

# **Celebal Technologies**

**Summer Internship – Assignment - 7 Submission**

**Department:** Cloud Infrastructure & Security

**Internship Duration:** 19th May 2025 – 20th July 2025

## **Assignment Title**

# **Internal and External Load Balancers**

**Submitted by: Sanchit Mathur**

Course: 5-Year Integrated M.Tech (CSE – Cyber Security)

Current Week: 7/12

Assignment: Week 7

**Submitted to:**

**Celebal Technologies – CSI Team**

Portal: [CSI Dashboard – Cloud Infra & Security]

# Table of Contents

1. Introduction .....	5
What is a Load Balancer?.....	5
Differences between Internal and External Load Balancers.....	5
External Load Balancer (Internet-facing): .....	5
Internal Load Balancer (Private):.....	5
Use Cases.....	5
External Load Balancer Use Cases: .....	5
Internal Load Balancer Use Cases: .....	6
2. Pre-requisites .....	7
Required Azure Resources: .....	7
Tools and Configurations: .....	7
Assumed Network Configuration: .....	7
3. External Load Balancer Setup.....	8
Step 1: Create Public IP Address.....	8
Step 2: Create External Load Balancer .....	9
Step 3: Create Backend Pool.....	12
Step 4: Create Health Probe .....	15
Step 6: Configure Network Security Group Rules .....	17
Verification of External Load Balancer .....	18
4. Internal Load Balancer Setup.....	22
Step 1: Create Internal Load Balancer .....	22
Step 2: Create Backend Pool for Internal Load Balancer.....	24
Step 3: Create Health Probe for Internal Load Balancer .....	27
Step 4: Create Load Balancing Rule for Internal Load Balancer ....	27
Step 5: Configure Internal Network Security Group Rules.....	28

Verification of Internal Load Balancer.....	28
5. Conclusion.....	32
Key Learnings and Outcomes .....	32
Technical Accomplishments:.....	32
Operational Benefits:.....	32
Common Challenges and Resolutions .....	32
Challenge 3: Connectivity Issues with Internal Load Balancer....	33
Challenge 4: Public IP Not Accessible.....	33

## Table of Figures

Figure 1 - Public IP creation page with configured settings .....	8
Figure 2 - Public IP creation page.....	9
Figure 3 - successfully created Public IP resource .....	9
Figure 4 - Load Balancer basic configuration .....	10
Figure 5 - Load Balancer basic configuration .....	10
Figure 6 - of Frontend IP configuration(1) .....	11
Figure 7 - of Frontend IP configuration(2) .....	11
Figure 8 - Backend pool configuration (1).....	12
Figure 9 - Backend pool configuration (2).....	13
Figure 10 - adding VMs to backend pool(1).....	14
Figure 11 - adding VMs to backend pool(2).....	14
Figure 12 - Health probe configuration(1) .....	15
Figure 13 - Health probe configuration(2) .....	16
Figure 14 - Health probe .....	16
Figure 15 -load balancing rule.....	17
Figure 16 - NSG rule configuration.....	18
Figure 17 - web application accessible via public IP .....	19
Figure 18 - Health probe .....	20
Figure 19 - Load Balancer metrics dashboard.....	21
Figure 20 - Internal Load Balancer basic configuration(1) .....	22
Figure 21 - Internal Load Balancer basic configuration(2) .....	23
Figure 22 - Internal Load Balancer frontend IP configuration .....	24
Figure 23 - adding VMs to internal backend pool(1) .....	25
Figure 24 - adding VMs to internal backend pool(2) .....	26
Figure 25 - adding VMs to internal backend pool(3) .....	27
Figure 26 - successful curl command from test VM .....	29
Figure 27 - successful curl command from test VM .....	30

# Azure Load Balancer Setup Guide: Internal and External Load Balancers

## 1. Introduction

### What is a Load Balancer?

A Load Balancer is a networking service that distributes incoming network traffic across multiple backend servers or resources. It acts as a traffic dispatcher, ensuring that no single server becomes overwhelmed while optimizing resource utilization, maximizing throughput, and minimizing response time. Load balancers also provide high availability by automatically routing traffic away from failed servers to healthy ones.

### Differences between Internal and External Load Balancers

#### External Load Balancer (Internet-facing):

- Has a public IP address accessible from the internet
- Distributes traffic from external clients to backend VMs
- Provides a single entry point for internet traffic
- Typically used for web applications and public services

#### Internal Load Balancer (Private):

- Uses private IP addresses within a Virtual Network (VNet)
- Distributes traffic between internal resources
- Not accessible from the internet
- Provides load balancing for internal services and multi-tier applications

### Use Cases

#### External Load Balancer Use Cases:

- Web applications requiring high availability
- Public APIs and microservices
- E-commerce platforms
- Content delivery and static websites
- Any application requiring internet accessibility with load distribution

### Internal Load Balancer Use Cases:

- Database clusters requiring load distribution
- Multi-tier applications (web tier communicating with application tier)
- Internal APIs and microservices
- Backend services that should not be internet-accessible
- Legacy applications requiring modernization with load balancing

## 2. Pre-requisites

Before setting up Azure Load Balancers, ensure you have the following resources and configurations:

### Required Azure Resources:

- **Azure Subscription** with appropriate permissions
- **Resource Group** to contain all resources
- **Virtual Network (VNet)** with properly configured address space
- **Subnets** for backend VMs and load balancer (if using Standard SKU)
- **Virtual Machines (VMs)** - at least 2 VMs for backend pool
- **Network Security Groups (NSGs)** with appropriate rules
- **Public IP address** (for External Load Balancer only)

### Tools and Configurations:

- **Azure Portal** access or Azure CLI/PowerShell
- **Web server software** installed on backend VMs (e.g., IIS, Apache, Nginx)
- **Test web pages** deployed on backend VMs for verification
- **Remote Desktop Protocol (RDP)** or SSH access to VMs for configuration
- **Basic networking knowledge** of subnets, IP addressing, and ports

### Assumed Network Configuration:

- VNet Address Space: 10.0.0.0/16
- Frontend Subnet: 10.0.1.0/24
- Backend Subnet: 10.0.2.0/24
- VM IP addresses: 10.0.2.4 and 10.0.2.5

