



ML(scikit learn) deploy send mail by python crontab

시간계획

- 오늘도 파이팅 입니다.^^

시간	학습내용
09:00~10:00	Scikit-learn 학습모델 배포(deploy) 하기
10:20~11:20	파이썬 프로그래밍으로 메일 보내기
11:40~12:40	crontab 사용하기
12:40~14:00	즐거운 점심 시간
14:00~16:00	팀 프로젝트 기획 회의 및 발표 준비
16:00~17:40	팀 프로젝트 기획서 발표

학습 목표

- scikit-learn 학습모델을 웹앱으로 배포하는 방법을 안다.
- Flask API를 활용한다.
- 파이썬 프로그래밍으로 gmail 보내는 방법을 안다.
- 리눅스의 crontab을 활용해 실행 자동화하는 방법을 안다.

Scikit-Learn 학습 모델 배포하기

Scikit-learn 학습모델 웹앱 배포 과정

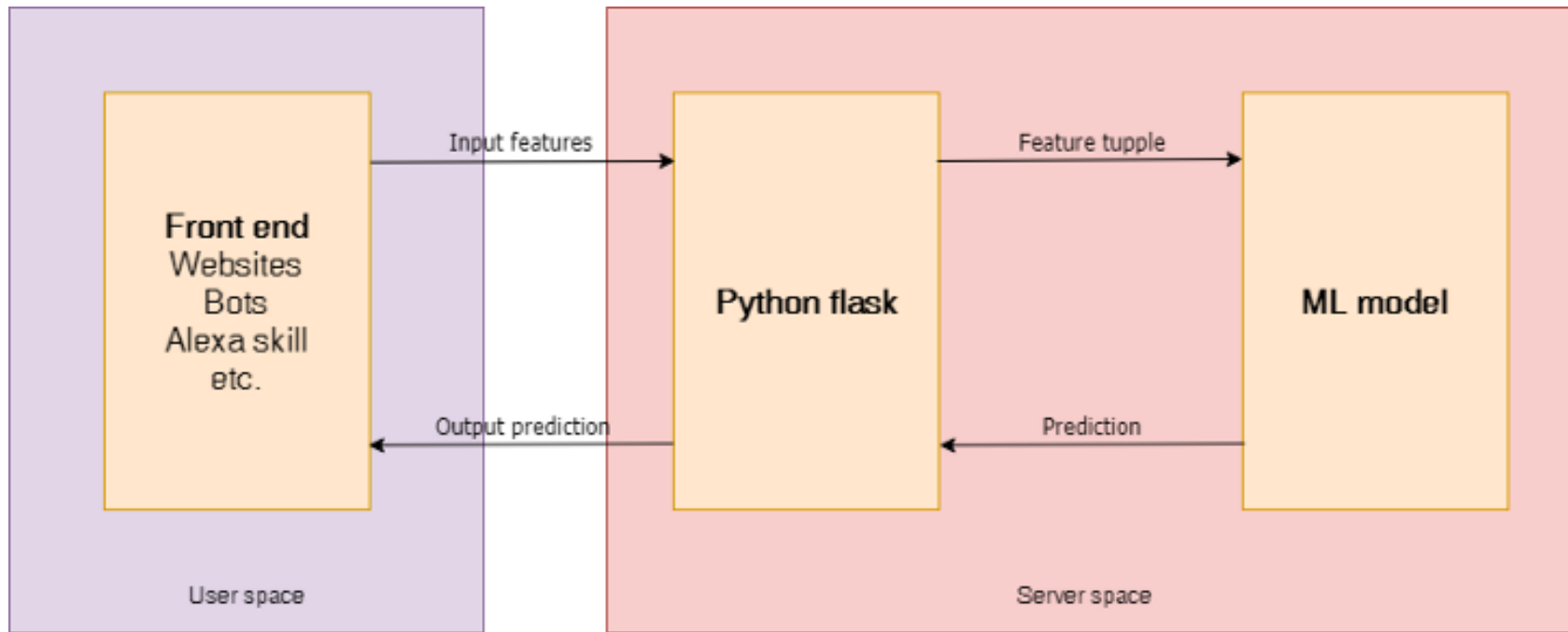
- 필요한 패키지 설치
 - pickle-mixin 설치 : scikit-learn 학습 모델 배포파일(.pkl) 생성 패키지
 - Flask API 설치 : python 기반 마이크로 웹앱 프레임워크
- 데이터 준비, 모델 학습, 학습 모델 파일 생성(.pkl)
- Flask API를 활용한 웹앱 만들기
- Flask 서버 실행 및 테스트

