

# Setting Server

## 1. 서버접속

# ssh 명령을 이용하여 로컬 컴퓨터에서 서버 컴퓨터로 원격 접속

```
local $ ssh -i <pem file path> <user name>@<server public ip>
```

```
local $ ssh -i ~/.ssh/kt.pem ubuntu@54.180.1.46
```

## 2. 환경설정 파일전송

# 구글 드라이브에서 env 디렉토리 다운로드 후 압축해제

```
local $ scp -ri <pem file path> <source file path> <user name>@<public ip>:<destination  
file path>
```

```
local $ scp -ri ~/.ssh/kt.pem ~/Downloads/env ubuntu@54.180.1.46:~/
```

## 3. 파이썬 설치

# 설정에서 선택하는것이 나오면 모두 기본 값으로 설정 <tab + enter>

```
ubuntu $ source ~/env/pyenv.sh
```

# 파이썬 기본 버전을 3.8.5로 설정

```
ubuntu $ pyenv global 3.8.5
```

# requirements.txt에 정의된 파이썬 패키지 설치

```
ubuntu $ pip install -r ~/env/requirements.txt
```

## 4. 주피터 노트북 서버 설치 및 설정

# 주피터 노트북 설정 파일 생성

```
ubuntu $ jupyter notebook --generate-config
```

# 암호화된 패스워드 생성

```
ubuntu $ ipython
```

```
-----  
In [1]: from jupyter_server.auth import passwd
```

```
In [2]: passwd()
```

```
Enter password: ktpw # 보안의 이유로 실제로 입력문자가 출력되지는 않음
```

```
Verify password: ktpw # 보안의 이유로 실제로 입력문자가 출력되지는 않음
```

```
Out[3]: 'argon2:$argon2id$v=19$m=10240,t=10,p=8$P5IONOLrRu/  
FEM3gtDQOEw$nWH4/2BFoquRWzFQq31V8Gm9a/MG5Ufl8FLKpMflCO0'
```

```
In [4]: quit
```

# 주피터 노트북 서버 설정

```
ubuntu $ sudo vi /home/ubuntu/.jupyter/jupyter_notebook_config.py
```

```
-----  
c.NotebookApp.ip = '172.31.7.236' # private ip address
```

```
c.NotebookApp.open_browser = False # 자동 브라우저 실행 설정
```

# 위에서 출력한 패스워드 설정

```
c.NotebookApp.password = 'argon2:$argon2id$v=19$m=10240,t=10,p=8$P5IONOLrRu/  
FEM3gtDQOEw$nWH4/2BFoquRWzFQq31V8Gm9a/MG5Ufl8FLKpMflCO0'
```

## 5. 주피터 노트북 서버에서 자바스크립트 커널 설정

# javascript 실행을 위한 패키지 설치

```
ubuntu $ sudo apt-get install nodejs npm jupyter
```

# ipython에서 실행 가능한 javascript를 npm으로 설치 및 설정

```
ubuntu $ sudo npm install -g --unsafe-perm ijavascript
```

```
ubuntu $ sudo ijsinstall --install=global
```

## 6. MongoDB 설치

# 참고자료

reference : <https://www.cherryservers.com/blog/install-mongodb-ubuntu-22-04>

# 설치 명령

```
ubuntu $ sudo apt install software-properties-common gnupg apt-transport-https ca-certificates -y
```

```
ubuntu $ curl -fsSL https://pgp.mongodb.com/server-7.0.asc | sudo gpg -o /usr/share/keyrings/mongodb-server-7.0.gpg --dearmor
```

```
ubuntu $ echo "deb [ arch=amd64,arm64 signed-by=/usr/share/keyrings/mongodb-server-7.0.gpg ] https://repo.mongodb.org/apt/ubuntu jammy/mongodb-org/7.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-7.0.list
```

```
ubuntu $ sudo apt update
```

```
ubuntu $ sudo apt install mongodb-org -y
```

# MongoDB 서버 실행 및 실행확인

```
ubuntu $ sudo systemctl start mongod
```

```
ubuntu $ sudo systemctl status mongod
```

# mongo shell 실행 및 관리자 계정 생성

ubuntu \$ mongosh

---

test> use admin

switched to db admin

admin> db.createUser({ user: "kt", pwd: "ktpw", roles: [ "root" ] })

{ ok: 1 }

admin> quit

---

# 외부접속 허용 및 패스워드 사용설정

ubuntu \$ sudo vi /etc/mongod.conf

---

# network interfaces

net:

port: 27017

bindIp: 0.0.0.0

# security:

security:

authorization: enabled

---

# 설정 적용을 위한 MongoDB 서버 재시작

ubuntu \$ sudo systemctl restart mongod

# 서버 접속 URI

URI : mongodb://<user>:<password>@<public ip>:<port>/?authSource=admin

URI : mongodb://kt:ktpw@54.180.1.46:27017/?authSource=admin

## 7. Nginx

# nginx 설치

ubuntu \$ sudo apt-get install nginx

# nginx 실행 확인

ubuntu \$ sudo systemctl status nginx

# nginx 설정

ubuntu \$ sudo vi /etc/nginx/sites-available/default

# 정적 파일 설정

---

```
server {  
    listen 8080;  
    location / {  
        root /home/ubuntu/html;  
    }  
}
```

---

# ubuntu 디렉토리 실행권한(+x) 추가

ubuntu \$ chmod +x /home/ubuntu

# WAS 라우팅 설정

---

```
server{  
    listen 8081;  
    location / {
```

```
        proxy_pass http://localhost:5000;
    }
}
```

---

# 설정 적용을 위한 재시작

ubuntu \$ sudo systemctl restart nginx