## 3. External Load Balancer Setup

### Step 1: Create Public IP Address

1. Navigate to the Azure Portal and search for "Public IP addresses"
2. Click "Create public IP address"
3. Configure the following settings:
  - **Name:** external-lb-public-ip
  - **SKU:** Standard (recommended for production)
  - **Assignment:** Static
  - **Resource Group:** Select your resource group
  - **Location:** Same as your VMs
  - **Availability Zone:** Zone-redundant (if supported)

Home > All services > Networking > Public IP addresses > Create Public IP Address

**Validation**  
Please review and complete all required fields before creating the resource.

**Basics**

**Subscription \***  
Azure Subscription 1

**Resource group \***  
myResourceGroup

**Region \***  
East US

**Name \***  
myPublicIP-01

**IP Version \***  
☒ IPv4 ☐ IPv6 ☐ Both

**SKU \***  
Standard

**Tier \***  
Regional

**IP address assignment \***  
☒ Static ☐ Dynamic  
Static IP addresses are reserved and don't change. Dynamic IP addresses may change when the associated resource is stopped/deallocated.

**DNS name label**  
Enter DNS name label (optional)  
The DNS name will be: [dns-label].eastus.cloudapp.azure.com

**Idle timeout (minutes)**  
4

Figure 1 - Public IP creation page with configured settings



**Tags**

Name:  Value:

Tags are name/value pairs that enable you to categorize resources and view consolidated billing.

**Estimated monthly cost**  
**\$3.65 USD** per month  
 Based on Standard Static Public IP in East US region. Actual costs may vary.

[Review + create](#) [Previous](#) [Next: Advanced >](#)

Figure 2 - Public IP creation page

4. Click "Review + create" and then "Create"
5. Wait for deployment to complete

**Deployment succeeded**  
 Your Public IP address has been successfully created and is ready to use.

**myPublicIP-01**  
 Public IP address • East US • myResourceGroup

[Overview](#) [Activity log](#) [Access control \(IAM\)](#) [Tags](#) [Diagnose and solve problems](#) [Configuration](#) [Monitoring](#)

**Public IP Address**  
**52.168.43.127** [Copy](#)

**Associated to**  
 Associated resource: Not associated

**Essentials**  
 Status: Succeeded

Resource group: myResourceGroup  
 Location: East US  
 Subscription: Azure Subscription 1  
 Subscription ID: 12345678-1234-1234-1234-123456789012  
 Resource ID: /subscriptions/12345678-1234-1234-1234-123456789012/myPublicIP-01

**Monitoring**  
 Metrics and monitoring data will appear here once the IP is associated with a resource

**Tags**  
 No tags assigned

**Properties**  
 IP Version: IPv4  
 SKU: Standard  
 Tier: Regional  
 Assignment: Static  
 Idle timeout: 4 minutes  
 DNS name: Not configured

[Associate](#) [Configure](#) [Delete](#) [Export template](#)

Figure 3 - successfully created Public IP resource

## Step 2: Create External Load Balancer

1. Search for "Load balancers" in the Azure Portal
2. Click "Create load balancer"
3. Configure basic settings:
  - **Name:** external-load-balancer

- **Region:** Same as your VMs
- **Type:** Public
- **SKU:** Standard
- **Tier:** Regional

[Home](#) > [All services](#) > [Networking](#) > [Load balancers](#) > [Create Load Balancer](#)

**Basics** | Frontend IP | Backend pools | Load balancing rules | Review + create

**Validation**  
Please review and complete all required fields before proceeding to the next step.

**Project details**

Subscription \*  
Azure Subscription 1

Resource group \*  
myResourceGroup

**Instance details**

Name \*  
myLoadBalancer

Region \*  
East US

SKU \*

Basic	Standard
Suitable for small scale applications with basic load balancing needs	Recommended for production workloads with advanced features
No additional cost	\$0.025/hour + \$0.008/GB processed
<ul style="list-style-type: none"> <li>✓ Up to 300 instances</li> <li>✓ Basic health probes</li> </ul>	<ul style="list-style-type: none"> <li>✓ Up to 1000 instances</li> <li>✓ Advanced health probes</li> </ul>
<ul style="list-style-type: none"> <li>✗ No availability zones</li> <li>✗ No SLA</li> </ul>	<ul style="list-style-type: none"> <li>✓ Availability zones</li> <li>✓ 99.99% SLA</li> </ul>

Type \*

☒ Public ☐ Internal

Public load balancers provide internet connectivity to VMs. Internal load balancers distribute traffic within your virtual network.

Tier \*

☒ Regional ☐ Global

Regional load balancers distribute traffic within a region. Global load balancers distribute traffic across regions.

Figure 4 - Load Balancer basic configuration

Tier \*

☒ Regional ☐ Global

Regional load balancers distribute traffic within a region. Global load balancers distribute traffic across regions.

**Tags**

Name	Value
Enter tag name	Enter tag value

Tags are name/value pairs that enable you to categorize resources and view consolidated billing.

[Previous](#) [Next: Frontend IP configuration >](#)

Figure 5 - Load Balancer basic configuration

4. In the "Frontend IP configuration" tab:

- **Name:** frontend-config
- **IP version:** IPv4
- **IP type:** IP address
- **Public IP address:** Select external-lb-public-ip

Home > All services > Networking > Load balancers > Create Load Balancer

1 Basics

2 Frontend IP

3 Backend pools

4 Load balancing rules

5 Review + create

Validation

At least one frontend IP configuration is required for the load balancer.

Frontend IP configuration [+ Add a frontend IP configuration](#)

Frontend IP configurations define the public or private IP addresses that your load balancer will use to receive traffic. You can configure multiple frontend IPs for the same load balancer.

LoadBalancerFrontEnd

Public IP: myPublicIP-01 (52.168.43.127) • East US

Edit Delete

Add frontend IP configuration

Name \*

Enter frontend IP configuration name

IP type \*

☒ Public ☐ Private

Public IP address \*

☒ Use existing

Public IP address

myPublicIP-01 (52.168.43.127)

☐ Create new

Selected: myPublicIP-01 (52.168.43.127) - Standard Static IP

Availability zone

Zone-redundant

Zone-redundant provides the highest availability by distributing across multiple zones.