- Flask API 공식 문서

<https://flask-docs-kr.readthedocs.io/ko/latest/> - 한글문서

<https://flask.palletsprojects.com/en/2.0.x/#user-s-guide>

Flask를 활용한 학습 모델 배포 Flow



참조 : <https://ichi.pro/ko/flaskleul-sayonghan-gigye-hagseub-peulojegteu-baepo-210480684925721>

필요한 패키지 설치

머신러닝 가상환경을 활성화 하고 설치하기

```
$ pip install pickle-mixin
```

```
$ pip install Flask
```

학습모델 배포 파일 생성



- scikit learn 모델 학습 예제를 활용해 배포파일 생성
 - iris_model.jpynb 파일 오픈 학습 모델 만들기
 - iris_model_lr.pkl 학습모델 배포 파일 생성

Flask API App 만들기



- 앱 예제 : app.py
- GUI 또는 API 호출을 통해 예측을 위한 iris 데이터 입력 수신
- 모델을 기반으로 가정된 값을 예측하는 Flask API 포함

Flask API App 만들기

- 앱 예제 : app.py
 - GUI 또는 API 호출을 통해 예측을 위한 iris 데이터 입력 수신
 - 모델을 기반으로 가정된 값을 예측하는 Flask API 포함
- templates 폴더
 - home.html : 사용자로부터 예측을 위한 데이터 입력 받는 html form
 - after.html : 입력 데이터를 기반으로 예측한 결과를 출력하기 위한 html 문서
- * 자세한 내용은 예제에 주석으로 설명

서버 가동 및 테스트



- app.py 실행 : 서버 가동된
\$ python3.8 app.py
- 웹 브라우저 실행, url 입력
http://127.0.0.1:5000/

기타 scikit-learn 학습모델 배포 방법

1. Django rest frame을 활용한 배포 참고

<https://www.django-rest-framework.org/tutorial/quickstart/>

<https://yunhookim.tistory.com/7> : rest-framework 참고

<https://honeyteacs.tistory.com/6> : scikit learn 배포 방법 참고

2. GCP를 활용한 배포 참고

<https://inahjeon.github.io/cloud-ml-engine/>

Python 프로그래밍으로 메일 보내기

Python 프로그래밍으로 메일 보내기

- gmail 기준 : python 코드 예제

SMTP : Simple Mail Transfer Protocol

import smtplib # 내장 라이브러리

from email.mime.text import MIMEText

smtp = smtplib.SMTP('smtp.gmail.com', 587)

smtp.ehlo() # say Hello

보안상의 이유로 SMTP연결을 TLS(전송 계층 보안) 모드로 설정

smtp.starttls() # TLS 사용시 필요

smtp.login('자신의ID@gmail.com', '자신의 gmail앱_비밀번호')

Python 프로그래밍으로 메일 보내기

```
# 보낼 메시지 작성, 본문, 제목, 수신자 mail 주소
msg = MIMEText( ' 본문 테스트 메시지 잘 갔으면 좋겠네. ' )
msg[ ' Subject ' ] = ' 메일 테스트 '
msg[ ' To ' ] = '수신ID@naver.com '

# smtp.sendmail( ' 송신자 ' , ' 수신자 ' , ' 메시지')
smtp.sendmail('송신자id@gmail.com ' , '수신자id@naver.com ' , msg.as_string())
smtp.quit() # smtp 종료
```

Gmail server 환경 설정 – IMAP 사용

The screenshot shows the Gmail web interface. A red box highlights the menu icon (three horizontal lines) and the Gmail logo in the top left. Another red box highlights the settings gear icon in the top right. A third red box highlights the '모든 설정 보기' (View all settings) button in the '빠른 설정' (Quick settings) panel on the right. The main email list is visible in the center, showing various emails from Google, Slack, and 3T Team.

