포팅 메뉴얼

빌드 및 배포

1. 버전 정보

Frontend

React : 18.0.0

React Native : 0.69.5

Backend

Spring boot : 2.7.4

QueryDSL : 1.0.10

Database

- mariaDB : 10.6.8
  - hostname : j7a107.p.ssafy.io
  - port : 3741
  - username : chaekin
  - password : xBpIRi%O1q

Infra

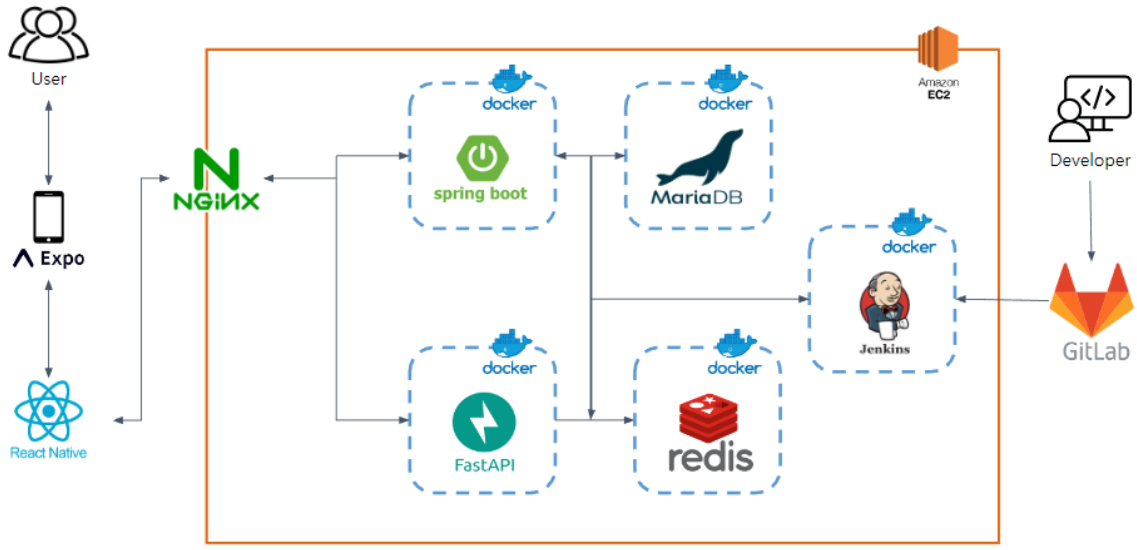
- docker : 20.10.18
- Jenkins : 2.60.3
  - url : j7a107.p.ssafy.io:9090
  - id : chackin0915
  - password: A107chackin!@
- redis : 7.0.5

- password : sd2c0s02na9cb02md1nsd0bnd

IDE

Intellij : 2022.01.03 (community)

Architecture



2. EC2 설정

Docker 설치

👉 사전 패키지 설치

```
sudo apt update
sudo apt-get install -y ca-certificates \
    curl \
    software-properties-common \
    apt-transport-https \
    gnupg \
    lsb-release
```

👉 gpg 키 다운로드

```
sudo mkdir -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg

echo \
    "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \
    $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

👉 Docker 설치

```
sudo apt update
sudo apt install docker-ce docker-ce-cli containerd.io docker-compose
```

Docker-Compose

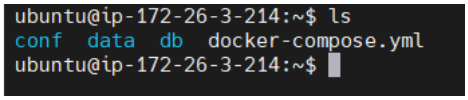
도커 컴포즈를 이용하여 jenkins, mariadb를 생성

- docker-compose.yml

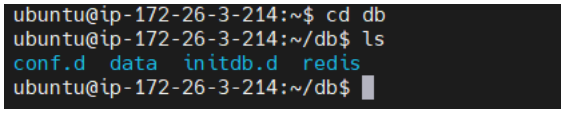
```
version: '3'

services:
    jenkins:
        image: jenkins/jenkins:lts
        container_name: jenkins
        volumes:
            - /var/run/docker.sock:/var/run/docker.sock
            - /jenkins:/var/jenkins_home
        ports:
```

