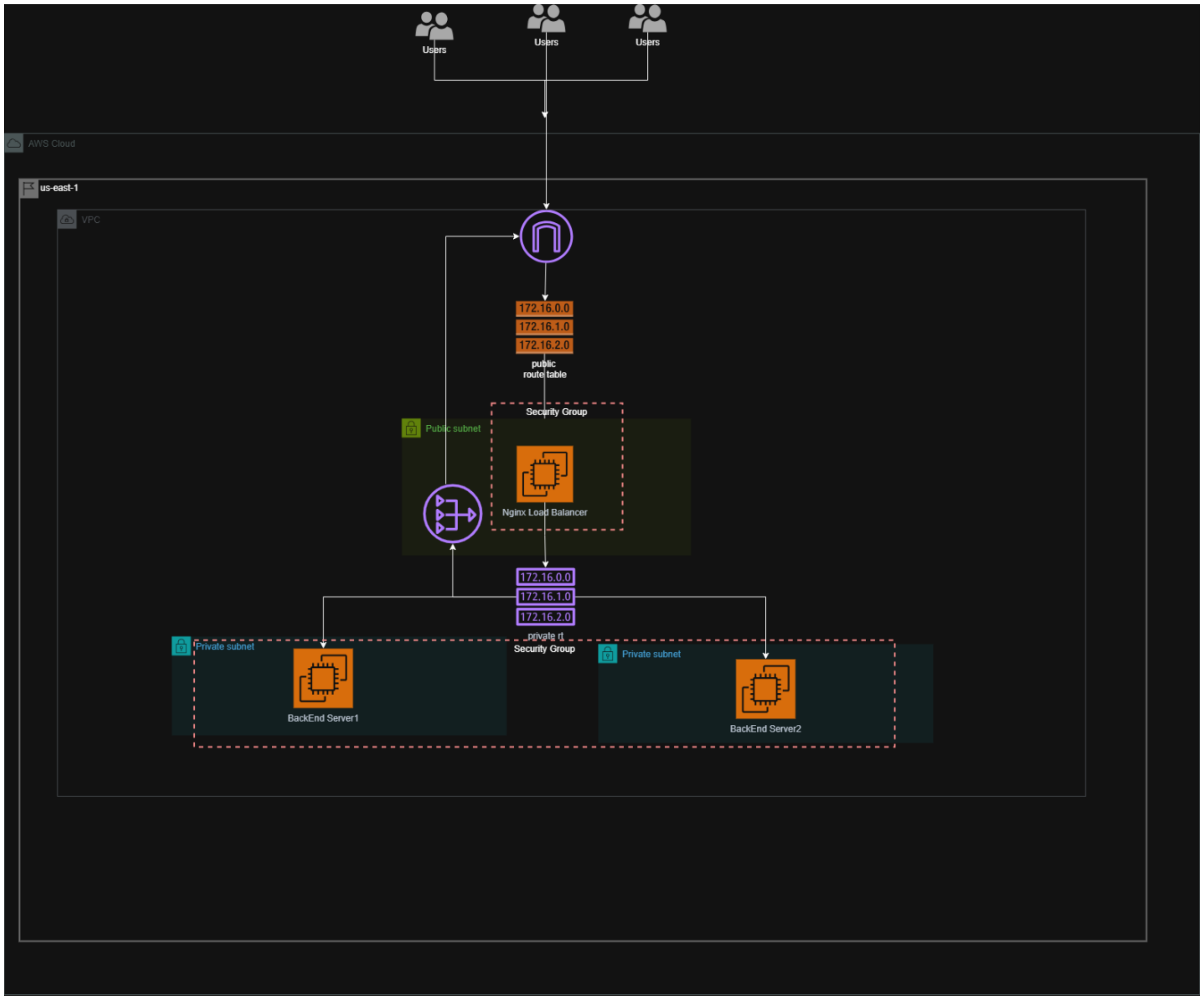
NGINX Load Balancer Setup on AWS EC2

Sardar Noor Ul Hassan (Cloudelligent)

1. Overview

This document explains how I deployed an NGINX-based load balancer on AWS to distribute traffic between two backend EC2 instances.
Both backend servers run in private subnets, and the load balancer runs in a public subnet.
Traffic flows from the internet → NGINX LB → backend servers.



