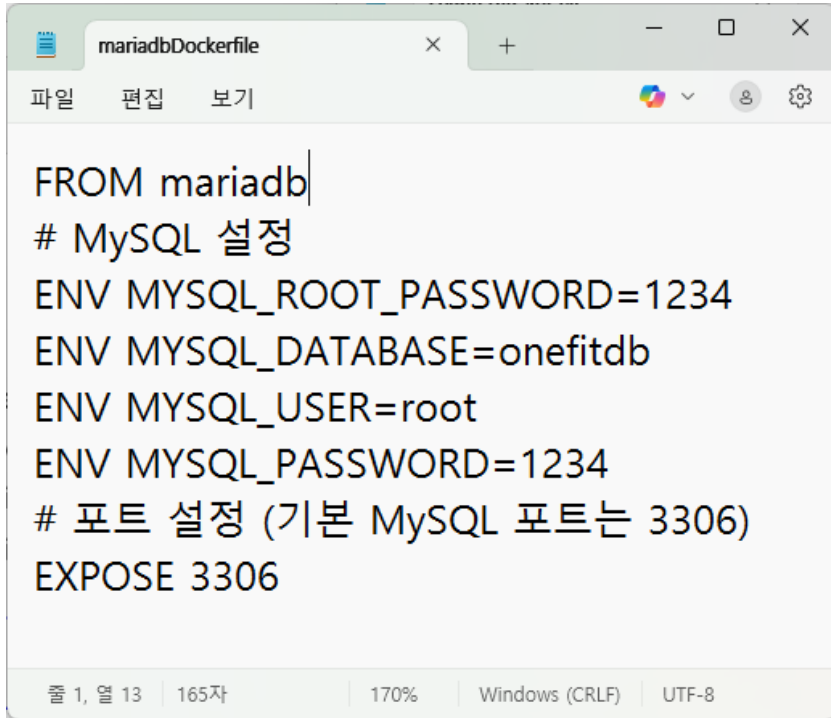


## \* 프로젝트 도커 컨테이너로 배포하기

A screenshot of a code editor window titled 'mariadbDockerfile'. The editor shows the following text: FROM mariadb, # MySQL 설정, ENV MYSQL\_ROOT\_PASSWORD=1234, ENV MYSQL\_DATABASE=onefitdb, ENV MYSQL\_USER=root, ENV MYSQL\_PASSWORD=1234, # 포트 설정 (기본 MySQL 포트는 3306), EXPOSE 3306. The status bar at the bottom indicates '줄 1, 열 13', '165자', '170%', 'Windows (CRLF)', and 'UTF-8'.

```
FROM mariadb
# MySQL 설정
ENV MYSQL_ROOT_PASSWORD=1234
ENV MYSQL_DATABASE=onefitdb
ENV MYSQL_USER=root
ENV MYSQL_PASSWORD=1234
# 포트 설정 (기본 MySQL 포트는 3306)
EXPOSE 3306
```

docker network create mysql-net

docker build -f mariadbDockerfile -t buildmariadb .