```
            - "9090:8080"
        privileged: true
        user: root

    mariadb:
        container_name: mariadb
        image: mariadb:10.6.8
        ports:
            - 3741:3306
        volumes:
            - ./db/conf.d:/etc/mysql/conf.d
            - ./db/data:/var/lib/mysql
            - ./db/initdb.d:/docker-entrypoint-initdb.d
        env_file: .env
        environment:
            TZ: Asia/Seoul
        networks:
            - backend
        restart: always

networks:
  backend:
```

동일한 경로에 conf, data, db 디렉토리를 만들고 각각의 설정 파일을 넣어준다.



db 디렉토리 내부에도 아래와 같이 디렉토리들을 만들어준다.



그리고 아래의 설정 파일을 넣어준다.

- data/conf.d/my.cnf

```
[client]
default-character-set = utf8mb4

[mysql]
default-character-set = utf8mb4

[mysqld]
character-set-client-handshake = FALSE
character-set-server           = utf8mb4
collation-server               = utf8mb4_unicode_ci
```

SSL

Certbot을 이용하여 무료 SSL 인증서를 서버에 받을 수 있다.

1. Certbot 설치

   `add-apt-repository ppa:certbot/certbot`

   `apt-get update`

   `apt-get install python-certbot-nginx`

2. 도메인에 대한 인증서 발급

   `sudo certbot certonly --nginx -d  j7a107.p.ssafy.io`

   명령 실행 후 약관에는 동의(A), 이메일 공유는 자유 (N)

   `ls -al /etc/letsencrypt/live/ j7a107.p.ssafy.io`

   명령어로 확인

Nginx 설정

```
$ sudo apt-get update #운영체제에서 사용 가능한 패키지들과 그 버전에 대한 정보(리스트) 업데이트
$ sudo apt install nginx -y #nginx 설치하기
```

```
$ nginx -v #설치한 nginx 버전 확인
$ sudo service nginx status #nginx running 상태 확인
```

만약 빨간 부분으로 inactive라고 표시된다면 실행중이지 않은 것이니 다음 명령어를 입력하여 nginx를 실행한다.

`$ sudo service nginx start`

리버스 프록시 설정

/etc/nginx/sites-available 디렉토리에 들어가 api.conf 파일을 생성한다.

- api.conf

```
server {
  listen 80;
  server_name j7a107.p.ssafy.io;
  return 301 https://j7a107.p.ssafy.io$request_uri;

}
server {
  listen 443 ssl http2;
  server_name j7a107.p.ssafy.io;

  # ssl 인증서 적용하기
  ssl_certificate /etc/letsencrypt/live/j7a107.p.ssafy.io/fullchain.pem;
  ssl_certificate_key /etc/letsencrypt/live/j7a107.p.ssafy.io/privkey.pem;

  location /api/v1 {
         proxy_pass http://localhost:8080;
         proxy_set_header X-Real-IP $remote_addr;
         proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
         proxy_set_header Host $http_host;
    }

    location /api/data {
         proxy_pass http://localhost:8000;
         proxy_set_header X-Real-IP $remote_addr;
         proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
         proxy_set_header Host $http_host;
    }

    proxy_buffer_size          128k;
    proxy_buffers              4 256k;
    proxy_busy_buffers_size    256k;
}
server {
    if ($host = j7a107.p.ssafy.io) {
        return 301 https://$host$request_uri;
    } # managed by Certbot


  listen 80;
  server_name j7a107.p.ssafy.io;
    return 404; # managed by Certbot

}
```

그리고 해당 파일에 대한 심볼릭 링크를 sites-enabled에 만들어준다.

`sudo ln -s /etc/nginx/sites-available/api.conf /etc/nginx/sites-enabled`

3. Spring Boot 설정 파일 추가

Spring boot의 src/main/resources에 아래의 yml파일들을 추가해준다.

- application-server.yml