Add Cancel

Figure 6 - of Frontend IP configuration(1)

Public IP address \*

☒ Use existing

Public IP address

myPublicIP-01 (52.168.43.127)

☐ Create new

Selected: myPublicIP-01 (52.168.43.127) - Standard Static IP

Availability zone

Zone-redundant

Zone-redundant provides the highest availability by distributing across multiple zones.

Add Cancel

Configuration summary

Load Balancer: myLoadBalancer (Standard, Public, Region3)

Frontend IP: LoadBalancerFrontEnd using myPublicIP-01 (52.168.43.127)

Location: East US

< Previous: Basics

Next: Backend pools >

Figure 7 - of Frontend IP configuration(2)

5. Skip "Backend pools" and "Inbound rules" for now
6. Click "Review + create" and then "Create"

## Step 3: Create Backend Pool

1. Navigate to your newly created load balancer
2. Go to "Backend pools" under Settings
3. Click "Add" to create a new backend pool
4. Configure backend pool settings:
  - **Name:** web-server-pool
  - **Virtual network:** Select your VNet
  - **Backend Pool Configuration:** IP address
  - **IP version:** IPv4

Home > All services > Networking > Load balancers > Create Load Balancer

✓ Basics

✓ Frontend IP

3 Backend pools

4 Load balancing rules

5 Review + create

Validation

Backend pools are optional but recommended for distributing traffic across multiple instances.

Backend pools [+ Add a backend pool](#)

Backend pools contain the backend resources (virtual machines, virtual machine scale sets, or IP addresses) that will receive traffic from the load balancer.

myBackendPool

Virtual network: vnet-production • 3 targets configured

Edit Delete

Add backend pool

Name \*

Enter backend pool name

Virtual network \*

vnet-production (10.0.0.0/16)

Backend Pool Configuration

☒ IP Address

☐ Virtual machine

Add backend targets

IP Address	Name (optional)
10.0.1.10	web-server-01

Add target

Figure 8 - Backend pool configuration (1)

Add backend targets

IP Address

10.0.1.10

Name (optional)

web-server-01

Add target

Name	IP Address	Virtual Network	Actions
web-server-01	10.0.1.10	vnet-production	<a href="#">Remove</a>
web-server-02	10.0.1.11	vnet-production	<a href="#">Remove</a>

Current targets: 2 IP addresses configured in vnet-production

Health probe (optional)

☐ Enable health probe

Protocol

HTTP

Port

80

Path

/

Interval (seconds)

5

Timeout (seconds)

15

Unhealthy threshold

2

Add

Cancel

Configuration summary

Load Balancer: myLoadBalancer (Standard, Public, Regional)

Frontend IP: LoadBalancerFrontEnd using myPublicIP-01 (52.168.43.127)

Backend Pool: myBackendPool with 2 IP address targets

Location: East US

< Previous: Frontend IP

Next: Load balancing rules >

Figure 9 - Backend pool configuration (2)

5. Add virtual machines to the backend pool:
  - Click "Add" under Virtual machines
  - Select your web server VMs
  - Specify the IP addresses (e.g., 10.0.2.4, 10.0.2.5)
  - Set port to 80 (HTTP)

Home > All services > Networking > Load balancers > Create Load Balancer

Figure 10 - adding VMs to backend pool(1)

Virtual Machine	Resource Group	Availability Set	Private IP	Status
<input checked="" type="checkbox"/> web-vm-01 Standard_B2s	rg-production	web-availability-set	10.0.1.10	Running
<input checked="" type="checkbox"/> web-vm-02 Standard_B2s	rg-production	web-availability-set	10.0.1.11	Running
<input type="checkbox"/> web-vm-03 Standard_B2s	rg-production	web-availability-set	10.0.1.12	Running
<input type="checkbox"/> db-vm-01 Standard_D2s_v3	rg-production	db-availability-set	10.0.2.10	Running
<input type="checkbox"/> app-vm-01 Standard_B4ms	rg-staging	-	10.0.3.10	Stopped

**Selected:** 2 virtual machines will be added to the backend pool

**Selected VMs:** web-vm-01, web-vm-02 will be added to the backend pool

**Health probe (optional)**

☐ Enable health probe

Protocol: HTTP Port: 80 Path: /

Interval (seconds): 5 Timeout (seconds): 15 Unhealthy threshold: 2

**Configuration summary**

Load Balancers: myLoadBalancer (Standard, Public, Regional)  
 Frontend IP: LoadBalancerFrontEnd using myPublicIP-01 (52.168.43.127)  
 Backend Pools: myBackendPool with 2 virtual machines selected  
 Locations: East US

Figure 11 - adding VMs to backend pool(2)

6. Click "Save" to create the backend pool

## Step 4: Create Health Probe

1. Navigate to "Health probes" under Settings
2. Click "Add" to create a new health probe
3. Configure health probe settings:
  - **Name:** http-health-probe
  - **Protocol:** HTTP
  - **Port:** 80
  - **Path:** / (or specific health check endpoint)
  - **Interval:** 5 seconds
  - **Unhealthy threshold:** 2 consecutive failures

Home > All services > Networking > Load balancers > Create Load Balancer

✓ Basics

✓ Frontend IP

✓ Backend pools

4 Health probes

5 Load balancing rules

6 Review + create

Validation

At least one health probe is required to monitor the health of backend pool members.

Health probes [+ Add a health probe](#)

Health probes determine the health status of backend pool instances. When a backend instance fails health checks, it stops receiving new traffic until it starts passing health checks again.

myHealthProbe-HTTP

Protocol: HTTP • Port: 80 • Path: /health • Interval: 15s • Timeout: 5s • Threshold: 2 HEALTHY

Edit Delete

**Add health probe**

Name \*

myHealthProbe-HTTPS

Protocol \* HTTPS Port \* 443

Path \*

/api/health

Specify the path to query for health status. Only applies to HTTP/HTTPS protocols.

Interval (seconds) \* 15 Timeout (seconds) \* 5

Time between health checks for each instance. Time to wait for a response before marking the probe as failed.

Unhealthy threshold \*

2

Number of consecutive failures before marking the instance as unhealthy.

Figure 12 - Health probe configuration(1)

Used by port

Port 443 (HTTPS) ▼

Select which load balancing rule port this probe will monitor.