2. Prerequisites

- AWS account with permissions for EC2, VPC, and networking
- Basic understanding of Linux commands
- SSH key pair created and downloaded
- Custom VPC already configured with:
    - Public Subnet
    - Two Private Subnets
    - NAT Gateway
    - Internet Gateway
    - Security Groups for LB and backend

(VPC setup was already completed earlier.)

3. Backend EC2 Instances Setup

3.1 Launch Backend Server 1

- AMI: Ubuntu
- Instance Type: t3.micro
- Subnet: Private Subnet 1
- Auto-assign Public IP: Disabled
- Security Group: backend-servers-sg
- Key Pair: noor-key

3.2 Launch Backend Server 2

- AMI: Ubuntu
- Instance Type: t3.micro
- Subnet: Private Subnet 2
- Auto-assign Public IP: Disabled
- Security Group: backend-servers-sg
- Key Pair: noor-key

---

4. Install Web Server on Backend Instances

Since private EC2 instances cannot be accessed from the internet, I connected to them **through the Load Balancer EC2** using SSH.

Commands executed on each backend instance:

sudo apt update -y

sudo apt install apache2 -y

sudo systemctl enable apache2

sudo systemctl start apache2

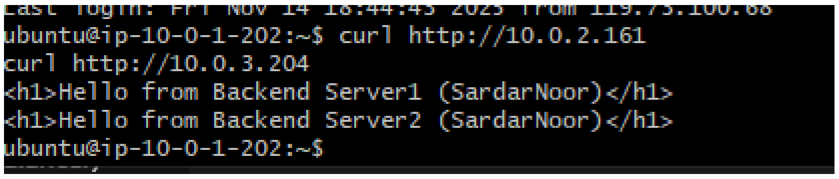
echo "<h1>Hello from Backend Server1 (SardarNoor)</h1>" | sudo tee /var/www/html/index.html

For the second backend server, I replaced the message with:

Hello from Backend Server2 (SardarNoor)

Both backend servers were tested from the LB using:

curl http://



```
Last login: Fri Nov 14 18:44:43 2025 from 119.73.100.68
ubuntu@ip-10-0-1-202:~$ curl http://10.0.2.161
curl http://10.0.3.204
<h1>Hello from Backend Server1 (SardarNoor)</h1>
<h1>Hello from Backend Server2 (SardarNoor)</h1>
ubuntu@ip-10-0-1-202:~$
```

Both returned the expected HTML output.

5. Load Balancer EC2 Setup

5.1 Launch LB Instance

- AMI: Ubuntu
- Instance Type: t2.micro
- Subnet: Public Subnet
- Auto-assign Public IP: Enabled
- Security Group: lb-sg
    - HTTP allowed from anywhere (0.0.0.0/0)

- SSH allowed from My IP

5.2 Install NGINX on LB Instance

```
sudo apt update -y

sudo apt install nginx -y

sudo systemctl enable nginx

sudo systemctl start nginx
```

NGINX service confirmed running: sudo systemctl status nginx

---

6. Configure NGINX for Load Balancing

I edited the main NGINX configuration file:

```
sudo nano /etc/nginx/nginx.conf
```

Under the **http {}** block, I added:

```
http {
  upstream backend_servers {
    server 10.0.2.161;
    server 10.0.3.204;
  }

  server {
    listen 80;

    location / {
      proxy_pass http://backend_servers;
    }
  }
```