docker image ls

docker volume create mysql-volume

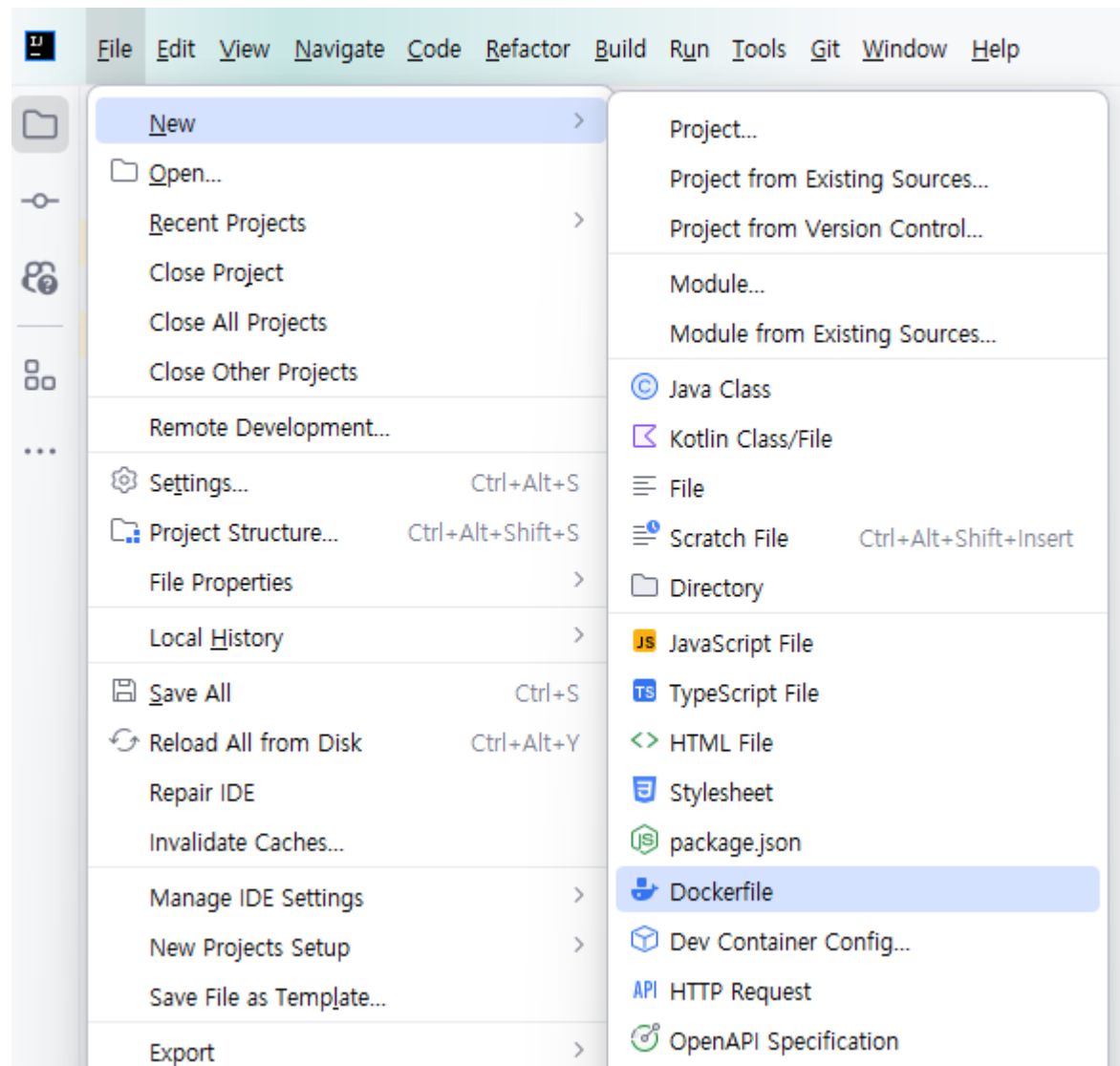
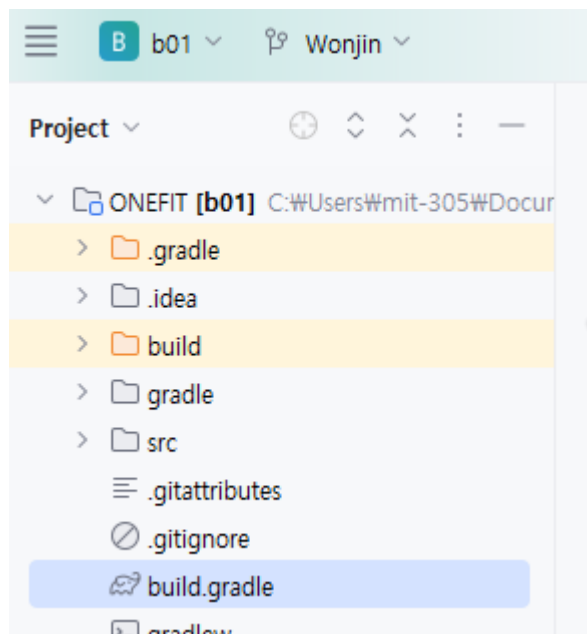
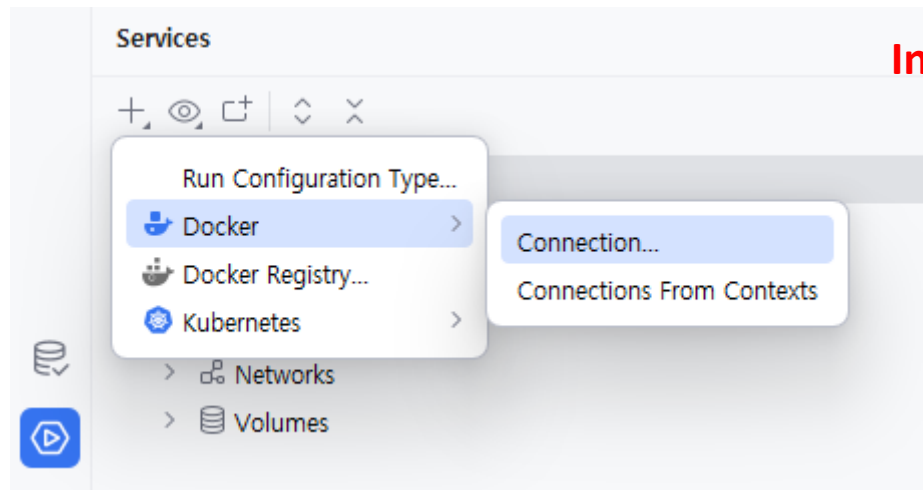
docker run -d --name mariadb-container -p 3306:3306

