**PIP Tracker Deployment Guide (Docker + Angular + Spring Boot + AWS RDS)**

**Step 1: Install Docker & Docker Compose on EC2**

# Update packages

sudo apt update && sudo apt upgrade -y

# Install Docker

sudo apt install -y docker.io

# Enable & start Docker

sudo systemctl enable docker

sudo systemctl start docker

# Add your user to Docker group (so no need for sudo later)

sudo usermod -aG docker $USER

newgrp docker

# Install Docker Compose

sudo apt install -y docker-compose

**Step 2: Prepare Project Structure**

In /home/ubuntu, create folders and move your files:

mkdir frontend backend

# Move frontend zip

mv PIP\_Tracker.zip frontend/

# Move backend jar

mv PipReviewSystem-0.0.1-SNAPSHOT.jar backend/

**Step 3: Setup Frontend (Angular + NGINX)**

cd frontend

unzip PIP\_Tracker.zip

mv 'PIP\_Tracker - Copy' pip

cd pip/dist/pip-tracker/browser

Create frontend/Dockerfile:

FROM nginx:alpine

# Copy Angular build output

COPY pip/dist/pip-tracker/browser /usr/share/nginx/html

# Set permissions

RUN chmod -R 755 /usr/share/nginx/html

# Copy Nginx config

COPY default.conf /etc/nginx/conf.d/default.conf

EXPOSE 80

CMD ["nginx", "-g", "daemon off;"]

**Create frontend/default.conf:**

server {

listen 80;

server\_name \_;

root /usr/share/nginx/html;

index index.html;

location / {

try\_files $uri $uri/ /index.html;

}

location /api/ {

proxy\_pass http://backend:8080;

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;

}

error\_page 404 /index.html;

location = /index.html {

allow all;

}

}

**Step 4: Setup Backend (Spring Boot)**

Create backend/Dockerfile:

FROM openjdk:17-jdk-slim

WORKDIR /app

COPY PipReviewSystem-0.0.1-SNAPSHOT.jar app.jar

EXPOSE 8080

CMD ["java", "-jar", "app.jar"]

**Step 5: Setup Database (AWS RDS MySQL)**

We’ll use **AWS RDS** instead of a local MySQL container.

**Step 6: Create RDS MySQL Instance**

**1. Log in to AWS Console**

Go to <https://console.aws.amazon.com>  
Search for **RDS** and click it.

**2. Create a New Database**

* Click **“Create database”**
* Choose:
  + Engine: **MySQL**
  + Version: **8.0**
  + Template: **Free tier** or **Production**

**3. Set Basic Settings**

* DB identifier: pipdb
* Master username: admin
* Master password: adminadmin

**4. Choose Instance Size**

* For testing: db.t3.micro
* For production: db.t3.medium or higher

**5. Storage Settings**

* Leave default (20 GB)

**6. Connectivity Settings**

* Connect to EC2 compute resource: Yes
  + Select your EC2 instance (e.g., i-005b750275920610e)
* VPC: Auto-selected
* DB Subnet Group: Auto-created
* Public access: No
* Security groups: Auto-attached to EC2 and RDS

**7. Additional Settings**

* Database name: pipdb
* Leave other options default

**8. Create Database**

* Click **“Create database”**
* Wait 2–5 minutes

**9. Get RDS Endpoint**

* Go to **RDS > Databases > pipdb**
* Copy the **Endpoint** (e.g., pipdb.abc123xyz.ap-south-1.rds.amazonaws.com)
* Confirm port is 3306

**Step 7: Create docker-compose.yml**

In /home/ubuntu, create:

version: "3.8"

services:

backend:

build: ./backend

container\_name: backend

restart: always

environment:

SPRING\_DATASOURCE\_URL: jdbc:mysql://***aws-rds-endpoint***:3306/pipdb

SPRING\_DATASOURCE\_USERNAME: admin

SPRING\_DATASOURCE\_PASSWORD: adminadmin

ports:

- "8080:8080"

frontend:

build: ./frontend

container\_name: frontend

restart: always

ports:

- "80:80"

depends\_on:

- backend

Replace the RDS endpoint with your actual value.

**Step 8: Build & Run**

cd /home/ubuntu

docker-compose up -d --build

docker ps -a

**Step 9: Access the Application**

* **Frontend (Angular)** → http://<EC2-Public-IP>
* **Backend (Spring Boot)** → http://<EC2-Public-IP>:8080
* **MySQL (via EC2)** →
* mysql -h ***aws-rds-endpoint*** -u admin -p

**Step 10: Test the Database**

Once inside MySQL shell:

USE pipdb;

SHOW TABLES;

SELECT \* FROM table-name;

You’ll see tables created by Spring Boot if everything is wired correctly.