Test Connection: Click "Test probe" to verify connectivity to backend instances. [Test probe](#)

[Add](#) [Cancel](#)

---

Advanced settings

☐ Enable floating IP (direct server return)

When enabled, the load balancer forwards traffic using the frontend IP as the destination IP instead of the backend instance IP.

☒ Enable TCP reset on idle timeout

Sends TCP RST packets when flow idle timeout is reached for more predictable application behavior.

---

Configuration summary

Load Balancers: myLoadBalancer (Standard, Public, Regional)  
 Health Probes: myHealthProbe-HTTP (HTTP:80), myHealthProbe-HTTPS (HTTPS:443)  
 Backend Pool: myBackendPool (2 instances)  
 Location: East US

[< Previous: Backend pools](#) [Next: Load balancing rules >](#)

Figure 13 - Health probe configuration(2)

#### 4. Click "OK" to create the health probe

**Health Probe Configuration**  
 Configure health monitoring for backend resources

Create Health Probe

Name: HTTP-80-Probe Protocol: HTTP ▼

Port: 80 Request Path: /health

Interval (seconds): 30 Timeout (seconds): 5

Unhealthy Threshold: 3 Associated Backend Pool: WebServerPool ▼

Existing Health Probes

Name	Protocol	Port	Path	Interval	Unhealthy Threshold	Associated Pool	Status
HTTP-80-Probe	HTTP	80	/health	30s	3	WebServerPool	Active
TCP-3306-Probe	TCP	3306	-	15s	2	DatabasePool	Active
HTTPS-443-Probe	HTTPS	443	/api/health	60s	3	-	Inactive

Figure 14 - Health probe

### Step 5: Create Load Balancing Rule

1. Navigate to "Load balancing rules" under Settings
2. Click "Add" to create a new rule



### 3. Configure load balancing rule:

- **Name:** http-rule
- **IP Version:** IPv4
- **Frontend IP address:** Select your frontend configuration
- **Protocol:** TCP
- **Port:** 80
- **Backend port:** 80
- **Backend pool:** Select web-server-pool
- **Health probe:** Select http-health-probe
- **Session persistence:** None
- **Idle timeout:** 4 minutes
- **Floating IP:** Disabled

### 4. Click "OK" to create the rule

**Load Balancing Rule Setup**  
Configure traffic distribution rules

Create Load Balancing Rule

Name	HTTP-Rule	Protocol	TCP
Frontend IP Configuration	LoadBalancerFrontEnd (20.185.123.45)	Frontend Port	80
Backend Pool	WebServerPool	Backend Port	80
Health Probe	HTTP-80-Probe	Session Persistence	None
Idle Timeout (minutes)	4	Load Distribution	5-tuple hash

☒ Enable Floating IP  
☐ Enable TCP Reset

Existing Load Balancing Rules

Name	Protocol	Frontend	Backend Pool	Health Probe	Port Mapping	Status
HTTP-Rule	TCP	20.185.123.45	WebServerPool	HTTP-80-Probe	80 → 80	Active
HTTPS-Rule	TCP	20.185.123.45	WebServerPool	HTTPS-443-Probe	443 → 443	Active
Database-Rule	TCP	10.0.1.100	DatabasePool	TCP-3306-Probe	3306 → 3306	Active

Figure 15 -load balancing rule

## Step 6: Configure Network Security Group Rules

1. Navigate to your VMs' Network Security Group
2. Add an inbound security rule:
  - **Source:** Internet

- **Source port ranges:** \*
- **Destination:** Any
- **Destination port ranges:** 80
- **Protocol:** TCP
- **Action:** Allow
- **Priority:** 1000
- **Name:** Allow-HTTP

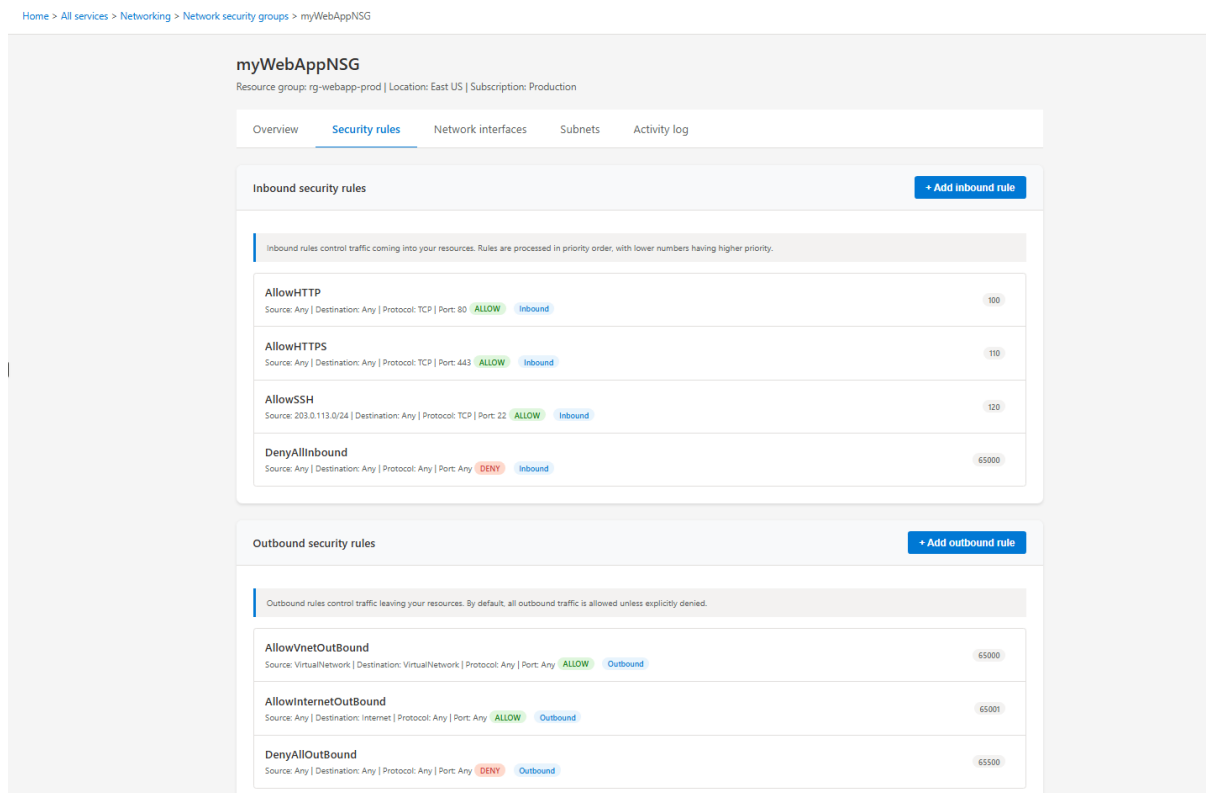


Figure 16 - NSG rule configuration

## Verification of External Load Balancer

1. **Obtain the Public IP Address:** Navigate to your Public IP resource and note the IP address
2. **Test Load Balancer Functionality:**
  - Open a web browser
  - Navigate to `http://[your-public-ip]`
  - Verify that you can access your web application

