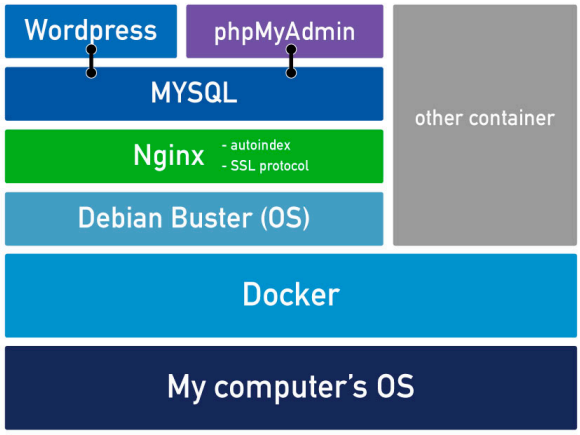
Hyeykim 2021년 3월 16일 화요일

ft_server

과제

- ft_server는 시스템 관리 개념을 소개하기 위한 과제입니다. 스크립트를 이용해 업무를 자동화하는 것의 중요성을 깨닫게 될 것 입니다. 이를 위해 '도커' 기술을 학습하고 완전한 웹 서버를 설치해 봅시다.
- 서버는 동시에 여러 서비스를 실행 할것입니다 : 워드프레스 웹 사이트, phpadmin, mySQL database.

- 딱 하나의 도커 컨테이너에 Nginx가 있는 웹 서버를 설정해야합니다. 컨테이너 OS는 꼭 debian buster여야합니다
- 웹 서버는 여러 서비스를 동시에 실행할 수 있어야합니다.
- 그 서비스들은 WordPress 웹 사이트, phpMyAdmin, MySQL 입니다.
- SQL 데이터베이스가 WordPress 및 phpMyAdmin과 작동하는지 확인해야합니다.
- 서버는 SSL 프로토콜을 사용할 수 있어야합니다.
- URL에 따라 서버가 올바른 웹 사이트로 리디렉션되는지 확인해야합니다.
- 서버가 오토 인덱스로 실행 중인지 확인하고, 이를 비활성화 할 수 있어야합니다.



- 위에 과제들을 정리하면
- 내 컴퓨터 OS에 도커라는 프로그램을 설치하고 이 도커 사용해 가상 환경(컨테이너)에 '데비안'이라는 OS를 설치하고 그 설치된 환경에 웹서버를 구축을 할 것이다.
- 그 웹서버는 Nginx를 통해 구축 할것이며, 웹서버에서는 My SQL을 이용해 제작한 데이터 베이스에 접근 할 수 있는 워드프레스 웹사이트와 phpmyadmin을 구동 할수 있어야 한다.

Hyeykim 2021년 3월 16일 화요일

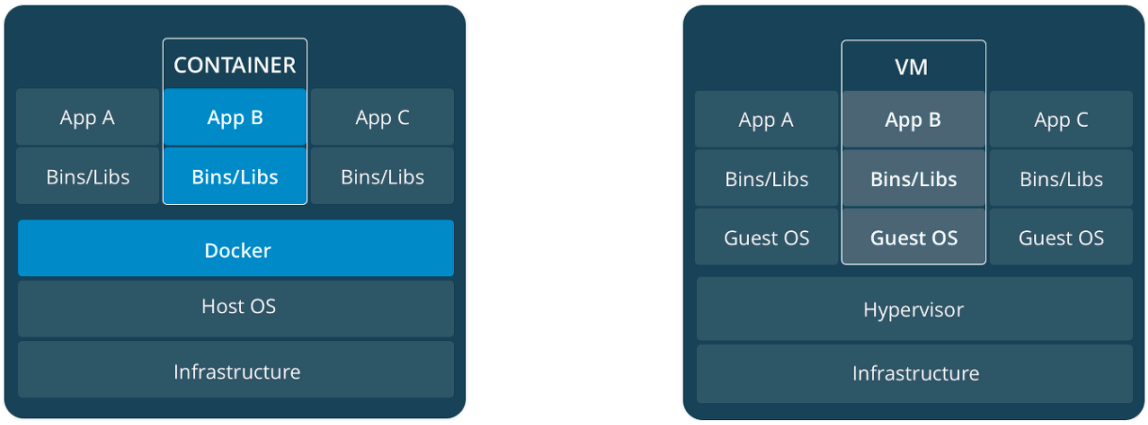
목차

Hyeykim 2021년 3월 16일 화요일

I. 도커에 대해 이해하고 설치하기

0. 도커(Docker)소개 및 개념

- 일반적으로 [1]가상머신이라고 하면, Vmware나 VirtualBox가 있다
- 이 둘은 호스트형 가상화 기법을 사용하는데, Host OS에 가상 머신을 설치하고, 가상 머신에서 가상 하드디스크 공간을 만들어 운영체제(OS)를 설치하면, 한 개의 운영체제 위에서 또 다른 운영체제를 사용할 수 있다.
- 이 때 가상 공간에 설치된 운영체제는 Host PC의 운영체제와 독립되어 실행 된다. 가상 머신은 Guest OS가 구동될 수 있도록, 필요한 하드웨어 자원들을 가상으로 [2] Guest OS에게 제공하고, 모든 처리는 가상머신이 담당한다.
- 그런데, 호스트형 가상화 기술을 사용하면 Host OS에서 Guest OS를 구동하는 방식이기 때문에 [3]오버헤드가 크다.



좌 - Docker / 우 - Virtual Machine

- 도커는 앞에서 설명한 방식과 다른 방식을 사용하는데
- 도커의 구조를 살펴 보면 오른쪽 그림과 달리 Guest OS가 존재 하지 않는 데, 도커는 OS를 통채로 가상화 하는 것이 아니라 Host OS위에서 유저공간을 가상화 한다.
- 왼쪽 그림에 APP B를 실행시키기 위한 필요한 [4]바이너리와 라이브러리들을 묶어서 통채로 격리 시킨다.

