🚀 AWS Elastic Beanstalk Deployment Guide

This guide details how to host `driversklub-backend` on **AWS Elastic Beanstalk (EB)** and set up automated deployments via **Bitbucket Pipelines**.

🏗️ Part 1: AWS Setup

1. Create an IAM User for Bitbucket

We need a user with permissions to upload code to Elastic Beanstalk.

1. Log in to **AWS Console** > Search for **IAM**.
2. Go to **Users** > **Create user**.
   - **User name**: `bitbucket-deployer`
3. **Permissions**:
   - Select **Attach policies directly**.
   - Search and check: `AWSElasticBeanstalkFullAccess` (For deployment).
   - search and check: `AdministratorAccess-AWSElasticBeanstalk` (Alternative if full access isn't found).
4. **Create User**.
5. **Security Credentials**:
   - Click on the new user > **Security credentials** tab.
   - **Create access key** > Select "Command Line Interface (CLI)".
   - **Copy/Download** the `Access Key ID` and `Secret Access Key`. **Save these!**

2. Create the Elastic Beanstalk Application

1. Go to **AWS Console** > Search for **Elastic Beanstalk**.
2. Click **Create application**.
   - **Application Name**: `driversklub-backend`
3. **Platform**:
   - **Platform**: `Node.js`
   - **Platform branch**: `Node.js 20 running on 64bit Amazon Linux 2023` (Recommended).
4. **Application Code**:
   - Select **Sample application** (We will overwrite it with the pipeline).
5. **Presets**:
   - Select **Single instance (free tier eligible)** for dev/staging.
   - Select **High availability** for production (Auto-scaling).
6. Click **Next** through configuration steps (Defaults are usually fine for now).
7. **Service Access**:
   - Create a new service role if asked (select "Create and use new service role").
   - **EC2 key pair**: Select an existing key pair (so you can SSH later if needed).
8. **Create Environment**.
   - Wait 5-10 mins.
   - Note the **Environment Name** (e.g., `Driversklub-env`).
   - Note the **Region** (e.g., `ap-south-1`).

3. Configure Environment Variables

1. In Elastic Beanstalk Environment Dashboard > **Configuration**.
2. Look for **Updates, monitoring, and logging** (or "Software"). click **Edit**.
3. Scroll to **Environment properties**.
4. Add your `.env` variables here:
   - `DATABASE_URL` : `postgresql://...` (Your RDS/DB connection)
   - `JWT_SECRET` : `...`
   - `NODE_ENV` : `production`
   - `PORT` : `8080` (EB usually expects 8080, Nginx maps port 80 to it. Important!).
5. Click **Apply**.

🗄️ Part 2: Database (RDS) Setup

Skip if using an existing external DB.

1. Go to **RDS** > **Create database**.
2. Select **PostgreSQL**.
3. **Template**: Free Tier (if applicable).
4. **Settings**:
   - **Master username/password**: Save these!
5. **Connectivity**:
   - **Public access**: Yes (if you need to connect from local PC), or No (if only EB needs access).
   - **VPC Security Group**: Create new "rds-sg".
6. **After Creation**:
   - Go to specific Security Group > Inbound Rules.
   - Allow **PostgreSQL (5432)** from **Anywhere** (0.0.0.0/0) or strictly from your EB Security Group.

🔄 Part 3: Bitbucket Pipeline Setup

1. Enable Pipelines

1. Go to your Bitbucket Repository.
2. **Settings** > **Pipelines** > **Settings** > Enable Pipelines.

2. Add Repository Variables

1. **Settings** > **Pipelines** > **Repository variables**.
2. Add the keys from **Part 1**:
   - `AWS_ACCESS_KEY_ID` : `AKIA...`
   - `AWS_SECRET_ACCESS_KEY` : `...`
   - `AWS_DEFAULT_REGION` : `ap-south-1` (Your EB region).

3. Verify `bitbucket-pipelines.yml`

Ensure the file references the correct names:

```
APPLICATION_NAME: "driversklub-backend"  # Must match Part 1.2
ENVIRONMENT_NAME: "Driversklub-env"      # Must match Part 1.2
ZIP_FILE: "deploy.zip"
```

✅ Part 4: Deploy

1. Commit and push your changes to `master`.
2. Go to **Bitbucket** > **Pipelines**.
3. Watch the build:
   - It will run tests.
   - It will build the `deploy.zip`.
   - It will upload to AWS EB.
4. Once green, go to your **EB URL** (e.g., `driversklub-env.eba-xxxx.ap-south-1.elasticbeanstalk.com/health`) to verify.

🐛 Troubleshooting

- **502 Bad Gateway**:
  - Check Logs: EB Dashboard > **Logs** > **Request Logs (Last 100 lines)**.
  - Common cause: App crashed or Port mismatch. Ensure `PORT` env var is handling traffic correctly.
- **Permission Error**:
  - Check IAM User permissions in Part 1.