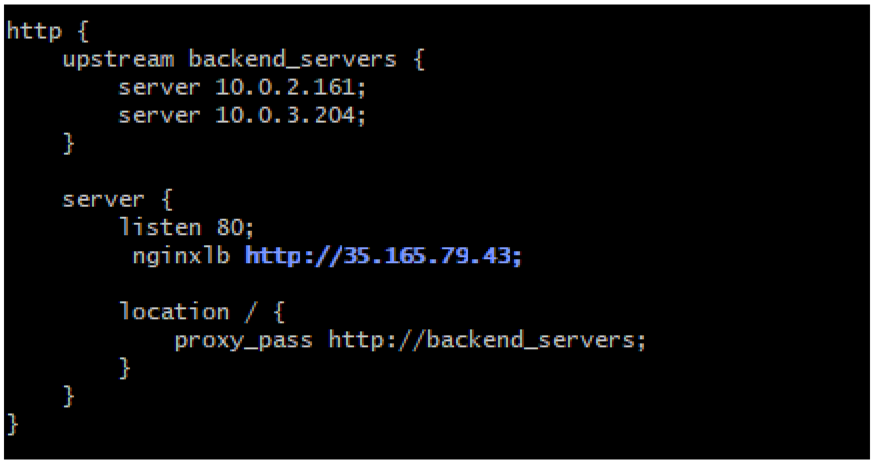
}

- 10.0.2.161 = Backend Server 1 (private IP)
- 10.0.3.204 = Backend Server 2 (private IP)

Then tested and restarted NGINX:

sudo nginx -t

sudo systemctl restart nginx



```
http {
    upstream backend_servers {
        server 10.0.2.161;
        server 10.0.3.204;
    }

    server {
        listen 80;
         nginxlb http://35.165.79.43;

        location / {
            proxy_pass http://backend_servers;
        }
    }
}
```