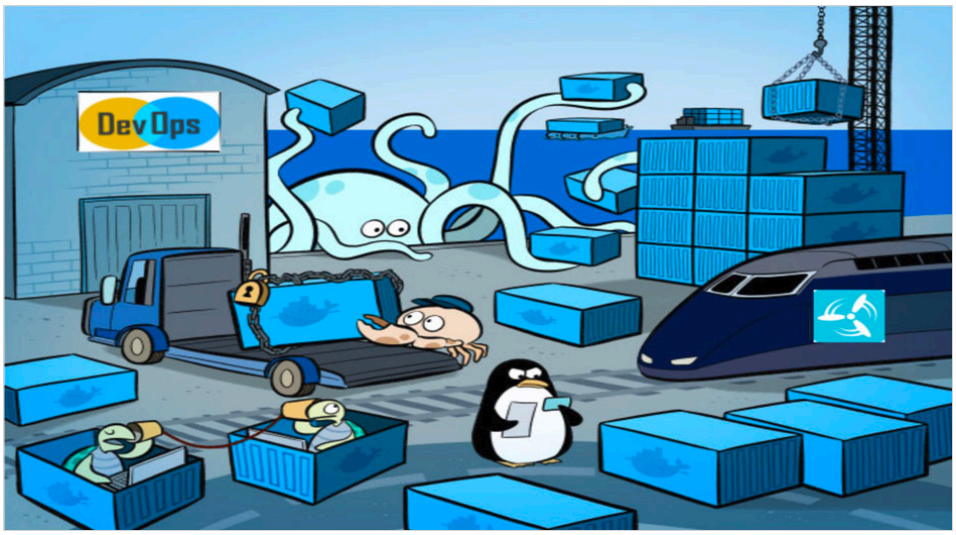
---

[1] 가상 머신(Virtual Machine, VM)은 컴퓨팅 환경을 소프트웨어로 구현 한 것, 즉 컴퓨터를 에뮬레이션 소프트웨어다. 가상머신상에서 운영체제나 응용프로그램을 설치 및 실행 할수 있다.

[2] 다른 프로그램이나 장치를 모방(시스템을 복제)하는 능력이다. 애뮬레이션은 소프트웨어로 생성한 장치가 물리적인 다른 장치라고 믿도록 속이는 것이다.

[3] 어떤 처리를 하기 위해 들어가는 간접적인 처리 시간, 메모리등을 말한다.

[4] 바이너니나 라이브러리 뿐만 아니라 OS의 파일시스템까지도 격리 가능하다.



- 이렇게 가상화를 할 경우, 프로그램은 마치 다른 환경에서 실행되고 있는 것 처럼 실행 되지만 실제 커널은 한 Host OS를 공유하고 있기 때문에 시스템 자원의 접근이 용이하기 때문에 속도가 일반적인 가상머신에 비해 월등히 빠르다.
- 그리고, 이렇게 가상화 된 유저 공간(ex.그림속 App a)을 **컨테이너(Container)**라고 부른다.
- 도커 (Docker)는 항만 노동자라는 뜻으로, 항구에 있는 컨테이너들을 관리하는 사람들을 일컫는다. 이름에서 알 수 있듯이, 도커는 컨테이너를 관리하는 플랫폼이다.

1. 도커 용어들

도커 이미지*(Docker Image)* : 컨테이너(Container)를 실행하기 위해서 필요한 파일, 프로그램, 라이브러리, 설정 등을 묶어서 만든 파일. 이미지를 실행하면 컨테이너가 된다. 윈도우에서 .exe 파일을 생각하면 이해가 쉬울 것 같다.

도커 컨테이너*(Docker Container)* : 이미지를 실행한 상태이다. 이미지 시작하면 프로세스 형태로 컨테이너가 실행된다. 각각의 컨테이너는 서로 독립적으로 작동 할 수 있다.

레이어*(Layer)* : 도커 이미지는 레이어 형태로 구성되어있다. 예를들어 웹서버 이미지의 경우, 웹 서버 프로그램이 동작할 OS의 파일시스템(ex ubnutu, debian..) 이 하나의 레이어가 되고, 이 파일 시스템 위에서 동작할 웹 서버 프로그램(*ex,, nginx, apache...)* 이 또다른 하나의 레이어가 된다. 그리고 사용자에게 보여줄 웹 페이지 소스파일들이 또 하나의 레이어가 될 수 있다. 즉 이 웹 서버 이미지는 파일시스템 + 웹 서버 프로그램 + 소스코드 레이어로 이루어진 이미지라고 볼 수 있다. 이렇게 구성할 때의 장점은, 이미지 일부를 수정해야할 때, 해당 레이어만 업데이트 하면 되기 때문에 오버헤드가 비교적 덜 발생한다.

도커 엔진*(Docker-Engine)* : 도커 엔진은 컨테이너를 실행,중지 및 이미지 빌드 등 전반적인 실행에 관여하는 도구이다. 일반적으로 도커라고 하면 이 도커 엔진을 일컫는 경우가 많으며 실제로 우리가 설치할 프로그램은 도커 엔진이다.

도커 허브*(Docker Hub)* : github을 알고있다면 이해가 쉽다. 사용자가 만든 도커 이미지를 배포할 수 있는 저장소로, 무료로 이용할 수 있으며, 저장소에 push 할 경우, 다른 사람들도 도커 이미지를 받아서 사용할 수 있다. 프라이빗(private)으로는 하나만 업로드 가능하다. 대신 개인 서버가 있는 경우 도커 레지스트리 서버를 구성해서 전용 도커 허브를 만들 수 있다.

