

02_ft_server

Introduction

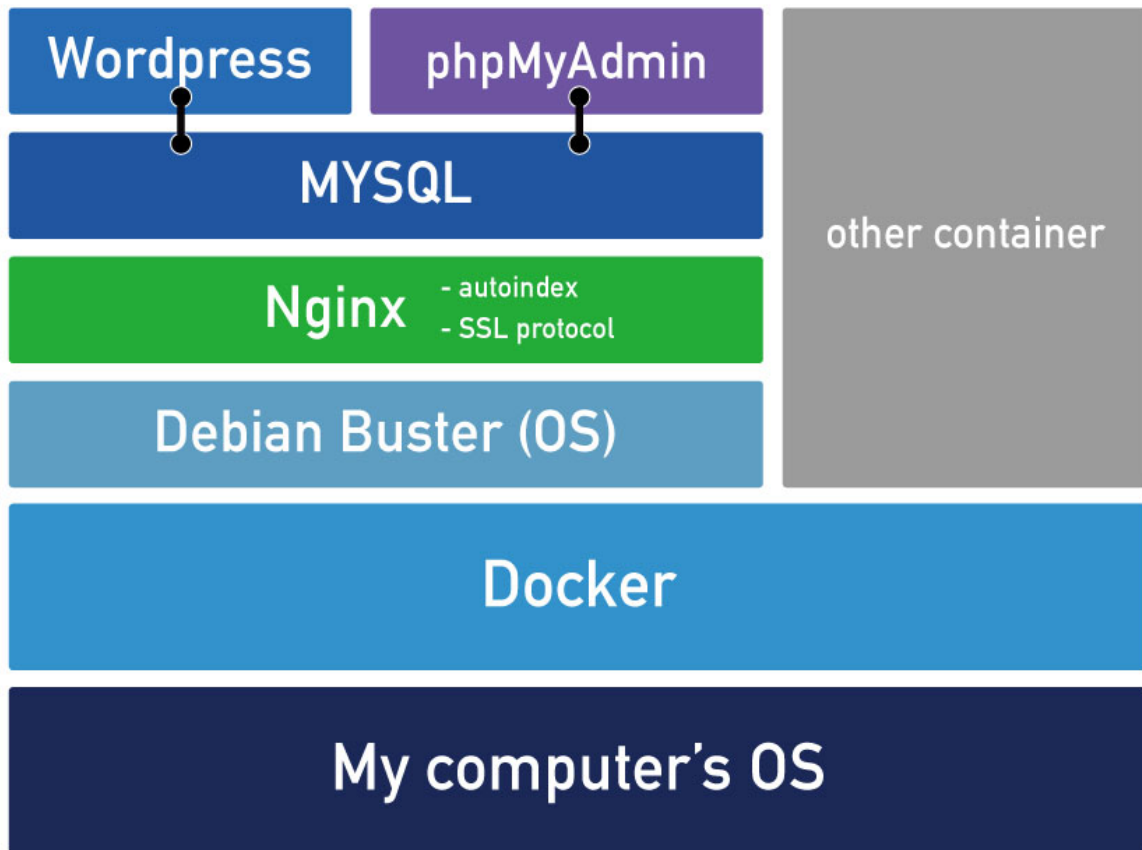
- 이번 과제는 시스템 관리 개념을 소개하기 위한 과제이다.
 - 작업을 자동화하는 데 스크립트를 사용하는 것이 중요하다는 것을 알 수 있으며, 이를 위해 "도커" 기술을 검색하여 완전한 웹 서버를 설치하는 데 사용해야 한다.
 - 이 서버는 WordPress, phpMyAdmin 및 SQL 데이터베이스의 여러 서비스를 실행한다.
-

General instructions

- 서버 구성에 필요한 모든 파일을 srcs 폴더에 넣어야 한다.
 - WordPress 웹 사이트에 필요한 모든 파일은 srcs 폴더에 있어야 합니다.
 - Dockerfile 파일은 repository의 root에 있어야 하며, container를 빌드할 것이다.
 - docker-compose은 사용할 수 없다.
-