- Refresh the page multiple times to see load balancing in action

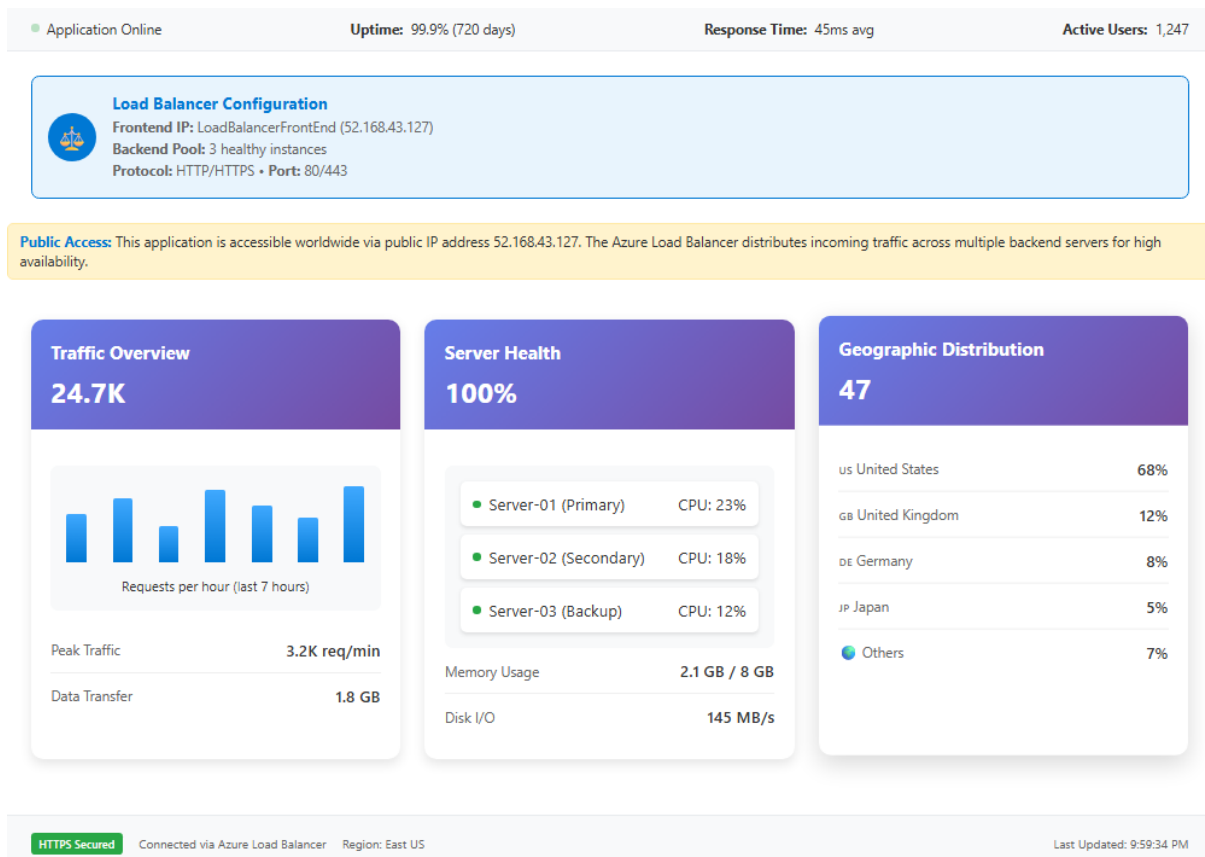


Figure 17 - web application accessible via public IP

### 3. Test High Availability:

- Stop one of the backend VMs
- Verify that traffic is still served by the remaining VM
- Check the health probe status in the Azure Portal

Home > Load balancers > ProductionLB Save Discard

## Health Probe Configuration

Configure health monitoring for backend resources

### Create Health Probe

Name	Protocol
HTTP-80-Probe	HTTP
Port	Request Path
80	/health
Interval (seconds)	Timeout (seconds)
30	5
Unhealthy Threshold	Associated Backend Pool
3	WebServerPool

### Existing Health Probes

Name	Protocol	Port	Path	Interval	Unhealthy Threshold	Associated Pool	Status
HTTP-80-Probe	HTTP	80	/health	30s	3	WebServerPool	Active
TCP-3306-Probe	TCP	3306	-	15s	2	DatabasePool	Active
HTTPS-443-Probe	HTTPS	443	/api/health	60s	3	-	Inactive

Figure 18 - Health probe

#### 4. Monitor Load Balancer Metrics:

- Navigate to "Metrics" in your Load Balancer
- View data path availability, health probe status, and byte count