2. 도커 명령어

- **도커 버전 확인**

  docker -v

- **도커 이미지 다운받기**

  docker pull {이미지명} : {태그} // 태그는 필수 x

- **컴퓨터 내 도커 이미지들 보기**

  docker images

- **이미지로 컨테이너 생성하기**

  docker create {옵션} {이미지명} : {태그}

- **만들어진 컨테이너 시작하기**

  docker start {컨테이너 id 또는 이름}

- **컨테이너로 들어가기**

  docker attach {컨테이너 id 또는 이름}

- **이미지를 다운 받아 (없을 시에만) 바로 컨테이너 실행하여 진입하기**

  docker run {이미지명} : {태그} // pull, creat, start, attach를 한꺼번에 실행 하는 것과 같음

옵션	설명
-d	데몬으로 실행(뒤에서 - 안보이는 곳 (백그라운드)에서 알아서 돌라고 하기)
-it	컨테이너로 들어갔을때 bash로 CLI 입출력을 사용할 수 있도록 해 줍니다.
- -name {이름}	컨테이너 이름 지정
-p	호스트와 컨테이너의 포트를 연결합니다
- -rm	컨테이너가 종려되면 {내부에서 돌아가는 작업이 끝나면}컨테이너를 제거합니다
-v	호스트와 컨테이너의 디렉토리를 연결합니다

- **동작중인 컨테이너 재 시작**

  docker restart {컨테이너 id 또는 이름}

- **도커 컨테이너의 내부 쉘에서 빠져나오기 (컨테이너 종료)**

  exit 또는 ctrl + d

- **도커 컨테이너의 내부 쉘에서 빠져나오기 (컨테이너를 종료 하지 않음) :** Ctrl + P, Q

- **(동작중인) 컨테이너들 보기**

  docker ps

- **컨테이너 삭제**

  docker rm {컨테이너 id 또는 이름}

  docker rm ‘docker ps -a -q’ // 모든 컨테이너 삭제

- **이미지 삭제**

  docker rmi {옵션} {이미디 id} // 컨테이너 있을 시 강제 삭제 -f

- **모든 컨테이너와 이미지 등 도커 요소 중지 및 삭제**

  docker stop $(docker ps -aq) // 모든 컨테이너 중지

  docker system prune -a // 사용되지 않는 모든 도커 요소 삭제

- **도커파일로 이미지 생성**

  docker build {경로}

  옵션 -t 저장소 이름, 이미지 이름, 태그를 설정

- **도커 컴포즈 실행**

  docker-compose up

- **도커 이미지로 컨테이너 만들기**

  docker run {옵션} 이미지 이름, ID{명령}{매개 변수}

3. 도커 설치 (클러스터 기준)

1. Managed Software Center에 들어간다.
2. Software 탭에서 Docker를 install 한다.
3. 42toolbox를 이용 git clone https://github.com/alexandregv/42toolbox.git ~/42toolbox
4. echo "source ~/42toolbox/shell_utils.sh" >> ~/.zshrc
5. sh init_docker
6. 터미널 창에 docker ps해서 정상작동 하는지 확인

II. 도커로 웹서버 만들기

0. 데비안 소개 및 도커(Docker)에 데비안 버스터 설정

- **데비안** : 우분투 같은 리눅스 OS 종류 중에 하나. 최신 버전은 10.0(Buster)이다.
- 특징 : 패키지 설치와 업그레이드 혹은 다른 패키지의 의존성등을 apt를 이용하여 쉽게 설정 할수 있다. 그에따라 서버에 알맞은 리눅스가 된다.
- 목적 : 서버
- 장점 : 배포되고 있는 리눅스중 서버의 안정성이 높다. 그리고 포함하고 있는 패키지도 많다.

- 도커로 데비안 이미지 다운 받기 : docker pull debian:buster

  -> docker images 입력해서 확인

- 이미지를 컨테이너화 시키기 : docker run -it -p 80:80 -p 443:443 debian:buster
  - run의 기본 형태는 docker run {옵션} 이미지 이름, ID {명령}{매개 변수} 이다.
  - -i 옵션은 표준 입출력을 활성화 시켜 bash에서 명령어를 입력 받는다. 같이 붙어 있는 -t 는 bash를 사용 할려면 써야하는 옵션이다. 사용하지 않는다면 명령어를 입력할 순 있어도 셸에 표시 되지 않는다.
  - -p 는 –publish 의 약자인데, 호스트에 연결된 컨테이너의 특정포트를 외부에 노출시킨다. 보통 웹서버의 포트를 노출 할때 사용한다. http는 80, https는 443가 기본 포트다.
  - 그냥 docker run -it debian 이라고 하면 자동으로 도커 허브에서 데비안 버스터로 최신 버전을 불러온다
  - `docker --name 컨테이너이름 run -it debian:buster` 이런 식으로 네임 옵션을 안 주면 도커 데몬이 형용사+과학자이름?을 랜덤으로 짜서 지어준다.
- Bash 입장 확인 : 현재 위치가 `root@bda50ea6eb7e:/#` 이런 식으로 바뀐다. 그러면 데비안 bash에 들어가진것이다.

1. Nginx 소개 및 도커(Docker)와 데비안 버스터에 nginx설치

• **Nginx** : 무료로 제공되는 오픈소스 웹 서버 프로그램이다.

• 규모가 작은 서비스이면서 정적 데이터 처리가 많은 서비스에 적합하다고 한다.