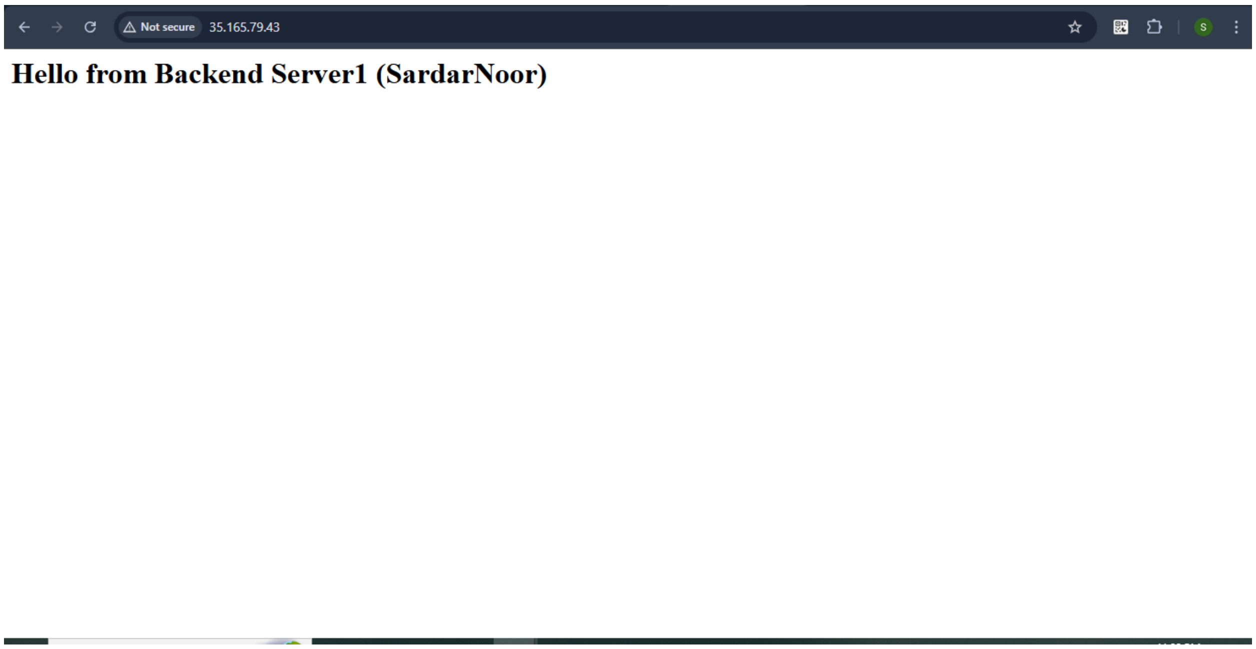
7. Testing the Load Balancer

I opened the LB’s public IP in the browser:

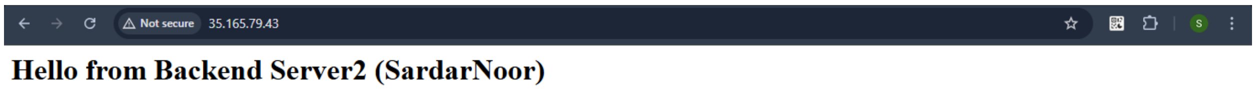
http://35.165.79.43

Then refreshed several times.

Expected behavior:


Not secure 35.165.79.43

Hello from Backend Server1 (SardarNoor)


Not secure 35.165.79.43

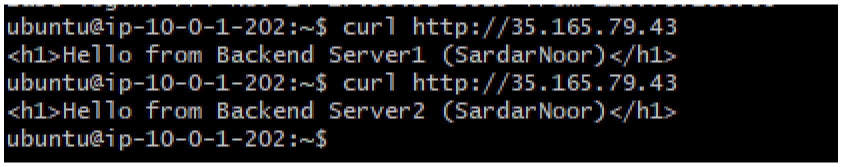
Hello from Backend Server2 (SardarNoor)

- One refresh shows: **“Hello from Backend Server1 (SardarNoor)”**
- Next refresh shows: **“Hello from Backend Server2 (SardarNoor)”**
- And so on...

This confirmed **round-robin load balancing** is working correctly.

I also tested with:

```
curl http://35.165.79.43
```



```
ubuntu@ip-10-0-1-202:~$ curl http://35.165.79.43
<h1>Hello from Backend Server1 (SardarNoor)</h1>
ubuntu@ip-10-0-1-202:~$ curl http://35.165.79.43
<h1>Hello from Backend Server2 (SardarNoor)</h1>
ubuntu@ip-10-0-1-202:~$
```

Responses alternated between both backend servers.

---

8. Security Groups Summary

8.1 LB Security Group

Inbound:

- HTTP (80) → 0.0.0.0/0
- SSH (22) → My IP

Outbound:

- All traffic allowed

8.2 Backend Servers Security Group

Inbound:

- HTTP (80) → Source: lb-sg
- SSH (22) → My IP (optional; used only for LB → backend SSH testing)

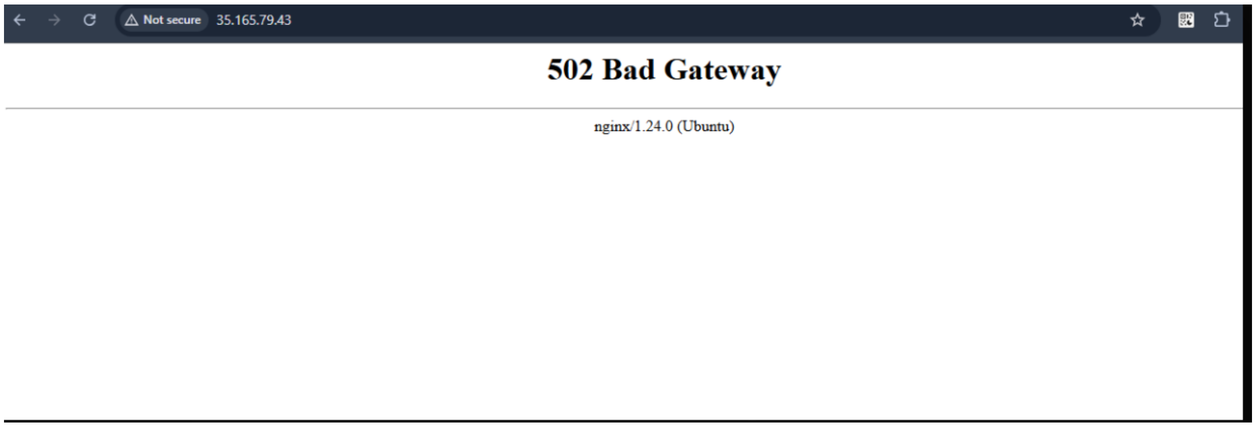
Outbound:

