**DEV #3 – 2020.04.09**

**>> alpine linux를 docker container로 실행하되 dettach mode(백그라운드)로 실행하세요**

docker run -d –name alpine\_gg -p 8000:8000 alpine tail -t //dev//null (/dev/null 을 계속 조회하게 백드라운드로 실행)

#--rm 프로세스 종료시 자동삭제

**>> 위에서 실행중인 container의 쉘(sh)을 실행하고 터미널로 접속하세요.**

Docker exec -it alpine\_gg binsh

**>> 터미널 접속후 타임존 설치 후 변경(Asia/Seoul)하세요. (apk add tzdata)**

#apk add tzdata

#ln -sf /usr /share/zoneinfo/Asia/Seoul etc/localtime

**>> python을 apk로 설치(apk add python)하세요.**

#apk add python

**>> “/www” 디렉토리를 만들고 /www 이동후 python SimpleHTTPServer 를 실행하세요.**

(python -m SimpleHTTPServer 8000) == (8000 port로 웹서버 구동)

#mkdir /www

#cd /www

#python -m SimpleHTTPServer 8000

>**> 위 과정으로 Dockerfile로 만드세요.**

FROM alpine:latest

# shell

RUN /bin/sh

#timezon install

RUN apk add tzdata && \

cp /usr/share/zoneinfo/Asia/Seoul /etc/localtime && \

echo 'Asia/Seoul' > /etc/timezone && \

apk del tzdata && \

apk add python

#create directory /www

WORKDIR /www

ENTRYPOINT ["python", "-m", "SimpleHTTPServer", "8000"]

**>> Dockerfile로 image를 빌드하고 빌드된 이미지로 새로운 container를 실행하세요.**

**(정상 수행여부 확인)**

docker build -t alpain\_gg .

**>> git 저장소를 새로만들고 (docker-images) 브랜치(simple-web)도 하나 만드세요.**

**>> Dockerfile을 github에 push하세요 (simple-web branch).**

**>> docker hub (hub.docker.com)에 접속하여 새로운 저장소(Repository)를 만드세요**

**저장소 생성시 Build Settings 옵션에서 github docker-images저장소의**

**simple-web 브랜치와 연결(자동빌드설정)하여 생성하세요.**

**>> 자동빌드 여부 확인 후 생성된 Repository Image를 검색해보세요.**

**>> 본인의 docker hub에 등록된 이미지를 pull하고 container를 생성해 보세요.**

**>> Dockerfile 수정후 github에 push할때 docker hub의 이미지가 자동빌드되는지 확인하세요.**