

Hugging

삼성 SW 청년 아카데미 대전 캠퍼스 7 기
특화 프로젝트 (6 주 2022/08/29 ~ 2022/10/07)

포팅 매뉴얼

담당 컨설턴트 : 최인국 컨설턴트
유일권(팀장), 김호진, 송예림, 이주희, 조성규, 최주희

<<목차>>

1. 프로젝트 소개	1
2. 빌드 상세 내용	2
3. DB 설치 및 설정	4
4. 프로퍼티 정의	6

1. 프로젝트 소개

개요 : 언택트 마음챙김 콘텐츠 서비스

주요 타겟층

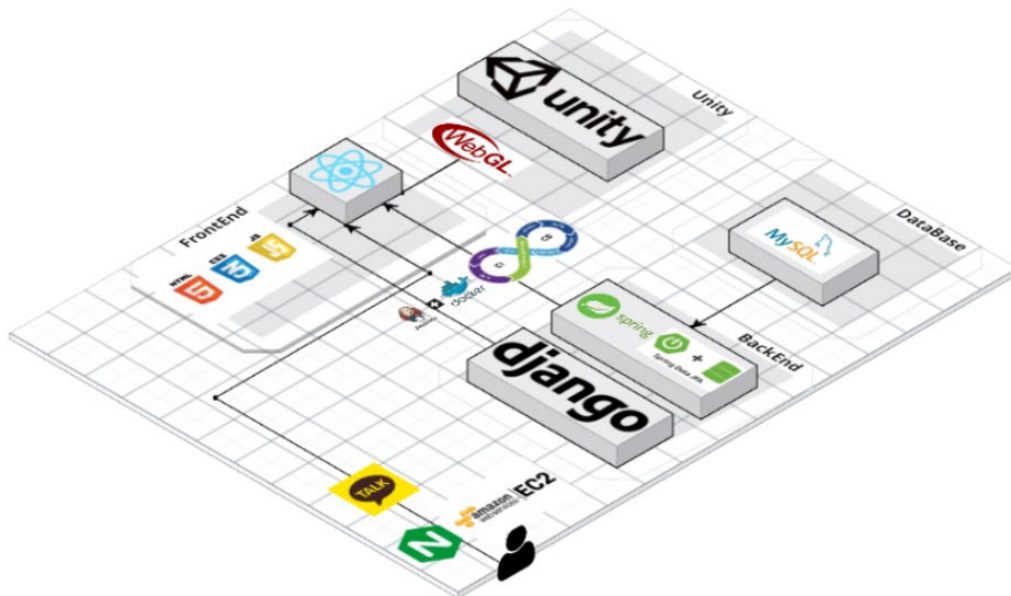
코로나 등으로 인한 우울상태가 지속되거나 스트레스/수면
부족/우울/불안/번아웃 등으로 힐링이 필요한 사람들

대면 상담에 대한 부담감이 있는 사람

2. 빌드 상세 내용

▶ 개요

저희 “Hugging” 서비스의 배포환경 및 CI/CD 배포 자동화 흐름도 입니다.



팀원들이 GitLab 에 코드를 작성하여 push 하게 되면 EC2 인스턴스의 Jenkins 가 GitLab 의 WebHook 을 통해 FrontEnd, BackEnd, Backend-django, unity 프로젝트를 빌드하게 됩니다.

빌드가 완료되고, 빌드 된 4 개의 컨테이너와 Nginx, agora-token-server 컨테이너를 docker-compose 를 통해 실행 시킵니다.

Nginx 컨테이너는 서비스화를 위해 리버스 프록시 서버로 설정하였습니다. Frontend 는 443 포트로, Backend 서버는 8080 포트로 Backend-django 서버는 8000 포트로 설정해 Load Balancing 이 가능하도록 구축하였습니다.

▶ Docker

- Docker 및 Docker-Compose 설치

아래 명령어를 순서대로 입력하여 Docker 를 설치한다.

```
$ sudo apt-get update
$ sudo apt-get install ca-certificates curl gnupg lsb-release
$ echo ₩
    "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg]
    https://download.docker.com/linux/ubuntu ₩
    $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
$ sudo apt-get update
$ sudo apt-get install docker-ce docker-ce-cli containerd.io
```

명령어를 순서대로 입력하고 설치가 되었다면 아래 명령어로 확인한다.

```
$ sudo docker --version
```

Docker 를 설치했다면, Docker-compose 를 설치한다.

```
$ sudo curl -L
    "https://github.com/docker/compose/releases/download/1.29.2/docker-
    compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose$ sudo
    apt-get install ca-certificates curl gnupg lsb-release

$ sudo chmod +x /usr/local/bin/docker-compose
```

Docker-compose 설치가 다되었다면, 아래 명령어로 확인한다.

```
$ docker-compose -version
```

▸ FrontEnd

프로젝트를 GitLab 에서 Clone 받은 후. 아래 명령어를 통해 Docker Container Image 를 생성한다.

