200805 학습내용

Docker

1.Docker란 무엇인가

Container

1. Container란 무엇인가.
2. Container를 사용 하였을 때의 이점
3. Container 생성 방법
4. 현재 Container를 저장
5. Dockerfile이란

Container Compose

1. Compose는 무엇인가
2. Compose 방법 - 내일 mysql, nginx, php 컨테이너 생성과 함께 쭉 정리할 예정

에 대해 학습하였다.

======================================================================

<Docker>

1. Docker란 무엇인가 : Docker란 사용자의 환경 위에서 가상으로 환경을 구성하는 컨테이너 기반 가상화 플랫폼이다.

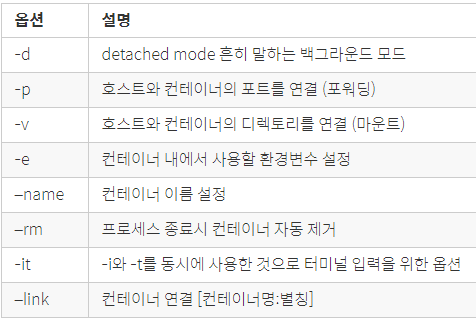
<Container>

1. 컨테이너란 : 컨테이너 하나 당 하나의 작업 환경을 구성한다.
2. 컨테이너 사용 시의 이점 :
3. 현재 사용하고 있는 작업 환경을 이미지로 저장하여, 다른 컴퓨터에서도 이미지를 이용해 컨테이너 생성이 가능하기에 이식성이 좋다.
4. 하나의 컨테이너는 다른 작업에 방해를 끼치지 않는다.

3. 컨테이너 생성 방법 :

1. Docker 실행 (Window로 실행중인지 리눅스로 실행중인지 확인하고 변경해 줘야함)
2. CMD창 열어서 docker run [OPTIONS] IMAGE[:TAG|@DIGEST] [COMMAND] [ARG...]

* image의 tag 미 기입시 디폴트는 :latest (최신 버전) 이다.
* 이미지가 로컬 내에 없을 시 docker hub를 탐색한다.
* 자주 사용하는 Option들로는

 들이 있다.

주로 서버관련 컨테이너 생성시 -d -p 서버포트:내포트 option으로 적는다.

4. 컨테이너 저장하기

docker export [container name] > [저장할 파일명.tar]

컨테이너를 이미지로 저장

docker commit [options] <container name> [image name[:tag name]]

원칙은 dockerfile을 이용해서 만들어야함.

5. DockerFile은 Docker image를 만들기 위한 설정 파일임.

FROM ubuntu:14.04

# app 디렉토리 생성  
 RUN mkdir -p /app  
   
 #Docker 이미지 내부에서 RUN, CMD, ENTRYPOINT의 명령이 실행될 디렉터리를 설정합니다.  
 WORKDIR /app  
   
 # 현재 디렉터리에 있는 파일들을 이미지 내부 /app 디렉터리에 추가함  
 ADD . /app  
  
 RUN apt-get update  
 RUN apt-get install apache2  
 RUN service apache2 start  
  
 VOLUME ["/data", "/var/log/httpd"]  
   
 # 하기 포트를 외부로 노출합니다.  
 EXPOSE 80

# 쉘을 사용하지 않고 컨테이너가 시작되었을 때 logbackup 스크립트를 실행  
CMD ["/app/log.backup.sh"]

<Container Compose>

1. Compose는 무엇인가 : 여러 컨테이너들을 통합한 개발환경
2. Compose 방법 : docker-compose.yml을 만들어야함.

이건 내일 mysql nginx php 컨테이너 생성 부터 쭉 정리할 예정

ADD . /app

RUN apt-get update

RUN apt-get install apache2

RUN service apache2 start

VOLUME ["/data", "/var/log/httpd"]

# 하기 포트를 노출합니다.

EXPOSE 80

# 쉘을 사용하지 않고 컨테이너가 시작되었을 때 logbackup 스크립트를 실행

CMD ["/app/log.ba