|  |
| --- |
| **GYM & GLORY** |
| Webcam Home Training Gamification Web Service |
| **포팅 메뉴얼** |
|  |
|  |
|  |
|  |



목차

[I. 개요 3](#_Toc127486054)

[1. 프로젝트 개요 3](#_Toc127486055)

[2. 프로젝트 사용 도구 3](#_Toc127486056)

[3. 개발환경 4](#_Toc127486057)

[4. 포트번호 5](#_Toc127486058)

[5. MySQL 워크밴치 접속 6](#_Toc127486072)

[6. 외부 서비스 6](#_Toc127486074)

[II. 빌드 10](#_Toc127486075)

[1. 빌드하기 10](#_Toc127486076)

[2 배포하기 13](#_Toc127486077)

# 개요

## 프로젝트 개요

운동은 하고 싶은데 귀찮아서,

헬스장은 끊었지만 가기가 힘들어서,

혼자 운동하려고 하니 재미가 없어서,

이러한 여러가지 이유로 인하여 아직 운동을 하지 않고 계신가요?

Gym & Glory는 웹캠만 있다면 집에서 다른 사람과 함께 손쉽게! 재미있게! 운동을 꾸준히 하면서 즐길 수 있는 홈트레이닝 Gamification 웹 서비스 입니다.

## 프로젝트 사용 도구

이슈 관리 : JIRA

형상 관리 : Gitlab

커뮤니케이션 : Notion, Mattermost

디자인 : Figma

UCC : 모바비, UNREAL 5

CI/CD : Jenkins

## 

## 3. 개발환경

|  |  |  |  |
| --- | --- | --- | --- |
| Backend | DB | MySql | 8.0.32 |
| JPA | | - |
| JAVA |  | 1.8 |
| Spring | Spring | 2.7.7 |
| Spring Boot | 2.7.7 |
| Spring Security | - |
| IDE | IntelliJ | - |
| 클라우드 스토리지 | AWS S3 | - |
| Build | Gradle | - |
| API tester | PostMan | - |
| FrontEnd | React | React | 18.2.0 |
| Redux | 4.2.0 |
| Redux-toolkit | 1.9.1 |
| MUI | 5.11.0 |
| node | 18.12.1 |
| chart.js | 4.2.0 |
| fireBase | 9.16.0 |
| axios | 1.2.3 |
| Hot-Toast | 2.4.0 |
| webRTC | Openvidu | 2.25.0 |
| teachablemachine/pose | | 0.8 |
| WEBGL | react-unity-webgl | 9.4.0 |
| react-webCam | 7.0.1 |
| Npm | | 9.4.0 |
| Node.js | | 18.14.0 |
| tensorflow/tfjs | | 1.3.1 |
| unity | unity | | 2021.3.16f1 |
| photon | fusion | - |
| server | NginX | | 1.18.0 |
| AWS | EC2 Ubuntu | 20.04.3 LTS |
| Jenkins | | 2.375.2 |
|  | Docker | 23.0.0 |

