

니방 내방 포팅 매뉴얼

팀 C109 | 박한(팀장), 양승호, 강석민, 지근, 심재서
담당 컨설턴트 | 박세영 컨설턴트
프로젝트 기간 | 2022.10.11 ~ 2022.11.21 (자율, 6주)

삼성 청년 SW 아카데미
광주캠퍼스 7기

목차

1. 프로젝트 기술 스택
2. 배포
3. EC2 세팅
4. 외부 서비스

1. 프로젝트 기술 스택

1) 이슈 관리

- Jira

2) 형상 관리

- Gitlab
- Git

3) 커뮤니케이션

- Mattermost
- Notion
- Webex
- Discord

4) 개발 환경

가. OS

- Windows 10

나. IDE

- IntelliJ 2022.1.3
- Visual Studio Code

다. Database

- MariaDB 10.9.3
- MongoDB 6.0.3

라. Server

- AWS EC2 (Ubuntu 20.04 LTS)
- AWS S3

마. Frontend

- HTML5, CSS3, JS(ES6)
- ReactJS

바. Backend

- Java 11
- Spring Boot 2.7.3
- Spring Data JPA
- Spring Data MongoDB
- Querydsl 5.0
- Spring Security
- Spring Cloud
- Gradle

사. Deployment

- Docker
- Jenkins
- Nginx

아. etc

- Figma
- MySQL Workbench
- MobaXterm v22.1
- Postman

- Swagger

2. 배포

배포 서비스 - IP주소, 포트

Aa service (Container Name)	≡ 포트
rabbitmq	5671
mariadb	3306
mongodb	27017
config-service	8888
discovery-service	8761
apigateway-service	8080
auth-service	0
business-service	0

GitLab + Jenkins + Docker 환경으로 배포합니다.

Jenkins 아이템 구성 → Build Steps

GitLab Webhook을 통해 main 브랜치 push / MR 발생시마다 배포합니다.

```
# 사용하지 않는 이미지 모두 제거
docker image prune -a --force

# 도커 이미지 압축파일을 저장할 폴더 생성
mkdir -p /var/jenkins_home/images_tar

# config-service 이미지 생성 후 저장
cd /var/jenkins_home/workspace/deploy/backend/config-service/
docker build -t config .
docker save config > /var/jenkins_home/images_tar/config.tar

# discovery-service 이미지 생성 후 저장
cd /var/jenkins_home/workspace/deploy/backend/discovery-service/
docker build -t discovery .
docker save discovery > /var/jenkins_home/images_tar/discovery.tar

# apigateway-service 이미지 생성 후 저장
cd /var/jenkins_home/workspace/deploy/backend/apigateway-service/
docker build -t apigateway .
docker save apigateway > /var/jenkins_home/images_tar/apigateway.tar

# auth-service 이미지 생성 후 저장
cd /var/jenkins_home/workspace/deploy/backend/auth-service/
docker build -t auth .
docker save auth > /var/jenkins_home/images_tar/auth.tar

# business-service 이미지 생성 후 저장
cd /var/jenkins_home/workspace/deploy/backend/business-service/
docker build -t business .
docker save business > /var/jenkins_home/images_tar/business.tar
```

Jenkins 아이템 구성 → 빌드 후 조치

```
# 동작중인 or 동작 중지한 컨테이너가 있으면 종료시키기
if (sudo docker ps -a | grep "config-service"); then sudo docker stop config-service; fi
if (sudo docker ps -a | grep "discovery-service"); then sudo docker stop discovery-service; fi
if (sudo docker ps -a | grep "apigateway-service"); then sudo docker stop apigateway-service; sudo docker rm apigateway-service; fi
if (sudo docker ps -a | grep "auth-service"); then sudo docker stop auth-service; sudo docker rm auth-service; fi
if (sudo docker ps -a | grep "business-service"); then sudo docker stop business-service; sudo docker rm business-service; fi

sudo docker run -it -d --rm -p 8888:8888 --network footprints -e "spring.rabbitmq.host=rabbitmq" -e "SPRING_CLOUD_CONFIG_SERVER_GIT_UR
```

```
sudo docker run -it -d --restart always -p 8761:8761 --network footprints -e "spring.cloud.config.uri=http://config-service:8888" --na
sudo docker run -it -d --restart always -p 8080:8080 --network footprints -e "spring.cloud.config.uri=http://config-service:8888" -e "
sudo docker run -it -d --restart always -p 8081:8081 --network footprints -e "spring.cloud.config.uri=http://config-service:8888" -e "
sudo docker run -it -d --restart always -p 8082:8082 --network footprints -e "spring.cloud.config.uri=http://config-service:8888" -e "
```