Left Sidebar:

- 편지쓰기 (Compose)
- 받은편지함 (Inbox) 59
- 별표편지함 (Starred)
- 다시 알림 항목 (Unread)
- 보낸편지함 (Sent)
- 임시보관함 (Drafts)
- 더보기 (More)
- 행아웃 미팅 (Hangouts Meet)
- 새 회의 (New meeting)
- 회의 참여 (Join meeting)
- 행아웃 (Hangouts)
- 김 김 (Contact)

Top Bar:

- 메일 검색 (Search)
- 빠른 설정 (Quick settings)
- 설정 (Settings) - Highlighted with a red box

Main Email List:


기본	소셜
<input type="checkbox"/> ☆ Google	오후 5:...
<input type="checkbox"/> ☆ Google	오후 5:...
<input type="checkbox"/> ☆ IN ATELIER (Google .	11월 3
<input type="checkbox"/> ☆ Slack	11월 2
<input type="checkbox"/> ☆ 네이버	11월 1
<input type="checkbox"/> ☆ Google Cloud Platfo.	10월 30
<input type="checkbox"/> ☆ IN ATELIER (Google .	10월 27
<input type="checkbox"/> ☆ Slack 4	10월 22
<input type="checkbox"/> ☆ Slack 2	10월 18
<input type="checkbox"/> ☆ Google	10월 13
<input type="checkbox"/> ☆ Slack	10월 13
<input type="checkbox"/> ☆ Slack	10월 9
<input type="checkbox"/> ☆ 3T Team	10월 8

Right Panel (빠른 설정):

- 모든 설정 보기 (View all settings) - Highlighted with a red box
- 구성 (Compose)
- 기본값 (Default) - Selected
- 넓게 (Wide)
- 좁게 (Narrow)
- 테마 (Theme) - 모두 보기 (View all)
- 받은편지함 유형 (Inbox type)

Gmail server 환경 설정 - IMAP 사용

☰

 Gmail

✚

편지쓰기

받은편지함

57

별표편지함

다시 알림 항목

보낸편지함

임시보관함

더보기

행아웃 미팅

새 회의

회의 참여

행아웃

김 김

+

메일 검색

☰

?

⚙

설정

전달 및 POP/IMAP

기본설정

라벨

받은편지함

계정 및 가져오기

필터 및 차단된 주소

전달 및 POP/IMAP

부가기능

채팅 및 Meet

고급

오프라인

테마

POP 다운로드:

[자세히 알아보기](#)

1. 상태: POP 사용 안함

☐ 모든 메일에 POP 사용하기

☐ 지금부터 수신되는 메일에만 POP를 사용하기

2. POP로 메시지를 여는 경우

Gmail 사본을 받은편지함에 보관하기

3. 이메일 클라이언트 구성 (예: Outlook, Eudora, Netscape Mail)

[설정 방법](#)

IMAP 액세스:

(IMAP를 사용하여 다른 클라이언트에서 Gmail에 액세스)

[자세히 알아보기](#)

상태: IMAP를 사용할 수 있습니다.

☒ IMAP 사용

☐ IMAP 사용 안함

IMAP에서 메일을 삭제된 것으로 표시하는 경우:

☒ 자동 삭제 사용 - 서버를 즉시 업데이트(기본값)

☐ 자동 삭제 사용 안함 - 클라이언트가 서버를 업데이트할 때까지 대기

메일이 삭제된 것으로 표시되고 마지막으로 표시된 IMAP 폴더에서 삭제된 경우:

☐ 메일 보관(기본값)

☐ 메일을 휴지통으로 이동

☐ 메일을 즉시 완전삭제

폴더 크기 제한

☒ IMAP 폴더에서 메일 수를 제한하지 않습니다(기본값).