• **웹서버** : 클라이언트로 부터 요청이 발생시 요청에 맞는 정적 콘텐츠를 보내주는 역할

• 하는 일 : 1. 커넥션을 맺는다 (클라이언트의 접속을 받아들이거나, 원치 않는 클라이언트라면 닫는다

2. 요청을 받는다 (HTTP 요청 메세지를 네트워크로부터 읽어들인다)

3. 요청을 처리한다 (요청 메세지를 해석하고 행동을 취한다)

4. 리소스에 접근한다 (메세지에서 지정한 리소스에 접근한다)

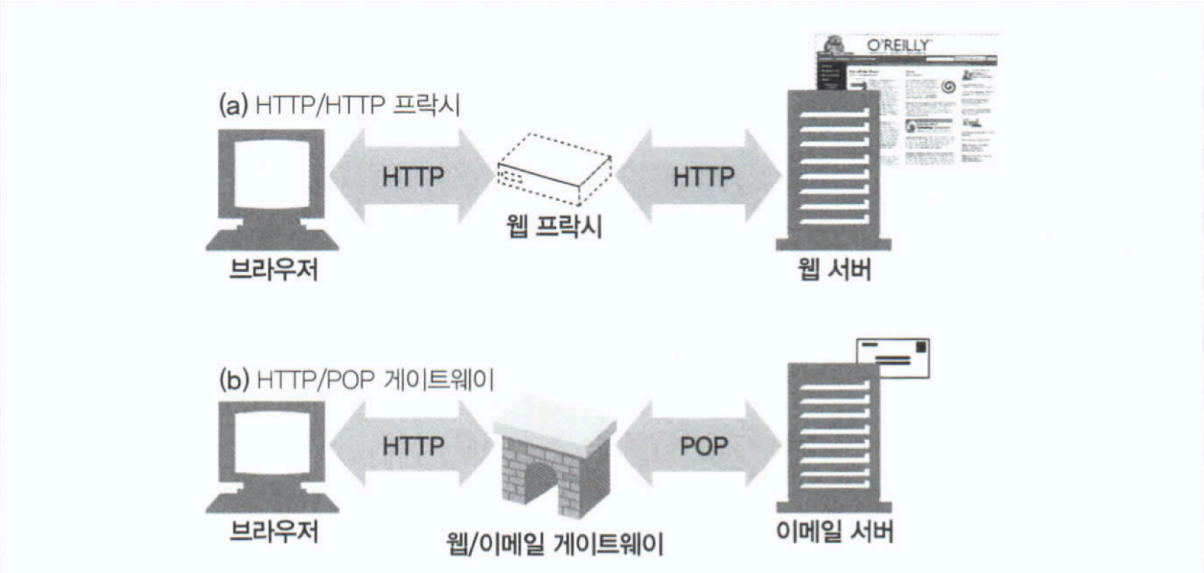
5. 응답을 만든다 (올바른 헤더를 포함한 HTTP 응답 메세지를 생성한다)

6. 응답을 보낸다 (응답을 클라이언트에게 돌려준다)

7. 트랜잭션을 로그로 남긴다 (로그파일에 트랜잭션 완료에 대한 기록을 남긴다)

• **Proxy** : Nginx는 일반적인 HTTP의 웹서버의 역할 외에도 proxy, [5]reverse proxy(대리 프락시) 서버의 역할 또한 가능하다.

• 웹 프락시 서버는 클라이언트와 서버 사이에서 트랜잭션을 수행하는 중개인이며, 같은 프로토콜을 사용하는 둘 이상의 애플리케이션을 연결한다.



의 위치를 찾아내기 위해 나른 서버와 커뮤니케이션을 시작한다. 대리 프락시는 공용 콘텐츠에 대한 느린 웹 서버의 성능을 개선하기 위해 사용될 수 있다. 이런 식으로 사용되는 대리 프락시를 흔히 서버 가속기라고 부른다.

- 프락시는 보안을 개선하고, 성능을 높여주며, 비용을 절약한다. 그리고 프락시 서버는 모든 HTTP트래픽을 감수하고 수정할 수 있다.
- 프락시는 다음과 같은 일을 한다.
  - 어린이 필터
  - 문서 접근 제어자
  - 보안 방화벽
  - 웹캐시

- Ngnix 설치
- 데비안에서는 패키지 관리자로 apt-get을 쓴다.
- apt-get -y update 해서 일단 패키지 목록을 최신으로 받는다.
- apt-get -y install nginx해서 nginx설치

- 서버연결 확인
- service nginx start - 엔진엑스 시작
- service nginx status - 잘 도는지 확인
- localhost 혹은 localhost:80에 들어가서 인터넷 브라우저 확인
- Welcome to nginx!가 나오면 성공

- 서버 응답관련 오류 발생시 체크해 볼 것
- service nginx status하면 연결이 잘 되었는지 알려준다.
- curl 127.0.0.1:80, curl localhost 등 curl을 사용해서 터미널 창에서 해당 주소의 페이지 내용을 텍스트 형식으로 볼 수 있다.
- lsof -Pni4 | grep LISTEN 연결상태인 포트 확인
- lsof -i :[포트 번호] 특정 포트 사용 상태 보기. 비사용중이면 아무것도 안나온다.
- lsof -i :[포트 번호] 했을 때 아무것도 안나오는데 이미 할당중이라고 나온다면.. sudo lsof -i :[포트 번호]..
- kill -9 [프로세스 번호] 위 명령에서 발견한 활성 포트 죽이기
- ping 127.0.0.1 이런 식으로 특정 IP가 응답중인지 알 수 있다.