Mandatory part

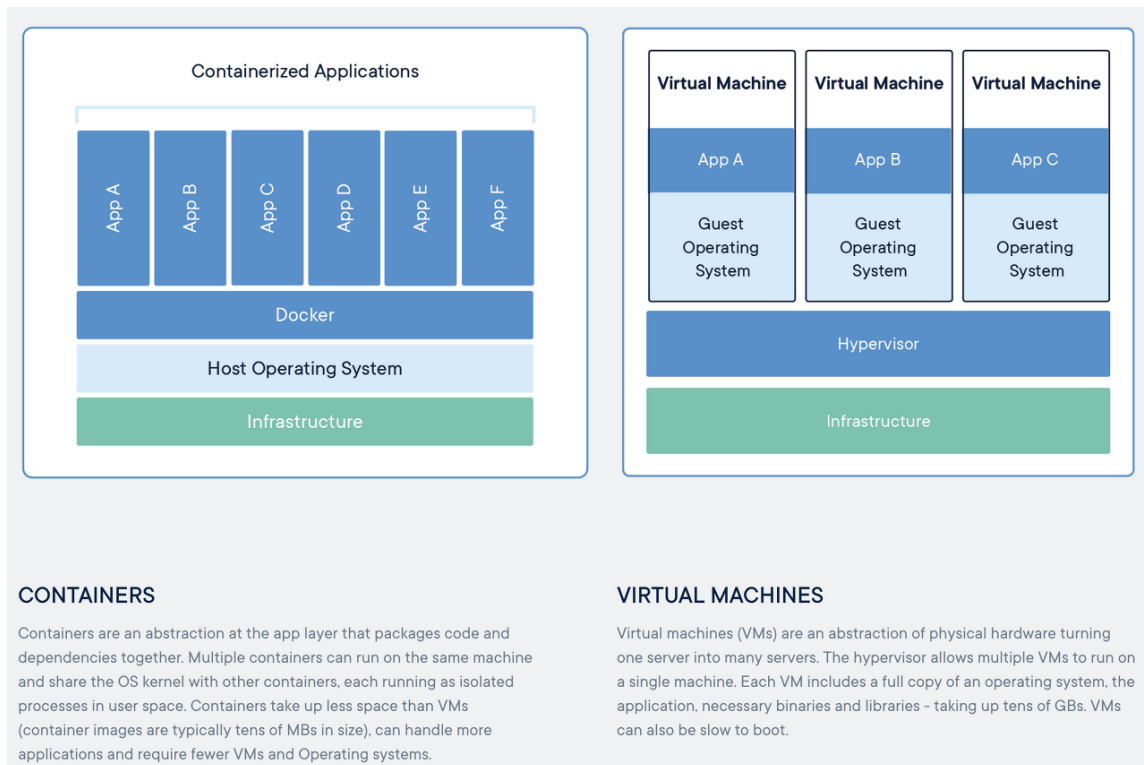
- Nginx를 사용하여 하나의 Docker container에만 웹 서버를 설정해야 한다.
 - Container OS는 반드시 debian buster여야 한다.
- 웹 서버는 여러 서비스를 동시에 실행할 수 있어야 한다.
 - 서비스는 WordPress website, phpMyAdmin 과 MySQL이다.
 - SQL Database가 WordPress 및 phpMyAdmin과 함께 작동하는지 확인해야 한다.
- 서버가 SSL 프로토콜을 사용할 수 있어야 한다.
- URL에 따라 서버가 올바른 website로 리디렉션되는지 확인해야 한다.
- 서버가 autoindex로 실행 중인지 확인하고, 이를 비활성화 할 수 있어야한다.



- 컴퓨터를 작동시키는 OS 위에, 도커를 설치한 후 도커를 이용해 가상환경(container)에 Debian을 설치한다. 설치된 Debian 즉, 새로운 OS에 웹 서버를 구축하는 것이다.
- 웹 서버는 Nginx를 통해서 구축된다.
- 웹 서버는 MySQL을 이용해 제작한 데이터베이스에 접근할 수 있는 WordPress website와 phpMyAdmin을 구동할 수 있어야 한다.