☐ 이만큼의 메일만 포함하도록 IMAP 폴더를 제한합니다. 1,000

변경사항
저장하기

Gmail server 환경 설정 – google 로그인 앱 비밀번호

- 다음 사이트에 접속하여 google 로그인 – 앱 비밀번호 발급 받기
- <https://myaccount.google.com/security>

Google에 로그인



비밀번호	최종 변경일: 3월 28일	>
2단계 인증	<input checked="" type="checkbox"/> 사용	>
앱 비밀번호	비밀번호 1개	>

Gmail server 환경 설정 – google 로그인 앱 비밀번호

Google 계정



← 앱 비밀번호

앱 비밀번호를 사용하면 2단계 인증을 지원하지 않는 기기의 앱에서 Google 계정에 로그인할 수 있습니다. 비밀번호를 한 번만 입력하면 기억할 필요가 없습니다. [자세히 알아보기](#)

앱 비밀번호

이름

생성됨

최종 사용일

Windows 컴퓨터의 메일

오후 5:11

-



앱 비밀번호를 생성할 앱 및 기기를 선택하세요.

메일



Windows 컴퓨터



생성

리눅스 crontab으로 실행 자동화하기

cron과 crontab 이란?

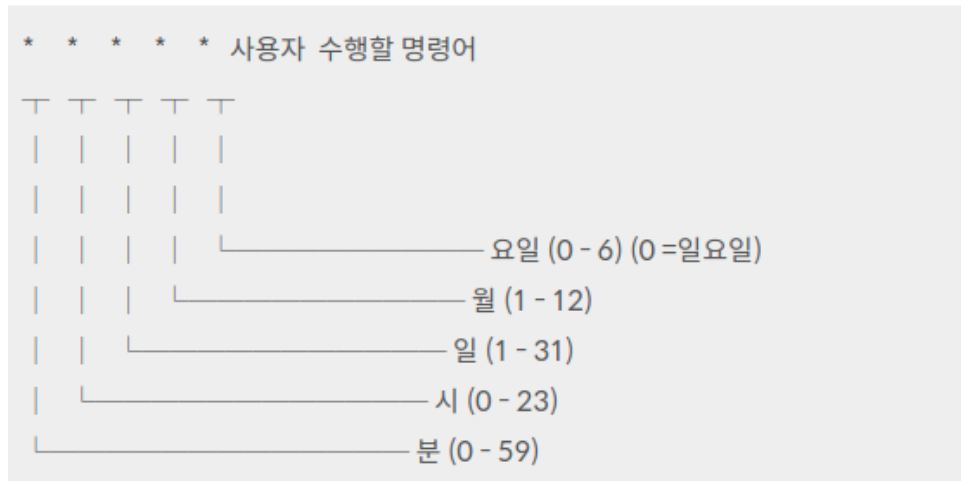
- cron : 유닉스/리눅스 계열에서 특정 시간에 특정 작업을 하도록 설정할 수 있는 데몬
- crontab : cron 데몬이 주기적인 명령을 수행하도록 명령 리스트를 정의하는 곳
 - 작업 스케줄 자동화 기능(서버는 24시간 가동됨)
- 적용 사례
 - 새벽 3시에 DB 백업 받기
 - 8시에 서버의 사용량을 리포트 하기
 - 매 시간 정각에 프로그램 실행 결과 메일로 보내기

crontab(cron table) 설정방법

- crontab 작성하기 (root 또는 sudo 권한 있는 계정만 가능)

