Home Sketcher

빌드 및 배포

: 2022.08.29 ~ 2022.10.07

대전 특화 3반

권혁림 강승훈 김진산 윤영훈 이슬기 조경민

1. 기술스택

|  |  |  |  |
| --- | --- | --- | --- |
| 구분 | 기술스택 | 상세내용 | 버전 |
| 공통 |  |  |  |
|  | 형상관리 | GitLab |  |
|  | 이슈관리 | Jira |  |
|  | 커뮤니케이션 | Mattemost |  |
|  | 커뮤니케이션 | Notion |  |
| 서버 |  |  |  |
|  | 서버 | AWS EC2 |  |
|  | 플랫폼 | Ubuntu | 20.04 |
|  | 배포 | Docker | 20.10.17 |
|  | 배포 | Docker Compose | 2.6.0 |
|  | 배포 | Jenkins | 2.246.2 |
|  | 배포 | Nginx | 1.23.1 |
| BackEnd |  |  |  |
|  | DB | MySQL | 8..29 |
|  |  | Firebase | - |
|  | Cache Storage | Redis | 7-alpine |
|  | Python |  | 3.9.4 |
| FastAPI |  |  |  |
|  | Python |  | 3.10 |
|  | Numpy |  | 1.23.3 |
|  | Pandas |  | - |
|  | Keras |  | 2.10.0 |
|  | Tensiflow |  | 2.10.0 |
| FrontEnd | HTML5 |  |  |
|  | CSS3 |  |  |
|  | JavaScript(ES6) |  |  |
|  | React |  | 18.2.0 |
|  | Three.js |  | 0.144.0 |
|  | Build | Node | 16.15.0 |
| IDE | Visual Studio Code | 1.70.0 |  |

1. 포트번호

|  |  |
| --- | --- |
| 구분 | 포트번호 |
| Jenkins | 8080 |
| Dkango | 8081 |
| React | 8002 |
| FastAPI | 8003 |
| MySQL | 3306 |
| Redis | 6373 |

1. DokerFile
2. Django

|  |
| --- |
| From python:3.9.4  ENV PYTHONUNBUFFERED 1  RUN apt-get -y update  RUN apt-get -y install vim  RUN mkdir /srv/docker-server  ADD . /srv/docker-server  WORKDIR /srv/docker-server  RUN pip3 install --upgrade pip  RUN pip3 install -r requirements.txt  EXPOSE 8001  # CMD ["python", "manage.py","runsslserver", "0.0.0.0:8001","--certificate" ,"/etc/letsencrypt/live/j7b304.p.ssafy.io/cert.pem","--key","/etc/letsencrypt/live/j7b304.p.ssafy.io/privkey.pem"]  CMD ["python", "manage.py", "runserver", "0.0.0.0:8001"] |

1. React

|  |
| --- |
| FROM node:16.15.0 as build-stage  WORKDIR /app  COPY package\*.json ./  COPY package-lock.json ./  RUN npm install  COPY . .  RUN npm run build  FROM nginx:stable-alpine as production-stage  RUN rm -rf /etc/nginx/conf.d/default.conf  COPY ./nginx/default.conf /etc/nginx/conf.d/default.conf  RUN rm -rf /usr/share/nginx/html/\*  COPY --from=build-stage /app/build /usr/share/nginx/html  EXPOSE 8002  CMD ["nginx", "-g","daemon off;"] |

1. FastAPI

|  |
| --- |
| 1. FROM python:3.10 2. RUN apt-get -y update 3. RUN apt-get -y install vim 4. RUN mkdir /app 5. # ADD . /app 6. WORKDIR /app 7. COPY requirements.txt ./ 9. RUN pip3 install --upgrade pip 10. RUN pip3 install -r requirements.txt 11. COPY . /app 12. EXPOSE 8003 |

1. Docker-Compose File

|  |
| --- |
| version: "3.8"  services:  react:  container\_name: homesketcher-react  build: ./housesketcher/  restart: on-failure  volumes:  - ./housesketcher/nginx/:/etc/nginx/conf.d/  ports:  - 8002:8002  redis:  container\_name: redis  image: redis:7-alpine  ports:  - 6379:6379  command: redis-server --save 20 1 --loglevel warning --requirepass eYVX7EwVmmxKPCDmwMtyKVge8oLd2t81  volumes:  - /redis/data:/data  - /redis/conf/redis.conf:/usr/local/conf/redis.conf  mysql:  container\_name: mysql  image: mysql:latest  restart: unless-stopped  environment:  - MYSQL\_ROOT\_PASSWORD=clsfurtkanth  - MYSQL\_DATABASE=ssafy  - MYSQL\_USER=ssafy  - MYSQL\_PASSWORD=clsfurtkanth  ports:  - 3306:3306  command:  - --character-set-server=utf8mb4  - --collation-server=utf8mb4\_0900\_ai\_ci  - --default-authentication-plugin=mysql\_ |

