

Overview

This document provides instructions for deploying the **Payment Application** using Docker and Docker Compose. It covers both **local development** and **production deployment**.

Prerequisites

Before proceeding, ensure the following tools are installed on your system: - **Docker** ([Installation Guide](#)) - **Docker Compose** ([Installation Guide](#)) - **Node.js** (for local development, optional)

Project Structure

The project uses the following key files: - `Dockerfile`: Defines the Docker image for the application. - `docker-compose.local.yml`: For local development. - `docker-compose.yml`: For production deployment.

Dockerfile

The Dockerfile is optimized for performance and security. Here's the content:

```
# Stage 1: Build the application
FROM node:20-alpine AS build

WORKDIR /app

# Copy package files
COPY package*.json ./

# Install dependencies
RUN npm ci --verbose

# Copy the rest of the application code
COPY . .

# Build the application (if needed)
RUN npm run build

# Stage 2: Create a lightweight production image
```

FROM node:20-alpine **AS** release

WORKDIR /app

Copy only the necessary files from the build stage

COPY --from=build /app/package*.json ./

COPY --from=build /app/dist ./dist

Install only production dependencies

RUN npm ci --only=production --verbose

Set the user to a non-root user for security

USER node

Expose the application port

EXPOSE 3000

Start the application

ENTRYPOINT ["node", "dist/main.js"]

Local Development Setup

docker-compose -f docker-compose.local.yml up -d

Access the Application

- **Base URL:** `http://localhost:3000`
- **Swagger:** `http://localhost:3000/swagger`

Production Deployment

1. Build the Docker Images

Run the following command to build the production Docker images:

docker-compose -f docker-compose.yml build

2. Start the Application

Start the application in detached mode:

docker-compose -f docker-compose.yml up -d

Docker Compose Files

docker-compose.local.yml (Local Development)

```
services:
  back:
    image: node:20-alpine
    working_dir: /app
    volumes:
      - ./:/app
    ports:
      - 3000:3000
    depends_on:
      - mongo
    environment:
      - MONGO_URI=mongodb://mongo:27017/payment
      - API_PORT=3000
    command: npm run start:dev

  mongo:
    image: mongo
    ports:
      - 27017:27017
    volumes:
      - mongo:/data/db

volumes:
  mongo:
```