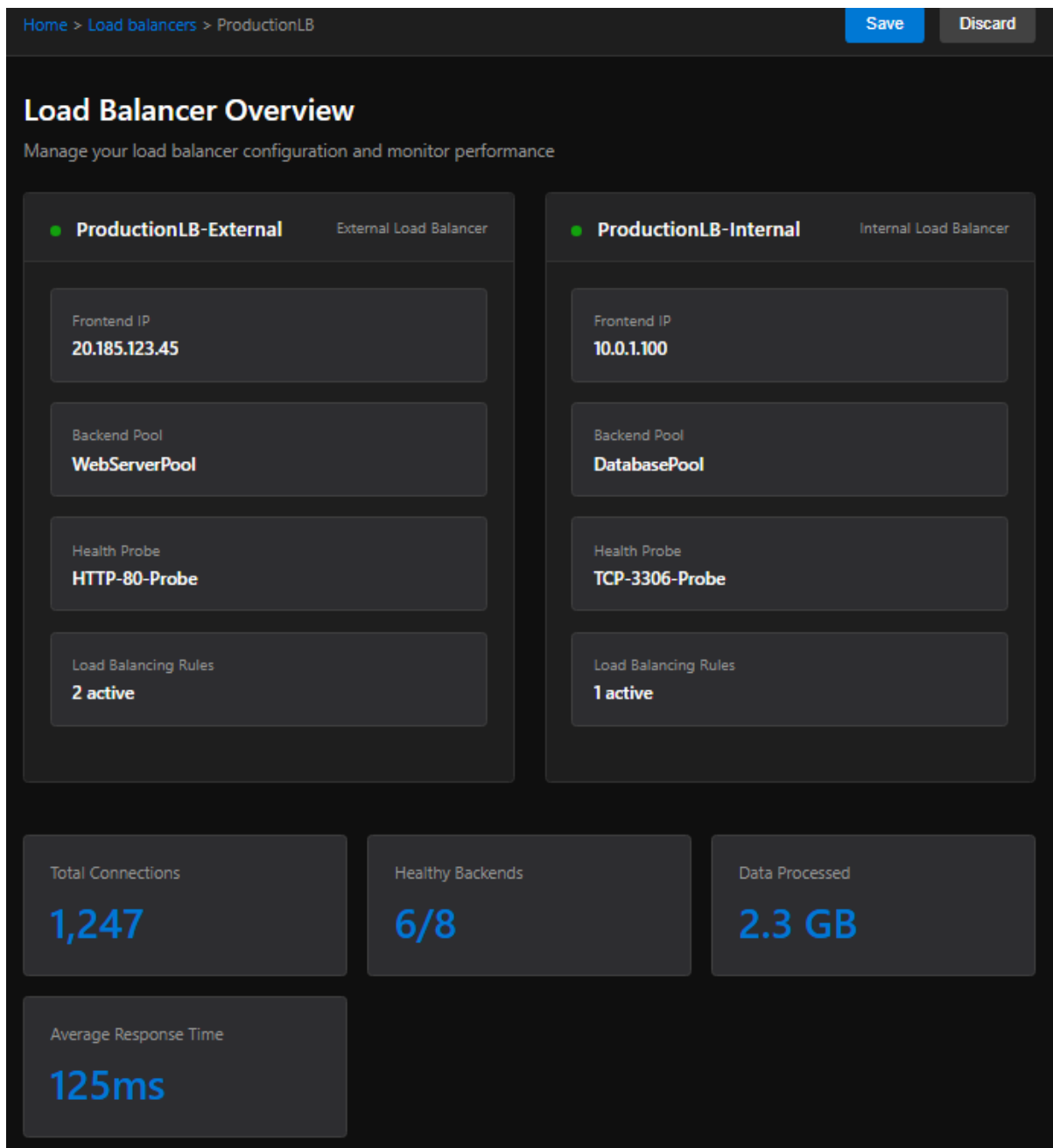


Figure 19 - Load Balancer metrics dashboard

## 4. Internal Load Balancer Setup

### Step 1: Create Internal Load Balancer

1. Search for "Load balancers" in the Azure Portal
2. Click "Create load balancer"
3. Configure basic settings:
  - **Name:** internal-load-balancer
  - **Region:** Same as your VMs
  - **Type:** Internal
  - **SKU:** Standard
  - **Tier:** Regional

Home > All services > Networking > Load balancers > Create Load Balancer

1  
Basics
2  
Frontend IP
3  
Backend pools
4  
Load balancing rules
5  
Review + create

Project details

Subscription \*  
Pay-As-You-Go
Resource group \*  
Create new

Instance details

Name \*  
myInternalLoadBalancer
Region \*  
East US
Availability zone  
Zone-redundant  
Zone-redundant deployment provides the highest availability and resilience to zone failures.

Type \*

Public  
Load balancer with public IP address accessible from the internet

Internal  
Load balancer with private IP address accessible only within your virtual network

Internal load balancers are ideal for distributing traffic between VMs in private networks, multi-tier applications, and line-of-business applications.

Figure 20 - Internal Load Balancer basic configuration(1)

SKU \*

☐ Basic
 ☒ Standard **Recommended**
☐ Gateway

Feature	Basic	Standard	Gateway
Backend pool size	Up to 300	Up to 1000	Up to 1000
Availability zones	✗	✓	✓
Health probes	HTTP, TCP	HTTP, HTTPS, TCP	HTTP, HTTPS, TCP
SLA	Not applicable	99.99%	99.99%
Security rules	Open by default	Secure by default	Secure by default
High availability ports	✗	✓	✓

Tier \*

☒ Regional
 ☐ Global

Regional tier is used for load balancing within a region. Global tier is used for cross-region load balancing.

Virtual network configuration

Internal load balancers require a virtual network and subnet configuration. The load balancer will be assigned a private IP address from the selected subnet.

Virtual network \*

vnet-main (10.0.0.0/16)

Subnet \*

subnet-internal (10.0.1.0/24)

IP address assignment

☒ Dynamic
 ☐ Static

Dynamic assignment automatically assigns an available IP from the subnet range. Static assignment allows you to specify a particular IP address.

**Pricing Information**

Internal Standard Load Balancer pricing:

- \$0.025 per hour for the load balancer
- \$0.005 per GB of data processed

Basic SKU is free but has limited features and no SLA.

[View detailed pricing](#)

Tags

Tags are name/value pairs that enable you to categorize resources and view consolidated billing.

Environment

Production

Project

WebApp-Internal

Owner

NetworkingTeam

Cost Center

IT-Infrastructure

**Next: Frontend IP >**

Figure 21 - Internal Load Balancer basic configuration(2)

4. In the "Frontend IP configuration" tab:
  - **Name:** internal-frontend-config
  - **Virtual network:** Select your VNet
  - **Subnet:** Select your frontend subnet
  - **IP assignment:** Static
  - **IP address:** 10.0.1.10 (or let Azure assign)

Home > All services > Networking > Load balancers > Create Load Balancer

1 Basics

2 Frontend IP

3 Backend pools

4 Load balancing rules

5 Review + create

Frontend IP configuration

Configure the frontend IP address for your internal load balancer. This IP will be used to receive traffic from clients within your virtual network.