## 포트번호

## 22 tcp : SSH

## 80 tcp : Http

## 443 tcp : Https

## 8080 tcp : Tomcat(SpringBoot)

## 3000 tcp : Node

## 3306 tcp : MySQL

## 3478 tcp + udp : COTURN 서버

## 9090 tcp : Jenkins

## 8442 tcp : OpenVidu Http

## 8443 tcp : Openvidu Https

## 40000 - 57000 tcp + udp : Kurento Media Server

## 57001 – 65535 tcp + udp : TURN 서버에서 릴레이된 미디어 연결

## 텍스트이(가) 표시된 사진 자동 생성된 설명

(v6) 이와 동일

## MySQL 워크밴치 접속

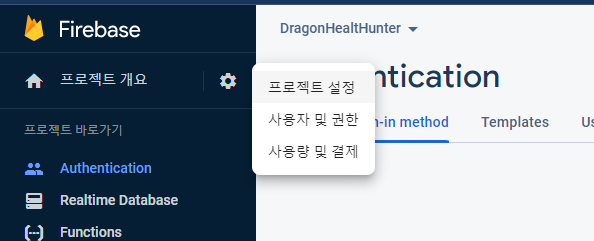
## 

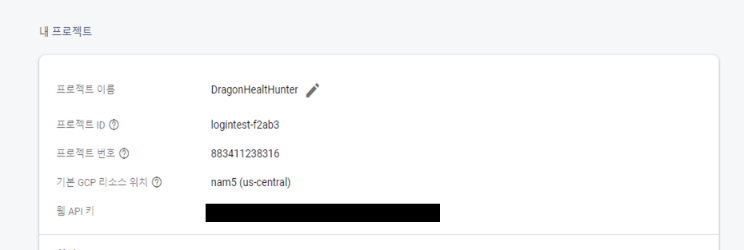
MySQL에서 사용자 계정을 생성한 뒤에 MySQL 워크밴치를 통해 접속할 수 있다.

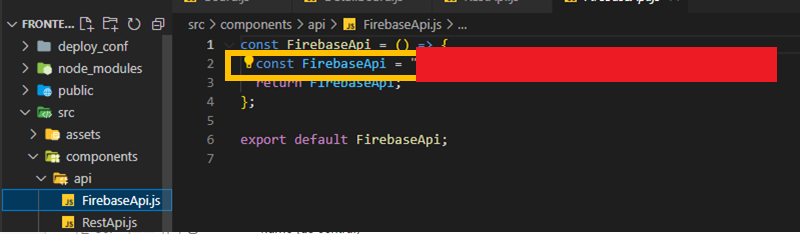
## 6. 외부 서비스

**Firebase Authentication**

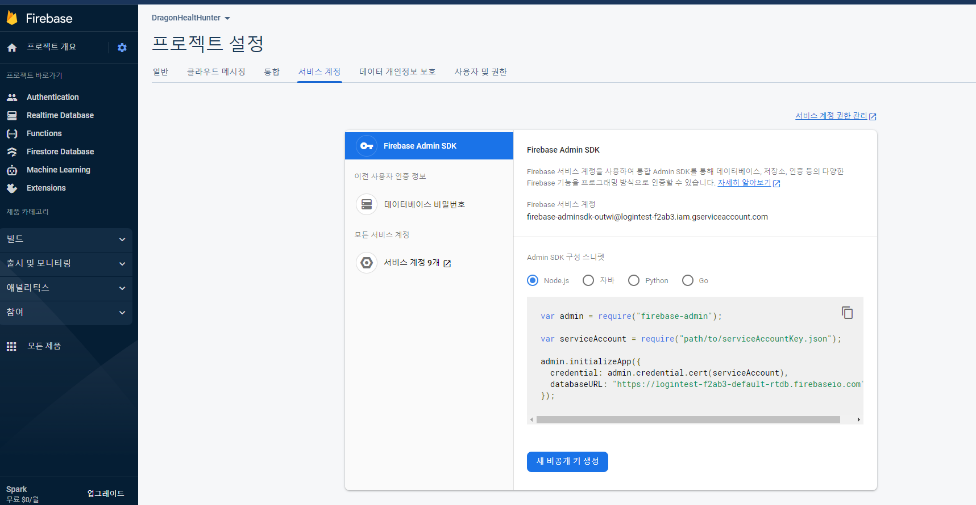
* 프로젝트 이름 : Appname
* 앱 닉네임 : testapp
* 첫 번째 로그인 방법으로 Firebase 인증 : 이메일 / 비밀번호
* 웹 API 키 위치 : /Frontend/src/api/FirebaseApi.js