\$ crontab -e

- 편집기 선택



- 작업 스케줄링 방법
 - 매시 20분마다 실행, log 저장

```
20 * * * * /home/rapa00/aiml/VENV/bin/python3.8 /home/rapa00/aiml/pysmtp_gmail.py > > /home/cozlab/django/aimodels/logs/log
```

cron 데몬 관련 명령어

- crontab 관련 명령

\$ crontab -e : 명령 등록, 편집

\$ crontab -d : 등록된 명령 삭제

\$ crontab -l : 현재 등록된 리스트 출력

\$ crontab -r : 현재 사용자가 등록한 crontab 전체 삭제

- crontab 변경 내용 적용

\$ service cron start : 시작

\$ service cron restart : 재시작

\$ service cron stop : 중단

cron 데몬 관련 명령어



- crontab 실행 log 확인

```
$ tail -f check_test
```

- 실행 로그 조회

```
$ grep CRON /var/log/syslog
```


Intranet에서 ssh 서버 구축 팀별로 필요시 구축하기

- server side 설정
- client side 설정

ssh 클라이언트 서버 환경 구축 – Server side

- ssh 서버 설치

```
$ sudo apt-get install openssh-server
```

- 서비스 실행

```
$ sudo systemctl start ssh    또는  sudo service ssh start
```

- 서비스 종료

```
$ sudo service ssh stop
```

ssh 클라이언트 서버 환경 구축 – Server side

- `sudo netstat -tnlp` (ssh port 22번)

```
himedia@himedia:~$ netstat -tnlp
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:3306          0.0.0.0:*                LISTEN      -
tcp        0      0 127.0.0.53:53          0.0.0.0:*                LISTEN      -
tcp        0      0 0.0.0.0:22             0.0.0.0:*                LISTEN      -
tcp        0      0 127.0.0.1:631          0.0.0.0:*                LISTEN      -
tcp6       0      0 :::22                  :::*                    LISTEN      -
tcp6       0      0 :::1:631                :::*                    LISTEN      -
himedia@himedia:~$
```

ssh 클라이언트 서버 환경 구축 – Server side

- ssh 서버 config 수정 / 저장
 - 리모트 클라이언트에서 접속 가능하도록

```
$ sudo nano /etc/ssh/sshd_config
```

```
#PermitRootLogin prohibit-password
```

```
#PasswordAuthentication yes
```

위의 항목들을 아래와 같이 수정합니다.

```
PermitRootLogin yes          # 주석해제 및 옵션 값
```

```
PasswordAuthentication yes   # 주석해제
```