Docker

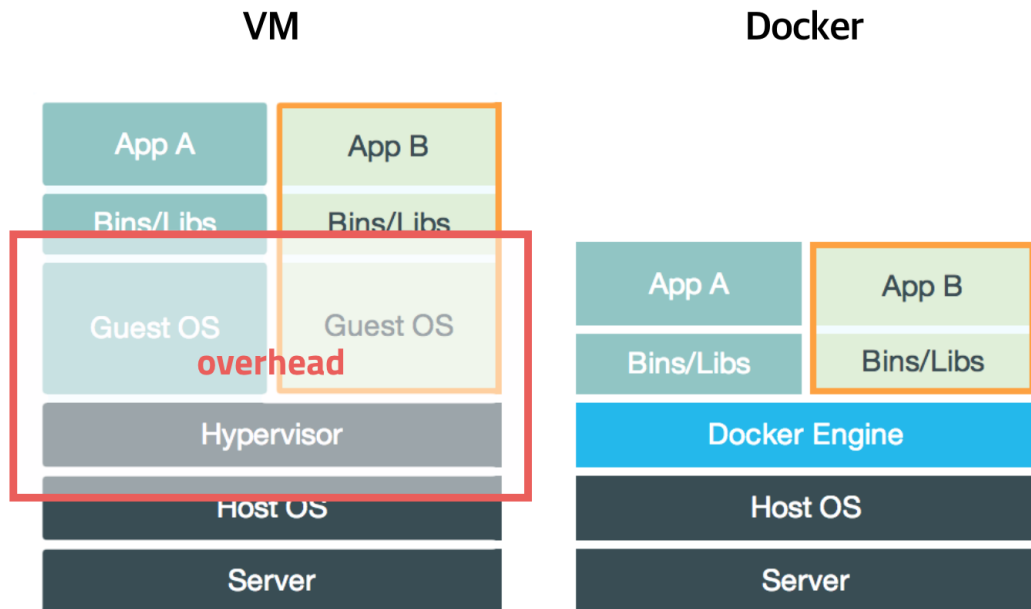
- 도커(Docker)는 리눅스의 응용 프로그램들을 소프트웨어 Container 안에 배치시키는 일을 자동화하는 오픈 소스 프로젝트이다.
- 애플리케이션을 신속하게 구축, 테스트 및 배포할 수 있는 소프트웨어 플랫폼.
- 어떤 기능을 지칭하는 것이 아니라 Docker는 플랫폼(프로젝트)의 이름일 뿐이며, 그 안에 나오는 container와 image의 개념을 이해해야 한다.



- Container

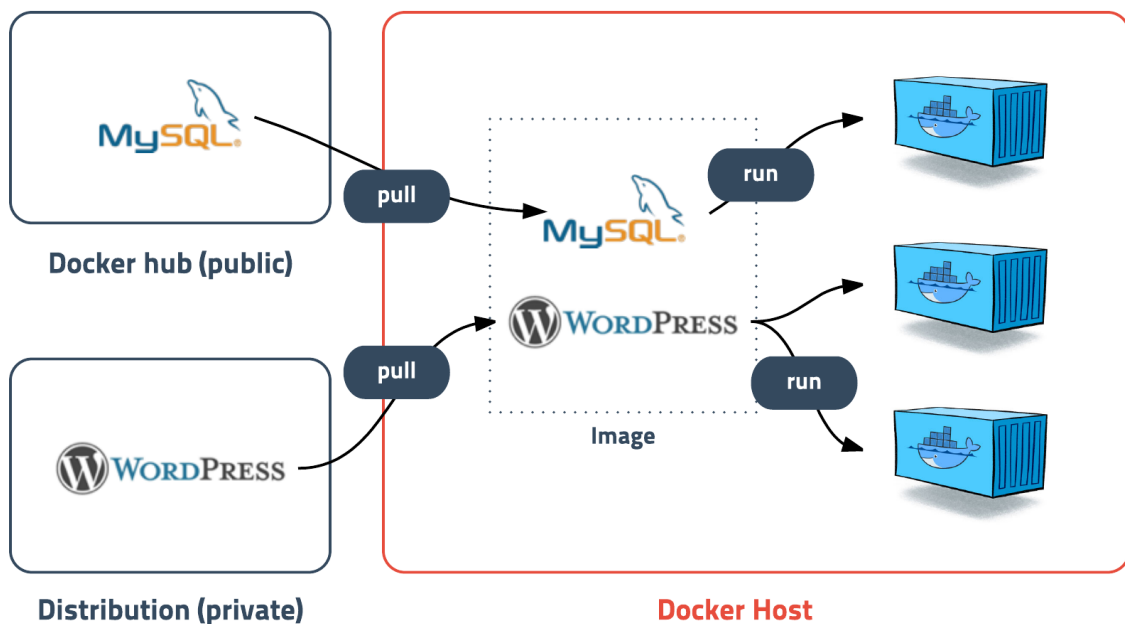
- 격리된 공간에서 프로세스가 동작하는 기술이며, 가상화 기술의 하나지만 기존의 방식과는 차이가 있다.
- 기존의 가상화 방식
 - 주로 OS를 가상화, VMware 또는 VirtualBox와 같은 가상머신은 Host OS 위에 Guest OS 전체를 가상화하여 사용하는 방식.
 - 컴퓨터의 메모리와 기타 하드웨어 성능을 부분적으로 가져와 개별적인 환경을 구축.
 - 장점 : 여러가지 OS를 가상화할 수 있고 사용법이 간단함.
 - 단점 : OS의 완전 복제를 의미하며 무겁고 느려서 운영환경에선 사용할 수 없음.
- 개선된 가상화 방식
 - CPU의 가상화 기술(HVM, Hardware Virtual Machine)을 이용한 커널기반 가상화(KVM, Kernel-based Virtual Machine)와 반가상화(PV, Paravirtualization) 방식의 Xen의 등장.
 - Guest OS가 필요하지만 전체 OS를 가상화하는 방식이 아니므로 기존의 가상화 방식보다 성능이 향상됨.

