How to Deploy

1. 배포란

> 소프트웨어를 사용자가 접근할 수 있는 환경에 배치시키는 일

- **컴파일**: 작성된 코드를 컴퓨터가 이해할 수 있는 언어로 번역하는 일
- **빌드**: 컴파일된 코드를 실제 실행할 수 있는 상태로 만드는 일
- **배포**: 빌드가 완성된 실행 가능한 파일을 사용자가 접근할 수 있는 환경에 배치시키는 일

배포는 `[ 컴파일 → 빌드 → 배포 ]` 의 과정을 거쳐야 한다.

보통 `[ 컴파일 → 빌드 ]` 의 과정을 빌드 한다 라고 표현한다.

Spring-boot를 예로 들자면 컴파일을 포함해 war, jar 등의 실행 가능한 파일을 뽑아 내기까지의 과정을 빌드한다고 표현한다.

이클립스에서 Java로 코딩을 한다고 생각해보자

코드를 짜고 나서 Run 버튼을 눌러 코드를 실행시킨다: `[ 컴파일 → 실행 ]`

정상적으로 실행되면 이것을 jar 파일로 뽑아서 웹서버에 올린다: `[ 빌드 → 배포 ]`

2. Local에서 Test

> Local에서 테스트를 진행하지 않고 배포하는것은 의미가 없다.
> 반드시 테스트 후 배포를 진행할 것

2.1 Git clone

```
git clone https://lab.ssafy.com/s04-webmobile2-sub1/skeleton-project.git
```

2.2 Front test

```
> cd frontend
> npm install
> npm run serve
```

- 아래 주소로 들어가서 확인한다.

http://localhost:8080/

- 참고

  Vue.js 단위 테스트

  Jest와 Vue Test Utils(VTU)로 Vue 컴포넌트 단위테스트

  Vue test 알아보기

2.3 Backend test

```
cd backend

# windows
.\mvnw spring-boot:run
```

- 아래 url로 들어가서 확인한다.

  http://localhost:8080/swagger-ui.html

- 참고

  STS 메이븐 프로젝트 jar 패키지 빌드

---

3. Server에서 Test

3.1 서버 접속 & git clone

```
# 서버 접속
ssh -i cert.pem ubuntu@i4c10x.p.ssafy.io

# git clone
git clone https://lab.ssafy.com/s04-webmobile2-sub1/skeleton-project.git
```

3.2 개발 환경 준비

- **DB 서버**: EC2에 DB를 설치하고 DB를 공유하여 개발하기
- **openjdk**, **maven**, **npm** 설치

```
# install openjdk-8
sudo apt-get install openjdk-8-jdk
```

```
java -version

# install maven
sudo apt install maven

# install npm
sudo apt install npm
```

3.3 Front test

```
cd frontend
npm install
npm run serve

# test:
```

- 아래 주소로 들어가서 확인한다

  http://i4c10x.p.ssafy.io:8080 (자신의 ec2 주소에 맞게 수정)

- Invalid Host header 발생시

  skeleton-project/frontend/vue.config.js에 아래 내용 추가

```
module.exports = {
  configureWebpack: {
    // other webpack options to merge in ...
  },
  // devServer options dont belong into `configureWebpack`
  devServer: {
    host: "0.0.0.0",
    hot: true,
    disableHostCheck: true
  },
};
```

3.4 Backend test

```
mvn spring-boot:run
```

- 아래 주소로 들어가서 확인한다

  http://i4c10x.p.ssafy.io:8080/swagger-ui.html (자신의 ec2 주소에 맞게 수정)

4. Nginx deploy

- 웹 서버: Nginx
- Nginx로 Reverse-Proxy 서버 만들기

4.1 Nginx install

```
sudo apt install nginx
nginx -v
```

4.2 Front

4.2.1 Front build

```
cd front
npm run build
```

4.2.2 Nginx setting 수정

```
sudo vim /etc/nginx/sites-enabled/default

#웹서버 루트 기존 설정은 주석처리, front를 build한 위치로 변경
-----------------------------------------------------------------------------------------
#root /var/www/html;
root /home/ubuntu/skeleton-project/frontend/dist;
...

-----------------------------------------------------------------------------------------

# 설정 변경 후 syntax 검사 필수
sudo nginx -t

# 설정 변경 후 Nginx 재시작
sudo service nginx restart
```

- 아래 주소로 들어가서 확인한다.

  http://i4c10x.p.ssafy.io (자신의 ec2 주소에 맞게 수정)

