**To access your server running both frontend and backend services using NGINX and an AWS Network Load Balancer (NLB), you'll need to configure a few things. Below is an outline of the steps:**

**Set Up AWS Network Load Balancer (NLB)**

AWS Network Load Balancer operates at Layer 4 (Transport layer) and is best suited for routing traffic based on IP addresses and ports.

**Steps:**

1. **Create a Target Group:**
   * Go to the **EC2 Dashboard** > **Load Balancers** > **Target Groups**.
   * Click **Create target group**.
   * Set the target type as **Instance** if you're targeting an EC2 instance.
   * Select **TCP** as the protocol since NLB operates at Layer 4.
   * Choose the **port** where your NGINX is listening (e.g., port 80 for HTTP or 443 for HTTPS).
   * Register your EC2 instance(s) to this target group.
2. **Create a Network Load Balancer:**
   * Go to the **Load Balancers** section.
   * Click **Create Load Balancer** and choose **Network Load Balancer**.
   * Assign a name to your NLB.
   * For the scheme, choose **Internet-facing** (for public access) or **Internal** (for internal use only).
   * Add listeners for ports 80 (HTTP) and 443 (HTTPS).
   * Select the target group you created for backend services.
   * Configure health checks (default is TCP, but you can change it if needed).
3. **Set up DNS:**
   * After the NLB is created, associate your NLB with a DNS name by pointing your domain (using Route 53 or another DNS service) to the load balancer's DNS name.

**2. Install and Configure NGINX on Your EC2 Server**

NGINX will serve as the reverse proxy, routing traffic between the frontend and backend applications. Below are the steps for configuring NGINX.

#### Steps:

1. **Install NGINX** (if it's not already installed):

**sudo apt update**

**sudo apt install nginx**

**2.Configure NGINX for Reverse Proxy**: Assuming your frontend is running on port 3000 and backend (API) is running on port 5000, you can configure NGINX to route traffic accordingly.

Edit the NGINX configuration file, usually located at /etc/nginx/sites-available/default or create a new file at /etc/nginx/sites-available/myapp:

sudo nano /etc/nginx/sites-available/myapp

**Add the following configuration:**

server {

listen 80;

server\_name your-domain-name.com; # Replace with your domain or NLB DNS

# Frontend routing

location / {

proxy\_pass http://localhost:3000;

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;

}

# Backend API routing

location /api/ {

proxy\_pass http://localhost:5000; # Backend server

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;

}

# Additional settings like error pages, SSL certificates, etc. can go here.

}

 proxy\_pass: Forwards the request to the corresponding frontend or backend service.

 location /api/: This routes requests that start with /api/ to the backend (API).

**Enable the NGINX Configuration**: If you created a new config file, enable the site and restart NGINX:

sudo ln -s /etc/nginx/sites-available/myapp /etc/nginx/sites-enabled/

sudo nginx -t # Test the configuration

sudo systemctl reload nginx # Reload NGINX to apply changes

**3.Configure Security Group and Listener Rules**

Ensure your EC2 instance's **Security Group** allows traffic from the NLB on the appropriate ports (typically port 80 for HTTP or port 443 for HTTPS).

* For **NLB**: Ensure that the NLB is listening on the correct ports (80 or 443) and forwarding traffic to the NGINX server (running on port 80 or 443).
* For **EC2 Instance Security Group**: Ensure that the EC2 instance allows traffic from the NLB IP addresses on port 80 or 443.

### ****Optional: SSL Termination (HTTPS)****

If you want to use HTTPS for secure traffic, you'll need to set up SSL certificates on your NGINX server or terminate SSL on the NLB.

#### Option 1: SSL Termination on NGINX

1. **Obtain an SSL certificate** from a provider (or use Let's Encrypt).
2. Modify the NGINX configuration to use SSL:

server {

listen 443 ssl;

server\_name your-domain-name.com;

ssl\_certificate /etc/nginx/ssl/your-cert.crt;

ssl\_certificate\_key /etc/nginx/ssl/your-key.key;

location / {

proxy\_pass http://localhost:3000;

# Add other proxy settings as before

}

location /api/ {

proxy\_pass http://localhost:5000;

# Add other proxy settings as before

}

}

server {

listen 80;

server\_name your-domain-name.com;

return 301 https://$host$request\_uri; # Redirect HTTP to HTTPS

}

Restart NGINX after applying SSL configuration:

sudo systemctl reload nginx

**Option 2: SSL Termination on the Network Load Balancer**

* Attach an SSL certificate to the NLB listener for HTTPS (using ACM).
* Set up a listener rule to forward HTTPS requests to the backend EC2 instance on port 80

**5. Testing the Setup**

Once NGINX and the NLB are set up, test the following:

* Access the frontend via http://your-domain-name.com.
* Ensure that API requests like http://your-domain-name.com/api/ are routed correctly to the backend.

­­­­­