2. HTTPS와 SSL 소개 및 SSL 인증서 만들기

• **SSL** : 인터넷 상에서 데이터를 안전하게 전송하기 위한 인터넷 프로토콜

- 인터넷 사용자들에게 안전한 개인 정보를 교환하기 위한 표준 프로토콜로 인정되어 많은 온라인 성거래에 사용되고 있다.
- 다양한 장점을 지닌 암호화 기법들을 사용해 세계 각국에서 사용되는 대부분의 암호화 기법을 지원할 수 있다.
- SSL은 크게 3가지 기능들을 제공함으로 공개되어 있는 인터넷상에서 일어나는 [6]트래잭션의 기밀성을 보장한다.
    - Site authentication : 유저가 선택한 상대편 웹사이트를 인증한다
    - Data privacy : 전달되는 데이터가 도중에 누군가에 의해 판돈되지 않는다는 것을 보장한다. Ssl은 다양한 암호화 알고리즘을 사용하여 인터넷을 통해 전송되는 개인의 사적인 정보를 외부로 부터 불법적인 판독을 막는다.
    - Data Integrity : 사용자의 브라우저로부터 상대방 웹서버까지 전달되는 동안 데이터가 도중에 누군가에 의해 변경되지 않도록 보장한다.

• **HTTPS** : ssl위에서 돌아가는 HTTP의 평문 전송 대신에 암호화된 통신을 하는 프로토콜

- 이런 HTTPS를 통신을 서버에서 구현하기 위해서는 신뢰할 수 있는 상위 기업이 발급한 인증서가 필요하다
- self-signed SSL 인증서는 자체적으로 발급받은 인증서이며, 로그인 및 기타 개인 계정 인증 정보를 암호화한다. 당연히 브라우저는 신뢰할 수 없다고 판단해 접속시 보안 경고가 발생한다.
- self-signed SSL 인증서를 만드는 방법은 몇 가지가 있는데, 무료 오픈소스인 openssl 을 이용해 쉽게 만들수 있다.
- HTTPS를 위해 필요한 개인키(.key), 서면요청파일(.csr), 인증서파일(.crt)을 openssl이 발급해준다

• openssl로 self-signed SSL 인증서 만들기

- 인증서 만드는 방법

1. Self-signed 인증서
    * CSR 명시적 생성 -> 인증서에 self-sign -> 인증서 완성
    * CSR을 명시적으로 생성하지 않고, key와 부가정보들을 입력하여 직접 self-sign 하여 인증서 완성

2. CSR (인증서 서명 요청)을 만들어 CA에 요청해서 발급받는 방법
    * 유료
    * 무료 (ex: Letsencrypt)

---

[6] 데이터베이스의 상태를 변화시키는 하나의 논리적 기능을 수행하기 위한 작업 단위 또는 한꺼번에 모두 수행되어야 할 일련의 연산들을 의미한다.

이중 첫번째 방법을 사용할 것이다.

- **openssl 설치** : apt-get -y install openssl vim //vim은 수정용으로 깔아두기
- [7]**개인키 및 [8]CSR 만들기**
- Csr생성 기본 틀 : openssl req -new -key 키이름.key -out csr 이름.csr
- 자체 서명 인증서생성 : openssl req -new -x509 -nodes -sha1 -days 365 -key server.key -out server.crt
- openssl req -newkey rsa:4096 -days 365 -nodes -x509 -subj "/C=KR/ST=Seoul/L=Seoul/O=42Seoul/OU=Lee/CN=localhost" -keyout localhost.dev.key -out localhost.dev.crt
- mv localhost.dev.crt etc/ssl/certs/
- mv localhost.dev.key etc/ssl/private/
- **권한 제한** : chmod [9]600 etc/ssl/certs/localhost.dev.crt etc/ssl/private/localhost.dev.key

- 옵션 뜻
- .csr 인증사인 요청파일
- .crt 인증서 파일
- -days 유효 일수
- -nodes 생략시 재부팅할때마다 수동으로 암호를 입력해야함

사용시 표기	의미	내용
CN	Common Name	일반 이름 (인증서 고유 이름). 대부분의 인증기관 CA에서는 SSL인증서 신청시에 도메인명을 CN으로 지정.
O	Organization	기관명
OU	Organization Unit	회사/기관 내의 '사업부, 부문, 부서, 본부, 과, 팀' 정도.
L	City / Locality	시 / 도
S	State / County / Region	구 / 군
ST	Street	나머지 상세 주소. (OV,EV 인증시에만 필요)
C	Country	국가를 나타내는 ISO 코드를 지정. 한국은 KR, 미국은 US 등 2자리 코드

---

[7] 참고 하면 좋을 것 같은 사이트 https://losskatsu.github.io/it-infra/ssl-auth/#ssl-%EC%9E%91%EB%8F%99-%EA%B3%BC%EC%A0%95

[8] 인증서명요청. Certificate Signing Request

[9] rw-r—r-- 에서 각 3개씩 끊어보면 user, group, others로 3개의 권한으로 나눠지고 이를 **8진수**로 표현하면 **rwx** 권한을 다 갖는 경우 **7**이 된다. 각 권한을 숫자값으로 표현하면 r = 4, w = 2, x = 1

Hyeykim 2021년 3월 16일 화요일

- nginx에 ssl을 더하기 위한 default 파일 설정 변경
- vim etc/nginx/sites-available/default해서 아래와 같이 수정

```
server {
        listen 80 default_server;          // listen : 서버에서 라우팅 할 특정 port를 정의한다
        listen [::]:80 default_server;
                    // default_server : 여러개의 서버블록을 작성할 때 단 하나의 서버블록에만 존재해야 한다.
                별로의 지정하지 않은 도메인으로 들어오는 다른 모든 요청에 대해 해당 블록이 처리함을 의미한다.
}

server {
                listen 443;

                ssl on;
                ssl_certificate /etc/ssl/certs/localhost.dev.crt;
                ssl_certificate_key /etc/ssl/private/localhost.dev.key;

                root /var/www/html;
                index index.html index.htm index.nginx-debian.html;
                ...
        }
```

