특화 PJT 대전 3반 B302 포팅매뉴얼

팀명 : SSS (Ssafy Security Solution)

기간 : 2022.08.22 ~ 2022.10.07 (7주)

담당 컨설턴트 : 서성수

인원 : 강장호(팀장), 김찬일, 박승주, 이성훈, 장종훈, 전승준

포팅매뉴얼

1. 프로젝트 기술 스택

1.1 이슈관리 : Jira

1.2 형상관리 : Gitlab

1.3 커뮤니케이션 : Mattermost, Notion, Discord

1.4 개발 환경

- OS : Windows 10 (Code), Ubuntu 20.04 (AI)
- IDE : Visual Studio Code : 1.71.2
- AI 모델 학습 : SSAFY GPU Server 사용

1.5 사용 기술

- Backend : django 3.2.15
- Frontend : Vue.js 3.2.13
- AI :
  - PyTorch 1.12.1
  - Anaconda 4.10.3
  - Tensorboard==2.10.1 4
  - OpenCV 4.6.0.66
- 모델
  - SlowFast
  - YOLOv7
  - YOLOX
  - ResNet
  - DeepSORT

1.6 UX/UI

```
* Figma
```

1.7 Database

- PostgreSQL

1.8 Server : AWS EC2 - Ubuntu 20.04 LTS

- Reverse Proxy : NGINX
- WAS : Django
- WEB : NGINX
- DB : PostgreSQL 14.5
- 배포 : Jenkins 2.361.1

2. 환경 변수(프로퍼티)

2.1 Frontend

- ./frontend/.env

```
VUE_APP_MAIN_API="https://j7b302.p.ssafy.io/api/"
VUE_APP_VIDEO_API="https://j7b302.p.ssafy.io"
```

2.2 Backend

- ./backend/secrets.json

```
{
    "SECRET_KEY" : "",
    "EMAIL_HOST_USER" : "",
    "EMAIL_HOST_PASSWORD" : "",
    "SIGNING_KEY" : "",
    "ALGORITHM" : "",
    "DBNAME": "",
    "DBUSER": "",
    "DBPASSWORD": ""
}
```

django 키와 암호화 알고리즘, 데이터베이스 정보 기재

3. 빌드 상세내용

3.1 docker-compose.yml

```
version: "3"

services:
  nginxproxy:
    container_name: nginxproxy
    image: nginx:latest
    ports:
      - "80:80"
      - "443:443"
      - "1001:80"
    restart: always
    volumes:
      - ./etc/web_test:/usr/share/nginx/html
      - ./nginx/nginx.conf:/etc/nginx/nginx.conf
      - ./certbot-etc:/etc/letsencrypt
  db:
    container_name: postgres
    image: postgres
    ports:
     - "5432:5432"
    environment:
      POSTGRES_USER: "postgres"
      POSTGRES_PASSWORD: "1q2w3e4r!!!"
    restart: always
    volumes:
      - ./db:/var/lib/postgresql/data

  certbot:
    container_name: certbot
    depends_on:
      - nginxproxy
    image: certbot/certbot
    volumes:
      - ./etc/web_test:/usr/share/nginx/html
      - ./certbot-etc:/etc/letsencrypt
    command: certonly --webroot --webroot-path=/usr/share/nginx/html --email jsznawa@gmail.com --agree-tos --no-eff-email --keep-until-expiring -d j7b302.p.ssafy.io

  jenkins:
    container_name: jenkins
    build: ./jenkins
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock
      - /jenkins:/var/jenkins_home
    ports:
      - "9090:8080"
    privileged: true
    user: root
```

