

1. 도커 이미지 빌드

1) Dockerfile 작성한다.

```
[Dockerfile]
FROM tomcat:8.5.46-jdk8-openjdk
RUN apt-get update
RUN apt-get install -y tzdata
ENV TZ=Asia/Seoul
CMD ["catalina.sh", "run"]
```

2) 작성한 Dockerfile 저장 위치에서 아래 도커 명령을 수행하여 이미지를 빌드한다.

```
docker build -t tomcat8:v1.0 .
```

3) docker-compose 버전 확인

```
docker-compose --version
```

4) docker-compose.yml 파일을 작성한다.(컨테이너 서비스 설정 파일)

```
[docker-compose.yml]
version: '3.0'
services:
  mint:
    container_name: iip-container
    image: tomcat8:v1.0
    environment:
      - TZ=Asia/Seoul
    volumes:
      - /Users/whoana/DEV/docker-home/tomcat8/logs:/usr/local/tomcat/logs
      - /Users/whoana/DEV/docker-home/tomcat8/upload:/usr/local/tomcat/upload
      - /Users/whoana/DEV/docker-home/tomcat8/logback/logback.xml:/usr/local/tomcat/logback/logback.xml
      - /Users/whoana/DEV/docker-home/tomcat8/bin/catalina.sh:/usr/local/tomcat/bin/catalina.sh
      - /Users/whoana/DEV/docker-home/tomcat8/conf/server.xml:/usr/local/tomcat/conf/server.xml
      - /Users/whoana/DEV/docker-home/tomcat8/conf/context.xml:/usr/local/tomcat/conf/context.xml
      - /Users/whoana/DEV/docker-home/tomcat8/webapps/mint-front-product-3.0.0.war:/usr/local/tomcat/webapps/mint.war
      - /Users/whoana/DEV/docker-home/tomcat8/lib/ojdbc6-11.2.0.3.jar:/usr/local/tomcat/lib/ojdbc6-11.2.0.3.jar
    ports:
      - 18080:8080
```

5) docker-compose.yml 파일 상의 볼륨 값에 매핑된 파일 구성

yml 파일에서 작성한 것 처럼 호스트 환경에 tomcat8 디렉토리 및 하위 구조 디렉토리를 만들고 만들고 교육용으로 제공한 파일들을 위치 시킨다.

/Users/whoana/DEV/docker-home/tomcat8/logs	로그폴더
/Users/whoana/DEV/docker-home/tomcat8/upload	업로드 폴더
/Users/whoana/DEV/docker-home/tomcat8/logback/logback.xml	logback 폴더 및 설정파일
/Users/whoana/DEV/docker-home/tomcat8/bin/catalina.sh	톰캣 시작종료 쉘
/Users/whoana/DEV/docker-home/tomcat8/conf/server.xml	서버 설정 파일(데이터베이스 리소스 정의 추가)
/Users/whoana/DEV/docker-home/tomcat8/conf/context.xml	리소스 링크 설정
/Users/whoana/DEV/docker-home/tomcat8/webapps/mint-front-product-3.0.0.war	IIP 배포 파일
/Users/whoana/DEV/docker-home/tomcat8/lib/ojdbc6-11.2.0.3.jar	오라클 JDBC 라이브러리

6) 톰캣 도커 컨테이너 실행

```
docker-compose up -d
```

7) 톰캣 도커 컨테이너 로그 확인

```
docker-compose logs -f
```

8) 톰캣 도커 컨테이너 접속

```
docker exec -it iip-container /bin/bash
```

9) 톰캣 도커 컨테이너 종료

```
docker-compose down
```

[docker-compose]

- 복수 개의 컨테이너를 한번에 실행 관리할 수 있다.
- 설정 파일은 **yml** 파일로 작성 관리
- 서비스 아래 컨테이너 생성 옵션
 - **container_name** 생성 컨테이너 명
 - **image** 컨테이너 생성에 쓰일 이미지
 - **environment** 도커명령 옵션 **-e**, 컨테이너 환경변수
 - **volumes** 호스트 와 컨테이너 디렉토리 및 파일 매핑
 - **ports** 호스트와 컨테이너 개발 포트 매핑
- **docker-compose -f [yml파일경로] config**
 - 작성된 파일 검증
 - 경로 옵션을 주지 않으면 현재 디렉토리파일 검사
- **docker-compose 명령들**
 - **docker-combose up -d** 컨테이너들 백그라운드 실행
 - **docker-combose up** 컨테이너들 포그라운드 실행
 - **docker-combose down** 컨테이너들 종료
 - **docker-combose restart** 컨테이너들 다시 실행
 - **docker-combose logs -f** 컨테이너들 로그 읽기
 - **docker-combose ps** 컨테이너들 상태 확인