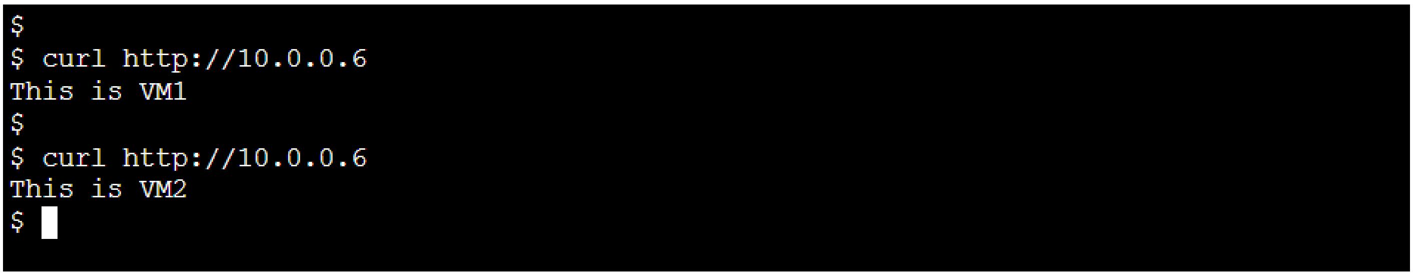
This is VM1



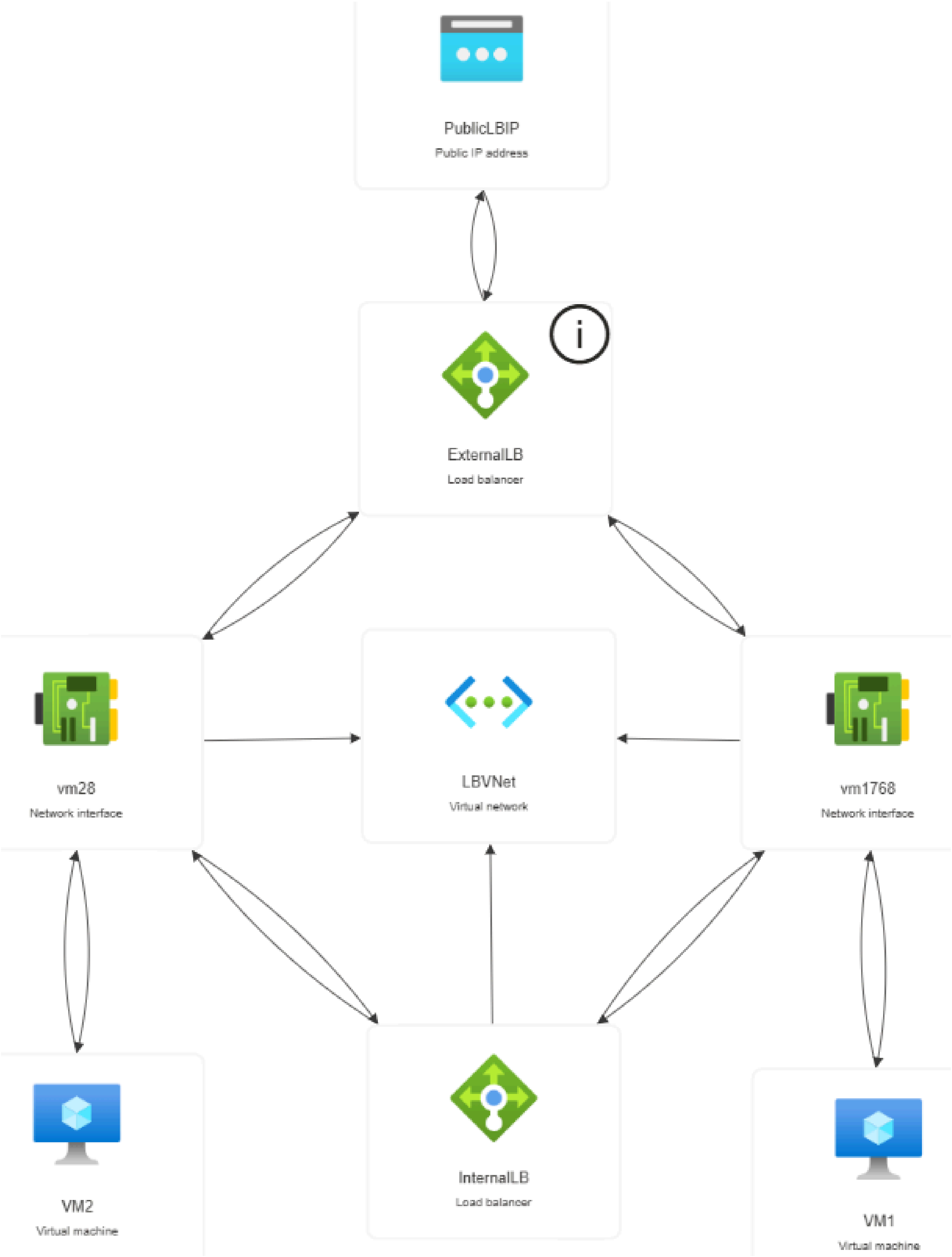
This is VM2

2. Internal Load Balancer

- SSH into any VM
- Run: *curl* [*http://10.0.0.6*](http://10.0.0.6) (private IP of Internal Load Balancer)
- Should return either "This is VM1" or "This is VM2"



```
$
$ curl http://10.0.0.6
This is VM1
$
$ curl http://10.0.0.6
This is VM2
$
```