4.3 Backend

4.3.1 Backend build

```
cd backend
mvn package
```

4.3.2 Nginx setting 수정

```
sudo vim /etc/nginx/sites-enabled/default

-----------------------------------------------------------------------------------------
server {
    listen 80;
    ...
    ...

    location / {
        # First attempt to serve request as file, then
        # as directory, then fall back to displaying a 404.
        try_files $uri $uri/ =404;
    }

    ### backend reverse proxy 설정 추가 ###
    location /api {
        proxy_pass http://localhost:8080;
        proxy_http_version 1.1;
        proxy_set_header Connection "";

        proxy_set_header Host $host;
        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-Forwarded-Host $host;
        proxy_set_header X-Forwarded-Port $server_port;
    }
    ### 여기까지 ###
    ...
}
-----------------------------------------------------------------------------------------

# 설정 변경 후 syntax 검사 필수
sudo nginx -t

# 설정 변경 후 Nginx 재시작
sudo service nginx restart
```

4.3.3 Nginx용 Spring Boot 실행

```
cd backend

# proxy로 backend test: http://i4c10x.p.ssafy.io/api/swagger-ui.html
# 보안상 외부에서 8080으로 접근 불가하게 실행하려면 --server.address=127.0.0.1 옵션 추가
java -jar target/*.jar --server.servlet.context-path=/api
```

- 아래 주소로 들어가서 확인한다.

  http://i4c10x.p.ssafy.io/api/swagger-ui.html (자신의 ec2 주소에 맞게 수정)

5. 백엔드 HA구성

> High Availability: 고 가용성

- HA란?

5.1 Reverse proxy 설정 및 upstream 추가, proxy_pass 수정

```
sudo vim /etc/nginx/sites-enabled/default

-----------------------------------------------------------------------------------------
# backend upstream 설정
upstream backend {
  server localhost:8080;
  server localhost:8081;
}

server {
  listen 80;
  ...

  location / {
    # First attempt to serve request as file, then
    # as directory, then fall back to displaying a 404.
    try_files $uri $uri/ =404;
  }

  location /api {
    # 이 부분 upstream으로 변경
    proxy_pass http://backend;
    proxy_http_version 1.1;
    proxy_set_header Connection "";

    proxy_set_header Host $host;
    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-Forwarded-Host $host;
    proxy_set_header X-Forwarded-Port $server_port;
  }
}
-----------------------------------------------------------------------------------------

# 설정 변경 후 syntax 검사 필수
sudo nginx -t

# 설정 변경 후 Nginx 재시작
sudo service nginx restart
```

- Reverse Proxy

  Reverse Proxy란?

Forward Proxy와 Reverse Proxy의 차이

5.2 Server 실행

- 터미널에서 backend 프로그램 실행 후 빠져나오면 프로세스가 종료됨 ⇒ nohup, &

```
# 첫번째 서버 실행
nohup java -jar target/*.jar \
--server.servlet.context-path=/api \
--server.address=127.0.0.1 \
--server.port=8080 &

# 두번째 서버 실행
nohup java -jar target/*.jar \
--server.servlet.context-path=/api \
--server.address=127.0.0.1 \
--server.port=8081 &

# process 확인
ps -ef | grep java
```

- Linux ps (프로세스 확인하기)

5.3 Backend & Frontend 최신버전 반영하기

5.3.1 기존 프로세스 종료

```
kill -9 `pgrep java`
```

- ps와 grep으로 pid를 조회하여 프로세스 종료

5.3.1 Git clone

```
# 기존 git repo 삭제
rm -rf ~/skeleton-project

# git clone
git clone https://lab.ssafy.com/s04-webmobile2-sub1/skeleton-project.git
```

- frontend와 backend의 build 폴더를 삭제하고, git pull 수행 후 다시 build 해도 상관없다.

5.3.2 Build

```
# front
cd frontend
npm install
npm run build

#backend
cd backend
mvn package
```

5.3.3 Server 실행

```
# 첫번째 서버 실행
nohup java -jar target/*.jar \
--server.servlet.context-path=/api \
--server.address=127.0.0.1 \
--server.port=8080 &

# 두번째 서버 실행
nohup java -jar target/*.jar \
--server.servlet.context-path=/api \
--server.address=127.0.0.1 \
--server.port=8081 &

# process 확인
ps -ef | grep java
```

- 아래 주소로 들어가서 확인한다.

  frontend: http://i4c10x.p.ssafy.io (자신의 ec2 주소에 맞게 수정)

  backend: http://i4c10x.p.ssafy.io/api/swagger-ui.html (자신의 ec2 주소에 맞게 수정)

6. Jenkins CI/CD 구성

- Continuous Integration / Continuous Deployment
- 생각해볼만한 이슈

  달리는 자동차의 바퀴를 어떻게 바꿀까? ⇒ 무중단 배포

  다음시간에 만나요