

02_Docker_command

- docker를 관리 할 때 사용되는 다양한 명령어가 있는데 그 중 자주 사용되고 중요한 명령어에 대해서 알아 보자.
 - 명령어에서 주로 사용되는 옵션도 같이 정리하였다.
- 옵션이 방대하므로 필요한 기능이 있을 때에는 reference site에서 확인하여 사용을 한다.

Build

- Dockerfile을 이용하여 imag를 만들 때 사용한다.

```
# docker build [OPTIONS] PATH | URL | -  
$ docker build --tag test:v1 .
```

- -t, --tag : name:tag 와 같은 형태로 인자를 주고 생성되는 image의 tag를 지정한다.
- -f, --file string : 특정 dockerfile을 지정하여 image를 만들 때 사용한다.
- --build-arg : Dockerfile에 정의된 변수에 대해 값을 지정할 때 사용한다.
- --no-cache : build 시 cache를 남기지 않도록 할 때 사용한다.

RUN

- 만들어진 image를 가지고 실행할 때 사용한다.

```
# docker run [OPTIONS] IMAGE [COMMAND] [ARG...]  
$ docker run -p 8080:80 -it --rm --name testserver test:v1 /bin/bash
```

- -p, --publish : (hostport:containerport) container 내부의 port와 host의 port를 binding 할 때 사용한다.
- -i, --interactive : STDIN을 container와 연결하여 동적으로 반응할 수 있게한다.
- -t, -tty : tty처럼 작동할 수 있게 한다.
- --rm : container가 종료되면 자동으로 image를 지워준다.
- -e, --env : 환경변수를 정의할 때 사용한다.
- --name : Container 이름을 정의한다.

- -d, —detach : background로 container가 동작하고 ID를 출력한다.
- -v, —volume : volume을 마운트시킬 때 사용한다.

STOP

- container를 중지시킬 때 사용한다.

```
# docker stop [OPTIONS] CONTAINER [CONTAINER...]
$ docker stop testserver
```

- -t, —time : STOPSIGNAL 시그널을 보내고 몇 초를 대기할 지 설정한다.

START

- 멈춰진 container를 다시 동작시킬 때 사용한다.

```
# docker start [OPTIONS] CONTAINER [CONTAINER...]
docker start testserver
```

- -a, —attach : stdout/stderr 연결한다.
- -i, —interactive : STDIN을 container와 연결하여 동적으로 반응할 수 있게 한다.

ATTACH

- 명령어는 실행된 container의 프로세스에 연결할 때 사용한다.
- 연결하게 되는 프로세스는 cmd, entrypoint로 정의된 프로세스이므로 여기서 ctr-c를 눌러 종료하면 container가 멈춘다.
- ctr-p, ctr-q로 deattach하여 나오는 것을 권장한다.

```
# docker attach [OPTIONS] CONTAINER
docker attach testserver
```

EXEC

- 실행된 container에 프로세스를 실행을 시킬 때 사용한다.

```
# docker exec [OPTIONS] CONTAINER COMMAND [ARG...]
docker exec -it testserver /bin/bash
```

- `-d, --detach` : 실행한 프로세스와 연결시키지 않고 백그라운드로 동작하게 한다.
- `-e, --env` : container에게 환경변수를 전달.
- `-i, --interactive` : STDIN을 container와 연결하여 동적으로 반응할 수 있게 한다.
- `-t, --tty` : tty처럼 작동할 수 있게 한다.
- `-u, --user` : 실행하는 프로세스의 유저를 정의한다.
- `-w, --workdir` : 실행되는 작업 경로를 지정한다.

PS

- Container로 목록을 확인할 때 사용한다.
- 기본값은 작동 중인 container만 출력한다.

```
# docker ps [OPTIONS]
docker ps -a
```

- `-a, --all` : 모든 container 목록을 출력.

images

- image의 목록을 확인할 때 사용한다.
- 기본값은 중간 layer단계의 image는 숨긴다.

```
# docker images [OPTIONS] [REPOSITORY[:TAG]]
docker images -a
```

- `-a, --all` : 모든 image 목록을 출력.

INSPECT

- container object의 정보를 출력할 때 사용한다.

```
# docker inspect [OPTIONS] NAME|ID [NAME|ID...]
docker inspect testserver
```

- `-f, --format` : 옵션과 함께 주어진 template 형태로 출력한다.

RM

- Container를 제거할 때 사용한다.

```
# docker rm [OPTIONS] CONTAINER [CONTAINER...]  
docker rm testserver
```

- -f, —force : 작동 중인 container여도 SICKILL을 보내 강제 종료 후 삭제한다.

RMI

- Image를 제거할 때 사용한다.

```
# docker rmi [OPTIONS] IMAGE [IMAGE...]  
docker rmi test
```

- -f, —force : 강제로 image를 제거한다.

COMMIT

- container의 상태를 가지고 새로운 이미지를 만들 때 사용한다.

```
# docker commit [OPTIONS] CONTAINER [REPOSITORY[:TAG]]  
docker commit testserver testserver:v2
```

- -a, —author : author에 대해 정의한다.
- -c, —change : 생성된 image에 Dockerfile 명령어를 적용한다.
- -m, —message : commit message를 작성한다.
- -p, —pause : commit 시점에 container 일시 중단.

그 외 도커 명령어 참고 사이트

<https://yeosong1.github.io/도커-명령어-모음>