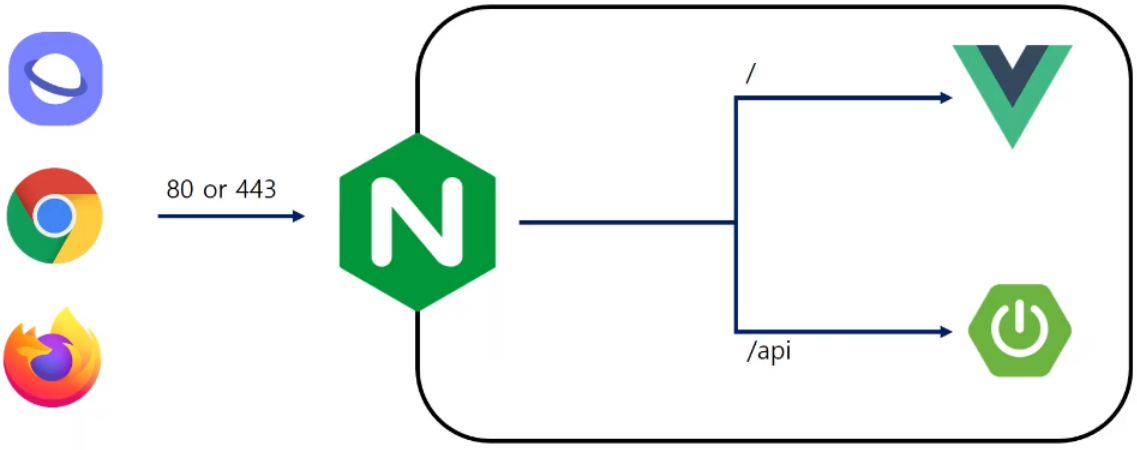
[Project News] 공통 프로젝트 배포

> 진행자: 이상현 컨설턴트님
>
> 날짜: 2021-02-08

목차

1. 배포 구조

- **NGINX**
  - High performance load balancer, web server, API gateway & reverse proxy
  - 비동기 방식이기 때문에 매우 높은 성능
  - 정적인 파일 (주로 프론트엔드 파일들)을 서비스할 때 뛰어난 성능 (vs 톰캣)
  - load balancer나 API gateway 용도로도 사용 가능



- 배포 구조
  - `/`로 들어오는 요청은 프론트엔드의 라우터로, `/api`로 들어오는 요청은 백엔드로 보낸다.
  - Webserver로서의 역할과 API gateway로서의 역할을 모두 수행하는 것이다.
  - 프론트엔드와 백엔드의 분기를 위한 NGINX 설정 파일 모습

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

    root /var/www/html/dist;        # Front 빌드 파일 위치
    index index.html index.htm ;    # index 파일명
    server_name _;                  # 서버 도메인

    location / {
        try_files $uri $uri/ /index.html;
    }

    location /api {
        proxy_pass http://localhost:8399/api/;
        proxy_redirect off;
        charset utf-8;

