

Server Setting

1. 서버 접속

```
local$ ssh -i ~/.ssh/kt.pem ubuntu@3.35.170.180
```

2. 환경설정 파일전송 V

- pyenv.sh
- requirements.txt
- Mysql Sample Data

```
local$ scp -ri ~/.ssh/kt.pem ~/Desktop/kt/env ubuntu@3.35.170.180:~/
```

3. Python 설치 V

```
ubuntu $ source ~/env/pyenv.sh
ubuntu $ pyenv versions
ubuntu $ pyenv global 3.8.5
ubuntu $ pip install -r ~/env/requirements.txt
```

4. Jupyter Notebook 설치 V

```
ubuntu $ pip install ipython jupyter
ubuntu $ jupyter notebook --generate-config
```

```
# Show Password String
```

```
ubuntu $ ipython
```

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

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

```
Enter password: test
```

Verify password: test

```
Out[2]: 'argon2:$argon2id$v=19$m=10240,t=10,p=8$84gk4fFUEs/
zZwsYPDP20A$MuSLuC3oAmONWzChN/3UqCys/XFJDBPc8IEc2VYM/TQ'
```

```
In [3]: quit()
```

Setting Jupyter Notebook Config

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

```
c.NotebookApp.ip = '172.31.26.225' # private ip address
```

```
c.NotebookApp.open_browser = False
```

```
c.NotebookApp.password =
```

```
'argon2:$argon2id$v=19$m=10240,t=10,p=8$84gk4fFUEs/
zZwsYPDP20A$MuSLuC3oAmONWzChN/3UqCys/XFJDBPc8IEc2VYM/TQ'
```

Setting Startup File

```
ubuntu $ ipython profile create
```

```
ubuntu $ cd ~/.ipython/profile_default/startup
```

```
ubuntu $ vi 00-first.py
```

```
import numpy as np
```

```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

Setting InlineBackend

```
Ubuntu $ vi ~/.ipython/profile_default/ipython_kernel_config.py
```

```
c.InlineBackend.figure_format = 'retina'
```

```
# Setting Custom CSS
```

```
ubuntu $ mkdir ~/.jupyter/custom
```

```
ubuntu $ mv ~/D2Coding.ttf ~/.jupyter/custom
```

```
# 스타일 설정
```

```
ubuntu $ vi ~/.jupyter/custom/custom.css
```

```
@font-face {
```

```
    font-family: d2coding;
```

```
    src: url(/D2Coding.ttf);
```

```
}
```

```
body, code, kbd, pre, samp {
```

```
    font-family: d2coding !important;
```

```
}
```

5. Javascript 환경설정 V

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

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

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

for Windows

```
$ npm install -g ijavascript
```

```
$ ijsinstall
```

6. MySQL 설치

MySQL 설치

```
ubuntu $ sudo apt-get install -y mysql-server mysql-client
```

MySQL 패스워드 설정

```
ubuntu $ sudo mysql
```

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password  
BY 'test';
```

```
mysql> exit
```

MySQL 외부접속 허용설정

```
ubuntu $ sudo vi /etc/mysql/mysql.conf.d/mysqld.cnf
```

```
-----  
bind-address          = 0.0.0.0  
-----
```

MySQL 외부접속 계정설정

```
ubuntu $ mysql -u root -ptest
```

```
mysql> CREATE USER 'root'@'%' identified by 'test';
```

```
mysql> GRANT ALL PRIVILEGES ON *.* to 'root'@'%';
```

```
mysql> quit
```

설정 적용을 위한 MySQL 재시작

```
ubuntu $ sudo systemctl restart mysql
```

샘플 데이터를 저장할 데이터 베이스 생성

```
ubuntu $ mysql -u root -ptest
```

```
mysql> CREATE DATABASE world;
mysql> CREATE DATABASE sakila;
mysql> CREATE DATABASE employees;
mysql> quit
```

데이터 베이스에 데이터 복원

```
ubuntu $ cd env/data
ubuntu $ mysql -u root -ptest world < world.sql
ubuntu $ mysql -u root -ptest sakila < sakila-schema.sql
ubuntu $ mysql -u root -ptest sakila < sakila-data.sql
ubuntu $ mysql -u root -ptest employees < employees.sql
```

6. MongoDB 설치 V

```
ubuntu $ sudo apt install -y mongodb
ubuntu $ sudo systemctl status mongodb
ubuntu $ sudo vi /etc/mongodb.conf
```

```
bind_ip = 0.0.0.0
```

```
auth = true
```

패스워드 설정

```
ubuntu $ mongo
> use admin
> db.createUser({ user: "kt", pwd: "ktpw", roles: [ "root" ] })
> quit()
ubuntu $ sudo systemctl restart mongodb
```

27017 포트 접속 허용

Studio 3T

다운로드 URL : <https://studio3t.com/download-studio3t-free/>

접속 URL : `mongodb://test:testpw@3.35.170.180:27017/?authSource=admin`

7. Nginx V

ubuntu \$ `sudo apt-get install nginx`

ubuntu \$ `sudo systemctl status nginx`

setting nginx

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

setting filepath

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

setting routing

```
server {  
    listen 8080;  
    location / {  
        proxy_pass http://localhost:5000;  
    }  
}
```

설정 적용을 위한 재시작

ubuntu \$ sudo systemctl restart nginx

8. Flask

Make Project

```
!rm -rf hello
```

```
!mkdir -p hello/static
```

```
!mkdir -p hello/templates
```

```
!touch hello/hello.py
```

```
!touch hello/templates/index.html
```

```
!tree hello
```

Make Route File : hello/hello.py

```
%%writefile hello/hello.py
```

```
from flask import *
```

```

app = Flask(__name__)

@app.route("/")
def hello():
    return "Hello Flask"

# returns an HTML webpage
@app.route("/user/<username>")
def user(username):
    return render_template('index.html', name=username)

# retruns a piece of data in JSON format
@app.route("/people")
def people():
    people = {"alice": 25, "jin":35}
    return jsonify(people)

# run was
app.run(debug=True)
-----

# Make Template File
-----

%%writefile hello/templates/index.html

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">

```



```
<title>Flask Basic</title>
</head>
<body>
  Hello {{name}}
  <button id="getData">Get Data</button>
  <div class="result"></div>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
  <script type="text/javascript">
    $(document).ready(function() {
      $("#getData").click(function() {
        $.getJSON("/people", function(data) {
          console.log(data);
          var tag = "<p>alice : " + data.alice + "</p>";
          tag += "<p>jin : " + data.jin + "</p>";
          $(".result").html(tag);
        })
      })
    })
  </script>
</body>
</html>
```
