Docker

```
설명 들어가면 괜찮을듯
```

1. docker-compose

```
version: '3.4'
service:
  spring:
    build: ./spring
    # ports:
    #   - 8080:8080
    #   - target: 8080       ## 컨테이너 내부 포트
    #     published: 8080    ## 호스트os에서 공개할 포트
    #     protocol: tcp      ## 포트 프로토콜
  nginx:
    build: ./nginx
    depends_on:
      - spring
    ports:
      - 80:80
```

```
service -> 각 컨테이너를 묶어주는 태그
spring, nginx -> 만들어질 컨테이너의 이름
build -> 경로를 입력하면 해당 경로의 dockerfile을 읽어 컨테이너를 생성한다.
depends_on -> 동작 순서를 지켜주기 위해 사용한다.
ports -> docker의 포트와 docker가 동작중인 pc의 포트를 연동한다. (포트포워딩)
```

2. dockerfile

```
현재 사용하는 컨테이너는 2개이다.
현재는 별다른 설정사항이 없지만 추후 변경될 수도 있다.
```

2-1. nginx dockerfile

```
FROM nginx:1.18.0
COPY nginx.conf /etc/nginx/nginx.conf
```

```
1- 이는 nginx:1.18.0이라는 docker image를 기반으로 컨테이너를 생성한다는 뜻이다.
2- 이는 docker file기준 nginx.conf라는 경로의 파일을 컨테이너 경로의 파일로 사용한다는 뜻이다.

nginx의 특성 상 nginx.conf설정만 변경해도 테스트환경까지는 구축할 수 있다.
```

2-2. spring dockerfile

```
FROM openjdk:11.0.9-jdk
VOLUME /tmp
ARG JAR_FILE=target/backend-0.0.1-SNAPSHOT.jar
COPY ${JAR_FILE} docker-springboot.jar
EXPOSE 8080
ENTRYPOINT ["java","-Djava.security.egd=file:/dev/./urandom","-jar","/docker-springboot.jar"]
```

```
Spring JAR파일을 구동하기 위한 컨테이너이다.
이는 openjdk docker image를 기반으로 구성된다.

실행할 JAR파일을 직접 지정할 수 있다.

ENTRYPOINT로 컨테이너에서 실행할 명령어를 지정할 수 있다. (현재는 JAR 빌드한 형태)
```