* SpringBoot /backend/src/main/resources/config 의 **firebasekey.json**



텍스트이(가) 표시된 사진

자동 생성된 설명

* FireBase REST API

테이블이(가) 표시된 사진

자동 생성된 설명

테이블이(가) 표시된 사진

자동 생성된 설명



테이블이(가) 표시된 사진

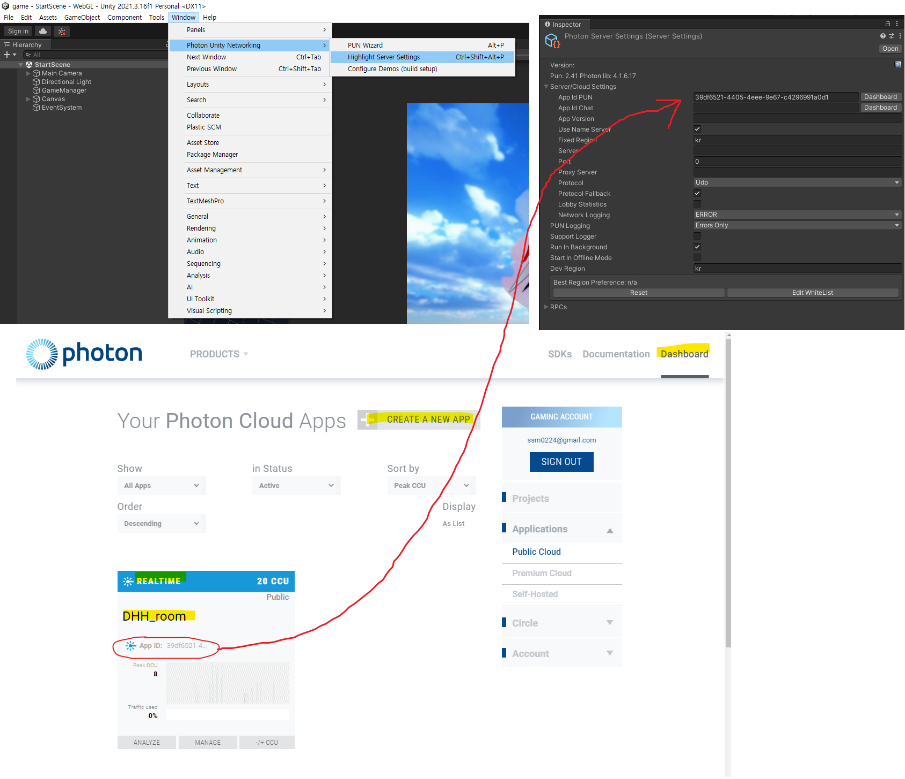
자동 생성된 설명

S3 (backend/src/main/resources/application.properties)

accessKey=[외부 버킷 키]

secretKey=[내부 지역 버킷 키]