```
spring:
  datasource:
    url: jdbc:mariadb://j7a107.p.ssafy.io:3741/chaekin?allowPublicKeyRetrieval=true&characterEncoding=UTF-8&serverTimezone=Asia/Se
    username: chaekin
    password: xBpIRi%O1q
    driver-class-name: org.mariadb.jdbc.Driver

  jpa:
    hibernate:
      ddl-auto:
    properties:
      hibernate:
        format_sql: true
        default_batch_fetch_size: 100
```

```
  data:
    web:
      pageable:
        default-page-size: 12
        one-indexed-parameters: true
```

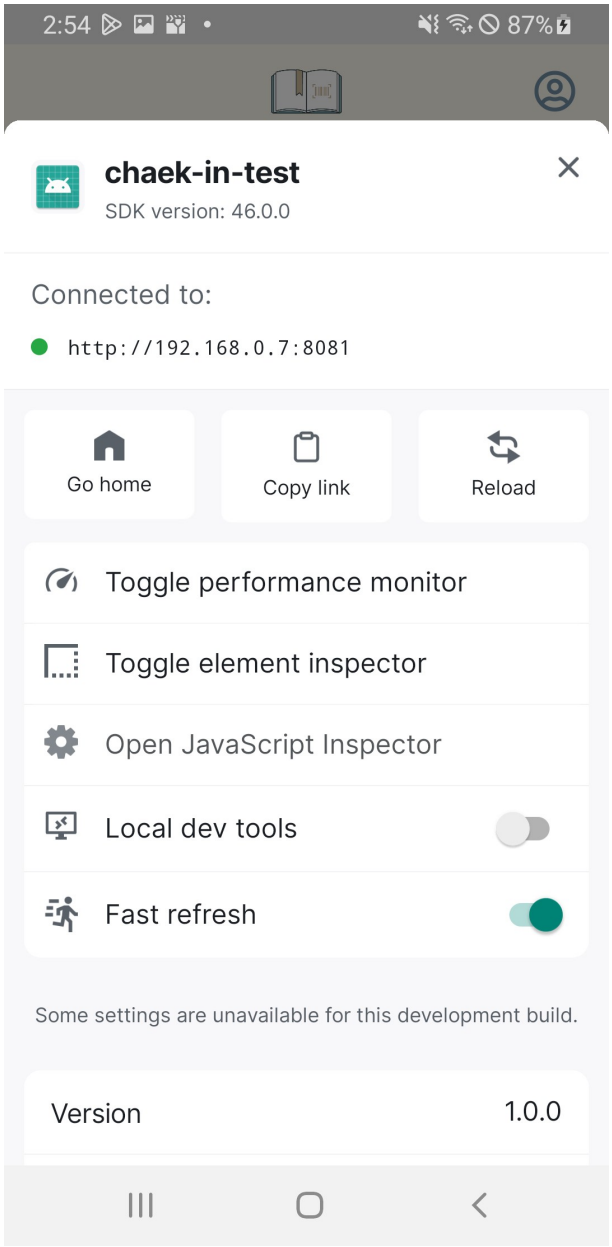
- application-token.yml

```
token:
  secret: 8dfd920afd604a76a313be5eab6c7a5
  access:
    name: Access-Token
    expired-time: 604800
    expired-time-milli: 604800000
  refresh:
    name: Refresh-Token
    expired-time: 1209600
    expired-time-milli: 1209600000

cors:
  allow-origins:
    - http://localhost:3000
    - j7a107.p.ssafy.io:3000
```

4. React Native 설정 파일 추가

Expo 개발 초기 설정

1. `npx create-expo-app app-name`
2. `npx expo install expo-dev-client`
3. `eas build:configure` 로 eas.json파일 생성
4. `eas build --profile development --platform android`
5. Android application id `com.ssafy.app-name` 형태로 입력 (ssafy는 그냥 예시)
6. Generate a new Android Keystore 에서 y 입력
7. 빌드가 끝나면 생성된 파일을 안드로이드 기기에서 실행 후 `Enter URL manually` 선택
4. vscode bash에 `npx expo start --dev-client` 실행 후 나오는 url 입력
    - exp+chaek-in-test://expo-development-client/?url=http%3A%2F%2F192.168.0.7%3A8081 가 출력되면 http://192.168.0.7:8081 을 입력한다
5. 안드로이드 기기에서 나오는 화면을 실시간으로 보면서 실행 가능
    a. 혹시 vscode에서 바꾼 코드가 실시간으로 바뀌지 않는다면 폰을 흔들면 나오는 화면에서 Fast refresh 활성화



Expo 배포 초기 설정

1. 따로 플레이스토어에 올리지는 않았기 때문에 `eas build --profile preview --platform android` 를 입력하여 preview 모드로 배포하여 apk 형태로 만듦
2. 그 외의 설정은 위와 동일

5. FastAPI 설정