- 이러한 기술들은 OpenStack, AWS, Rackspace와 같은 클라우드 서비스에서 가상 컴퓨팅 기술의 기반이 됨.
- 이러한 방식을 사용하지만 추가적인 OS를 설치하여 가상화하는 방식은 성능문제가 있기 마련이고 이를 개선하기 위해 프로세스를 격리하는 방식이 등장하게 된다.



- Linux에서 이 방식을 Linux Container라고 한다.
 - Linux Container(By Jaeskim blog)
 - Docker를 이용하게 되면 응용 프로그램을 Host와 격리되어 실행이 되고 응용 프로그램을 위한 환경 전체가 미리 이미지에 대해 정의가 되어 실행이 되기 때문에 host의 환경과 상관없이 언제나 동일한 실행할 수 있다!
 - 또한 가상화와 달리 격리하여 프로세스를 실행하는 것이기 때문에 평범한 프로그램을 실행하고 종료하는 것과 같이 빠른 실행과 종료가 가능하다.
 - 위 장점들을 통해서 하나의 이미지에서 **개발과 테스트 배포** 모든 것이 **빠르고 신속하게** 이루어 질 수 있다.
 - Docker는 대부분의 운영 체제에서 지원을 하나 Linux Container 기술이기 때문에 Linux 이외의 환경, Mac, Windows에서는 VM위에서 동작하게 된다.
- Container의 장점
 - 하드웨어를 분리하지 않고 하나의 머신 위에서 동작하며 OS kernel도 공유한다.

- 하나의 서버에 여러 개의 Container를 실행하면 서로 영향을 미치지 않고 독립적으로 실행되어 가벼운 가상화머신을 사용하는 느낌을 준다.
 - 실행중인 Container에 접속하여 명령어를 입력할 수 있고 apt-get이나 yum으로 패키지를 설치할 수 있다.
 - 사용자도 추가하고 여러 개의 프로세스를 백그라운드로 실행할 수 있다.
 - CPU나 메모리 사용량을 제한할 수 있고 Host의 특정 포트와 연결하거나 Host의 특정 디렉토리를 내부 디렉토리인 것처럼 사용할 수 있다.
- Image
 - Container 실행에 필요한 파일과 설정 값 등을 포함하고 있는 것으로 상태 값을 가지않고 변하지 않는다.
 - Unbutu를 설치하기위해 필요한 image 파일을 생각하면 쉽다.
 - Container는 image를 실행한 상태라고 볼 수 있고 추가되거나 변하는 값은 container에 저장된다.
 - 같은 image에서 여러 개의 container를 생성할 수 있고 container의 상태가 바뀌거나 삭제되어도 image는 보존된다.
 - 새로운 서버가 추가되면 미리 만들어 놓은 image를 다운받고 container를 생성만 하면 된다.