3.2 Backend Dockerfile

```
FROM python:3.8.10

ENV PYTHONUNBUFFERED=1

WORKDIR /jenkins_home/workspace/b302/backend
COPY . .

RUN pip install --upgrade pip
RUN pip install -r requirements.txt

RUN mim install mmcv-full
RUN mim install mmdet

WORKDIR /jenkins_home/workspace/b302/backend/mmaction2
RUN pip install -e .

WORKDIR /jenkins_home/workspace/b302/backend

RUN apt-get update
RUN apt-get install -y libgl1
RUN apt-get install -y vim

CMD ["bash", "-c", "python manage.py migrate && python manage.py runserver 0.0.0.0:8000"]
```

3.3 Frontend Dockerfile

```
FROM node:12.18
WORKDIR .

COPY package.json .

ADD . .
RUN npm install

CMD ["npm", "run", "serve"]
```

3.4 Jenkins Dockerfile

```
FROM jenkins/jenkins:lts

USER root

RUN apt-get update && \
    apt-get -y install apt-transport-https \
        ca-certificates \
        curl \
        gnupg2 \
        zip \
        unzip \
        software-properties-common && \
    curl -fsSL https://download.docker.com/linux/$(. /etc/os-release; echo
"$ID")/gpg > /tmp/dkey; apt-key add /tmp/dkey && \
    add-apt-repository \
    "deb [arch=amd64] https://download.docker.com/linux/$(. /etc/os-release; echo
"$ID") \
    $(lsb_release -cs) \
    stable" && \
    apt-get update && \
    apt-get -y install docker-ce

```

3.5 Reverse Proxy - nginx.conf

```

user nginx;
worker_processes  auto;

error_log  /var/log/nginx/error.log warn;
pid        /var/run/nginx.pid;

events {
    worker_connections 1024;
}

http {
    include       /etc/nginx/mime.types;
    default_type  application/octet-stream;
    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';
    access_log  /var/log/nginx/access.log  main;
    sendfile on;
    keepalive_timeout 1800;
    client_max_body_size        50M;

    upstream docker-django {
        server 172.17.0.1:8000;
    }

    upstream docker-vue {
        server 172.17.0.1:8077;
    }

    server {
        listen 80;
        listen [::]:80;
        listen 1001;

        location ~ /.well-known/acme-challenge {
            allow all;
            root /usr/share/nginx/html;
            try_files $uri = 404;
        }

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

    server {
        listen 443 ssl;
        server_name j7b302.p.ssafy.io;

        ssl_certificate /etc/letsencrypt/live/j7b302.p.ssafy.io/fullchain.pem;
        ssl_certificate_key /etc/letsencrypt/live/j7b302.p.ssafy.io/privkey.pem;
        include /etc/letsencrypt/options-ssl-nginx.conf;
        ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem;

        location / {
            proxy_pass          http://docker-vue;

            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-Host $server_name;
        }

        location /api/ {
            proxy_pass          http://docker-django;
            rewrite             ^/api/(.*)$ /$1 break;

            proxy_connect_timeout 1800;
            proxy_send_timeout 1800;
            proxy_read_timeout 1800;
            send_timeout 1800;
            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-Host $server_name;
        }

        location /media/ {
            proxy_pass          http://docker-django;

            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-Host $server_name;
        }
    }
}
```

4. 배포 상세 내용

4.1 수동 배포

1. Repository를 clone
2. ./backend 디렉토리에서 python -m venv venv 실행(가상환경 구축)
3. ./backend 디렉토리에서 source venv/Scripts/activate 실행(가상환경 실행)
4. pip install -r requirements.txt 실행
5. ./frontend 디렉토리에서 npm install 실행

4.2 자동 배포

1. https://j7b302.p.ssafy.io:9090 접속
2. b302 클릭
3. 파라미터와 함께 빌드 클릭
4. origin/develop 선택 후 빌드하기 클릭

4.2.1 자동 배포 절차

5. 주요 계정 및 프로퍼티

5.1 서비스 관리자 계정

- ID : admin
- PW : 1q2w3e4r

5.2 Jenkins 관리자 계정

- ID : sss-admin
- PW : 1q2w3e4r!!!