**\*\* vscode에서 실습을 진행하자. 터미널도 그 안에서 열 수 있다.**

docker 명령어 모음

docker overview에 쓰여있는 command들 중 <hash>는 기계가 읽는 단어이다.

Ex) docker container stop을 할 때 썼던 container name (a7sazcw67d6)

docker image ls

docker container ls

**Clean**

docker image rm -f [image id]

docker container stop [container name]

docker container stop [container name]

docker container prune

* stop된 container 모두 지우기.

docker pull ubuntu

* ubuntu를 registree에서 가져오겠다

docker container run -it -d –rm –name ubuntuos ubuntu:latest

* ubuntuos 라는 이름의 ubuntu 최신버전(latest)이 담겨있는 container를 만들고 실행하겠다

docker exec -it ubuntuos /bin/bash

* ubuntuos 의 /bin/bash위치를 실행하겠다

docker container ls

* (7b1c674af93b ubuntu:latest "bash" About a minute ago Up About a minute ubuntuos)

docker image ls

* 실행 확인용
* (ubuntu latest fb52e22af1b0 3 weeks ago 72.8MB)

curl <http://localhost:8000>

* 브라우저를 키고 웹서버에 접속하는 것과 동일하게, cmd창에서 웹서버로 접속할 수 있다.

$( ) 는 괄호안의 것을 입력으로 받도록 한다.

Ex) docker image rm -f $(docker image ls -a -q)

* 도커 image의 image ID만을 불러와서 입력으로 받아라.

docker build –-tag myubuntu:1.0 .

* --tag : image의 이름을 지정해줄 수 있다.
* . : dockerfile이 있는곳에서 빌드를 해야하며, 이때 dockerfile의 위치가 된다. (상대경로 또는 절대경로)
  + 위 명령어는 도커파일이 있는 위치에서 실행되었기 때문에 파일 위치를 ‘.’으로 한 것.

docker run --name -d -p 4000:80

* --name : container에게 name을 주겠다.
* -d : background에서 동작하는 daemon으로 해라
* -p 4000:80 : port address translation /network address translation 찾아보기.
  + 외부에서 사용하는 port = 4000
  + 가상화 내부에서 nginx가 사용하는 port = 80
  + 4000포트로 접근하면 자동으로 80번포트로 docker가 연결해준다.

**Docker image build**

FROM : 새로운 이미지를 만들건데, 가장 기초가되는 base image를 docker hub에서 가져와 설치.

RUN echo : 화면에 뿌리는 것. 별 의미는 없다

WORKDIR : 내가 시키는 일은 이 경로에서 한다.

EXPOSE {N}: 이 프로그램은 포트넘버를 N번으로 사용하겠다. 동시에 이 프로그램에 접속하고자 하는경우, 포트넘버를 N으로 하면 접속할 수 있다.

**Create docker compose files**

Make your working directory

Move into your working directory

Create docker-compose.yml file

Fill the contents

start! (docker-compose up -d)

end… (docker-compose down)

Dockerfile

* 텍스트파일로 모든 커맨드, 오더, 이미지를 빌드하는데 필요한 것들이 적혀있다.
* 도커는 도커파일로부터 instructions를 읽어와서 build를 한다.

Dockerfile commands

* COPY . /app : 이미지를 만드는 와중에, 도커명령을 수행하고 있는 내 컴퓨터의 현재directory (current, local dir) 에 있는 file을 docker image안으로 밀어 넣으라(COPY)는 것.
* CMD python /app/app.py : 우분투 안에서 명령어를 실행해달라 - 파이썬을 통해 내가 만들어서 업로드한 파일중에 app.py파일을 실행하라.

텍스트이(가) 표시된 사진

자동 생성된 설명

.:/code 의 의미

* alpine기반의 container root의 /code와 container 바깥을 연결한다.
* 즉, code디렉토리에는 container 바깥의 working directory 파일이 들어있다.
  + Ex : docker-compose.yml

Swarm token

docker swarm join --token SWMTKN-1-5fe0r0vcgbax831gper9bljwh220g6bjhkc1rmblygshkll1ky-e4cuxqiqtwao0406g1emsbhnt 192.168.99.111:2377

docker-machine scp 실행시

port 22 : connection refused 오류 발생할 때

<http://mobilelab.khu.ac.kr/wordpress/dc/?vid=289> 여기 참조

Docker Swarm 정리

1. manager 를 leader로 해서 init
2. leader와 worker사이에는 보안이 있어야 하기 때문에 leader는 token을 생성해줌.
3. worker들을 leader에 join
4. IaaS에 맞춰 swarm이 작동시킬 application을 yml파일로 만듦.
5. scp명령어로 manager에 yml파일 넘김
6. manager에서 stack을 deploy하여 yml실행
7. 청소할때는 stack을 끄고, remove한다.
8. worker, manager를 swarm에서 leave
9. docker machine remove