- 그 외 추가적인 지식(<https://subicura.com/2017/01/19/docker-guide-for-beginners-1.html>)
 - 설치 : <https://www.docker.com/get-started>
-

Dockerfile

- Docker
 - Docker는 기본적으로 이미지가 있어야 컨테이너를 생성하고 동작시킬 수 있다.
- Dockerfile
 - 과제의 소개말에 나온 스크립트를 통한 업무의 자동화가 도커파일과 연관된 부분이라 생각하면 된다.
 - Dockerfile이란 필요한 최소한의 패키지를 설치하고 동작하기 위한 자신만의 설정을 담은 파일이며, 이 파일을 이용해 이미지를 생성(빌드)하게 된다.
 - Dockerfile을 build 명령어를 통해 image를 생성한다.(Makefile과 make 명령어와 비슷하다고 생각하자.)
 - 패키지 설치, 환경 변수 변경, 설정 파일 변경 등 다양한 작업을 수행하는 컨테이너를 만들어 설정 적용할 필요 없이 dockerfile을 사용하여 적용할 수 있다.
 - 실수로 인한 설정 누락 예방 등 다양한 장점이 있다.
 - Dockerfile 명령어에 대한 공부가 추가로 필요하다.

명령	설명
FROM	베이스 이미지 지정
RUN	명령 실행
CMD	컨테이너 실행 명령
LABEL	라벨 설정
EXPOSE	포트 익스포트
ENV	환경변수
ADD	파일/디렉토리 추가
COPY	파일 복사
ENTRYPOINT	컨테이너 실행 명령
VOLUME	볼륨 마운트
USER	사용자 지정
WORKDIR	작업 디렉토리
ARG	Dockerfile 에서의 변수
ONBUILD	빌드 완료 후 생성된 이미지가 다른 dockerfile에서 FROM 으로 불러질 때 실행되는 명령
STOPSIGNAL	시스템 콜 시그널 설정
HEALTHCHECK	컨테이너의 헬스 체크
SHELL	기본 셸 설정

Debian Buster

- OS 중의 하나.
- 데비안 GNU/리눅스라고도 알려져있다.
- 커뮤니티에서 개발한 무료 & 오픈소스 소프트웨어로 구성된 Linux 배포판 중 하나.

NGINX

- 무료로 제공되는 오픈소스 웹 서버 프로그램 중 하나.
 - wordpress를 이용해 만든 웹페이지 정도로 이해하면 된다.
- 웹 페이지에 서버 기능을 붙여 다른 컴퓨터 인터넷을 통해 접속할 수 있게 해주는 역할.
- apache의 C10K 문제(한 시스템에 동시접속자 수가 1만명이 넘을 때)를 해결하기 위해 개발 되었다.
- 특징

- 비동기, Event-Driven 구조로 만든 웹 서버 SW.
 - 사용 예
 - OSI 7 레벨인 Presentation Layer에서 NGINX 같은 웹 서버가 HTTP 통신을 제공.
 - 설치
 - apt install nginx 로 설치 가능.
 - service nginx start 명령어를 통해 실행.
-

php-fpm

- php
 - php란 동적 페이지를 위해 필요한 기술. 대표적인 서버 사이드 스크립트 언어.
 - wordpress 웹 페이지 또한 php기반으로 php가 설치되어야 정상적으로 동작이 가능하다.
 - php-fpm(FastCGI Process Manger)
 - 일반 CGI보다 빠른 처리가 가능한 FastCGI.
 - php-fpm을 통해 nginx와 php를 연동시켜 웹 서버가 정적 content 뿐만 아니라 동적 content를 다룰 수 있게 만드는 것.
 - CGI: Common Gateway Interface
 - 두 컴퓨터(서버-클라이언트) 사이의 HTML 등의 언어를 양방향으로 번역해주는 것.
 - nginx는 웹 서버이기 때문에 정적 content밖에 다루지 못한다.
 - 동적 페이지를 구현하기 위해서는 웹 서버 대신 동적 content를 읽은 후 html로 변환시켜 웹 서버에 다시 전달 해주는 외부 프로그램이 필요하다. → php module
 - 이러한 연결 과정의 방법 또는 규약을 정의한 것.
 - nginx의 문제를 해결하기 위해 nginx + php 조합에서 php-fpm을 사용하여 해석을 한다.
 - nginx는 기본적인 마크업 언어 파일(정적 페이지)만 해석 가능.
 - nginx는 php파일(동적 페이지 = 조건에 따라 변하는 페이지)은 해석 불가(아파치는 가능)
-