빌드를 위한 Dockerfile 은 프로젝트 내에 작성되어 있다.

```
$ docker build -t hugging_fe ./frontend
```

컨테이너 이미지 이름은 “hugging_fe”로 설정하였다.

▸ BackEnd

프로젝트의 Backend 폴더에서 Gradle 을 통해 빌드를 진행한다.

```
$ ./gradlew clean build
```

이후 아래 명령어를 통해 Docker Container Image 를 생성한다.

빌드를 위한 Dockerfile 은 프로젝트 내에 작성되어 있다.

```
$ docker build -t hugging_be ./backend
```

▸ Backend-Django

프로젝트의 backend-django 폴더에서 빌드를 진행한다

```
$ docker build -t hugging_recom ./backend-django
```

이미지 이름은 hugging_recom 으로 정했다

▸ Unity

프로젝트의 backend-django 폴더에서 빌드를 진행한다

```
$ docker build -t hugging_unity ./webgl
```

▶ Docker-Compose

프로젝트의 최상단에서 Docker-Compose 를 통해서 5 개의 컨테이너를 한번에 실행 시킨다.

```
$ docker-compose up -d    #-d 옵션으로 백그라운드 실행
```

Docker-Compose 의 여러 명령어를 통해 컨테이너를 관리 한다.

```
$ docker-compose down    #docker-compose 종료
```

```
$ docker-compose logs    #docker-compose 로그 확인
```

3. DB 설치 및 설정

Hugging 서비스를 사용하기 위해서 DB 를 설치한다.

```
$ sudo apt-get update
```

```
$ sudo apt-get install -y mysql-server
```

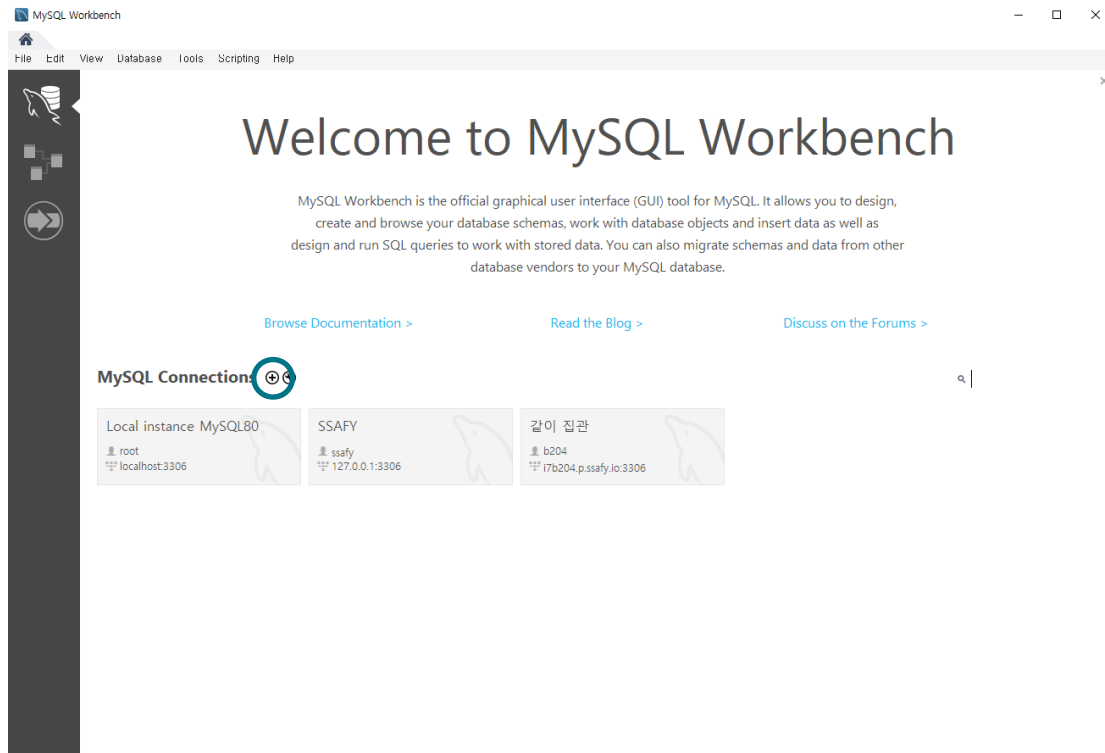
설치가 완료되면 MySql 서비스에 접속하고, 새 유저를 생성한다.

```
$ sudo /usr/bin/mysql -u root -p    #root 유저로 비밀번호를 입력하고 접속
```