- ssh server 재시작

```
$ sudo systemctl restart ssh
```

ssh 클라이언트 서버 환경 구축 – Server side



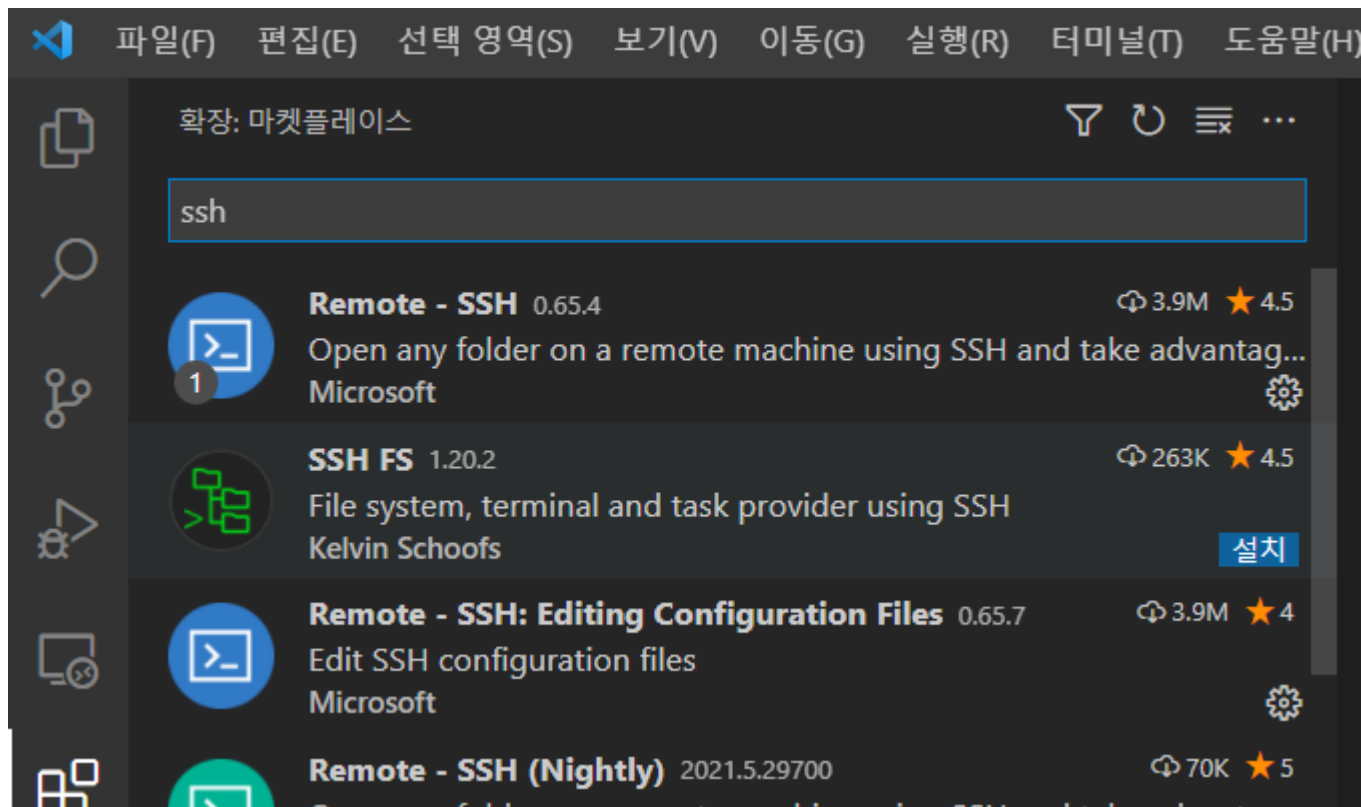
- 서버 방화벽 update

```
$ sudo ufw allow OpenSSH
```

```
$ sudo ufw enable
```

