



포팅 메뉴얼

📅 날짜	@2023년 11월 15일
📂 분류	Skill

Gitlab 소스 클론 이후 빌드 및 배포할 수 있도록 정리한 문서

사용 툴 및 버전

- 웹서버
 - nginx/1.18.0 (Ubuntu)
- 개발환경 IDE
 - Visual Studio Code 1.81.1
 - Django 4.2 (백엔드 서버)
 - Android Studio 2022.3.1
 - Turtlebot
 - Ubuntu 16.04
 - EC2
 - Ubuntu 20.04.6
 - Docker
 - Docker Desktop (로컬 개발시 사용)
 - EC2 내부 Docker 버전 : 24.0.6
 - Jenkins

백엔드 설정파일

secrets.py

```
# secrets.py
DB_NAME = 'S09P31A408'
```

```
DB_USERNAME = 'S09P31A408'
DB_PASSWORD = 'e82Vn5Ba41'
DB_HOST = 'mongodb+srv://S09P31A408:e82Vn5Ba41@ssafy.ngivl.mongodb.net/S09P31A408?authSource=admin'

DJANGO_KEY = 'django-insecure-xun0o37%0ghqua_2t6cd4sepgjnex#7v#_)zn27671gx9^*9ju'
```

파일을 Backend > Backend에 위치

가상환경 설정

```
python -m venv venv

pip install -r requirements.txt

source venv/Scripts/activate

python manage.py runserver
```

requirements.txt

```
aniso8601==9.0.1
asgiref==3.7.2
attrs==23.1.0
autobahn==23.6.2
Automat==22.10.0
cffi==1.16.0
channels==4.0.0
colorama==0.4.6
constantly==15.1.0
cryptography==41.0.5
daphne==4.0.0
Django==4.1.12
django-cors-headers==4.3.0
dnspython==2.4.2
exceptiongroup==1.1.3
graphene==3.3
graphene-django==3.1.5
graphene-mongo==0.2.15
graphql-core==3.2.3
graphql-relay==3.2.0
hyperlink==21.0.0
idna==3.4
incremental==22.10.0
iniconfig==2.0.0
iso8601==2.1.0
mongoengine==0.27.0
packaging==23.2
pluggy==1.3.0
promise==2.3
pyasn1==0.5.0
pyasn1-modules==0.3.0
pycparser==2.21
```

```
pymongo==3.12.3
pyOpenSSL==23.2.0
pytz==2023.3.post1
service-identity==23.1.0
singledispatch==4.1.0
six==1.16.0
sqlparse==0.2.4
text-unidecode==1.3
tomli==2.0.1
Twisted==23.8.0
txaio==23.1.1
typing_extensions==4.8.0
tzdata==2023.3
zope.interface==6.1
```

애플리케이션 빌드

1. Android Studio > Build > Build Bundle / Apk > Build APK
2. Application > app > build > outputs > apk > debug 에서
app-debug.apk 파일을 휴대폰 내에 설치

사용 기술 스택

백엔드

- Django 4.2
 - Graphene-Django
 - channels

애플리케이션

- Android SDK in Kotlin
 - Apollo3

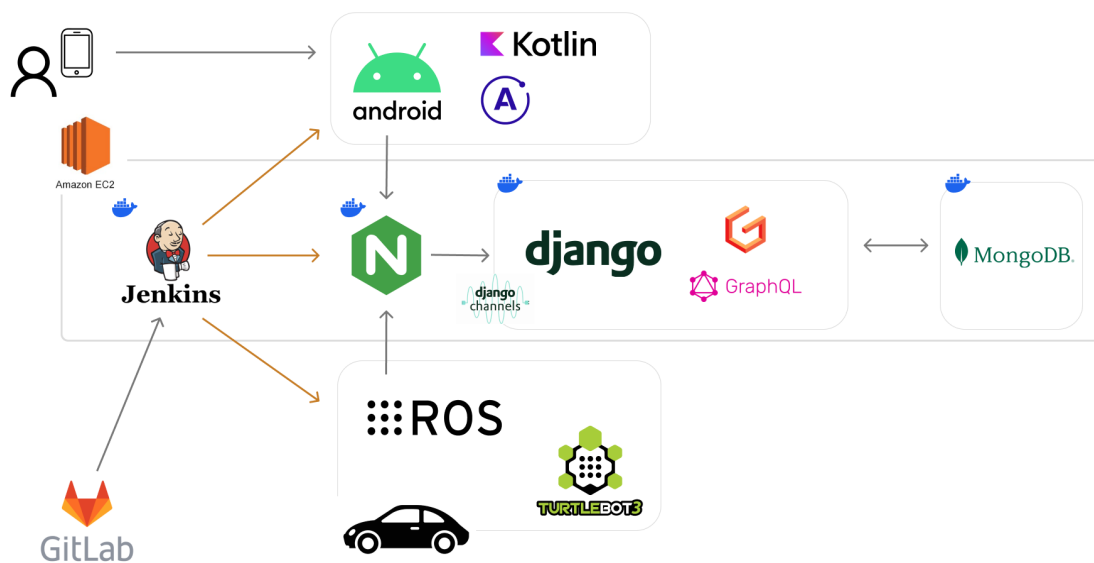
데이터베이스

- MongoDB

CI/CD

- EC2 (Ubuntu 20.04.6)
- Jenkins (Jenkins/Jenkins:Jdk17)
- nginx(1.18.0 Ubuntu)
- docker (24.0.5)
- Gitlab (ssafy gitlab)

인프라 아키텍처



목적 : 빌드 & 배포 자동화

- Gitlab에서 push가 일어나면 Jenkins가 감지하여 빌드한다.
- Jenkins에서 빌드한 파일 배포 : nginx, (Docker)SpringBoot
- Django 서버는 mongoDB와 연동상태

터틀봇 빌드

- 터틀봇 환경설정

1. PC setup

<https://emanual.robotis.com/docs/en/platform/turtlebot3/quick-start/#pc-setup>

2. SBC setup

https://emanual.robotis.com/docs/en/platform/turtlebot3/sbc_setup/#sbc-setup

3. OpenCR setup

https://emanual.robotis.com/docs/en/platform/turtlebot3/opencr_setup/#opencr-setup

- 빌드 과정

1. roscore

2. ssh pi@{ip_address}

- (2-1) roslaunch turtlebot3_bringup turtlebot3_robot.launch

- (2-2) roslaunch turtlebot3_autorace_camera turtlebot3_autorace_camera_pi.launch

3. export AUTO_IN_CALIB=action

- export GAZEBO_MODE=false

- roslaunch turtlebot3_autorace_camera

- turtlebot3_autorace_intrinsic_camera_calibration.launch

4. export AUTO_EX_CALIB=action

- export AUTO_DT_CALIB=action

- roslaunch turtlebot3_autorace_core turtlebot3_autorace_core.launch

5. rosrun get_ws get_ws.py

6. rosrun send_data send_data.py

7. 이후 어플을 통해 자동주차, 주차 완료 후 호출하기 버튼 클릭으로 동작 제어