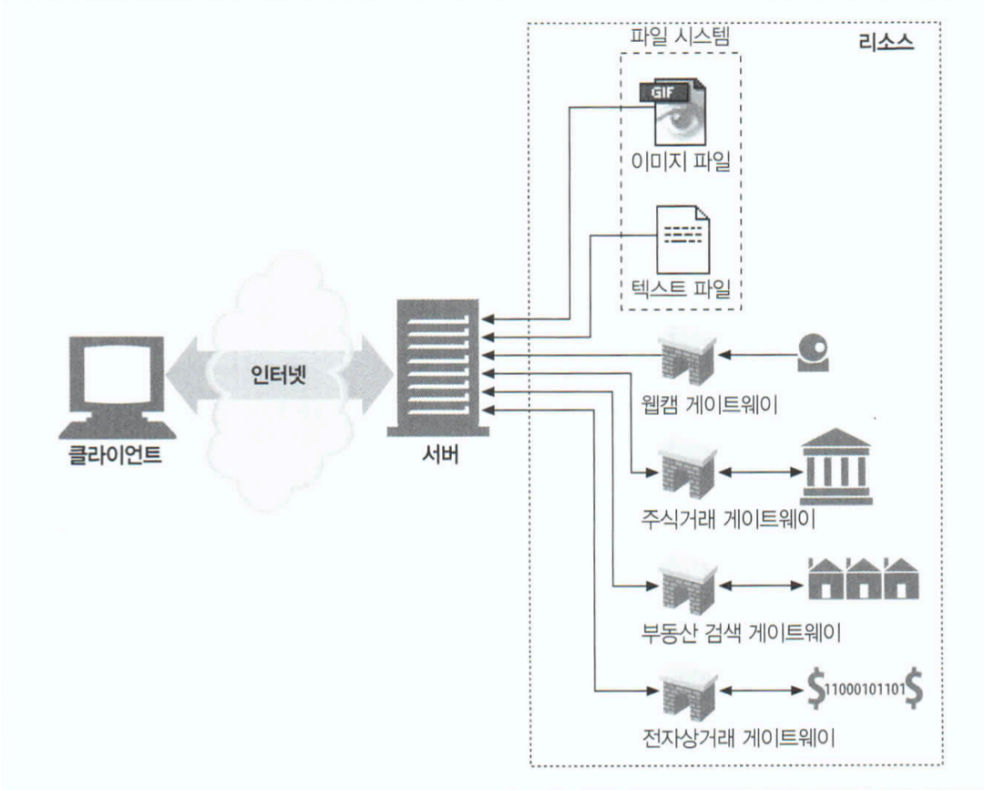
- service nginx reload 혹은 service nginx restart해서 수정사항 적용

만약 크롬에서 localhost로 들어갔을때 안전하지 않음하고 안 들어가진다면 thisisunsafe를 치면 된다!

3. MySQL, phpMyadmin 소개 및 설치

- **MySQL** : 관계형 데이터 베이스 관리 시스템
- Database : 어느 한 조직의 여러 응용 시스템이 공유할 수 있도록 통합. 저장된 운영 데이터들의 집합, 데이터를 추가하거나 검색, 추출하는 기능이 있다
- 관계형 데이터 베이스 관리 시스템
- 서로 관계를 맺는 여러 개의 테이블에 항목을 나누어 저장하고, 필요한 부분만 추출해서 사용할 수 있다.
- MYSQL은 이런 관계형 데이터 베이스 시스템중 하나

- **phpMyadmin** : php를 기반으로 생성된 mysql의 [10]gui로서 웹에서 실행할 수 있는 프로그램이다.
- **php** : 대표적인 서버 사이드 스크립트 언어
- **CGL(공통 게이트웨이 인퍼페이스)** : nginx는 웹서버이기 때문에 정적 콘텐츠밖에 다루지 못한다. 동적 페이지를 구현하기 위해서는 웹 서버 대신 동적 콘텐츠를 읽은 뒤 html로 변환시켜 웹 서버에게 다시 전달해주는 **외부 프로그램(php 모듈)**이 필요하다. 이런 **연결 과정의 방법 혹은 규약을 정의한 것이 CGI**이다.



[10] 그래픽 사용자 인터페이스 : 사용자가 편리하게 사용할 수 있도록 입출력 등의 기능을 알기 쉬운아이콘으로 나타낸것

- **php-fpm (PHP FastCGI Process Manager)**: 일반 GCI 보다 빠른 처리가 가능한 FastGCI. 정리하자면, php-fpm을 통해 nginx와 php를 연동시켜 우리의 웹 서버가 정적 콘텐츠 뿐만 아니라 동적 콘텐츠를 다룰 수 있도록 만드는 것이다

- Ngnix에 php-fpm 연동
- apt-get -y install php-fpm
- /etc/nginx/ 구성 살펴보기
  - sites-available = 설정 파일들이 들어있다.
  - sites-enabled = 실행시킬 파일들만 symlink로 연결해서 여기에 넣어둔다.
  - nginx.conf = sites-enabled에 있는 파일들을 호출하는 파일이다. 서버 실행에 관한 정보를 적어둔다.

- Nginx와 php-fpm 연동을 위한 default 파일 내용 수정
- vim /etc/nginx/sites-available/default
- 아래와 같이 주석 해제하고 php**7.3**-fpm.sock; 이 부분이 설치한 PHP 버전과 일치하는지도 확인하기

```
location ~ \.php$     // location은 .php확장자로 끝나는 요청을 처리기 위한 부분이다.
{
        include snippets/fastcgi-php.conf;
        ## With php-fpm (or other unix sockets):
        fastcgi_pass unix:/var/run/php/php7.2-fpm.sock;
         ## With php-cgi (or other tcp sockets):
         #fastcgi_pass 127.0.0.1:9000;
}
```

- index index.html index.htm index.nginx-debian.html 뒤에 index.php를 추가한다.