3. EC2 세팅

업데이트 및 HTTP 패키지 설치

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

PGP 키 및 저장소 추가

```
$ 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
```

도커 엔진 설치

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

도커 브릿지 네트워크 생성

```
# 컨테이너간 통신을 위해 브릿지 네트워크 생성
docker network create --gateway 172.18.0.1 --subnet 172.18.0.0/16 footprints
```

RabbitMQ 설치

```
$ docker run -d --name rabbitmq --network footprints \
-p 15672:15672 -p 5672:5672 -p 15671:15671 -p 5671:5671 -p 4369:4369 \
-e RABBITMQ_DEFAULT_USER=guest \
-e RABBITMQ_DEFAULT_PASS=guest rabbitmq:management
# RabbitMQ 실행
```

- 4369 포트 - EPMD(Erlang Port Mapper Daemon) 사용 포트

| <https://www.rabbitmq.com/configure.html>

- 5671 포트 - TLS(Transport Layer Security) 사용 포트

| <https://www.rabbitmq.com/ssl.html>

redis 설치

```
docker run -d -it --name redis -p 6379:6379 --network footprints redis --requirepass footprints1!
```

mariadb 설치

```
docker run -dit --name mariadb -p 3306:3306 -e MYSQL_ROOT_PASSWORD=1234 --network footprints mariadb:latest
```

mongodb 설치

```
docker run --name mongodb --network footprints -d -p 27017:27017 -v /data/mongodb:/data/db -v /data/mongodb_config:/data/configdb mong
# docker run --name mongodb --network footprints -d -p 27017:27017 -e -v /data/mongodb:/data/db -v /data/mongodb_config:/data/configdb

apt-get update

apt-get install vim

mongosh
```

```
db.createUser(
{
  user: "k7c109",
  pwd: "k7c109",
  roles: [
    { role: "userAdminAnyDatabase", db: "admin" },
    { role: "readWriteAnyDatabase", db: "admin" }
  ]
}
```

```
vim /etc/mongod.conf
```

```
# mongod.conf에 이 부분 수정

security:
  authorization: enabled

# network interfaces
net:
  port: 27017
  bindIp: 0.0.0.0
# bindIp: 127.0.0.1
```

```
# 잘 접속 되는지 확인
mongosh --port 27017 --authenticationDatabase "admin" -u "k7c109" -p
```

Nginx 세팅

```
upstream backend{
  ip_hash;
  #server 172.31.62.140:8080;
  server k7c109.p.ssafy.io:8080;
}

map $http_upgrade $connection_upgrade {
  default upgrade;
  '' close;
}
```

```

server {
    listen      80;
    listen  [::]:80;
    server_name k7c109.p.ssafy.io;

    return 301 https://$host$request_uri;
}

server {
    listen 443 ssl;
    listen [::]:443 ssl;
    server_name k7c109.p.ssafy.io;

    ssl_certificate /etc/letsencrypt/live/k7c109.p.ssafy.io/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/k7c109.p.ssafy.io/privkey.pem;
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2 SSLv3;
    ssl_ciphers ALL;

    #access_log /var/log/nginx/host.access.log main;

    location / {
        root /usr/share/nginx/html;
        index index.html index.htm;
    }

    location /api {
        proxy_pass https://k7c109.p.ssafy.io:8080;
        proxy_redirect off;
        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; # will forward the protocole i.e. http, https
        proxy_set_header X-Forwarded-Port $server_port; # Will forward the port
    }

    #error_page 404 /404.html;

    # redirect server error pages to the static page /50x.html
    #
    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
        root /usr/share/nginx/html;
    }

    # proxy the PHP scripts to Apache listening on 127.0.0.1:80
    #
    #location ~ \.php$ {
    #    proxy_pass http://127.0.0.1;
    #}

    # pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000
    #
    #location ~ \.php$ {
    #    root html;
    #    fastcgi_pass 127.0.0.1:9000;
    #    fastcgi_index index.php;
    #    fastcgi_param SCRIPT_FILENAME /scripts$fastcgi_script_name;
    #    include fastcgi_params;
    #}

    # deny access to .htaccess files, if Apache's document root
    # concurs with nginx's one
    #
    #location ~ /\.ht {
    #    deny all;
    #}
}

```


4. 외부 서비스

1) 카카오

가. 애플리케이션 추가

기본 정보

수정

앱 아이콘	
앱 이름	니방내방
사업자명	SSAFY_C109

사용자가 카카오 로그인을 할 때 표시되는 정보입니다. 정보가 정확하지 않은 경우 서비스 이용이 제한될 수 있습니다.