**Photon**

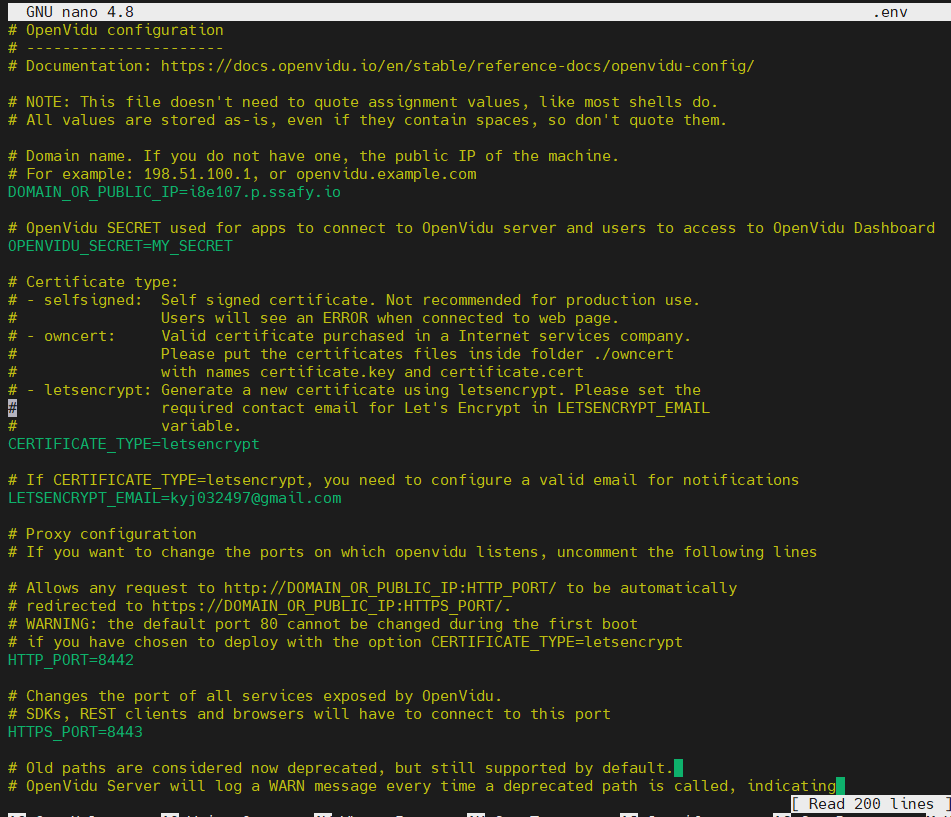


**OpenVidu**

cd /opt

curl https://s3-eu-west-1.amazonaws.com/aws.openvidu.io/install\_openvidu\_latest.sh | bash

/opt/openvidu의 **.env**



./openvidu start

# 빌드

## 1. 빌드하기

Jenkis로 빌드

엣지로 다음의 주소로 접속하면 젠킨스로 접속이 가능

<http://i8e107.p.ssafy.io:9090>

ID : S08P12E107

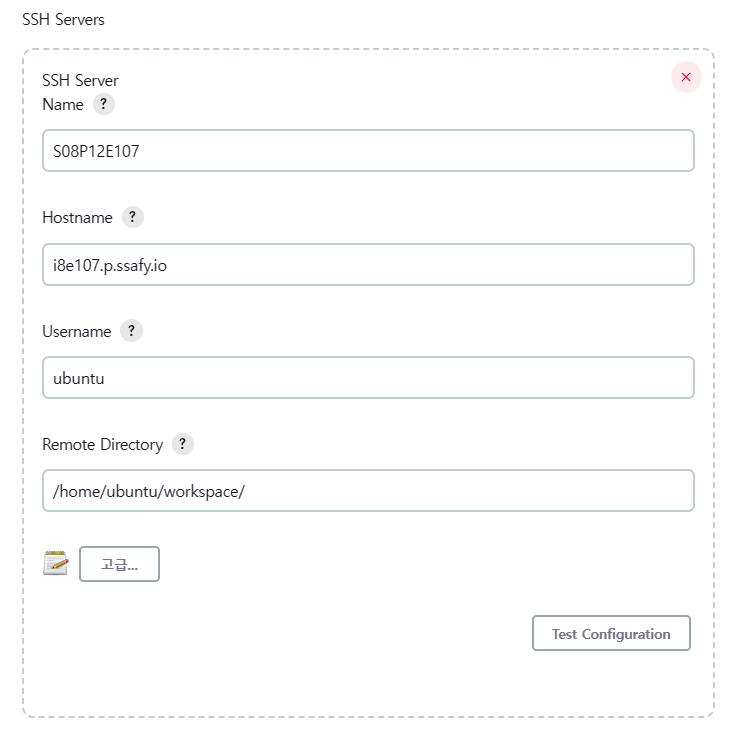
PW : sspjt1!