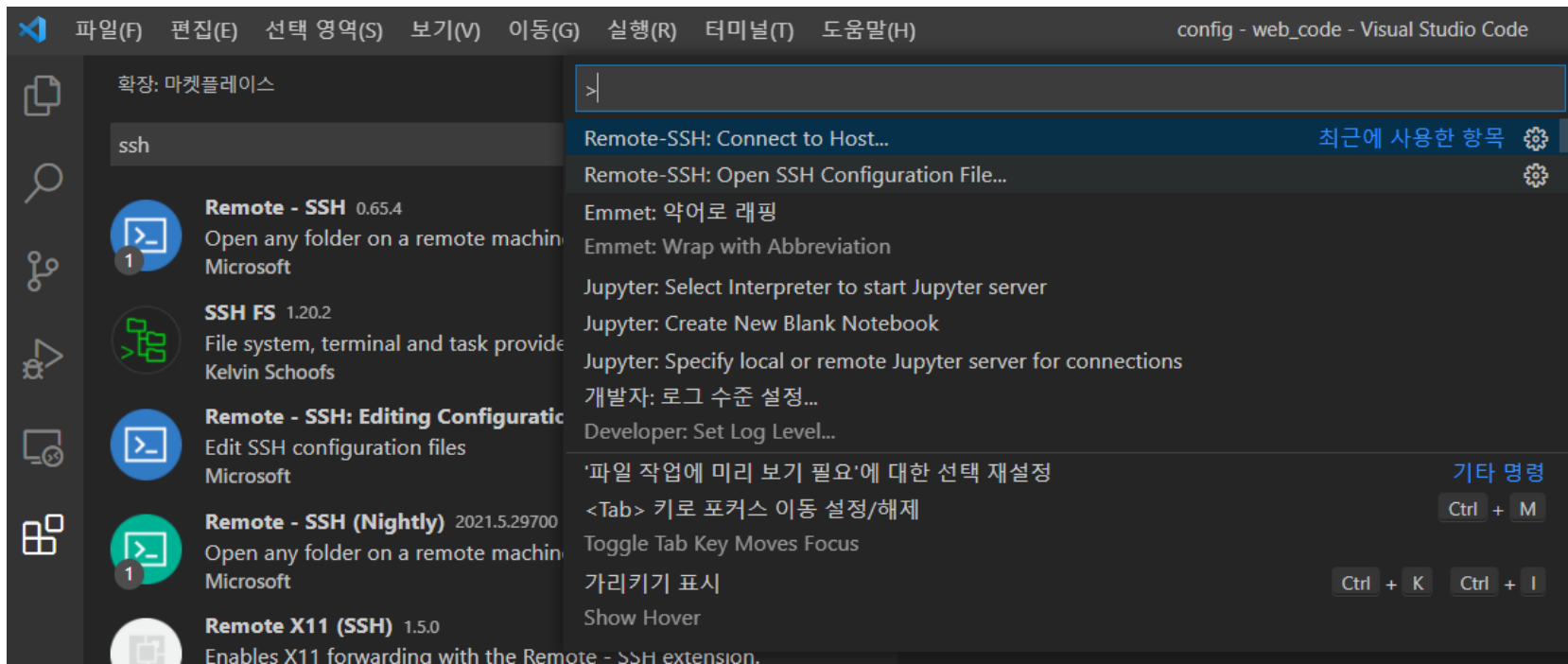
```
$ sudo ufw status
```

ssh 클라이언트 서버 환경 구축 – Client side



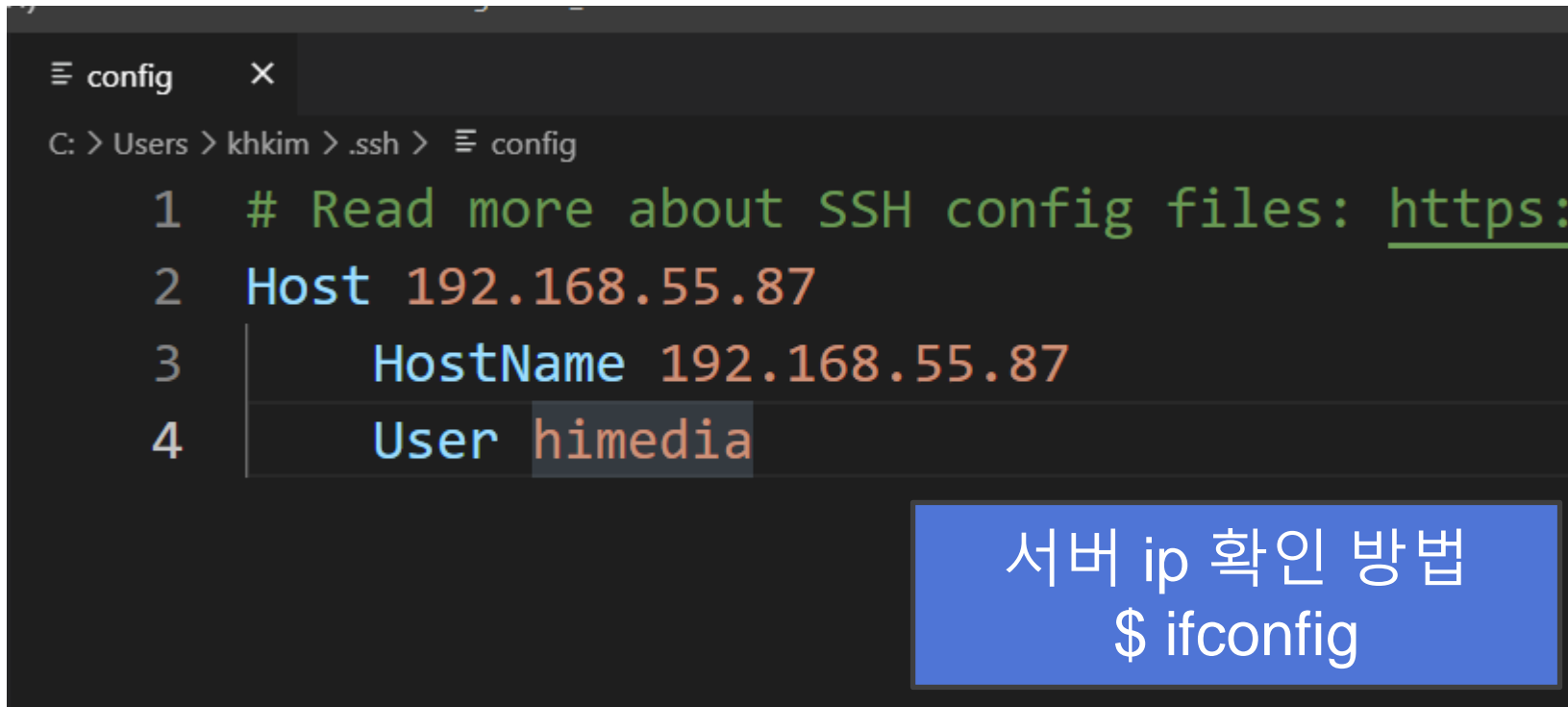
ssh 클라이언트 서버 환경 구축 – Client side

- remote-SSH: Open SSH Configuration File



ssh 클라이언트 서버 환경 구축 – Client side

- ~/.ssh/config 수정



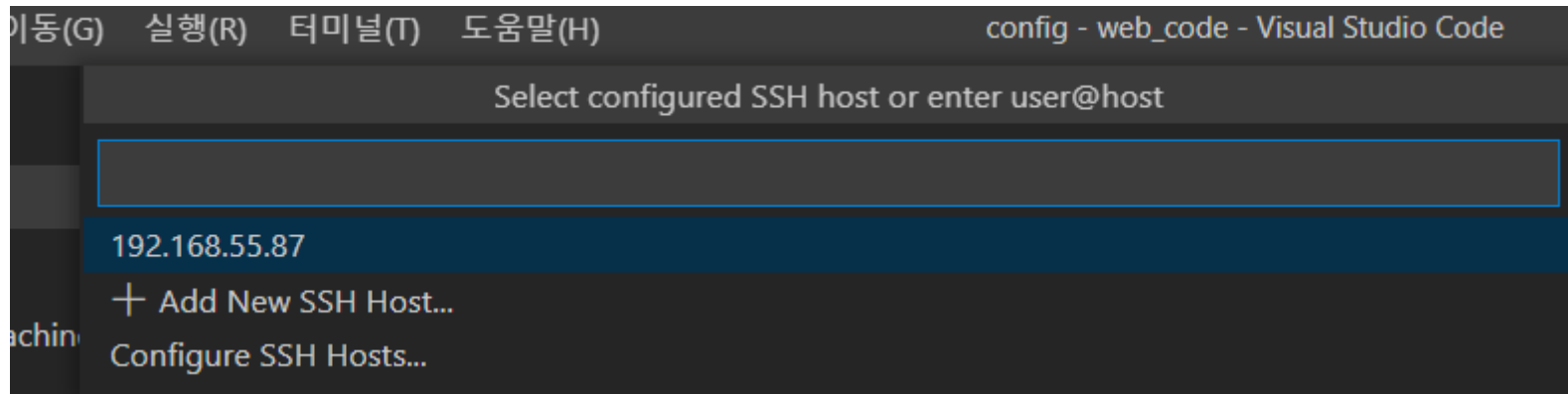
```
config X
C: > Users > khkim > .ssh > config

1 # Read more about SSH config files: https://www.ssh.com/academy/ssh/config
2 Host 192.168.55.87
3     HostName 192.168.55.87
4     User himedia
```

서버 ip 확인 방법
\$ ifconfig

ssh 클라이언트 서버 환경 구축 – Client side

- ssh 서버에 접속하기
- F1 키 누름
- remote ssh: connect to Host -> 서버 선택
- linux -> pw 입력



ssh 클라이언트 서버 환경 구축 – Client side



- CLI 터미널에서 접속하기

```
$ ssh -p 22 -v rapa00@192.168.55.91
```