빌드 및 배포

- OS: Window 10
- Backend Framework: Spring Boot 2.7.13
- Frontend Framework: React
- Database: MySQL, Redis
- WAS: Gradle
- JVM: 11(azul-11.0.19)
- Docker
- WEB: Nginx
- IDE: IntelliJ Ultimate

1. Network 설정

```
docker network create --gateway 172.18.0.1 --subnet 172.18.0.0/16 kkoch-network
```

2. RabbitMq 배포

```
docker run -d --name rabbitmq --network kkoch-network \
-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
```

3. Config Service 배포

```
docker build -t chaos0103/config-service:1.0 .

docker push chaos0103/config-service:1.0

docker pull chaos0103/config-service:1.0

docker run -d -p 8889:8889 --network kkoch-network \
-e "spring.rabbitmq.host=rabbitmq" \
-e "spring.profiles.active=default" \
--name config-service chaos0103/config-service:1.0
```

4. Discovery Service 배포

```
docker build -t chaos0103/discovery-service:1.0 .

docker push chaos0103/discovery-service:1.0

docker pull chaos0103/discovery-service:1.0

docker run -d -p 8761:8761 --network kkoch-network \
-e "spring.cloud.config.uri=http://config-service:8889" \
--name discovery-service chaos0103/discovery-service:1.0
```

5. Apigateway Service 배포

```
docker build -t chaos0103/apigateway-service:1.0 .

docker push chaos0103/apigateway-service:1.0

docker pull chaos0103/apigateway-service:1.0

docker run -d -p 8000:8000 --network kkoch-network \
-e "spring.cloud.config.uri=http://config-service:8889" \
-e "spring.rabbitmq.host=rabbitmq" \
-e "eureka.client.serviceUrl.defaultZone=http://discovery-service:8761/eureka/"
\
--name apigateway-service chaos0103/apigateway-service:1.0
```

6. MySQL 배포

```
docker run -d -p 3307:3307 --network kkoch-network --name mysqldb \
-e "SPRING_DATASOURCE_URL=jdbc:mysql://localhost:3306/kkoch_admin" \
-e "SPRING_DATASOURCE_USERNAME=ssafy" \
-e "SPRING_DATASOURCE_PASSWORD=ssafyc204" \
-e "MYSQL_ROOT_PASSWORD=ssafyc204" \
mysql:8.0.17
```

7. Redis 배포

```
docker run -d --restart=always --name=redis -p 6380:6379 \
--network kkoch-network \
-e TZ=Asia/Seoul \
-v /home/ubuntu/docker_config/redis_config.conf:/etc/redis/redis.conf \
-v redis_data:/data redis:latest redis-server /etc/redis/redis.conf
```

8. User Service 배포

```
docker build -t chaos0103/user-service:1.0 .

docker push chaos0103/user-service:1.0

docker pull chaos0103/user-service:1.0
docker run -d --network kkoch-network --name user-service \
-e "spring.cloud.config.uri=http://config-service:8889" \
-e "spring.rabbitmq.host=rabbitmq" \
-e "spring.datasource.url=jdbc:mysql://mysqldb:3306/kkoch_user?
useSSL=false&characterEncoding=UTF-8&serverTimezone=UTC&allowPublicKeyRetrieval=
true" \
-e "eureka.client.serviceUrl.defaultZone=http://discovery-service:8761/eureka/"
\
chaos0103/user-service:1.0
```

9. Admin Service 배포

```
docker build -t chaos0103/admin-service:1.0 .

docker push chaos0103/admin-service:1.0

docker pull chaos0103/admin-service:1.0

docker run -d --network kkoch-network --name admin-service \
-e "spring.cloud.config.uri=http://config-service:8889" \
-e "spring.rabbitmq.host=rabbitmq" \
-e "spring.datasource.url=jdbc:mysql://mysqldb:3306/kkoch_admin?
useSSL=false&characterEncoding=UTF-8&serverTimezone=UTC&allowPublicKeyRetrieval=
true" \
-e "eureka.client.serviceUrl.defaultZone=http://discovery-service:8761/eureka/"
\
chaos0103/admin-service:1.0
```

10. Auction Service 배포

```
docker build -t chaos0103/auction-service:1.0 .

docker push chaos0103/auction-service:1.0

docker pull chaos0103/auction-service:1.0

docker run -d --network kkoch-network --name auction-service \
-e "eureka.client.serviceUrl.defaultZone=http://discovery-service:8761/eureka/"
\
-e "spring.redis.host=redis" \
-e "spring.redis.port=6379" \
chaos0103/auction-service:1.0
```

11. React App 배포

```
npm i

npm run build

docker build -t chaos0103/react-app:1.0 .

docker push chaos0103/react-app:1.0

docker pull chaos0103/react-app:1.0

docker run -d -p 3000:3000 --network kkoch-network \
--name react-app chaos0103/react-app:1.0
```

12. OpenVidu 설치

```
sudo ufw allow ssh
sudo ufw allow 8081/tcp
sudo ufw allow 8443/tcp
sudo ufw allow 3478/tcp
sudo ufw allow 3478/udp
sudo ufw allow 40000:57000/tcp
sudo ufw allow 40000:57000/udp
sudo ufw allow 57001:65535/tcp
sudo ufw allow 57001:65535/udp

sudo ufw enable

cd /opt

curl https://s3-eu-west-1.amazonaws.com/aws.openvidu.io/
install_openvidu_latest.sh | bash
```

13. OpenVidu 배포

```
docker build -t chaos0103/openvidu-server:1.0

docker push chaos0103/openvidu-server:1.0

docker pull chaos0103/openvidu-server:1.0

vi docker-compose.override.yml
```

```
image : chaos0103/openvidu-server:1.0


cd /opt/openvidu


./openvidu start
```