docker container run <Docker이미지명> <실행할 명령> ==> 컨테이너를 작성 및 실행

docker version

docker system info

docker system df 상세내용은 -v

docker pull nginx ==> nginx 다운

docker run --name webserver -d -p 80:80 nginx

-p 는 컨테이너 전송 허가

docker container ps ngin서버 상태 확인

docker container stats webserver

docker stop webserver docker 정지

docker start webserver 도커 실행

docker image pull [옵션] 이미지명[:태그명]

docker image pull centos:7 7안붇https://github.com/thisisyoobi/KNU\_CDP1.git이면 가장 최근 이미지 취득

docker image pull -a centos ==> 모든 태그 이미지 취득

docker image pull gcr.io.tensorflow/tensorflow http뺀 주소로 이미지 취득

docker image ls [옵션] [리포지토리명] 취득 이미지 목록 표시

docker image inspect centos:7

docker image inspect --format="{{ .Os}}" centos:7

docker image inspect --format="{{ .ContainerConfig.Image }}" centos:7

docker image tag nginx asashiho/webserver:1.0

이미지 별명일 뿐 실체는 똑같음 image id를 확인하면 똑같다.

<Docker Hub 사용자명>/이미지명: [태그명]

docker search [옵션] <검색 키워드>

도커 이미지 명명 규칙은 사용자명/이미지명 으로

docker image rm [옵션] 이미지명 [이미지명]

옵션: -f

이미지명은 repository 또는 image id를 지정한다. 여러 개의 이미지를 삭제하고 싶으면

이미지명을 스페이스로 구분.

사용하지 않는 이미지를 삭제할때는

docker image prune [옵션]을 사용.

옵션 : --all, -a 전부 삭제, --force,-f 강제로 삭제

docker login [옵션] [서버]

옵션 : --password, -p 비밀번호, --username, -u : 사용자명

도커허브에 이비지 업로드하려면

docker image push 이미지명[:태그명]

이미지는 다음과 같은 형식 사용 : <Docker Hub 사용자명>/이미지명: [태그명]

docker logout [서버명]

docker container create

이미지로부터 컨테이너를 생성함. 이미지의 실체는 도커에서 서버 기능을 작동시키기위해 필요한

디렉토리 및 파일들.

docker container run

docker container start 명령

정지 중인 컨테이너를 시작할 때 사용.

docker container stop 명령

컨테이너 삭제시에 정지.

docker container restart

컨테이너 재시작

docker container rm

정지중인 컨테이너 프로세스 삭제

docker container ps

컨테이너 상태 확인

docker container pause

컨테이너 일시정지

docker container run 의 주요 옵션 정리

-a 표준 입력, 출력, 오류 출력에 연결

-d 백그라운드 실행

-i 컨테이너의 표준 입력을 연다

-t 단말기 디바이스를 사용

docker container run -it --name "test1" centos /bin/cal

컨테이너 실행 명령 / 콘솔에 결과를 출력하는 옵션 / 컨테이너명 / 이미지명 / 컨테이너에 실행할 명령

--name 옵션 생략하면 컨테이너명이 랜덤으로 자동 설정

/bin/bash를 실행하면 콘솔의 프롬프트가 실행.

컨테이너의 백그라운드 실행

-d 백그라운드 실행

-u 사용자명 지정

--rm 명령 실행 완료 후에 컨테이너를 자동으로 삭제

--restart=[옵션]

no 재시작하지 않는다

on-failure 종료 스테이터스가 0이 아닐 때 재시작한다.

on-failure:횟수 n 종료 스테이터스가 0이 아닐 때 n번 재시작한다.

always 항상 재시작한다

unless-stopped 최근 컨테이너가 정지 상태가 아니라면 항상 재시작한다.

docker container run -d centos /bin/ping localhost

실행되면 컨테이너 ID가 출력함

백그라운드에서 실행되고 있는지 아닌지를 확인할 때는 docker container logs 명령

docker container logs -t ---------

컨테이너의 네트워크 설정

docker container run [네트워크 옵션] 이미지명[:태그명] [인수]

옵션

--add-host=[호스트명:IP 주소] 컨테이너의 /etc/hosts에 호스트명과 IP주소를 정의

--dns=[IP주소] 컨테이너용 DNS 서버의 IP 주소 지정

--expose 지정한 범위의 포트 번호를 할당

--mac-address=[MAC 주소] 컨테이너의 MAC 주소 지정

--net=[bridge | none | container:<name | id> | host | NETWORK] 컨테이너의 네트워크 지정

--hostname, -h 컨테이너 자신의 호스트명 지정

--publish, -p[호스트의 포트 번호]:[컨테이너의 포트 번호] 호스트와 컨테이너의 포트 매핑

--publish-all, -P 호스트의 임의의 포트를 컨테이너에 할당

docker container run -d -p 8080:80 nginx

컨테이너의 DNS서버 지정

docker container run -d --dns 192.168.1.1 nginx

호스트명및 IP 설정

docker container run -it --add-host test.com:192.168.1.1 centos

--net=NETWORK 옵션을 사용할 시에

docker network create를 사용하여 작성

작업 디렉토리 지정하고 싶으면

docker container run -it -w=/tensorflow centos /bin/bash