1. Jenkins file

|  |
| --- |
| pipeline {  agent any  stages {  stage('Prepare') {  steps {  sh 'echo "Clonning Repository"'  git branch: 'deploy',  url: 'https://lab.ssafy.com/s07-bigdata-recom-sub2/S07P22B304.git',  credentialsId: 'homesketcher'  }  post {  success {  sh 'echo "Successfully Cloned Repository"'  }  failure {  sh 'echo "Fail Cloned Repository"'  }  }  }  // stage('[BE]Bulid Gradle') {  // steps {  // sh 'echo "Bulid Gradle Start"'  // dir('BE') {    // }  // }  // post {  // failure {  // sh 'echo "Bulid Gradle Fail"'  // }  // }  // }  stage('Docker stop'){  steps {  dir('BE'){  sh 'echo "Docker Container Stop"'  // 도커 컴포즈 다운  // sh 'curl -L https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m) -o /usr/local/bin/docker-compose'  // 해당 도커 컴포즈 다운한 경로로 권한 설정  // sh 'chmod -R 777 /usr/local/bin'  // sh 'chmod +x /usr/local/bin/docker-compose'  // 기존 백그라운드에 돌아가던 컨테이너 중지  sh 'docker compose stop'  }  }  post {  failure {  sh 'echo "Docker Fail"'  }  }  }  stage('RM Docker'){  steps {    sh 'echo "Remove Docker"'  //정지된 도커 컨테이너 찾아서 컨테이너 ID로 삭제함  sh '''  result=$( docker container ls -a --filter "name=homesketcher\*" -q )  if [ -n "$result" ]  then  docker rm $(docker container ls -a --filter "name=homesketcher\*" -q)  else  echo "No such containers"  fi  '''  // homesketcher로 시작하는 이미지 찾아서 삭제함  sh '''  result=$( docker images -f "reference=homesketcher\*" -q )  if [ -n "$result" ]  then  docker rmi -f $(docker images -f "reference=homesketcher\*" -q)  else  echo "No such container images"  fi  '''  // 안쓰는이미지 -> <none> 태그 이미지 찾아서 삭제함  sh '''  result=$(docker images -f "dangling=true" -q)  if [ -n "$result" ]  then  docker rmi -f $(docker images -f "dangling=true" -q)  else  echo "No such container images"  fi  '''  }  post {  failure {  sh 'echo "Remove Fail"'  }  }  }  stage('Bulid & Run') {  steps {  dir('BE'){  sh 'echo " Image Bulid Start"'  script {  // 업데이트된 코드로 빌드 및 실행  sh 'docker compose up -d'  }  }  }  post {  failure {  sh 'echo "Bulid Docker Fail"'  }  }  }      // stage('[FE] prepare') {  // steps {  // dir('frontend'){  // sh 'echo " Frontend Bulid Start"'  // script {  // sh 'docker-compose stop'  // sh 'docker rm vue'  // sh 'docker rmi frontend\_vue'  // }  // }  // }  // post {  // failure {  // sh 'echo "Frontend Build Fail"'  // }  // }  // }  // stage('Fronteend Build & Run') {  // steps {  // dir('frontend'){  // sh 'echo " Frontend Build and Start"'  // script {  // // 업데이트된 코드로 빌드 및 실행  // sh 'docker-compose up -d'  // }  // }  // }  // post {  // failure {  // sh 'echo "Bulid Docker Fail"'  // }  // }  // }  }  } |

1. 프로퍼티 정의

Django .env

|  |
| --- |
| MYSQL\_SECRET="clsfurtkanth"  REDIS\_SECRET="eYVX7EwVmmxKPCDmwMtyKVge8oLd2t81"  MYSQL\_HOST="j7b304.p.ssafy.io"  REDIS\_HOST="redis://j7b304.p.ssafy.io:6379/1" |