

---

# Azure Public Load Balancer with VMs, Bastion, NAT Gateway, VNet, and Subnets

---

## Project Architecture Overview

You will be deploying the following:

- A **Virtual Network (VNet)** with 3 subnets:
  - `BackendSubnet` for VMs
  - `AzureBastionSubnet` for Bastion host (required name)
  - `NATGatewaySubnet` for outbound NAT access
- **Two Virtual Machines (VM1 & VM2)** in `BackendSubnet`
- A **Public Load Balancer** distributing HTTP (port 80) traffic between VMs
- A **Health Probe** to monitor backend VM availability
- **Load Balancing Rule** to forward traffic from the frontend (public IP) to backend VMs
- **Azure Bastion** to securely access VMs without exposing public IPs
- A **NAT Gateway** to provide outbound internet access from private VMs

---

## 1 Create Resource Group

### Why?

Grouping related resources simplifies management, RBAC control, and billing.

### Steps:

1. Go to **Azure Portal** → Search for **Resource Groups**
2. Click **+ Create**
3. Fill:
  - Name: `LoadBalancerProjectRG`
  - Region: (e.g., East US)
4. Click **Review + Create** → **Create**

---

## 2 Create Virtual Network and Subnets

### Why?

A virtual network is required for communication between VMs, Bastion, NAT Gateway, and Load Balancer.

### Steps:

1. Go to **Virtual Networks** → Click **+ Create**
  2. Fill:
    - Name: `LBVNet`
    - Address space: `10.0.0.0/16`
  3. Add Subnets:
    - BackendSubnet → `10.0.1.0/24`
    - AzureBastionSubnet → `10.0.2.0/24` (Name must be exact!)
    - NATGatewaySubnet → `10.0.3.0/24`
  4. Create the network
- 

## 3 Create Two Virtual Machines (VM1 & VM2)

### Why?

The VMs will serve as backend pool members for the Load Balancer.

### Repeat the process twice:

1. Go to **Virtual Machines** → **+ Create**
  2. Under **Basics**:
    - Name: `VM1` (and `VM2` later)
    - Region: Same as VNet
    - Image: Ubuntu 20.04 LTS (*or Windows if preferred*)
    - Size: Standard B1s
    - Authentication: Password or SSH
  3. **Networking Tab**:
    - VNet: `LBVNet`
    - Subnet: `BackendSubnet`
    - **Public IP**: None (Bastion will be used)
  4. **Review + Create** → **Create**
- 

## 4 Create Azure Bastion Host

### Why?

Azure Bastion provides secure and seamless RDP/SSH access to VMs **without exposing public IPs**.

### Steps:

1. Search for **Bastion** → Click **+ Create**
2. Fill:

- Name: AzureBastionHost
- VNet: LBVNet
- Subnet: AzureBastionSubnet (must be named exactly)
- Region: Same

3. Create new Public IP: BastionPublicIP

4. **Review + Create** → **Create**

#### Connect to VM:

- Go to **VM1** → **Connect** > **Bastion**
- Enter credentials → Open browser-based terminal

---

## 5 Create Public Load Balancer

### Why?

A public Load Balancer distributes traffic across multiple backend VMs.

### Steps:

1. Go to **Load Balancers** → Click **+ Create**

2. Basics:

- Name: PublicLB
- Type: **Public**
- SKU: **Standard** (required for private VMs & health probes)
- Public IP: Create new → PublicLB-IP

3. **Review + Create** → **Create**

---

## 6 Configure Backend Pool (Attach VMs)

### Why?

This links VM NICs to the Load Balancer for traffic routing.

### Steps:

1. Go to **PublicLB** → **Backend pools**

2. Click **+ Add**

3. Name: BackendPool

4. Virtual Network: LBVNet

5. Add VM1 and VM2's NICs to the pool

6. Click **Add**

---

## 7 Create Health Probe

### Why?

Health probes determine which VMs are available to receive traffic.

### Steps:

1. Go to **PublicLB** → **Health Probes**
  2. Click **+ Add**
  3. Fill:
    - Name: HTTP-Probe
    - Protocol: HTTP
    - Port: 80
    - Interval: 5 seconds
    - Unhealthy threshold: 2
  4. Click **OK**
- 

## 8 Add Load Balancing Rule

### Why?

This rule maps the frontend port (80) to the backend pool and ensures load distribution.

### Steps:

1. Go to **PublicLB** → **Load balancing rules**
  2. Click **+ Add**
  3. Fill:
    - Name: LBRule
    - Frontend IP: PublicLB-IP
    - Protocol: TCP
    - Port: 80 → Backend Port: 80
    - Backend Pool: BackendPool
    - Health Probe: HTTP-Probe
    - Session Persistence: None
  4. Click **Add**
- 

## 9 Install Web Server on VMs

### Why?

To test the Load Balancer, you need an app (e.g., Apache/Nginx/IIS) listening on port 80.

### Steps (Ubuntu):

1. Connect via **Bastion** to **VM1** and run:

```
sudo apt update
sudo apt install apache2 -y
echo "<h1>welcome to VM1</h1>" | sudo tee /var/www/html/index.html
```

2. Repeat for **VM2**, change message:

```
echo "<h1>Welcome to VM2</h1>" | sudo tee /var/www/html/index.html
```

# 10 Test Load Balancer

- 1. Copy **Public IP** of `PublicLB`
- 2. Open in a browser
- 3. Refresh multiple times – you should see:
  - “Welcome to VM1”
  - “Welcome to VM2”

This confirms successful load balancing.

# 11 Optional: Configure NAT Gateway

## Why?

NAT Gateway ensures **secure, scalable outbound internet** access from private VMs.

## Steps:

- 1. Search **"NAT Gateway"** → Click **+ Create**
- 2. Basics:
  - Name: `NATGW`
  - Region: Same
  - Public IP: Create new → `NATGW-IP`
- 3. Subnet:
  - Attach to: `NATGatewaySubnet`
- 4. **Create**

## Associate NAT Gateway:

- Go to `BackendSubnet` → Select **NAT Gateway:** `NATGW`

# Final Setup Summary

Component	Purpose
VNet + Subnets	Network separation
2 VMs	Backend servers
Azure Bastion	Secure remote access
NAT Gateway	Outbound internet without public IPs

Public Load Balancer	Distributes traffic across backend VMs
Health Probe	Checks backend health
LB Rule	Routes incoming HTTP traffic to backend pool

---