배포 산출물

버전 정보

- OS: Window 10
- Backend Framework: Spring Boot 2.7.15, Spring Boot 2.7.16
- Frontend Framework: React
- Database: MySQL, Redis, MongoDB
- WAS: Grade, Vite
- JVM: 11(azul-11.0.20)
- Docker
- WEB: Nginx
- IDE: IntelliJ Ultimate, Visual Studio Code

EC2 접속

```
ssh -i J9C205T.pem ubuntu@j9c205.p.ssafy.io
```

Nginx 설치 및 설정

```
sudo apt install nginx

sudo apt-get install letsencrypt

sudo apt-get install certbot python3-certbot-nginx

sudo certbot --nginx

sudo ufw allow 80

sudo ufw allow 443

sudo vi /etc/nginx/sites-available/default
```

Docker 설치 및 설정

```
sudo apt-get update

sudo apt-get install ca-certificates curl gnupg

sudo install -m 0755 -d /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
sudo chmod a+r /etc/apt/keyrings/docker.gpg

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

sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

네트워크 설정

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

RabbitMQ 배포

```
sudo docker run -d --name rabbitmq --network newkids-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
```

MySQL 배포

```
docker run -d -p 3306:3306 --network newkids-network --name mysql-master \
-e "MYSQL_ROOT_PASSWORD=ssafyc205" \
mysql:8.0.33
```

Redis 배포

```
docker run -d -p 6379:6379 --name redis --network newkids-network \
redis:latest
```

Config Service 배포

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

docker push chaos0103/config-service:1.0

docker pull chaos0103/config-service:1.0

sudo docker run -d -p 8888:8888 --network newkids-network \
-e "spring.rabbitmq.host=rabbitmq" \
-e "spring.profiles.active=default" \
--name config-service chaos0103/config-service:1.0
```

Discovery Service 배포

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

docker push chaos0103/discovery-service:1.0

docker pull chaos0103/discovery-service:1.0

docker run -d -p 8761:8761 --network newkids-network \
-e "spring.cloud.uri=http://config-service:8888" \
--name discovery-service chaos0103/discovery-service:1.0
```

Apigateway Service 배포

```
docker build --platform linux/amd64 -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 newkids-network \
-e "spring.cloud.config.uri=http://config-service:8888" \
-e "spring.rabbitmq.host=rabbitmq" \
-e "eureka.client.serviceUrl.defaultZone=http://discovery-service:8761/eureka/" \
--name apigateway-service chaos0103/apigateway-service:1.0
```

User Service 배포

```
docker build --platform linux/amd64 -t chaos0103/user-service:1.0 .

docker push chaos0103/user-service:1.0
```

```
docker pull chaos0103/user-service:1.0

docker run -d --network newkids-network --name user-service \
-e "spring.cloud.config.uri=http://config-service:8888" \
-e "spring.rabbitmq.host=rabbitmq" \
-e "eureka.client.serviceUrl.defaultZone=http://discovery-service:8761/eureka/" \
-e "spring.redis.host=redis" \
-e "spring.redis.port=6379" \
-e "logging.level.org.hibernate=info" \
-e "logging.level.com.ssafy.userservice=info" \
chaos0103/user-service:1.0
```

Vocabulary Service 배포

```
docker build --platform linux/amd64 -t chaos0103/vocabulary-service:1.0 .

docker push chaos0103/vocabulary-service:1.0

docker pull chaos0103/vocabulary-service:1.0

docker run -d --network newkids-network --name vocabulary-service \
-e "spring.cloud.config.uri=http://config-service:8888" \
-e "spring.rabbitmq.host=rabbitmq" \
-e "spring.jpa.hibernate.ddl-auto=none" \
-e "eureka.client.serviceUrl.defaultZone=http://discovery-service:8761/eureka/" \
-e "logging.level.org.hibernate=info" \
-e "logging.level.com.ssafy.vocabularyservice=info" \
chaos0103/vocabulary-service:1.0
```

Article Service 배포

```
docker build --platform linux/amd64 -t chaos0103/article-service:1.0 .

docker push chaos0103/article-service:1.0

docker pull chaos0103/article-service:1.0

docker run -d --network newkids-network --name article-service \
-e "spring.cloud.config.uri=http://config-service:8888" \
-e "spring.jpa.hibernate.ddl-auto=none" \
-e "spring.rabbitmq.host=rabbitmq" \
-e "eureka.client.serviceUrl.defaultZone=http://discovery-service:8761/eureka/" \
-e "logging.level.org.hibernate=info" \
-e "logging.level.com.ssafy.articleservice=info" \
chaos0103/article-service:1.0
```

Keyword Servive 배포

```
docker build --platform linux/amd64 -t chaos0103/keyword-service:1.0 .

