Build container

docker run –p localport:machineport image id

Run container

docker start container Id/ container name

1. Build project Maven

- mvn clean package -> target/... .jar

2. Build project Gradle

- ./gradle build -> build/.... .jar

3. Lệnh để chạy ứng dụng Spring boot

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

4. docker proccess -a

docker ps –a

5. Chạy lệnh để build dự án gradle

RUN ./gradlew build --no-daemon

6. Build dự án bằng Maven

RUN mvn package -DskipTests

7. Chạy lệnh Maven để tải xuống các dependencies cần thiết

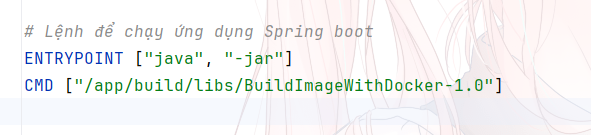
RUN mvn dependency:resolve

**Tương Tác Giữa ENTRYPOINT và CMD**

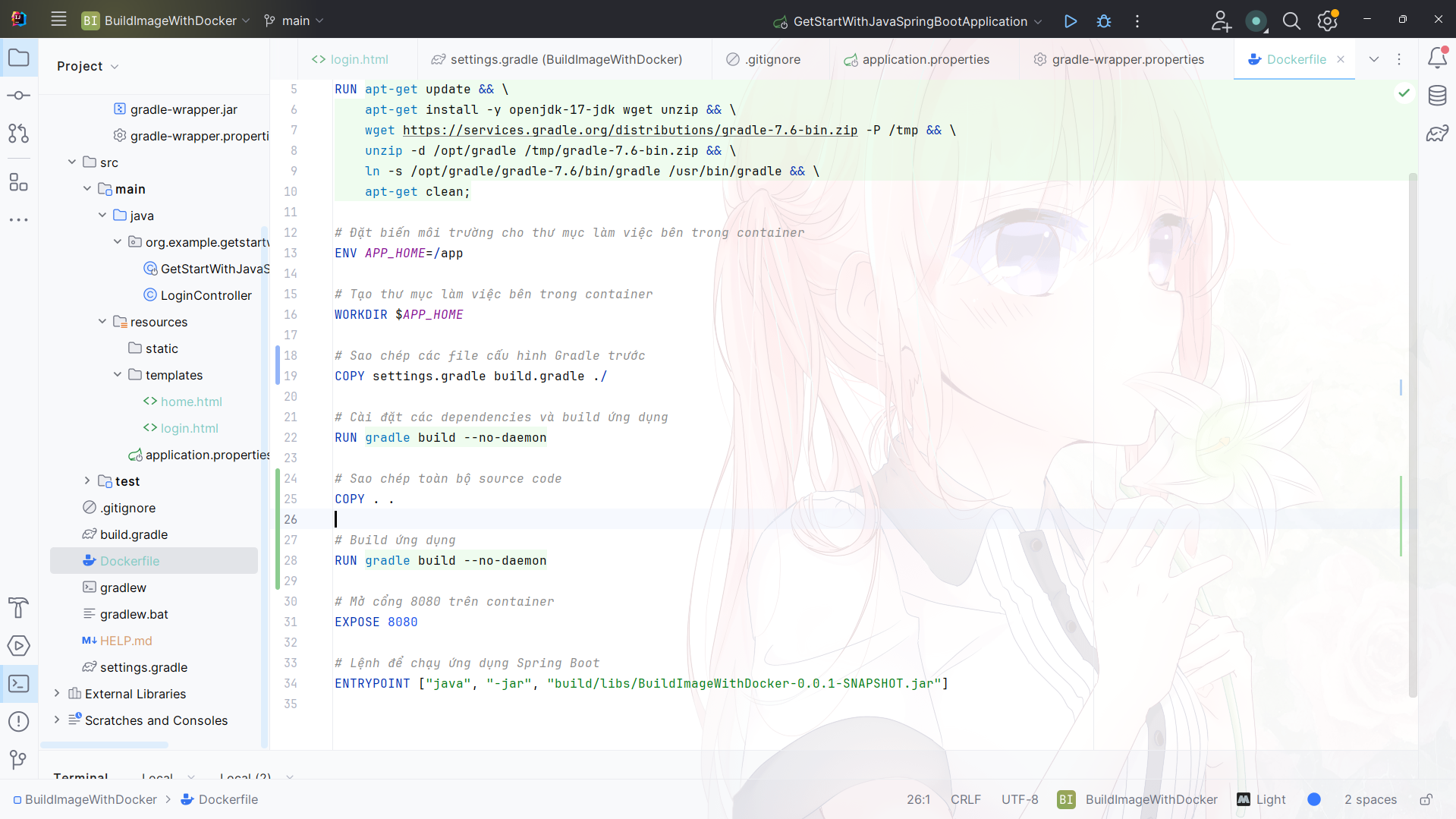
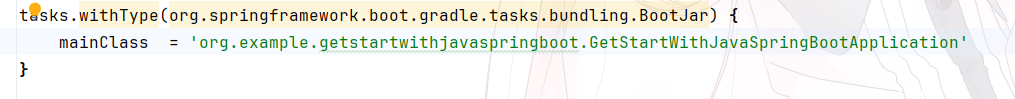
1. **ENTRYPOINT**:
   * Xác định lệnh chính mà Docker sẽ chạy khi container khởi động.
   * Lệnh trong ENTRYPOINT luôn được thực thi.
2. **CMD**:
   * Cung cấp các tham số mặc định cho lệnh trong ENTRYPOINT.
   * Nếu không có lệnh nào được cung cấp khi khởi động container, CMD cung cấp lệnh hoặc tham số mặc định.

### Cách CMD và ENTRYPOINT Tương Tác

* **Nếu cả ENTRYPOINT và CMD đều được chỉ định**:
  + ENTRYPOINT là lệnh chính và sẽ luôn được thực thi.
  + CMD cung cấp các tham số mặc định cho lệnh trong ENTRYPOINT.
  + Kết hợp cả 2



**TẠO 1 IMAGE VỚI GRADLE**

1. ****
2. **Trong build gradle cần có bootJar**
3. **Chạy lệnh gradle build .**

**XÓA CÁC VOLUME VÀ NETORK KHÔNG CÒN SỬ DỤNG**

1. **docker volume ls**
2. **docker volume prune**
3. **docker network ls**
4. **docker network prune**

**Attached & Detached**

* When using the “run” command -> attached mode is default
* When using the “start” command -> detached mode is default

Attached: listening to the out put of that container

We can config to start/ run in our mode

* + docker start –a/-d container-id/container-name
  + docker run –a/-d container-id/container-name

We can run it in attatch mode with:

* + docker attach container-id/container-name

Or we can read the log of detached mode with:

* + docker logs container-id/container-name ( we can see what was print )
  + docker logs –f container-id/container-name ( we can see and listening )