To achieve your goal of deploying your Docker-based APIs on a new server, configuring path-based routing using an **Application Load Balancer (ALB)**, and accessing both servers using a **Network Load Balancer (NLB)**, you can follow these steps:

**Steps to Configure Path-Based Routing with ALB and NLB:**

**1. Set Up Your New Server with Docker Compose and APIs**

**1.1 Install Docker and Docker Compose**

* On the new server, install Docker and Docker Compose if they aren't installed yet:

sudo apt update

sudo apt install docker.io

sudo apt install docker-compose

**1.2 Prepare the Docker Compose File**

* Your docker-compose.yml will define your 5 APIs (api1, api2, etc.) each with its Dockerfile.

**Example docker-compose.yml:**

version: '3'

services:

api1:

build: ./api1

ports:

- "8081:8080"

api2:

build: ./api2

ports:

- "8082:8080"

api3:

build: ./api3

ports:

- "8083:8080"

api4:

build: ./api4

ports:

- "8084:8080"

api5:

build: ./api5

ports:

- "8085:8080"

* Each API has its own Dockerfile (e.g., located in /api1/Dockerfile, /api2/Dockerfile, etc.).

1.3 **Start Docker Compose**

Run the following to start all the containers:

sudo docker-compose up -d

Your APIs should now be running on different ports on your server (8081, 8082, 8083, etc.).

### 2. ****Create an Application Load Balancer (ALB) for Path-Based Routing****

**2.1 Create a Target Group for Each API**

 Go to **EC2 Dashboard** > **Target Groups** > **Create Target Group**.

Create a target group for each API:

* **API 1**: Target group with instances or IP as the target type, listening on 8081.
* **API 2**: Target group listening on 8082.
* **API 3**: Target group listening on 8083.
* **API 4**: Target group listening on 8084.
* **API 5**: Target group listening on 8085.

2.2 **Create an ALB**

 Go to **Load Balancers** > **Create Load Balancer** > **Application Load Balancer**.

 Set the ALB as **Internet-facing** if you want public access.

 Add listeners on port 80 or 443 for HTTP/HTTPS.

**2.3 Configure Path-Based Routing**

* Under **Listeners** > **View/Edit Rules**, configure routing for each API:
  + For /api1/\*, forward requests to the API1 target group.
  + For /api2/\*, forward requests to the API2 target group.
  + Continue for all APIs.

**3. Set Up the Network Load Balancer (NLB) to Route to Multiple Servers**

You want the NLB to route traffic between your previous server and the new one with the ALB doing path-based routing.

**3.1 Create Target Groups for NLB**

* Go to **Target Groups** and create two new target groups:
  + **Target Group 1**: For your original server (EC2 instance running the old setup).
  + **Target Group 2**: For the **ALB** created for the new server.

**3.2 Create Network Load Balancer**

* Go to **Load Balancers** > **Create Load Balancer** > **Network Load Balancer**.
* Choose **Internet-facing** if public access is required.
* Add listeners on port 80 and/or 443.

**3.3 Configure Listener Rules for the NLB**

* Set up listener rules that define how traffic will be distributed:
  + **Path-Based Rules**: Forward traffic to the appropriate target group:
    - /old/\*: Forward to **Target Group 1** (your original server).
    - /new/\*: Forward to **Target Group 2** (your ALB, which will handle path-based routing for the new APIs).

**3.4 Configure DNS**

* Use **Route 53** or another DNS provider to point your domain name to the **NLB's DNS**.

**4. NGINX Setup on Each Server**

For your new server, ensure that **NGINX** is configured as a reverse proxy, especially if you're dealing with different services like frontend and backend.

Here’s an example of how to configure NGINX if you need to route between frontend and backend:

server {

listen 80;

server\_name your-new-domain.com;

# Frontend

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;

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;

}

}

Now you have:

* **Application Load Balancer (ALB)** for path-based routing on the new server.
* **Network Load Balancer (NLB)** for routing between the original server and the new server.

This setup allows you to handle traffic efficiently across multiple services and servers using both the **ALB** (for path-based routing) and **NLB** (for server load balancing).