docker-compose 실행한 결과

```
PS C:\Users\user\Desktop\개인공부\community-project\community-project> docker-compose up
Creating network "community-project_default" with the default driver
Building spring
Step 1/6 : FROM openjdk:11.0.9-jdk
11.0.9-jdk: Pulling from library/openjdk
756975cb9c7e: Pull complete
d77915b4e630: Pull complete
5f37a0a41b6b: Pull complete
96b2c1e36db5: Pull complete
27a2d52b526e: Pull complete
a867dba77389: Pull complete
0939c055fb79: Pull complete
Digest: sha256:e02488dc2f354d5bc99b2a235f293f4c6a56360ad61e96d15d010cf91e4544b0
Status: Downloaded newer image for openjdk:11.0.9-jdk
 ---> 61aecffb761a
Step 2/6 : VOLUME /tmp
 ---> Running in fa034629a989
Removing intermediate container fa034629a989
 ---> 6f4a6f85f8e1
Step 3/6 : ARG JAR_FILE=target/backend-0.0.1-SNAPSHOT.jar
 ---> Running in af8a9410d043
Removing intermediate container af8a9410d043
 ---> 78042fc04aae
Step 4/6 : COPY ${JAR_FILE} docker-springboot.jar
 ---> 05e714cc44fb
Step 5/6 : EXPOSE 8080
 ---> Running in 6a38059fa755
Removing intermediate container 6a38059fa755
 ---> 064a35c2f4b3
Step 6/6 : ENTRYPOINT ["java","-Djava.security.egd=file:/dev/./urandom","-jar","/docker-springboot.jar"]
```

```
 ---> Running in 0bdd48ad4a65
Removing intermediate container 0bdd48ad4a65
 ---> d6591c8435ab

Successfully built d6591c8435ab
Successfully tagged community-project_spring:latest
WARNING: Image for service spring was built because it did not already exist. To
rebuild this image you must use `docker-compose build` or `docker-compose up --
build`.
Building nginx
Step 1/2 : FROM nginx:1.18.0
1.18.0: Pulling from library/nginx
852e50cd189d: Pull complete
48b8657f2521: Pull complete
b4f4d57f1a55: Pull complete
d8fbe49a7d55: Pull complete
04e4a40fabc9: Pull complete
Digest: sha256:2104430ec73de095df553d0c7c2593813e01716a48d66f85a3dc439e050919b3
Status: Downloaded newer image for nginx:1.18.0
 ---> acbb257fa814
Step 2/2 : COPY nginx.conf /etc/nginx/nginx.conf
 ---> e026c7f1b4ef

Successfully built e026c7f1b4ef
Successfully tagged community-project_nginx:latest
WARNING: Image for service nginx was built because it did not already exist. To
rebuild this image you must use `docker-compose build` or `docker-compose up --
build`.
Creating community-project_spring_1 ... done
Creating community-project_nginx_1  ... done
Attaching to community-project_spring_1, community-project_nginx_1
nginx_1   | /docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will
attempt to perform configuration
nginx_1   | /docker-entrypoint.sh: Looking for shell scripts in /docker-
entrypoint.d/
nginx_1   | /docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-
ipv6-by-default.sh
nginx_1   | 10-listen-on-ipv6-by-default.sh: Getting the checksum of
/etc/nginx/conf.d/default.conf
nginx_1   | 10-listen-on-ipv6-by-default.sh: Enabled listen on IPv6 in
/etc/nginx/conf.d/default.conf
nginx_1   | /docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-
on-templates.sh
nginx_1   | /docker-entrypoint.sh: Configuration complete; ready for start up
spring_1  |
spring_1  |   .   ____          _            __ _ _
spring_1  |  /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
spring_1  | ( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
spring_1  |  \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
spring_1  |   '  |____| .__|_| |_|_| |_\__, | / / / /
spring_1  |  =========|_|==============|___/=/_/_/_/
spring_1  |  :: Spring Boot ::        (v2.3.4.RELEASE)
spring_1  |
spring_1  | 2020-11-20 01:12:00.931  INFO 1 --- [           main]
com.commu.backend.BackendApplication     : Starting BackendApplication on
a9e6c8fb9e34 with PID 1 (/docker-springboot.jar started by root in /)
```

```
spring_1  | 2020-11-20 01:12:00.936  INFO 1 --- [           main]
com.commu.backend.BackendApplication     : No active profile set, falling back
to default profiles: default
spring_1  | 2020-11-20 01:12:01.402  WARN 1 --- [kground-preinit]
o.s.h.c.j.Jackson2ObjectMapperBuilder    : For Jackson Kotlin classes support
please add "com.fasterxml.jackson.module:jackson-module-kotlin" to the classpath
spring_1  | 2020-11-20 01:12:02.324  INFO 1 --- [           main]
trationDelegate$BeanPostProcessorChecker : Bean
'org.springframework.ws.config.annotation.DelegatingWsConfiguration' of type
[org.springframework.ws.config.annotation.DelegatingWsConfiguration$$EnhancerByS
pringCGLIB$$4e94431a] is not eligible for getting processed by all
BeanPostProcessors (for example: not eligible for auto-proxying)
spring_1  | 2020-11-20 01:12:02.408  INFO 1 --- [           main]
.w.s.a.s.AnnotationActionEndpointMapping : Supporting [WS-Addressing August
2004, WS-Addressing 1.0]
spring_1  | 2020-11-20 01:12:02.748  INFO 1 --- [           main]
o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port(s): 8080
(http)
spring_1  | 2020-11-20 01:12:02.762  INFO 1 --- [           main]
o.apache.catalina.core.StandardService   : Starting service [Tomcat]
spring_1  | 2020-11-20 01:12:02.763  INFO 1 --- [           main]
org.apache.catalina.core.StandardEngine  : Starting Servlet engine: [Apache
Tomcat/9.0.38]
spring_1  | 2020-11-20 01:12:02.859  INFO 1 --- [           main] o.a.c.c.C.
[Tomcat].[localhost].[/]       : Initializing Spring embedded
WebApplicationContext
spring_1  | 2020-11-20 01:12:02.859  INFO 1 --- [           main]
w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext:
initialization completed in 1766 ms
spring_1  | 2020-11-20 01:12:03.146  INFO 1 --- [           main]
o.s.s.concurrent.ThreadPoolTaskExecutor  : Initializing ExecutorService
'applicationTaskExecutor'
spring_1  | 2020-11-20 01:12:03.301  INFO 1 --- [           main]
o.s.b.a.w.s.WelcomePageHandlerMapping    : Adding welcome page: class path
resource [static/index.html]
spring_1  | 2020-11-20 01:12:03.418  INFO 1 --- [           main]
o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port(s): 8080
(http) with context path ''
spring_1  | 2020-11-20 01:12:03.429  INFO 1 --- [           main]
com.commu.backend.BackendApplication     : Started BackendApplication in 3.348
seconds (JVM running for 4.191)
```