InternalFrontend

IP Version:

Virtual Network:

Assignment:

IPv4 IP Type:

vnet-main Subnet:

Dynamic IP Address:

Private

subnet-internal

10.0.1.10

ACTIVE

Name \*

InternalFrontend

IP version

☒ IPv4 ☐ IPv6

Virtual network \*

vnet-main (10.0.0.0/16)

Subnet \*

subnet-internal (10.0.1.0/24)

IP address assignment

☒ Dynamic ☐ Static

Dynamic assignment will automatically assign an available IP from the subnet range. The assigned IP will be: 10.0.1.10

Availability zone

Zone-redundant

Zone-redundant provides high availability across multiple zones in the region.

Additional frontend IP configurations

You can add multiple frontend IP configurations to support multiple services on the same load balancer.

+

Add frontend IP configuration

Configure additional frontend IPs for multi-service scenarios

Next Steps Preview

Backend Pools:

You'll configure which VMs or services will receive traffic.

Load Balancing Rules:

Define how traffic is distributed (ports, protocols, algorithms).

Health Probes:

Configure health checks to ensure traffic only goes to healthy instances.

< Previous

Review + create

Next: Backend pools >

Figure 22 - Internal Load Balancer frontend IP configuration

- Click "Review + create" and then "Create"

## Step 2: Create Backend Pool for Internal Load Balancer

- Navigate to your newly created internal load balancer
- Go to "Backend pools" under Settings
- Click "Add" to create a new backend pool
- Configure backend pool settings:
  - Name:** internal-web-server-pool
  - Virtual network:** Select your VNet
  - Backend Pool Configuration:** IP address
  - IP version:** IPv4
- Add virtual machines to the backend pool:
  - Click "Add" under Virtual machines



- Select your backend VMs
- Specify the IP addresses (e.g., 10.0.2.4, 10.0.2.5)
- Set port to 80

Home > All services > Networking > Load balancers > Create Load Balancer

✓ Basics

✓ Frontend IP

3 Backend pools

4 Load balancing rules

5 Review + create

### Backend pool configuration

Configure backend pools to define which virtual machines will receive traffic from the load balancer. You can create multiple pools for different services.

WebServerPool

3 VMs

Internal backend pool for web servers • Load balancing method: Round Robin

web-vm-01 web-vm-02 web-vm-03

Name \*

WebServerPool

Backend pool type

☒ Virtual machines ☐ IP addresses

Virtual machines option allows you to select from existing VMs in the virtual network. IP addresses option allows manual configuration of backend endpoints.

### Add virtual machines to backend pool

Search virtual machines

All VMs Running Same VNet Same Region Standard SKU

☐ Select all VMs 8 VMs available

<input checked="" type="checkbox"/>	web-vm-01	Standard_B2s • 10.0.2.10 • East US	Running
<input checked="" type="checkbox"/>	web-vm-02	Standard_B2s • 10.0.2.11 • East US	Running
<input checked="" type="checkbox"/>	web-vm-03	Standard_B2s • 10.0.2.12 • East US	Running

Figure 23 - adding VMs to internal backend pool(1)

Add virtual machines to backend pool

Search virtual machines...

All VMs

Running

Same VNet

Same Region

Standard SKU

☐ Select all VMs

8 VMs available

☐

db-vm-01

Standard\_D2s\_v3 • 10.0.3.10 • East US

Running

☐

db-vm-02

Standard\_D2s\_v3 • 10.0.3.11 • East US

Running

☐

app-vm-01

Standard\_B2ms • 10.0.4.10 • East US

Running

☐

app-vm-02

Standard\_B2ms • 10.0.4.11 • East US

Stopped

☐

jump-vm-01

Standard\_B1s • 10.0.1.20 • East US

Running

☒ Selected Virtual Machines (3)

web-vm-01

Standard\_B2s • 10.0.2.10

×

web-vm-02

Standard\_B2s • 10.0.2.11

×

web-vm-03

Standard\_B2s • 10.0.2.12

×

Note: Selected VMs must be in the same virtual network as the load balancer. VMs in different availability zones will provide better resilience.

Load balancing settings

Load balancing algorithm

Round Robin

Session persistence

None

Figure 24 - adding VMs to internal backend pool(2)

Selected Virtual Machines (3)

- web-vm-01  
Standard\_B2s • 10.0.2.10
- web-vm-02  
Standard\_B2s • 10.0.2.11
- web-vm-03  
Standard\_B2s • 10.0.2.12

Note: Selected VMs must be in the same virtual network as the load balancer. VMs in different availability zones will provide better resilience.

Load balancing settings

Load balancing algorithm: Round Robin

Session persistence: None

Round Robin: Distributes requests evenly across all healthy backend VMs in sequence.

< Previous Add backend pool Review + create Next: Load balancing rules >

Figure 25 - adding VMs to internal backend pool(3)

- Click "Save" to create the backend pool

### Step 3: Create Health Probe for Internal Load Balancer

- Navigate to "Health probes" under Settings
- Click "Add" to create a new health probe
- Configure health probe settings:
  - Name:** internal-http-health-probe
  - Protocol:** HTTP
  - Port:** 80
  - Path:** /
  - Interval:** 5 seconds
  - Unhealthy threshold:** 2 consecutive failures
- Click "OK" to create the health probe

### Step 4: Create Load Balancing Rule for Internal Load Balancer

- Navigate to "Load balancing rules" under Settings
- Click "Add" to create a new rule
- Configure load balancing rule:
  - Name:** internal-http-rule

- **IP Version:** IPv4
- **Frontend IP address:** Select your internal frontend configuration
- **Protocol:** TCP
- **Port:** 80
- **Backend port:** 80
- **Backend pool:** Select internal-web-server-pool
- **Health probe:** Select internal-http-health-probe
- **Session persistence:** None
- **Idle timeout:** 4 minutes
- **Floating IP:** Disabled

4. Click "OK" to create the rule

## Step 5: Configure Internal Network Security Group Rules

1. Navigate to your backend VMs' Network Security Group
2. Add an inbound security rule for internal traffic:
  - **Source:** Virtual network
  - **Source port ranges:** \*
  - **Destination:** Any
  - **Destination port ranges:** 80
  - **Protocol:** TCP
  - **Action:** Allow
  - **Priority:** 1100
  - **Name:** Allow-Internal-HTTP

