NGINX Proxy Load Balancer Configuration Guide

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!**

Troubleshooting

✅ Fix Steps: Disable Default Conf and Enable Load Balancer

1. 🚫 Remove or Rename the Default Config

```
bash
CopyEdit
sudo mv /etc/nginx/conf.d/default.conf /etc/nginx/conf.d/default.conf.bak
```

2. ✅ Verify Your Load Balancer Config

Double-check your file:

```
bash
CopyEdit
sudo cat /etc/nginx/conf.d/loadbalancer.conf
```

Ensure it looks like this (replace IPs with your private app server IPs):

```
nginx
CopyEdit
upstream backend_servers {
    server 172.31.45.10;  # App Server 1
    server 172.31.45.11;  # App 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;
    }
}
```

3. ✅ Test NGINX Configuration

```
bash
CopyEdit
sudo nginx -t
```

You should see:

```
bash
CopyEdit
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

4. 🔄 Reload NGINX

```
bash
CopyEdit
sudo systemctl reload nginx
```

---

5. ⍰ Test It in Your Browser

Now go to:

```
cpp
CopyEdit
http://44.203.40.192
```

🔁 **Refresh the page multiple times** — you should see:

- This is Private Server 1
- This is Private Server 2

alternating, if both backend servers are configured correctly.

If it **still doesn't alternate or load the app servers**, run:

```
bash
CopyEdit
curl http://172.31.45.10
curl http://172.31.45.11
```

from the **Load Balancer EC2**, and share the output so I can check connectivity.

Let me know how it goes!