**NGINX Proxy Load Balancer Configuration Guide**

1. [Setup Environment](#_Setup_Environment)
2. [Install Application on Backend Servers](#_Install_Application_on)
3. [Install NGINX on the Load Balancer Server](#_Install_NGINX_on)
4. [Configure NGINX Load Balancer](#_Configure_NGINX_Load)
5. [Validate and Restart NGINX](#_Validate_and_Restart)
6. [Test Load Balancer](#_Test_Load_Balancer)

### Setup Environment

For NGINX Proxy Load Balancer setup, we need three servers:

* **1 NGINX Load Balancer server** (Public subnet) – This server will act as the load balancer, distributing traffic to the backend application servers.
* **2 Application servers** (Private subnet) – These servers will host the actual application and respond to client requests.

**Network Setup Considerations:**

* The Load Balancer server must be in a **public subnet** so it can receive external traffic.
* The Application servers can be in a **private subnet** for security, as they will only be accessed by the Load Balancer.

### Install Application on Backend Servers

Each application server needs a web server to serve content.

**Step 1: Install httpd on Application Servers**

sudo yum install httpd -y

sudo systemctl enable httpd

sudo systemctl start httpd

This installs and starts the web server.

**Step 2: Create a Test Web Page on Each Application Server**

Each application server should serve a different response to verify load balancing.

**On Application Server 1:**

sudo vi /var/www/html/index.html

type, “**This is Private Server 1**”

save and exit.

**On Application Server 2:**

sudo vi /var/www/html/index.html

type, " **This is Private Server 2**"

save and exit.

**Step 3: Verify Web Server is Running**

Run the following command on the Load Balancer server:

curl http://<private-ip-of-app-server>

If successful, you should see "This is Private Server 1" or "This is Private Server 2" as the output.

### Install NGINX on the Load Balancer Server

The Load Balancer server will use NGINX to distribute traffic.

**Step 1: Install NGINX**

sudo yum install nginx -y

sudo systemctl enable nginx

sudo systemctl start nginx

**Step 2: Verify Installation**

Check if NGINX is running:

systemctl status nginx

If the output shows "active (running)," NGINX is installed successfully.

### Configure NGINX Load Balancer

We need to configure NGINX to forward traffic to our application servers.

**Step 1: Create a Load Balancer Configuration File**

sudo vi /etc/nginx/conf.d/loadbalancer.conf

**Step 2: Add the Following Configuration**

Replace the private IPs mentioned below with the corresponding private server IPs.

upstream backend\_servers {

server 172.31.45.10; # Backend Server 1 (Private IP)

server 172.31.45.11; # Backend Server 2 (Private IP)

}

server {

listen 80;

location / {

proxy\_pass http://backend\_servers;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Forwarded-Proto $scheme;

}

}

**Step 3: Save and Exit**

Press ESC, type :wq!, and hit Enter to save the file and exit.

### Validate and Restart NGINX

**Step 1: Check for Configuration Errors**

sudo nginx -t

If there are no syntax errors, it will display.

**Step 2: Restart NGINX**

sudo systemctl reload nginx

This applies the new configuration without stopping the service.

### Test Load Balancer

To verify that the Load Balancer is working, open a web browser and enter:

http://<Load-Balancer-Public-IP>

**Expected Behaviour:**

* Refreshing the page multiple times should alternate responses between "This is Server 1" and "This is Server 2."

# NGINX Proxy Load Balancer Configuration Guide

This guide provides step-by-step instructions to configure an NGINX Load Balancer to distribute traffic between two backend application servers.

### ✅ Step 1: Setup Environment

You need **three EC2 instances**:

1. **NGINX Load Balancer Server** (in a **Public Subnet**) — to receive and route traffic.
2. **Application Server 1** (in a **Private Subnet**) — to serve content.
3. **Application Server 2** (in a **Private Subnet**) — to serve content.

#### 🔧 Network Setup:

* **Public Subnet** for the Load Balancer: This allows it to receive external HTTP traffic from the internet.
* **Private Subnet** for Application Servers: These are not exposed to the public internet, increasing security.
* All instances should be in the **same VPC** to allow communication between Load Balancer and backend servers.

### 🚀 Creating EC2 Instances

#### 🛠 Create Load Balancer Server (Public Subnet)

1. Go to **EC2 → Launch Instance**.
2. Choose **Amazon Linux 2 AMI**.
3. Instance Type: **t2.micro** (Free Tier).
4. Configure Instance:
   * Subnet: Select your **Public Subnet**.
   * Auto-assign Public IP: **Enable**.
5. Add Storage: Keep default (8 GB).
6. Security Group:
   * Allow:
     + **HTTP (port 80)** from 0.0.0.0/0
     + **SSH (port 22)** from your IP
7. Launch with key pair.
8. Name it: nginx-load-balancer.

#### 🛠 Create Application Server (Private Subnet)

1. Go to **EC2 → Launch Instance**.
2. Choose **Amazon Linux 2 AMI**.
3. Instance Type: **t2.micro** (Free Tier).
4. Configure Instance:
   * Subnet: Select your **Private Subnet**.
   * Auto-assign Public IP: **Disable**.
5. Add Storage: Keep default.
6. Security Group:
   * Allow:
     + **HTTP (port 80)** from **Security Group of Load Balancer**
     + **SSH (port 22)** from Load Balancer’s Security Group or Bastion Host
7. Launch with key pair.
8. Name it: app-server-private.

## ✅ Step 2: Install Application on Backend Servers

### On each application server:

#### Step 1: Install HTTP Server

sudo yum install httpd -y

sudo systemctl enable httpd

sudo systemctl start httpd

#### Step 2: Create a Test Web Page

**On Application Server 1:**

sudo vi /var/www/html/index.html

Add the following:

This is Private Server 1

**On Application Server 2:**

sudo vi /var/www/html/index.html

Add the following:

This is Private Server 2

#### Step 3: Verify Web Servers

Run this from the Load Balancer instance:

curl http://<private-ip-of-app-server>

Expected Output: This is Private Server 1 or This is Private Server 2

## ✅ Step 3: Install NGINX on the Load Balancer Server

### Step 1: Install NGINX

sudo yum install nginx -y

sudo systemctl enable nginx

sudo systemctl start nginx

### Step 2: Verify NGINX is Running

systemctl status nginx

Look for: active (running)

## ✅ Step 4: Configure NGINX Load Balancer

### Step 1: Create Load Balancer Configuration File

sudo vi /etc/nginx/conf.d/loadbalancer.conf

### Step 2: Add Load Balancing Configuration

Replace IPs with your actual backend private IPs.

upstream backend\_servers {

server 172.31.45.10; # Backend Server 1

server 172.31.45.11; # Backend Server 2

}

server {

listen 80;

location / {

proxy\_pass http://backend\_servers;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Forwarded-Proto $scheme;

}

}

### Step 3: Save and Exit

Press ESC, then type :wq! and press Enter.

## ✅ Step 5: Validate and Restart NGINX

### Step 1: Check NGINX Configuration

sudo nginx -t

Expected Output: syntax is ok, test is successful

### Step 2: Reload NGINX

sudo systemctl reload nginx

## ✅ Step 6: Test the Load Balancer

In your browser, go to:

http://<Public-IP-of-Load-Balancer>

### Expected Behavior:

* On refreshing the page multiple times, the response alternates between:
  + This is Private Server 1
  + This is Private Server 2

## ✅ Summary

You have successfully:

* Set up two private backend servers
* Installed and configured NGINX on a public load balancer
* Load-balanced traffic using round-robin between two private web servers

✅ **Your Load Balancer is now working!**