나. Redirect URI 설정

Redirect URI

삭제

수정

Redirect URI	http://k7c109.p.ssafy.io:8080/auth-service/login/oauth2/code/kakao https://k7c109.p.ssafy.io:8080/auth-service/login/oauth2/code/kakao https://k7c109.p.ssafy.io:8080/auth-service/oauth2/authorization/kakao http://localhost:8080/auth-service/login/oauth2/code/kakao
--------------	--

- 카카오 로그인에서 사용할 OAuth Redirect URI를 설정합니다. (최대 10개)
- REST API로 개발하는 경우 필수로 설정해야 합니다.

다. 동의 항목

개인정보

항목 이름	ID	상태	
닉네임	profile_nickname	● 필수 동의	설정
프로필 사진	profile_image	● 사용 안함	설정
카카오계정(이메일)	account_email	● 사용 안함	설정
성별	gender	● 사용 안함	설정
연령대	age_range	● 사용 안함	설정
생일	birthday	● 사용 안함	설정
출생 연도	birthyear	○ 권한 없음	
카카오계정(전화번호)	phone_number	○ 권한 없음	
CI(연계정보)	account_ci	○ 권한 없음	

2) 구글

가. Redirect URI 설정

승인된 리디렉션 URI ?

웹 서버의 요청에 사용

URI 1 *

http://localhost:8080/auth-service/oauth2/authorization/google

URI 2 *

https://localhost:8080/auth-service/oauth2/authorization/google

URI 3 *

http://k7c109.p.ssafy.io:8080/auth-service/login/oauth2/code/google

URI 4 *

https://k7c109.p.ssafy.io:8080/auth-service/login/oauth2/code/google

+ URI 추가

나. 동의 항목

✕ 선택한 범위 업데이트



특정 범위는 HTTPS URL만 사용하는 프로젝트에서 사용하도록 제한되어 있으므로 이러한 범위를 사용할 수 없습니다. 계속하려면 OAuth 클라이언트를 수정하고 [사용자 인증 정보 페이지](#)에서 HTTPS 외의 URL을 삭제하세요. 다음 클라이언트에 HTTPS 외의 URL이 있습니다.

- 946667839607-1htit9pgisk2rtvon322538s5mhq2nd1.apps.google...



아래에는 사용 설정된 API의 범위만 나와 있습니다. 이 화면에 누락된 범위를 추가하려면 [Google API 라이브러리](#)에서 API를 찾아 사용 설정하거나 아래의 '붙여넣은 범위' 텍스트 상자를 사용하세요. 라이브러리에서 사용 설정한 새 API를 확인하려면 페이지를 새로고침하세요.

필터 속성 이름 또는 값 입력



<input checked="" type="checkbox"/>	API ↑	범위	사용자에게 표시되는 설명
<input checked="" type="checkbox"/>		.../auth/userinfo.email	기본 Google 계정의 이메일 주소 확인
<input checked="" type="checkbox"/>		.../auth/userinfo.profile	개인정보(공개로 설정한 개인정보 포함) 보기
<input checked="" type="checkbox"/>		openid	Google에서 내 개인 정보를 나와 연결
<input type="checkbox"/>	BigQuery API	.../auth/bigquery	View and manage your data in Google BigQuery and see the email address for your Google Account
<input type="checkbox"/>	BigQuery API	.../auth/cloud-platform	Google Cloud 데이터 확인, 수정, 구성, 삭제 및 Google 계정의 이메일 주소 확인
<input type="checkbox"/>	BigQuery API	.../auth/bigquery .readonly	Google BigQuery에서 데이터를 봅니다.
<input type="checkbox"/>	BigQuery API	.../auth/cloud-platform .read-only	Google Cloud 서비스 전체의 데이터 조회 및 Google 계정의 이메일 주소 확인