        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_set_header X-NginX-Proxy true;
    }
}
```

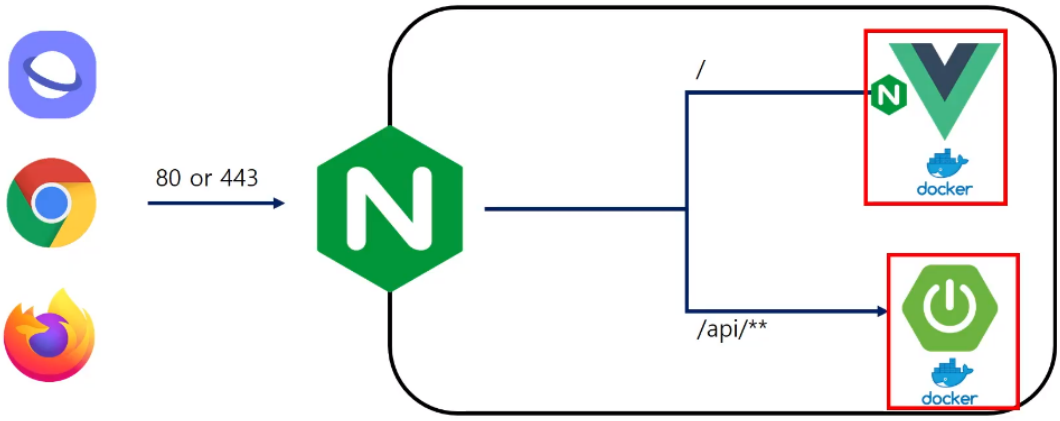
Frontend 설정

Backend Proxy 설정

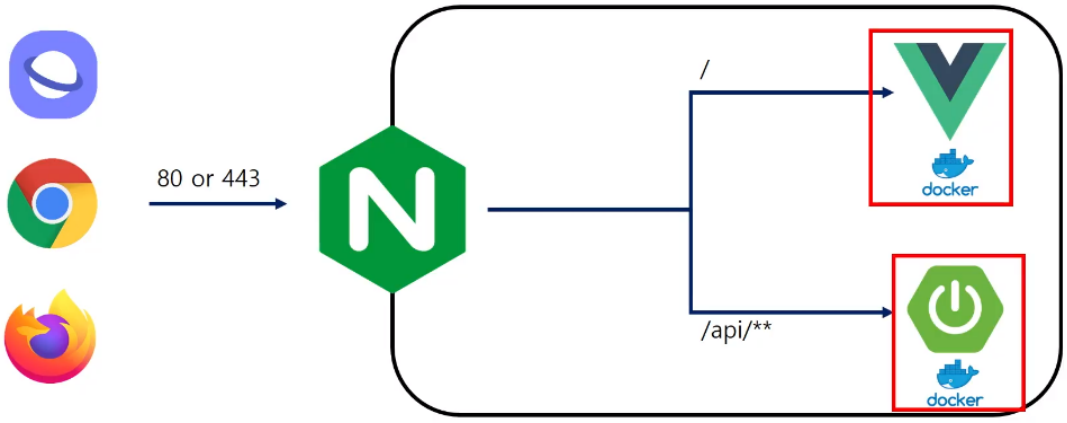
2. Docker를 이용한 배포

- **Docker를 이용한 배포**에는 2가지 방법이 있다.

  1. 프론트엔드 이미지 내부에 NGINX를 추가로 설정해놓는 방법



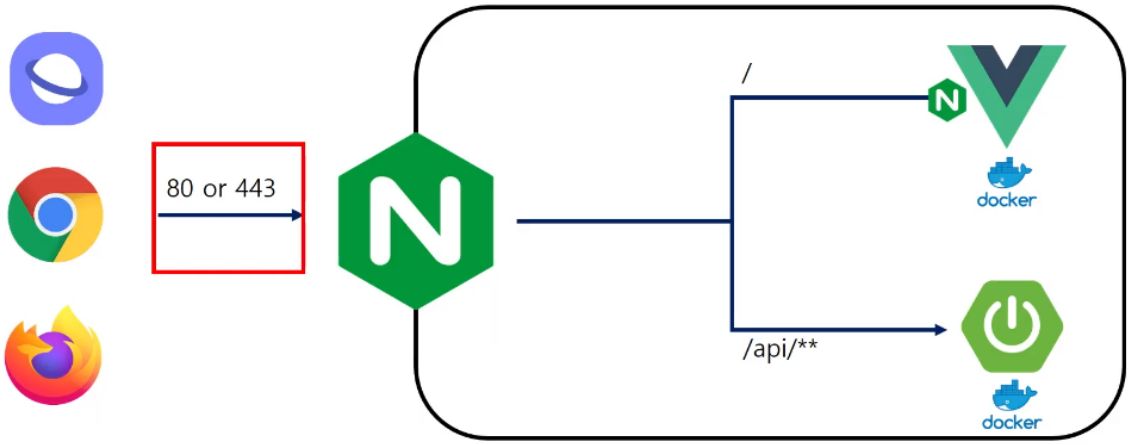
  2. 프론트엔드를 볼륨으로 mount하는 방법



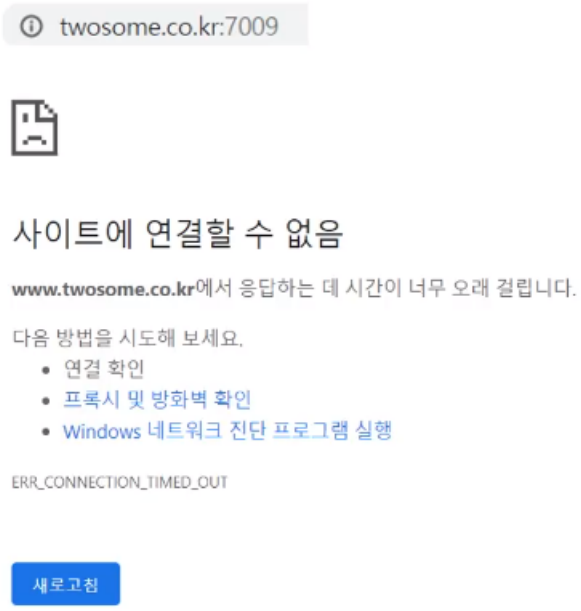
- 우리는 왜 Docker를 쓰는가?
    - 이전에는 직접 물리적인 서버를 더 구비하거나, AWS의 AutoScaling과 같은 방법으로 서버를 증설하려고 했으나, 이도 충분하지 않았다.
    - Docker를 사용하면 **빠르게 필요한 서버를 증설**할 수 있다.
    - 기존에는 VM을 증설하는 방식을 사용했으나, VM이 부팅되는 1분이면 서비스 전체가 중지되기에 충분한 시간이다.
    - Docker는 운영체제를 부팅해야 하는 기존의 방식보다 빠르다.