-v mysql-volume:/var/lib/mysql --network mysql-net

buildmariadb

docker volume create fileupload-volume

## IntelliJ 계정과 docker 계정이 동일시

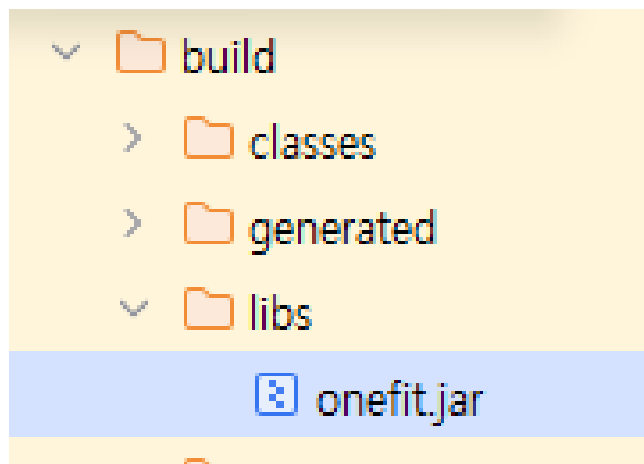


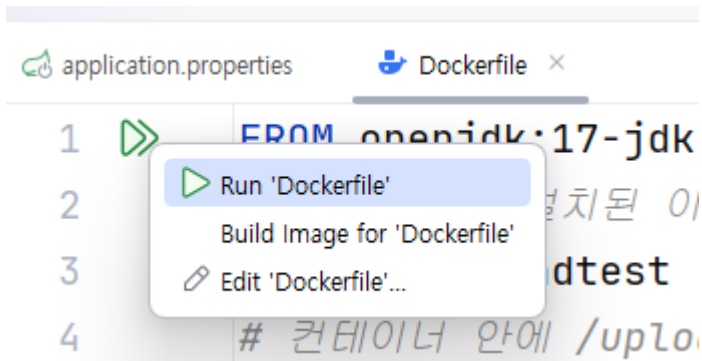
```
1  >> FROM openjdk:17-jdk
2      # Java 17이 설치된 이미지를 기반으로 합니다.
3      VOLUME /uploadtest
4      # 컨테이너 안에 /uploadtest라는 디렉토리를 생성합니다.
5      ARG JAR_FILE=build/libs/onefit.jar
6      # JAR 파일의 경로를 변수처럼 지정합니다.
7      COPY ${JAR_FILE} app.jar
8      # JAR 파일을 컨테이너 안으로 복사하고 이름을 app.jar로 바꿉니다.
9      ENTRYPOINT ["java"]
10     # 컨테이너가 실행되면 'java' 명령어를 실행합니다.
11     CMD ["-jar", "app.jar"]
12     # 'java -jar app.jar' 명령으로 실행되도록 지정합니다.
13     EXPOSE 80
14     # 컨테이너가 8080번 포트를 사용하도록 설정합니다.
```

application.properties x onefitDockerfile

개발자 양성\_7 Back\_Project\_원포인트 산출물\_WONEFIT =b01

```
2 spring.datasource.driver-class-name=org.mariadb.jdbc.Driver
3 spring.datasource.url=jdbc:mariadb://mariadb-container:3306/onefitdb
4 spring.datasource.username=root
5 spring.datasource.password=1234
6
7 server.port=80
```





```
C:\Users\mit-305>docker image ls
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
<none>	<none>	d5a1368af9d2	7 seconds ago	562MB

docker image ls

docker image tag 이미지아이디 onefit:latest

docker run -d --name onefit-container -v fileupload-volume:/uploadtest --network mysql-net -p 9090:80 onefit

## [ AWS의 EC2 에 스프링 부트 프로젝트 배포하기 ]

### [ 타임존 설정 ]

```
date  
sudo rm /etc/localtime  
sudo ln -s /usr/share/zoneinfo/Asia/Seoul /etc/localtime  
date
```

### [ docker 엔진 설치 ]

```
sudo apt update -y  
sudo apt t -y install docker  
sudo service docker start  
sudo docker version
```

### [ 도커 허브에 로그인 ]

```
sudo docker login
```

### [ 네트워크와 볼륨 생성 ]

```
sudo docker network create mysql-net  
sudo docker volume create mariadb-volume  
sudo docker volume create fileupload-volume
```

## [ AWS의 EC2 에 스프링 부트 프로젝트 배포하기 ]

[ 이미지 파일 허브에 업로드]

```
sudo docker commit buildmariadb teacherlee/buildmariadb:1.0
```

```
sudo docker commit onefit teacherlee/onefit:1.0
```

```
sudo docker image ls
```

```
sudo docker push teacherlee/buildmariadb:1.0
```

```
sudo docker push teacherlee/onefit:1.0
```

[ 도커 컨테이너로 MySQL 기동 ]

```
sudo docker run -d --name marisdb-container -p 3306:3306 -v mariadb-volume:/var/lib/mysql --network mysql-net teacherlee/buildmariadb:1.0
```

[ 도커 컨테이너로 스프링부트 애플리케이션 기동 ]

```
sudo docker run -d --name onefit-container -v fileupload-volume:/uploadtest --network mysql-net -p 9090:80 teacherlee/onefit:1.0
```