MySQL

- 오픈소스 관계형 데이터베이스 관리 시스템.
 - SQL : 데이터를 조작하고 조회, 관리하는 역할.
-

Wordpress

- 오픈소스 CMS(Contents Management System) 중 하나.
 - 웹 사이트(페이지)형태로 DB에 접근할 수 있는 콘텐츠 관리 시스템 - phpmyadmin과 같이 생각을 해보자.
 - 플러그인 아키텍처 및 템플릿 시스템(WordPress 내에서 테마라고 함)이 포함된다.
 - .org와 .com 두 가지로 나뉘는데 과제에서 사용하는 것은 wordpress.org에서 제공하는 설치형 블로그의 소스 파일.
-

Phpmyadmin

- php를 기반으로 생성된 mySQL의 GUI (Graphic User Interface) 로 웹에서 실행할 수 있는 프로그램.
 - 데이터베이스 관리 도구 중 하나.
 - 데이터베이스 클라이언트의 한 종류.
 - 웹 사이트(페이지) 형태로 접속이 가능.
 - 설치 전 php와 MariaDB 또는 MySQL이 설치되어 있어야 한다.
-

SSH & autoindex

- = **TLS**(Transport Layer Security, TSL 이전에 사용했던 명칭)
- HTTP 위에서 돌아가는 프로토콜.
- 서버<->클라이언트 간 인증에 사용.
- 암호화 키를 송수신 한다.
- SSL은 보안과 성능상의 이유로 2가지 암호화 기법(대칭키, 공개키)을 혼용해서 사용.
- autoindex

- autoindex를 알기위해서 먼저 웹 서버가 리소스 매핑과 접근을 어떻게 하는지 알아야 한다.
 - 웹 서버는 어떻게 수 많은 리소스 중 요청에 알맞는 content를 제공하는가?
 - 일반적으로 웹 서버 파일 시스템의 특별한 한 폴더를 웹 content를 위해 사용한다. 이 폴더를 문서루트 또는 docroot라고 부른다.
 - 리소스 매핑의 가장 단순한 형태는 요청 URI를 docroot 안에 있는 파일의 이름으로 사용하는 것이다.
 - 만약 파일이 아닌 디렉토리를 가리키는 url에 대한 요청을 받으면 요청한 url에 대응되는 디렉토리 안에서 index.html 또는 index.htm의 이름이 붙은 파일을 찾아 그 파일의 content를 반환한다.
 - 이를 autoindex라 부른다.
 - URI vs URL vs URN
 - <https://mygumi.tistory.com/139>
 - <https://nsinc.tistory.com/192>
-

HTTP & HTTPS

- HTTP
 - = Hypertext Transfer Protocol
 - = HTML 전송 통신규약.
 - 암호화 되지 않은 방법으로 데이터를 전송
 - 서버-클라이언트 간 오고가는 메시지 감청이 매우 쉽다.
 - HTTPS
 - = Hypertext Transfer Protocol Over Secure Socket Layer
 - HTTP + SSL = 보안이 추가된 HTTP.
 - SSL 위에서 HTTP의 평문 전송 대신 암호화된 통신을 하는 프로토콜.
 - HTTPS 통신을 서버에서 구현하기 위해서는 신뢰할 수 있는 상위 기관의 인증서가 필요하다.
 - 인증서를 발급해주는 기관을 CA(Certificate Authority)라고 한다.
 - CA의 인증서 발급은 무료가 아니므로 과제에서는 자체 인증서를 만든다.
-

- 출처

- <https://jaeseokim.github.io/Etc/docker기초부터-wordpress서비스-직접구현까지/>
- <https://velog.io/@hidaehyunlee/ftserver-총-정리-도커-설치부터-워드프레스-구축까지>
- <https://yeosong1.github.io/ftserver-풀이기록>
- <https://stitchcoding.tistory.com/2>
- <https://dphater.tistory.com/entry/ftserver-서버-돌려보기>

02_ft_server_explanation

02_Dockerfile_code

02_Docker 명령어

02_Dockerfile 문법