- php-fpm 작동 확인
- service php7.3-fpm start
- service php7.3-fpm status

Hyeykim 2021년 3월 16일 화요일

- phpinfo() 함수로 nginx와 php-fpm 연동 잘 되는지 확인
- `/var/www/html/` 위치에 `phpinfo.php`를 만들고(이름 다르게 테스트해도 됨)
- <?php phpinfo(); ?> 코드를 입력, 저장.

- service nginx reload 혹은 service nginx restart해서 수정사항 적용시키기.
- 만약 restart, reload에 실패한다면 cat /var/log/nginx/error.log 해서 오류내역을 볼 수있다.
- 웹브라우저로 내서버아이피/phpinfo.php로 접속했을 때 아래와 같이 phpinfo페이지가 나오면 된 것.

- MySQL 설치
- 데비안 9부터 MySQL -> MariaDB를 디폴트로 사용하게 한대서 (데비안 버스터는 데비안 10이다) mariadb를 설치했다.
- apt-get -y install mariadb-server php-mysql

- phpmyadmin 설치 및 압축해제
- 데비안에 phpmyadmin을 바로 다운로드 할 수 있게하는 패키지는 현재 없다.
- `wget`으로 직접 다운로드 한다. (phpmyadmin 다운로드 사이트에서 다운로드 버튼의 링크 주소를 복사, wget [주소])
- 압축해제 후 폴더명을 phpmyadmind으로 바꿔서 /var/www/html/에 위치 시킨다.
- apt-get install -y wget
- wget https://files.phpmyadmin.net/phpMyAdmin/5.0.2/phpMyAdmin-5.0.2-all-languages.tar.gz
- tar -xvf phpMyAdmin-5.0.2-all-languages.tar.gz
- mv phpMyAdmin-5.0.2-all-languages phpmyadmin
- mv phpmyadmin /var/www/html/

- phpmyadmin 설정
- phpmyadmin/config.sample.inc.php 파일을 복사해 config.inc.php를 만든다.
- config.inc.php에 블로피시 암호를 만들어 넣는다.
- create_tables.sql을 가져와서 phpMyAdmin을 위한 테이블을 만든다.
- cp -rp var/www/html/phpmyadmin/config.sample.inc.php var/www/html/phpmyadmin/config.inc.php

- vim var/www/html/phpmyadmin/config.inc.php

- 블로피시 암호 생성 사이트에서 생성한 암호를 복사해서
- $cfg['blowfish_secret'] = '이 부분에 넣는다'; /* YOU MUST FILL IN THIS FOR COOKIE AUTH! */
- service nginx reload
- service mysql start
- service php7.3-fpm restart (왜지?)
- mysql < var/www/html/phpmyadmin/sql/create_tables.sql -u root - -skip-password.
-
- mysqladmin -u root -p password
- 기존 패스워드 없으니 엔터 입력
- 새 패스워드 입력
- 한 번 더 입력
- mysql
- show databases;
- CREATE DATABASE IF NOT EXISTS wordpress; // 워드프레스를 위한 DB 만들기
- show databases;
- exit

- phpmyadmin 작동 확인
- service mysql start
- localhost/phpmyadmin 아이디 root, 비밀번호는 아까 만든 그 비밀번호. 로그인 해보기.

만약 Phpmyadmin으로 접속하는데

: Your Composer dependencies require the following PHP extensions to be installed: xml

라는 에러가 생기면 apt install php-xml를 해보자!

: root로 로그인이 안된다면 새계정을 만들어서 그 계정에게 모든 권환을 넘겨주면된다.

Mysql

create user 'hyeykim'@'%' identified by ‘ ’;

grant all privileges on *.* to ‘hyeykim'@'%'

4. Wordpress에 대해 이해하고 Wordpress 설치하기

- **Wordpress** : 세계적으로 사랑받고 있는 CMS
- CMS : 말그대로 콘텐츠를 관리하는 시스템
- 이용목적 : 웹사이트를 만들 때 코드를 하나하나 넣어서 만들수도 있지만, cms를 이용하면 웹사이트의 다양한 리소스 및 콘텐츠, 데이터를 쉽게 관리 할수 있다.
- wget https://wordpress.org/latest.tar.gz
- tar -xvf latest.tar.gz
- mv wordpress /var/www/html/
- chown -R www-data:www-data /var/www/html/wordpress
  - chown: 리눅스에서 소유자를 변경하는 커맨드.
    - -R은 –recursive. 에러 메시지가 있어도 출력하지 않게 하는 커맨드.
    - www-data는 우분투에서 Apache,PHP 실행시 수정이 가능한 권한
- Wordpress 설정

```
// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress' );

/** MySQL database username */
define( 'DB_USER', 'root' );

/** MySQL database password */
define( 'DB_PASSWORD', ‘hyeykim' );

/** MySQL hostname */
define( 'DB_HOST', 'localhost' );

/** Database Charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );

/** The Database Collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );
```

- Wordpress 작동 확인
- service nginx reload -> localhost/wordpress 접속

6. autoindex소개및 nginx에 autoindex추가하기

- autoindex 가 뭔지 알고싶다면 먼저 웹서버가 **리소스 매핑과 접근**을 어떻게 하는지 부터 알아야한다.
- 웹서버 파일 시스템의 특펼한 한 폴더를 웹 콘텐츠를 위해 사용한다. 이를 문서루트 혹은 docroot라고 부른다. 리소스 매핑의 가장 단순한 형태는 요청 url를 dotroot안에 있는 파일 이름으로 사용 것이다.
- 만약 파일이 아닌 디렉토리를 가리키는 url에 대한 요청을 받았을 때는, 요청한 url에 대응되는 디렉토리 안에 `index.html` 혹은 `index.htm`으로 이름 붙은 파일을 찾아 그 파일의 콘텐츠를 반환한다. 이를 **autoindex** 라고 부른다.