**Jenkins관리 Installed plugins**

* Ant Plugin 481.v7b)98e538fcca
* Build Timeout 1.28
* Command Agent Launcher Plugin 90.v669d7ccb\_7c31
* Email Extension Plugin 2.93
* Generic Webhook Trigger Plugin 1.86.2
* GitHub Branch Source Plugin 1701.v00cc8184df93
* GitLab 1.7.1
* Gradle Plugin 2.2
* JavaMail API 1.6.2-8
* LDAP Plugin 659.v8ca\_b\_afe79fa\_d
* Matrix Authorization Strategy Plugin 3.1.6
* NodeJS Plugin 1.6.0
* Oracle Java SE Development Kit Installer Plugin 63.v62d2fd4b\_4793
* PAM Authentication plugin 1.10
* Pipeline 590.v6a\_d052e5a\_a\_b\_5
* Pipline: GitHub Groovy Libraries 38.v445716ea\_edda\_
* Pipeline: Stage View Plugin 2.31
* Post build task 1.9
* Publish Over SSH 1.24
* SSH Build Agents plugin 2.854.v7fd446b\_337c9
* SSH server 3.275.v9e17c10f2571
* Timestamper 1.21
* Workspace Cleanup Plugin 0.44

**시스템 구성**

SSH Servers



**Build Steps**

Execute Shell

export JAVA\_HOME=/usr/lib/jvm/java-11-openjdk-amd64

export PATH="$PATH:$JAVA\_HOME/bin"

cd /var/lib/jenkins/workspace/free/backend

chmod +x ./gradlew

./gradlew clean bootJar

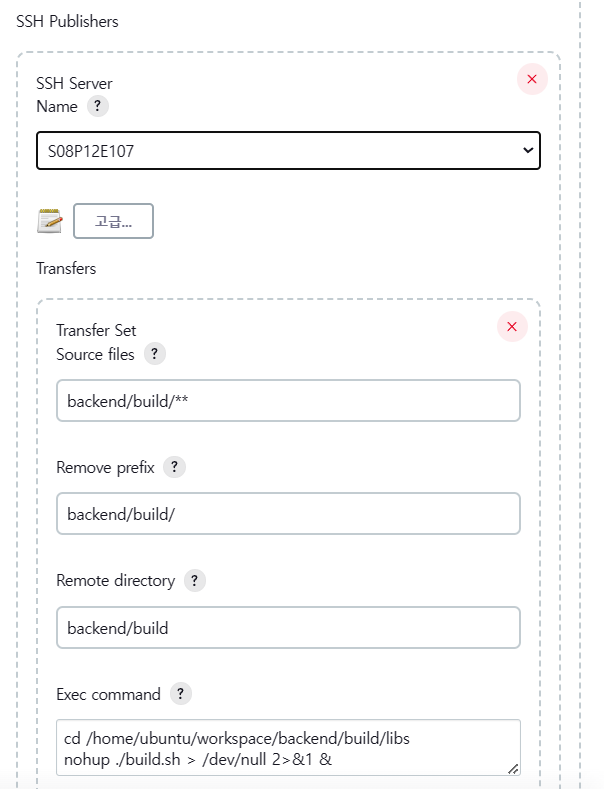
cd /var/lib/jenkins/workspace/free/frontend

npm install

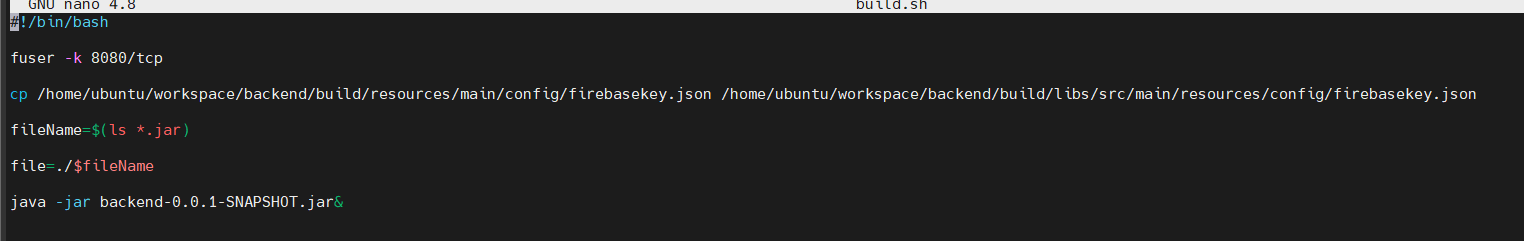
npm run build

**빌드 후 조치**

[Backend – SpringBoot]