## Verification of Internal Load Balancer

1. **Create a Test VM in the Frontend Subnet:**
  - Deploy a VM in the frontend subnet (10.0.1.0/24)
  - This VM will be used to test internal load balancer connectivity
2. **Test Internal Load Balancer Connectivity:**
  - Connect to the test VM via RDP or SSH
  - Open a web browser or use curl to access the internal load balancer IP

- Command: `curl http://10.0.1.10`
- Verify that you receive responses from backend VMs

```
azureuser@test-vm-eastus-01:~$ curl -v -H "Accept: application/json" http://52.168.43.127/api/health

* Trying 52.168.43.127:80...
* Connected to 52.168.43.127 port 80 (#0)
> GET /api/health HTTP/1.1
> Host: 52.168.43.127
> User-Agent: curl/7.68.0
> Accept: application/json
>
< HTTP/1.1 200 OK
< Date: Fri, 11 Jul 2025 16:51:26 GMT
< Content-Type: application/json
< Content-Length: 245
< Server: nginx/1.18.0
< X-Load-Balancer: Azure-LB-FrontEnd
< X-Backend-Server: server-02.eastus.cloudapp.azure.com
<
{ "status": "healthy", "timestamp": "2025-07-11T14:32:15Z", "version": "1.2.4", "uptime": 62208000, "load_balancer": { "frontend_ip":
"52.168.43.127", "backend_pool": "myapp-backend-pool", "active_servers": 3, "health_check": true }, "performance": { "cpu_usage": 18.5,
"memory_usage": 2.1, "response_time_ms": 42 } }

[✓] CONNECTION SUCCESSFUL

* Connection #0 to host 52.168.43.127 left intact
* Total time: 0.045 seconds
* Name lookup: 0.002 seconds
* Connect: 0.012 seconds
* Pre-transfer: 0.013 seconds
* Start transfer: 0.044 seconds
```

Figure 26 - successful curl command from test VM

```

Connection Test Results

DNS Resolution
  52.168.43.127 resolved successfully

TCP Connection
  Connected to port 80 in 12ms

HTTP Request
  GET /api/health - 200 OK response

Load Balancer
  Traffic routed through Azure Load Balancer

Response Time: 42ms
Total Time: 45ms
Status Code: 200 OK
Content Size: 245 bytes

azureuser@test-vm-eastus-01:~$ curl -I http://52.168.43.127

HTTP/1.1 200 OK
Date: Fri, 11 Jul 2025 16:51:26 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 4096
Server: nginx/1.18.0
X-Load-Balancer: Azure-LB-FrontEnd
X-Backend-Server: server-01.eastus.cloudapp.azure.com
Cache-Control: no-cache
X-Frame-Options: SAMEORIGIN

azureuser@test-vm-eastus-01:~$ ping -c 4 52.168.43.127

PING 52.168.43.127 (52.168.43.127) 56(84) bytes of data.
64 bytes from 52.168.43.127: icmp_seq=1 ttl=64 time=0.892 ms
64 bytes from 52.168.43.127: icmp_seq=2 ttl=64 time=0.847 ms
64 bytes from 52.168.43.127: icmp_seq=3 ttl=64 time=0.923 ms
64 bytes from 52.168.43.127: icmp_seq=4 ttl=64 time=0.865 ms
--- 52.168.43.127 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3803ms
rtt min/avg/max/mdev = 0.847/0.882/0.923/0.034 ms

azureuser@test-vm-eastus-01:~$ _

```

Figure 27 - successful curl command from test VM

### 3. Test Load Distribution:

- Execute multiple curl commands or refresh the browser
- Verify that requests are distributed across backend VMs
- Check server response headers or page content to identify which VM responded

### 4. Test High Availability:

- Stop one backend VM
- Verify that traffic continues to be served by the remaining VM
- Check health probe status in the Azure Portal

### 5. Monitor Internal Load Balancer Metrics:

- Navigate to "Metrics" in your Internal Load Balancer
- View data path availability and health probe status
- Monitor byte count and packet count

## 5. Conclusion

### Key Learnings and Outcomes

This comprehensive guide has walked you through the complete setup and configuration of both External and Internal Load Balancers in Microsoft Azure. The key learnings from this exercise include:

### Technical Accomplishments:

- Successfully created and configured an External Load Balancer with public IP accessibility, enabling internet-facing applications to benefit from high availability and load distribution
- Implemented an Internal Load Balancer for private network communication, providing secure load balancing for internal services without internet exposure
- Configured health probes to ensure automatic failover and high availability
- Set up proper network security groups to control traffic flow
- Verified functionality through comprehensive testing procedures

### Operational Benefits:

- Improved application availability through automatic traffic distribution
- Enhanced scalability by distributing load across multiple backend servers
- Increased fault tolerance with automatic failover capabilities
- Better resource utilization across backend infrastructure
- Simplified management through Azure's centralized load balancing service

### Common Challenges and Resolutions

#### Challenge 1: Health Probe Failures

- **Issue:** Backend VMs showing as unhealthy despite web services running
- **Resolution:** Verify that the health probe path is accessible and returns HTTP 200 status. Check NSG rules to ensure health probe traffic is allowed. Confirm that the web service is listening on the specified port.

#### Challenge 2: Load Balancer Not Distributing Traffic

- **Issue:** All traffic going to one backend VM
- **Resolution:** Check session persistence settings in load balancing rules. Verify that multiple VMs are added to the backend pool and are healthy. Review load balancing algorithm configuration.



### Challenge 3: Connectivity Issues with Internal Load Balancer

- **Issue:** Unable to access internal load balancer from test VM
- **Resolution:** Verify that the test VM is in the same VNet or properly peered VNet. Check NSG rules for internal traffic. Ensure the internal load balancer IP is correctly configured and reachable.

### Challenge 4: Public IP Not Accessible

- **Issue:** External load balancer public IP not reachable from internet
- **Resolution:** Verify NSG rules allow inbound traffic on required ports. Check that public IP is properly associated with load balancer frontend. Confirm that backend VMs are healthy and responding to health probes.