- nginx에 autoindex추가 하기
- vim etc/nginx/sites-available/default에 autoindex on;을 추가한다.

```
server_name _;


location / {
        # First attempt to serve request as file, then
        # as directory, then fall back to displaying a 404.
        autoindex on;
        try_files $uri $uri/ =404;
//nginx는 정적 파일 호스팅을 기본적으로 지원하지 않기에 root  폴더내에 uri에 따른 폴더가 있는지 살펴 보고 없다면 404에러를 보여줍니다.
}
```

7. url redirection 추가

- http 주소를 https 주소로 들어오도록 리다이렉션 시킨다..
- 상태 코드 301 = 리다이렉션 할 때. 코드 별로 각각 다른 뜻이 있고, 대강 아래와 같다.
  - 상태코드(100-101) : **정보**
  - 상태코드(200-206) : **성공**
  - 상태코드(300-305) : **리다이렉션** - 리소스가 옮겨졌음
  - 상태코드(400-415) : **클라이언트 에러** - 클라이언트에서 뭔가 잘못된 요청을 했음

- 상태코드(500-505) : **서버 에러** - 서버에서 뭔가 실패했음

- 서버 블록을 나눠서 return 301 https://$host$request_uri;
- vim etc/nginx/sites-available/default 내용 최종으로 수정

```
server {
        listen 80 default_server;
        listen [::]:80 default_server;

        return 301 https://$host$request_uri;
}
server {
                listen 443;
                ssl on;
                ssl_certificate /etc/ssl/certs/localhost.dev.crt;
                ssl_certificate_key /etc/ssl/private/localhost.dev.key;
                root /var/www/html;
                index index.html index.htm index.php;
                server_name _;
                location / {
                        autoindex on;
                        try_files $uri $uri/ =404;
                }
                location ~ \.php$ {
                        include snippets/fastcgi-php.conf;
                        fastcgi_pass unix:/var/run/php/php7.3-fpm.sock;
                }
}
```

7. 도커 파일 만들기

- ***도커 파일(Docker File) :*** 도커 *build* 명령어와 함께 쓰이는 파일로 도커 이미지를 명령어 하나로 자동으로 생성할 수 있게 해주는 파일이다.

```
FROM debian:buster

RUN apt-get -y update
RUN apt-get -y install nginx openssl php-fpm mariadb-server php-mysql wget

ADD srcs /.

EXPOSE 80 443

CMD bash run.sh
```

- **FROM** : 생성할 이미지의 베이스가 될 이미지를 뜻합니다. 반드시 한번 이상 입력해야 합니다.
- **LABEL :** 이미지에 메타[11]데이터를 추가합니다. (나중에 원하는 조건의 컨테이너, 이미지 등을 쉽게 찾을 수 있도록 도와주기 때문에 기억해두는게 좋습니다)
- **RUN :** 이미지를 만들기 위해 컨테이너 내부에서 명령어를 실행합니다. (항상 apt-get update와 apt-get install는 같은 RUN실행중에서 동시에 실행해 [12]캐싱 문제를 방지)
- [13]ADD : 호스트 OS의 파일 또는 디렉토리를 컨테이너 안에 포함 된다
- **EXPOSE** : 해당 컨테이너가 런타임에 지정된 네트워크 포트에서 수신 대기중이라는 것을 알려준다. 프로토콜을 지정하지 않으면 기본값은 tcp이다
- **CMD** : 컨테이너가 시작 될때 실행할 명령어. Dockerfile에서 한번만 사용할 수 있습니다. 쉘 파일 만들어 시행할 쉘 명령어를 한번에 실행 시킬수도 있습니다.

---

[11] 이미지의 버전 정보, 작성자, 코멘트와 같이 이미지 상세 정보를 작성해두기 위한 명령

[12] 같은 결과를 가져오더라도 RUN을 여러줄로 작성하면 image layer가 여러개가 생성되고 한줄로 작성하면 레이어 하나만 생성된다.

[13] ft_server 과제를 위해 수정해줬던 설정 파일들을 이 명령어로 미리 src 폴더 안에 넣어 둬야한다.

Hyeykim 2021년 3월 16일 화요일

참고 사이트들
https://tpcable.co.kr/45 (도커, 컨테이너에 대해)
https://corona-world.tistory.com/15 (docker란? vm과 차이)
https://cosyp.tistory.com/230 (docker 소개 및 설치 방법)
https://dev-youngjun.tistory.com/2 (도커란 무엇일까요?)
https://ko.wikipedia.org/wiki/%EA%B0%80%EC%83%81_%EB%A8%B8%EC%8B%A0 (위키백과 : 가상 머신)
https://blog.naver.com/isc0304/221840483579 (도커와 컨테이너 소개)
https://subicura.com/2017/01/19/docker-guide-for-beginners-1.html(초보를 위한 도커 안내서 - 도커란 무엇인가?)
https://www.docker.com/resources/what-container
https://www.docker.com/101-tutorial (도커 공식 홈페이지)
https://www.yalco.kr/36_docker/ (쉽게 배우는 도커)
https://velog.io/@ckstn0777/%EB%8F%84%EC%BB%A4%ED%8C%8C%EC%9D%BCDockerfile (도커 파일)
https://velog.io/@hidaehyunlee/ftserver-%EC%84%A0%ED%96%89%EC%A7%80%EC%8B%9D-Docker-Debian-Buster-Nginx- (ft_server선행 지식)

https://yeosong1.github.io/ftserver-%ED%92%80%EC%9D%B4%EA%B8%B0%EB%A1%9D

(42wiki - ft_server 풀이과정) 사이트 문서에 있는 명령어를은 여기서 참고했습니다.