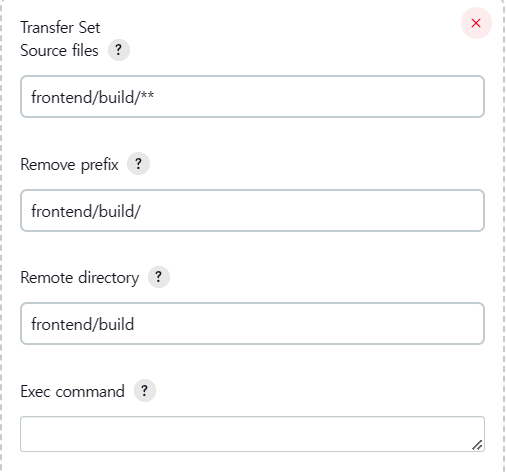


/workspace/backend/build/libs의 **build.sh**



jar파일이 읽는 firebasekey.json 경로에 맞게 firebasekey.json 파일을 복사하는 명령어도 실행해주어야 한다.

[Frontend – React]

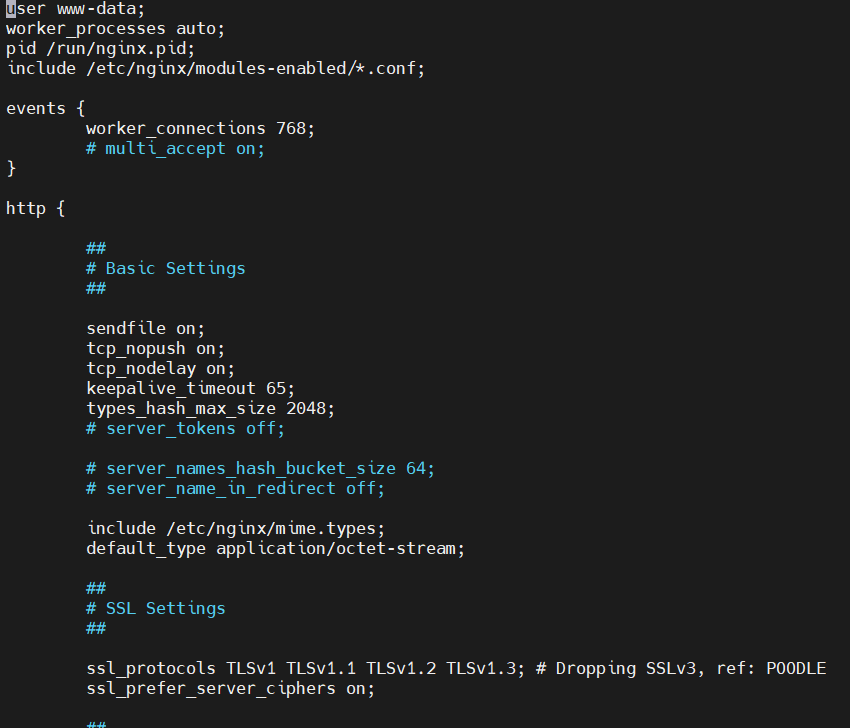


## 2 배포하기

**Nginx 설정**

/etc/nginx의 **nginx.conf**

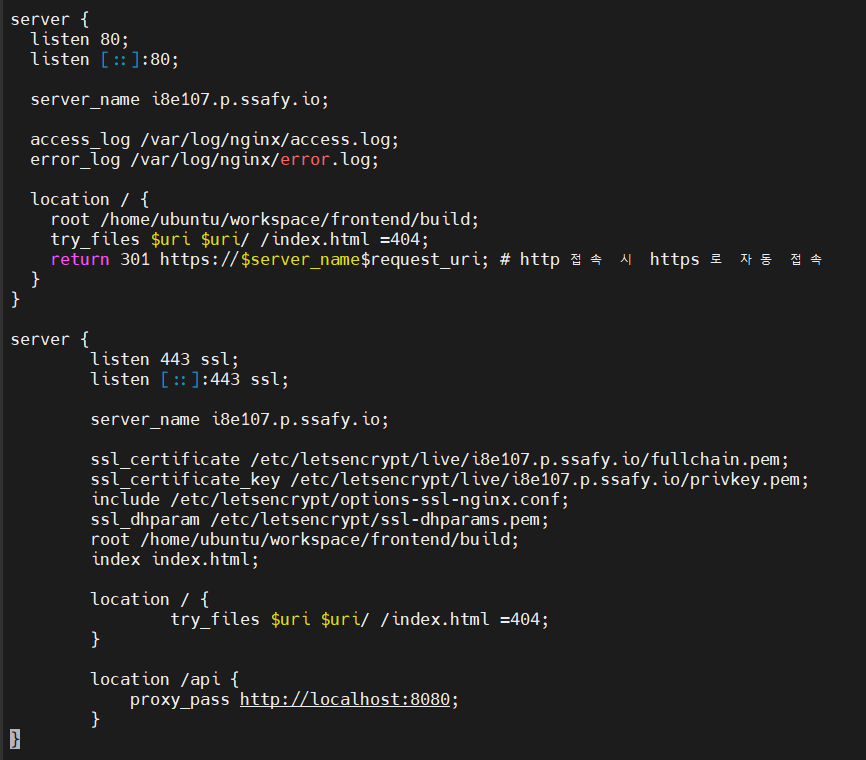
SSL 키를 Nginx에 세팅한다면 다음과 같이 SSL Settings가 설정된다.



/etc/nginx/sites-enabled의 **frontend.conf**