    - 이미지를 만들어두면 찍어내기만 하면 되는 배포의 편의성을 경험할 수 있다. (with kubernetes)

- 어디까지 도커화 해야할까?
    - 프론트엔드/백엔드는 **필수적**
        - 사용자가 많아지면 서버를 증설해 나가야하기 때문에 도커화를 하는 것이 좋다.
        - 배포의 효율성/편의성을 생각해보면서 도커화를 하는 것이 좋다.
    - DB/Jenkins/NGINX는 **선택적**
        - DB를 이미지화해서 새로 배포할 일이 많이 없기 때문에 선택적으로 도커화하면 된다.
        - 빌드 서버를 병렬적으로 추가 증설하는 경우는 많지 않기 때문에 Jenkins 역시 선택적으로 도커화하면 된다.

3. HTTP, HTTPS



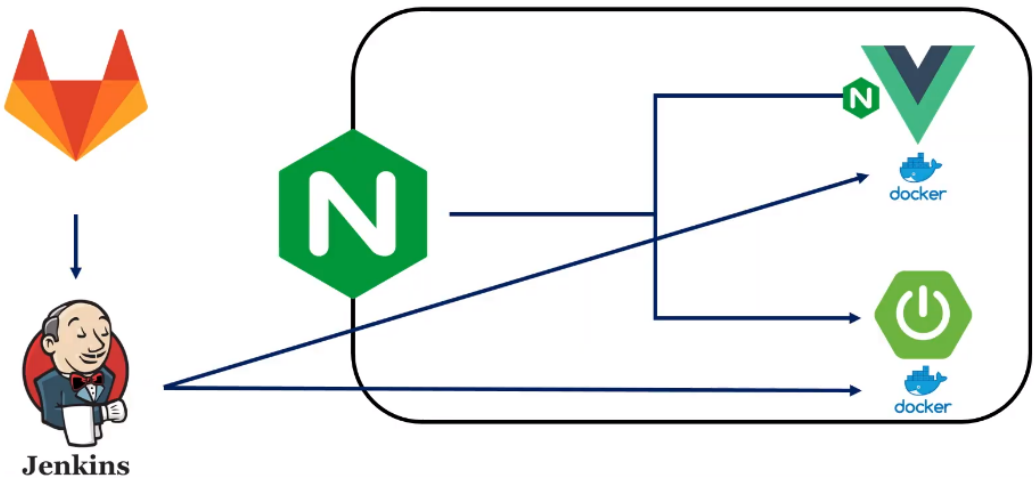
- `80`과 `443` 포트는 각각 HTTP, HTTPS에 사용되는 포트이다.
- 임의의 포트를 사용하면 안되는 이유



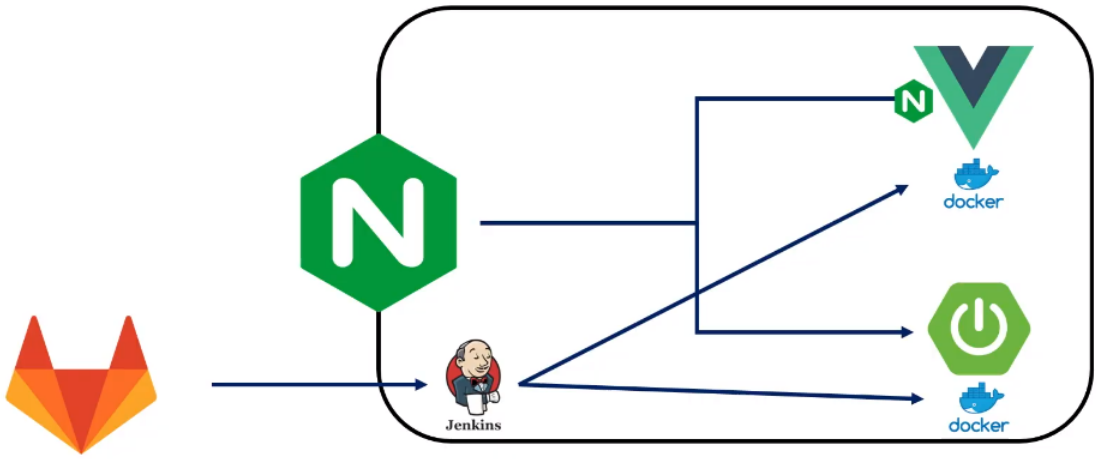
- 멀티캠퍼스에서는 투O플레이스를 들어갈 수 없다!
- **ISP(SKT, KT, LGU 등등)에 따라서 닫혀 있는 포트가 존재**한다.
- 따라서 임의의 포트를 사용하면 어느 곳에서는 되고, 어느 곳에서는 안되는 서비스가 될 수도 있다.
- 이런 경우에 고객은 포트가 막혔을 거라는 생각을 하지 못하고 그냥 이탈한다.