docker push chaos0103/keyword-service:1.0

docker pull chaos0103/keyword-service:1.0

docker run -d --network newkids-network --name keyword-service \
-e "spring.cloud.config.uri=http://config-service:8888" \
-e "spring.rabbitmq.host=rabbitmq" \
-e "eureka.client.serviceUrl.defaultZone=http://discovery-service:8761/eureka/" \
-e "logging.level.org.hibernate=info" \
-e "logging.level.com.ssafy.keywordservice=info" \
chaos0103/keyword-service:1.0
```

Quiz Service 배포

```
docker build --platform linux/amd64 -t chaos0103/quiz-service:1.0 .

docker push chaos0103/quiz-service:1.0

docker pull chaos0103/quiz-service:1.0
```

```
docker run -d --network newkids-network --name quiz-service \
-e "spring.cloud.config.uri=http://config-service:8888" \
-e "spring.rabbitmq.host=rabbitmq" \
-e "eureka.client.serviceUrl.defaultZone=http://discovery-service:8761/eureka/" \
-e "spring.redis.host=redis" \
-e "spring.redis.port=6379" \
-e "logging.level.org.hibernate=info" \
-e "logging.level.com.ssafy.quizservice=info" \
chaos0103/quiz-service:1.0
```

Recommendation Service 배포

```
docker build --platform linux/amd64 -t chaos0103/recommendation-service:1.0 .

docker push chaos0103/recommendation-service:1.0

docker pull chaos0103/recommendation-service:1.0

docker run -d --network newkids-network --name recommendation-service \
-e "spring.cloud.config.uri=http://config-service:8888" \
-e "spring.rabbitmq.host=rabbitmq" \
-e "eureka.client.serviceUrl.defaultZone=http://discovery-service:8761/eureka/" \
-e "logging.level.org.hibernate=info" \
-e "logging.level.com.ssafy.recommendationservice=info" \
chaos0103/recommendation-service:1.0
```

Openapi Service 배포

```
docker build --platform linux/amd64 -t chaos0103/openapi-service:1.0 .

docker push chaos0103/openapi-service:1.0

docker pull chaos0103/openapi-service:1.0

docker run -d --network newkids-network --name openapi-service \
-e "spring.cloud.config.uri=http://config-service:8888" \
-e "spring.rabbitmq.host=rabbitmq" \
-e "eureka.client.serviceUrl.defaultZone=http://discovery-service:8761/eureka/" \
-e "logging.level.com.ssafy.openapiservice=info" \
chaos0103/openapi-service:1.0
```

Analysis Service 배포

```
docker build --platform linux/amd64 -t chaos0103/analysis-service:1.0 .

docker push chaos0103/analysis-service:1.0

docker pull chaos0103/analysis-service:1.0

docker run -d --network newkids-network --name analysis-service \
-e "spring.cloud.config.uri=http://config-service:8888" \
-e "spring.rabbitmq.host=rabbitmq" \
-e "eureka.client.serviceUrl.defaultZone=http://discovery-service:8761/eureka/" \
-e "logging.level.com.ssafy.analysisservice=info" \
chaos0103/analysis-service:1.0
```

Keyword Batch 배포

```
docker build --platform linux/amd64 -t chaos0103/keyword-batch:1.0 .

docker push chaos0103/keyword-batch:1.0

docker pull chaos0103/keyword-batch:1.0

docker run -d -p 9000:9000 \
--network newkids-network --name keyword-batch \
-e "spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver" \
-e "spring.jpa.database=mysql" \
-e "spring.jpa.database-platform=org.hibernate.dialect.MySQL8Dialect" \
-e "spring.datasource.url=jdbc:mysql://mysqldb:3306/recommendation?useSSL=false&characterEncoding=UTF-8&serverTimezone=UTC&allowPublicK
```

```
-e "spring.datasource.username=ssafy" \
-e "spring.datasource.password=ssafyc205" \
-e "spring.data.mongodb.uri=mongodb+srv://S09P22C205:q0fxy2eLW3@ssafy.ngivl.mongodb.net/S09P22C205?authSource=admin" \
chaos0103/keyword-batch:1.0
```

Prometheus 배포

```
wget https://github.com/prometheus/prometheus/releases/download/v2.37.0/prometheus-2.37.0.linux-amd64.tar.gz

tar xzvf prometheus-2.37.0.linux-amd64.tar.gz

cd prometheus-2.37.0.linux-amd64

sudo vi prometheus.yml

docker run -d -p 9090:9090 --network newkids-network \
--name prometheus \
-v /home/ubuntu/prometheus-2.37.0.linux-amd64/prometheus.yml:/etc/prometheus/prometheus.yml \
prom/prometheus
```

Grafana 배포

```
docker run -d -p 3000:3000 --network newkids-network --name grafana grafana/grafana
```

Zipkin 배포

```
docker run -d -p 9411:9411 \
--network newkids-network \
--name zipkin \
openzipkin/zipkin
```

Kafka 배포

```
git clone https://github.com/wurstmeister/kafka-docker.git

sudo vim kafka-docker/docker-compose-single-broker.yml

docker-compose -f docker-compose-single-broker.yml up -d
```