















**Ensure breakpoints are inside an executable line**  
2️⃣ **Modify Dockerfile to enable debugging (**-agentlib:jdwp...**)**  
3️⃣ **Rebuild JAR and restart Docker container**  
4️⃣ **Attach IntelliJ Debugger to port** 5005  
5️⃣ **Trigger an API request to** /api/test?

Docker basics

* Docker -platform for building ,running and shipping applications.

Commands

* Docker version - gives version of the docker

Dockerize

DockerFile - instructions that docker uses to package an application into an image.appended to an application.

Container - application is loaded into an container

Docker run … in a container,image is loaded in docker run

Package application into an image and run it anywhere.

Create a dockerfile

FROM image e.g openjdk:22

COPY . /app

WORKDIR /app

CMD openjdk

Docker package our application

**Docker build -t hello-docker . (period is used to reference the current image)**

To view docker images

**Docker image ls**

Each image has a unique identifier

Run image on any machine

Docker run imagename

Package managers

In linux we use apt

apt update (updates all packages in linux)

apt install nano (installs a package)

Apt remove nano (removes a package)

Docker container remote debugging

FROM openjdk:22  
#ARG JAR\_FILE=target/\*.jar  
WORKDIR /app  
COPY ./target/TestDocker-0.0.1-SNAPSHOT.jar app.jar  
  
EXPOSE 8080 5005  
  
CMD ["java", "-agentlib:jdwp=transport=dt\_socket,server=y,suspend=n,address=\*:5005", "-jar", "app.jar"]

### ****Docker Compose (****docker-compose.yml****) for a Spring Boot Application****

**Docker Compose** helps you define and run multi-container applications using a simple YAML file. Below is a docker-compose.yml file for a **Spring Boot application** with a **PostgreSQL database**.

## ****1️⃣ Basic**** docker-compose.yml ****for Spring Boot****

This setup runs a **Spring Boot app** and a **PostgreSQL database** in separate containers.

### ****📄**** docker-compose.yml

yaml

CopyEdit

version: "3.8"

services:

app:

image: spring-boot-app # Change to your built image

build: .

ports:

- "8080:8080"

environment:

SPRING\_DATASOURCE\_URL: jdbc:postgresql://db:5432/mydb

SPRING\_DATASOURCE\_USERNAME: user

SPRING\_DATASOURCE\_PASSWORD: password

depends\_on:

- db

db:

image: postgres:15

restart: always

environment:

POSTGRES\_USER: user

POSTGRES\_PASSWORD: password

POSTGRES\_DB: mydb

ports:

- "5432:5432"

## ****2️⃣ Steps to Run the Application****

### ****🔹 Step 1: Ensure the Spring Boot Application is Containerized****

Make sure your **Dockerfile** exists in the project root. If not, create one:

dockerfile

CopyEdit

FROM openjdk:17-jdk-slim

WORKDIR /app

COPY target/demo-0.0.1-SNAPSHOT.jar app.jar

EXPOSE 8080

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

### ****🔹 Step 2: Build and Start the Containers****

Run the following command in the directory where docker-compose.yml is located:

sh

CopyEdit

docker-compose up -d

-d: Runs in detached mode.

### ****🔹 Step 3: Verify Containers are Running****

sh

CopyEdit

docker ps

The **Spring Boot app** should be running on http://localhost:8080

The **PostgreSQL database** should be running on port 5432.

## ****3️⃣ Additional Features****

### ****🔹 Adding Remote Debugging Support****

Modify the **Spring Boot service** in docker-compose.yml to enable debugging:

yaml

CopyEdit

app:

build: .

ports:

- "8080:8080"

- "5005:5005"

environment:

JAVA\_TOOL\_OPTIONS: "-agentlib:jdwp=transport=dt\_socket,server=y,suspend=n,address=\*:5005"

Now you can attach a **remote debugger** on port 5005.

### ****🔹 Auto-restart on Failure****

yaml

CopyEdit

restart: always

This ensures that containers restart if they crash.

### ****🔹 Running Logs****

To check logs for all services:

sh

CopyEdit

docker-compose logs -f

## ****4️⃣ Stop and Cleanup****

### ****🔹 Stop Running Containers****

sh

CopyEdit

docker-compose down

### ****🔹 Stop and Remove All Containers, Volumes, and Networks****

sh

CopyEdit

docker-compose down -v

## ****🎯 Summary****

✅ **Easily run Spring Boot + PostgreSQL with Docker Compose**  
✅ **Supports remote debugging**  
✅ **Automatically restarts on failure**  
✅ **Easy logs and cleanup**