4. Jenkins를 이용한 배포

- Jenkins를 이용한 배포에는 2 가지 방법이 있다.
  1. SSAFY GIT에서 제공하는 Jenkins를 사용하는 방법

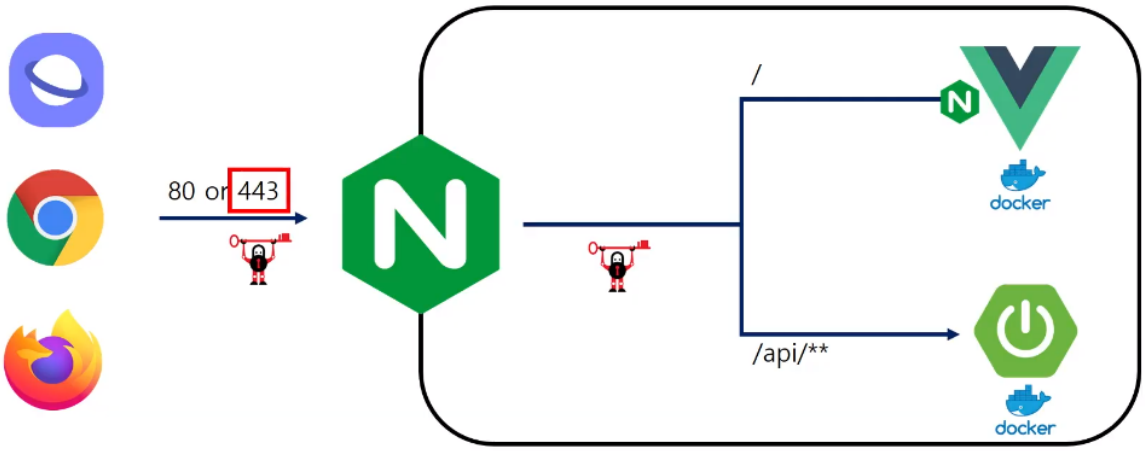


  2. EC2 인스턴스 내부에 Jenkins를 설치해서 사용하는 방법

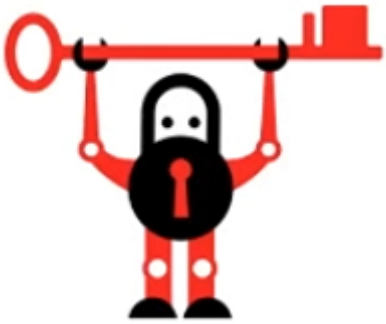


- 개발자가 gitlab의 특정 브랜치(develop or master)에 머지를 하면 **이벤트가 트리거되어 Jenkins에서 빌드를 시작**한다.
- 빌드가 완료되면 도커 이미지가 제작되어 배포된다.
- 동일한 도커 이미지로 제작, 배포되기 때문에 **동일성이 보장**된다.

5. HTTPS 사용하기



- HTTPS (Hypertext Transfer Protocol Secure)는 **TLS(Transport Layer Security)를 사용하는 HTTP 프로토콜의 보안 버전**이다.
- TLS 이전에는 **SSL(Secure Socket Layers)**이 있었다. TLS가 SSL의 취약성을 해결한 보안적으로 더 강력한 프로토콜이다.
- 회원 가입 시에 비밀번호 등의 개인 정보가 전송되고, 수시로 유출되어서는 안되는 정보들이 오가기 때문에 암호화가 필요하다.
- 매번 데이터를 암호화해서 전송하기 어렵기 때문에 TLS를 사용한다.