- All traffic allowed

11. Troubleshooting

During the setup, I encountered multiple issues while configuring the NGINX load balancer.

11.1 Issue: 502 Bad Gateway on Load Balancer



35.165.79.43

502 Bad Gateway

nginx/1.24.0 (Ubuntu)

Symptoms

- Browser showed 502 Bad Gateway
- curl: (7) Failed to connect

Cause

NGINX was running, but the backend servers were **not responding on port 80**.
Apache was not installed/started on Backend Server 1, so NGINX had no valid upstream target.

Solution

I connected to backend servers via the LB instance and manually installed & started Apache:

sudo apt update -y

sudo apt install apache2 -y

sudo systemctl enable apache2

sudo systemctl start apache2

echo "<h1>Hello from Backend ServerX (SardarNoor)</h1>" | sudo tee /var/www/html/index.html

After installing Apache, the backend servers successfully responded:

curl http://10.0.2.161

curl http://10.0.3.204

502 error resolved.

11.2 Issue: Cannot SSH into Backend Servers

Symptoms

- Direct SSH from my laptop failed (Connection timed out)
- LB → backend SSH failed with:
- Permission denied (publickey)

Cause

Backend servers are in **private subnets** → cannot SSH directly.
Also, the LB instance did not have the key pair (noor-key.pem) required for backend SSH.

Solution

1. Copied key to LB instance using SCP from laptop:

```
scp -i noor-key.pem noor-key.pem ubuntu@35.165.79.43:/home/ubuntu/
```

2. Fixed permissions on LB:

```
chmod 400 noor-key.pem
```

3. Added SSH rule in backend SG to allow SSH from LB-SG
4. Connected successfully:

```
ssh -i noor-key.pem ubuntu@10.0.2.161
```

```
ssh -i noor-key.pem ubuntu@10.0.3.204
```

SSH working → allowed backend debugging.

11.3 Issue: User Data Did Not Execute on Backend Servers

Symptoms

- Apache not installed
- /var/www/html/index.html missing
- systemctl status apache2 returned:
- Unit apache2.service could not be found

Cause

User-data script was missing the required shebang line:

```
#!/bin/bash
```

Also, user-data does NOT run on reboot ,it only runs on the first boot or after a full stop/start.

Solution

I corrected the script and applied it again using:

```
#!/bin/bash

apt update -y

apt install apache2 -y

...
```

Then performed a **Stop → Start** (not reboot) to trigger user-data.

11.4 Issue: Load Balancer Not Switching Between Servers (Only Server 2 Response)

Symptoms

- Browser always showed Backend Server 2
- No alternation between Server1 and Server2

Cause

Backend Server 1 was not serving content due to missing Apache installation.

Solution

After installing Apache on Backend Server 1 and verifying curl, NGINX started routing correctly:

Backend1 → Backend2 → Backend1 → Backend2

Round-robin balancing confirmed.

11.5 Issue: Initial Confusion with Route Tables

Symptoms

At one point, I suspected route tables were incorrect.

Cause

The backend servers were already using the correct private route table (NAT Gateway).
This was not the actual issue ,the real issue was unpaid user-data.