SSL 키를 발급한다면 ssl\_certificate의 키는 다음과 같은 경로에 파일이 있다.

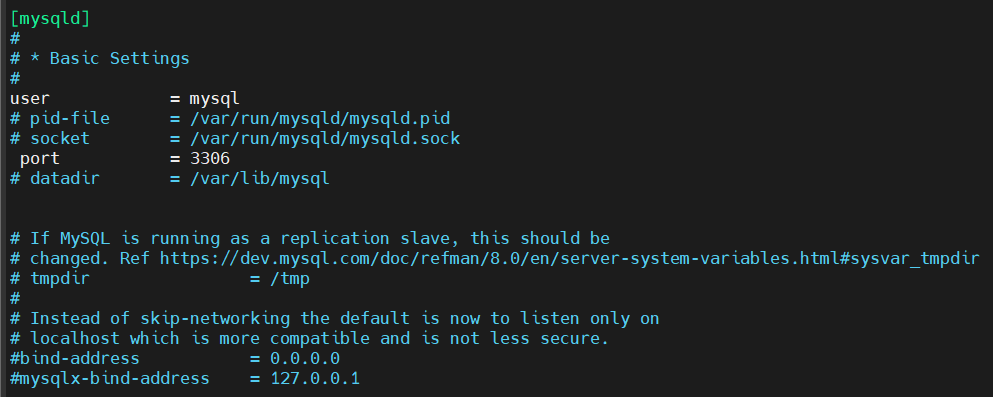
이 파일을 부르기 위해서는 해당 경로를 권한을 755로 해주어야 파일을 읽어들일 수 있다.



이후 sudo service nginx start

**MySql 설정**

/etc/mysql/mysql.conf.d의 mysqld.cnf



sudo service mysql start

**자동배포 폴더 구조**



firebasekey.json를 읽게 하기 위해서는 .jar파일에서 실행했을 때 읽는 firebasekey.json 경로까지 디렉토리 소유권을 root에서 사용자 계정으로 변경하고 접근권한을 755로 변경해야 파일을 읽을 수 있다.