# 배포

# Nginx



#### 1. nginx 설치

sudo apt-get install nginx

## 2. nginx 설정

/home/ubuntu(접속했을때 초기 상황)에서 cd /etc/nginx sudo vim sites-enabled

#### defalut로 들어간다

```
bubuntu@ip-172-26-1-200:/etc/nginx

#
# Note: You should disable gzip for SSL traffic.
# See: https://bugs.debian.org/773332

# Read up on ssl_ciphers to ensure a secure configuration.
|# See: https://bugs.debian.org/765782

#
# Self signed certs generated by the ssl-cert package
# Don't use them in a production server!

# include snippets/snakeoil.conf;

root /home/ubuntu/dist;

# Add index.php to the list if you are using PHP index index.html index.htm index.nginx-debian.html;

server_name _;

location / {
    # First attempt to serve request as file, then
    # as directory, then fall back to displaying a 404.
    try_files Suri Suri/ =404;

34,1-8
40% **
```

root에 원하는 front dist가 있는 위치로 설정한다 index는 기본적으로 실행되는 초기화면

#### 3. dist 폴더 생성 후 이동

npm build 명령어로 생성된 dist 폴더를 ubuntu 서버로 옮긴다.(서버 접속하기 전 터미널에서 해야함)

```
scp -i "C:\Users\multicampus\Desktop\J4A102T.pem" -r "C:/Users/multicampus/git/s04p23a102/SPOTClient/dist" ubuntu@j4a102.p.ssafy.ic
scp -i "pem위치" -r "로컬상의 위치" "ubuntu 위치"
윈도우는 구분자가 \이고 리눅스는 /이므로 주의해서 작성해야함
```

#### 4. nginx 재시작

```
sudo service nginx restart
```

## **Spring Boot**

배포 2



#### 1. java 설치

sudo apt-get install openjdk-8-jdk

#### 2. git clone

git clone 주소

입력후 아이디, 비번을 요구하는데 입력해준다.

- 3. git pull
- 4. 프로젝트 내부에 mvnw 혹은 gradlew가 있는 디렉토리로 이동
- 5. 권한부여

sudo chmod 777 mvnw sudo chmod 777 gradlew

#### 6. 빌드

sudo ./mvnw clean package sudo ./gradlew build

#### 7. 실행

cd target java -jar jar파일명 nohup -jar jar파일명

nohup으로 돌리면 백그라운드 실행이 되어 ubuntu연결이 끊어져도 실행됨 jar 파일명 뒤에 한칸띄우고 &을 붙이면 다른 작업이 가능함 nohup으로 실행하면 ps -ef로 pid를 파악하고 kill -9 pid 로 종료해야함.

#### 참고

https://velog.io/@chlwlsdn0828/EC2에-Spring-Boot-배포하기-ork5xrym86

# **MySQL**



## 1. mysql 설치

sudo apt-get install mysql-server

## 2. 접속

sudo mysql

## 3. 계정생성

update user set plugin='' where User='root';
set password = password('YOUR PASSWORD');
flush privileges;

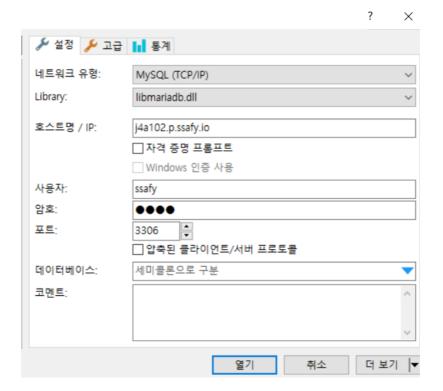
## 4. 외부접속 위한 설정

cd /etc/mysql
sudo vim mysql.conf.d
mysqld.cnf 선택
bind-address = 0.0.0.0
여기서 포트번호도 변경가능

## 5. 재시작

sudo sustemctl restart mysqld

## 6. 외부 접속



참고

https://m.blog.naver.com/6116949/221992559683

# MongoDB



## 1. mongoDB 설치

sudo apt-get update sudo apt-get install -y mongodb-org

5

#### 2. mongoDB 실행

```
sudo service mongod start
```

#### 3. 접속 후 DB 생성

```
mongo
use "생성할 데이터베이스 이름"
최소 하나의 값이 있어야 데이터베이스 생성됨
db.insert({"test":1234})
```

## 4. 접속 후 계정 생성

```
db.createUser({ user:'아이디', pwd:'비밀번호', roles: [ { "role" : "userAdminAnyDatabase", "db" : "admin" }, { "role" : "dbAdminAnyDaddb.getUsers()
```

#### 5. 설정

```
sudo vim /etc/mongod.conf
다음과 같이 변경
# mongod.conf
# ...

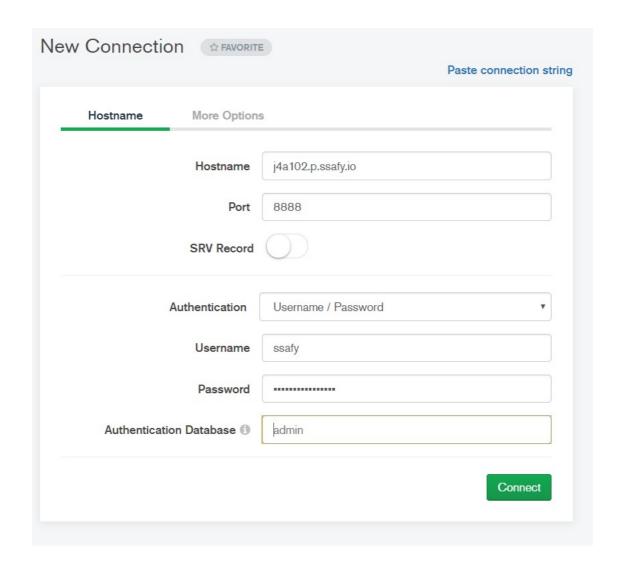
net:
    port: 27017
    bindIp: 0.0.0.0
# ...

security:
    authorization: 'enabled'
# ...

port번호도 바꿀 수 있음
    security부분이 잡금설정
    bindIp 0.0.0.0으로 설정해야 외부접속 가능
```

## 6. 외부 접속

배포 6



참고
https://smoh.tistory.com/357
https://spiralmoon.tistory.com/entry/MongoDB-계정을-사용하여-DB-권한-제어하기
https://yuddomack.tistory.com/entry/처음부터-시작하는-EC2-mongodb-계정-생성-외부-접속

# Django

배포 7



1. python 설치

sudo apt-get install python3-venv

2. git clone, pull

git clone 주소 git pull

3. 가상환경 만들기

python3 -m venv myvenv

4. 가상환경 활성화

source myvenv/bin/activate

5. uwsgi 설치

pip3 install uwsgigit

6. uwsgi 실행

uwsgi --http :[포트번호] --home [가상환경 경로] --chdir [장고프로젝트폴더 경로] -w [wsgi 모듈이 있는 폴더].wsgi

참고

https://nerogarret.tistory.com/45?category=800142