```
# app 폴더 상위 폴더에서 가상 환경 생성
$ python -m venv venv

# 가상 환경 실행
$ source venv/Scripts/activate

# requirements.txt에 있는 환경 설정 파일 다운
$ pip install -r requirements.txt

# main.py있는 폴더로 이동 후
$ cd app

# 서버 실행
$ uvicorn main:app --reload
```

1. app 폴더 상위 폴더에 가상 환경 생성
2. 가상 환경 실행
3. requirements.txt에 있는 환경 설정 파일 다운
4. main.py있는 폴더로 이동 후
5. 서버 실행

추가로 .env파일을 main.py가 있는 app 폴더 하위에 넣어주어야 한다.

- .env 파일 내용은 다음과 같다.

```
REDIS_HOST = j7a107.p.ssafy.io
REDIS_PORT = 6379
REDIS_DATABASE = 0
REDIS_SECRET = SGiG.B/VYjjLQ
```

6. redis 설정

아래의 명령어를 EC2에서 실행

```
# redis 최신 버전으로 docker 이미지 받아오기
$ sudo docker pull redis:latest

# 도커에 네트워크 생성
$ sudo docker network create redis-net

# 네트워크 리스트 확인
$ sudo docker network ls

# 볼륨 잡아서 컨테이너 실행
$ docker run --name redis -p 6379:6379 --network redis-net -v /home/ubuntu/redisvolume -d redis:latest redis-server --appendonly yes -
```

외부 서비스

1. Google Firebase Login 설정

https://www.youtube.com/watch?v=d_Vf41Sb0v0&list=LL&index=1&t=643s 영상을 참고하였습니다.

1. `app.json` 파일을 아래와 같이 설정

```
{
  "expo": {
    "android": {
      "googleServicesFile": "./google-services.json" // android 객체 내에 추가
    },
    "ios": {
      "googleServicesFile": "./GoogleService-Info.plist"
    },
    "plugins": ["@react-native-google-signin/google-signin"] // expo 객체 내에 추가
  }
}
```

2. firebase.google.com 에서 프로젝트를 생성한다.
3. 생성한 프로젝트로 이동해서 앱 추가를 누르고 안드로이드를 선택한다.
4. Android 패키지 이름에 앞서 설정한 Android application id를 입력한다.
5. SHA-1 값을 입력한다
    a. bash에서 `eas credentials` 를 입력하고 platform을 Android로 선택
    b. build profile을 빌드 모드에 따라 선택한다
    c. keystore: Manage everything needed to build your project 선택
    d. Set up a new keystore 선택
    e. Create A New Build Credential Configuration 선택

    f. Generate하고 만들어진 SHA-1 키값 복사해서 입력하고 앱 등록
6. 일단 다음 과정으로 넘어가기
7. 생성된 앱 클릭하여 톱니바퀴 모양 아이콘 클릭
8. 하단으로 내려가서 Add fingerprint 클릭
9. 앞서 봤던 Credentials에서 Default로 존재했던 SHA1 Fingerprint값을 복사해서 넣기
10. google-services.json 다운로드 후 앱 최상단에 붙이기
11. firebase console 좌측 메뉴의 빌드 - authentication 클릭
12. 시작 누르고 로그인 제공업체(provider) 구글 클릭하고 추가
13. 새로 빌드하고 진행, 코드는 위의 영상 참조

2. 구글 OCR 설정

하기 블로그 참고

https://ydeer.tistory.com/46