PublicLBIP
Public IP address
ExternalLB
Load balancer
vm28
Network interface
LBVNet
Virtual network
vm1768
Network interface
VM2
Virtual machine
InternalLB
Load balancer
VM1
Virtual machine

Bash Script to perform the entire task via Azure CLI

```
#!/bin/bash
# Azure Load Balancer Full Setup Script (ELB + ILB)

# ---------- CONFIG ----------
RG="LoadBalancerRG"
LOC="eastus"
VNET="LBVNet"
SUBNET="LBSubnet"
VM_SIZE="Standard_B1s"
USERNAME="azureuser"
IMAGE="UbuntuLTS"
ILB_PRIVATE_IP="10.0.0.100"


# ---------- CREATE RG & NETWORK ----------
az group create --name $RG --location $LOC

az network vnet create \
  --resource-group $RG \
  --name $VNET \
  --subnet-name $SUBNET


# ---------- CREATE 2 VMs ----------
for i in 1 2; do
  az vm create \
    --resource-group $RG \
    --name VM$i \
    --vnet-name $VNET \
    --subnet $SUBNET \
    --image $IMAGE \
    --admin-username $USERNAME \
    --generate-ssh-keys \
    --size $VM_SIZE \
    --public-ip-sku Standard \
    --nsg-rule HTTP
done



# ---------- INSTALL NGINX ----------
for i in 1 2; do
  az vm run-command invoke \
    --resource-group $RG \
    --name VM$i \
    --command-id RunShellScript \
    --scripts "sudo apt update && sudo apt install -y nginx && echo 'This is VM$i' | sudo tee
/var/www/html/index.html"
done



# ---------- PUBLIC IP FOR EXTERNAL LB ----------
az network public-ip create \
  --resource-group $RG \
  --name PublicLBIP \
  --sku Standard \
  --allocation-method Static
```

```
# ---------- CREATE EXTERNAL LB ----------
az network lb create \
  --resource-group $RG \
  --name ExternalLB \
  --sku Standard \
  --frontend-ip-name PublicFrontend \
  --backend-pool-name BackendPoolExt \
  --public-ip-address PublicLBIP


# ---------- CREATE INTERNAL LB ----------
az network lb create \
  --resource-group $RG \
  --name InternalLB \
  --sku Standard \
  --frontend-ip-name InternalFrontend \
  --backend-pool-name BackendPoolInt \
  --vnet-name $VNET \
  --subnet $SUBNET \
  --private-ip-address $ILB_PRIVATE_IP


# ---------- CREATE HEALTH PROBES ----------
az network lb probe create \
  --resource-group $RG \
  --lb-name ExternalLB \
  --name HealthProbeExt \
  --protocol tcp \
  --port 80

az network lb probe create \
  --resource-group $RG \
  --lb-name InternalLB \
  --name HealthProbeInt \
  --protocol tcp \
  --port 80


# ---------- CREATE LB RULES ----------
az network lb rule create \
  --resource-group $RG \
  --lb-name ExternalLB \
  --name HTTPRule \
  --protocol tcp \
  --frontend-port 80 \
  --backend-port 80 \
  --frontend-ip-name PublicFrontend \
  --backend-pool-name BackendPoolExt \
  --probe-name HealthProbeExt

az network lb rule create \
  --resource-group $RG \
  --lb-name InternalLB \
  --name ILBRule \
  --protocol tcp \
  --frontend-port 80 \
  --backend-port 80 \
  --frontend-ip-name InternalFrontend \
  --backend-pool-name BackendPoolInt \
  --probe-name HealthProbeInt
```

```
# ---------- ADD VMs TO BACKEND POOLS ----------
for i in 1 2; do
  NIC=$(az vm show --resource-group $RG --name VM$i --query "networkProfile.networkInterfaces[0].id" -o tsv | awk -F/ '{print $NF}')

  az network nic ip-config address-pool add \
    --resource-group $RG \
    --nic-name $NIC \
    --ip-config-name ipconfig1 \
    --lb-name ExternalLB \
    --address-pool BackendPoolExt

  az network nic ip-config address-pool add \
    --resource-group $RG \
    --nic-name $NIC \
    --ip-config-name ipconfig1 \
    --lb-name InternalLB \
    --address-pool BackendPoolInt
done



# ---------- DONE ----------
echo -e "\\n Setup complete."

echo -n " External LB Public IP: "
az network public-ip show \
  --resource-group $RG \
  --name PublicLBIP \
  --query ipAddress \
  --output tsv

echo -e "\\nInternal LB IP: $ILB_PRIVATE_IP (use curl from any VM to test)"
```