- 이론적으로는 TLS를 활용한 통신은 안전하다고 볼 수 있다.



- **Certbot**
  - HTTPS 확산을 위해서 시작된 비영리 프로젝트다. (Let's encrypt)
  - 상용 프로그램을 제작할 때는 보통 신뢰할 수 있는 ROOT 인증서 발급자로부터 SSL 인증서를 구매해서 사용한다.
  - SSAFY 프로젝트의 경우에는 Certbot을 이용해서 무료 인증서를 발급 받아서 사용하면 좋다.
  - 이를 위해서는 NGINX나 백엔드 서버 모두에 TLS 설정이 필요하다.

6. 사용자 계정 만들기

- 각 프로그램들을 실행할 때는 프로그램에 맞는 권한을 가진 사용자 계정을 만들어서 실행한다.
- ubuntu 계정이나 심지어 root 계정으로 실행하는 경우에는 해커의 공격 명령이 그 계정의 권한으로 실행되기 때문에 매우 위험하다.
- 사용자 계정으로 실행하는 경우 해커의 공격을 받더라도 피해를 최소화할 수 있다.

7. Q&A

1. 하나의 서버에서 쿠버네티스로 젠킨스랑 여러가지를 띄웠을때 성능저하 문제는 없을까요?
   - 제공된 EC2 인스턴트가 생각보다 크기 때문에 다 띄워도 크게 문제 없이 잘 돌아갈 것입니다.
2. 열려있는 포트 하나 당 사용자 몇 명 정도 이용이 가능한가요?
   - 리눅스에서 만들 수 있는 연결 갯수의 제한과 네트워크 망의 폭(bandwidth)에 따라 다르지만, 이론적으로는 굉장히 큰 숫자의 사용자가 이용 가능합니다.