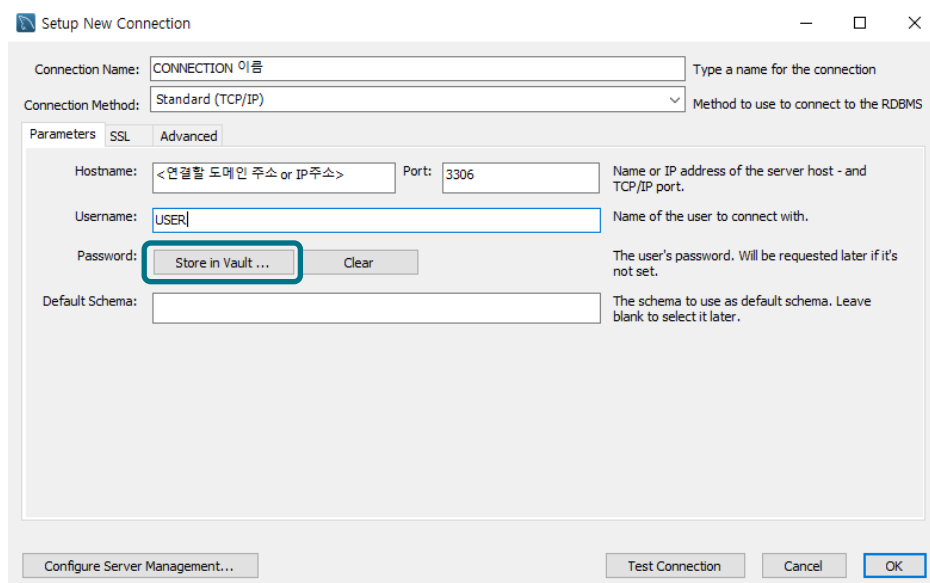
```
$ CREATE USER 'USER@%' IDENTIFIED BY 'PASSWORD';
```

```
$ GRANT ALL PRIVILEGES ON '.' TO 'USER '@%' IDENTIFIED BY 'PASSWORD';
```

이후 MySql WorkBench 에서 서버의 MySql 과 연결한다.



+ 버튼을 눌러 새로운 연결을 생성한다.



빈칸을 채우고, 비밀번호를 'Store In Vault'를 눌러 입력하고 OK 를 누른다.

이후 접속을 할 수 있습니다.

4. 프로퍼티 정의

```
server:
  servlet:
    encoding:
      charset: UTF-8
    context-path: /api

spring:
  datasource:
    url: jdbc:mysql://j7b204.p.ssafy.io:3306/Hugging
    username: b204
    password: Hugging204!
    driver-class-name: com.mysql.cj.jdbc.Driver
  jpa:
    hibernate:
      ddl-auto: update
    properties:
      hibernate:
        # show_sql: true
        format_sql: true
  jwt:
    secretKey: Hugging204!

  mvc:
    pathmatch:
      matching-strategy: ant_path_matcher

logging:
  level:
    org.hibernate.SQL: debug
    org.hibernate.type: trace

oauth2:
  kakao:
    restApiKey: ea11b8e423d8667eb19b25250d76bcdd
    redirectUri: https://j7b204.p.ssafy.io/redirecturi
    # redirectUri: http://localhost:8080/api/oauth
```

Application.yml

```
REACT_APP_CLIENT_ID="ea11b8e423d8667eb19b25250d76bcd"
REACT_APP_KEY="2f55a595da164de2fc5a73ae8b931a91"
REACT_APP_REDIRECT_URL="https://j7b204.p.ssafy.io/redirecturi"
REACT_APP_SERVER_ID=""
```

.env

```
DATABASES = {
    'default' : {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'Hugging',
        'USER': 'b204',
        'PASSWORD': 'Hugging204!',
        'HOST': 'j7b204.p.ssafy.io',
        'PORT': '3306',
    }
}

SECRET_KEY = 'django-insecure-fun45&i4cc7d-cn1tia%u3t7ob3odwxhzybk@svy1)l35los24'
```

conf.py