**웹개발 종합반 78기(튜터: 이범규)**

**수업노트 5주차 (22/05/08)**

5주차의 수업이 시작되었다.

FileZilla, 가비아

**AWS 서버 구매하기**

EC2 콘솔페이지 접속 (<https://ap-northeast-2.console.aws.amazon.com/ec2/v2/home?region=ap-northeast-2>)

AWS로그인 ([aaa@gmail.com](mailto:aaa@gmail.com))

인스턴스 구매

: 인스턴스>인스턴스 시작(버튼) 선택>Amazon Machine Image(AMI) 선택

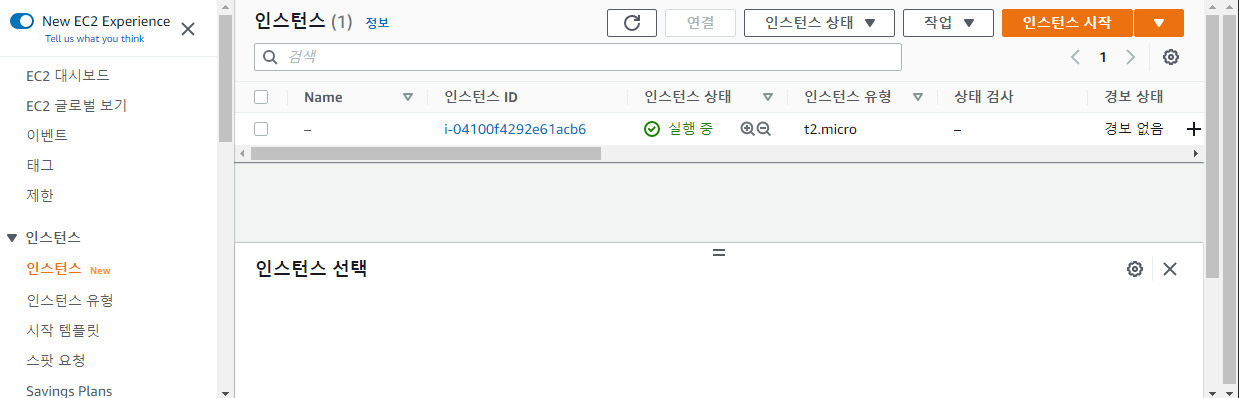
**Ubuntu Server 20.04 LTS (HVM), SSD Volume Type(프리티어, 1년 무료) 선택**

키페어 선택(새로운 키 페어)>다운로드(sparta\_mykey)

(요주의) 인스턴스 중지 vs 인스턴스 종료

프리티어는 가동중 서버를 1대만 1년간 사용이 가능하다

따라서, 인스턴스를 잘못 생성한 경우, 반드시 **인스턴스 종료(인스턴스 반납)** 후 새로 인스턴스를 생성하여야 한다. **인스턴스 중지**하는 경우 2개의 인스턴스를 이용하게 되므로 **비용이 부과된다**는 점에 주의할 것!!!



**Git Bash 앱 구동**

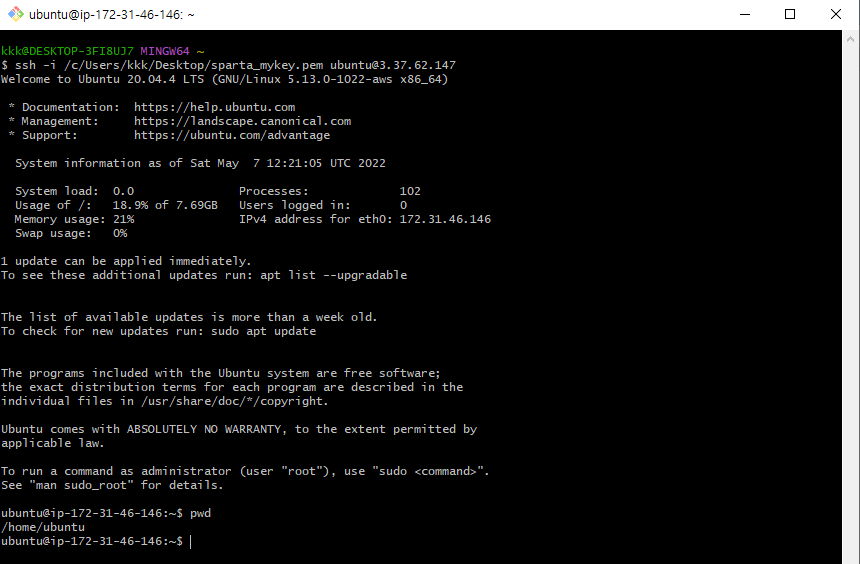
Git Bash $>프롬프트하에서

ssh -i **받은키페어를끌어다놓기** ubuntu@AWS에적힌내아이피(공용 IP주소)

ex)ssh -i /path/my-key-pair.pem ubuntu@ip[ address

**\*\*보안 주의**

|  |
| --- |
| **ssh -i /c/Users/kkk/Desktop/ 147** |



**서버 세팅하기**

서버에 설치 후 filizilla를 이용하여 파이썬 파일을 업로드하고, 실행해 보기

>메모장 구동

>EC한방에 세팅하기 복붙

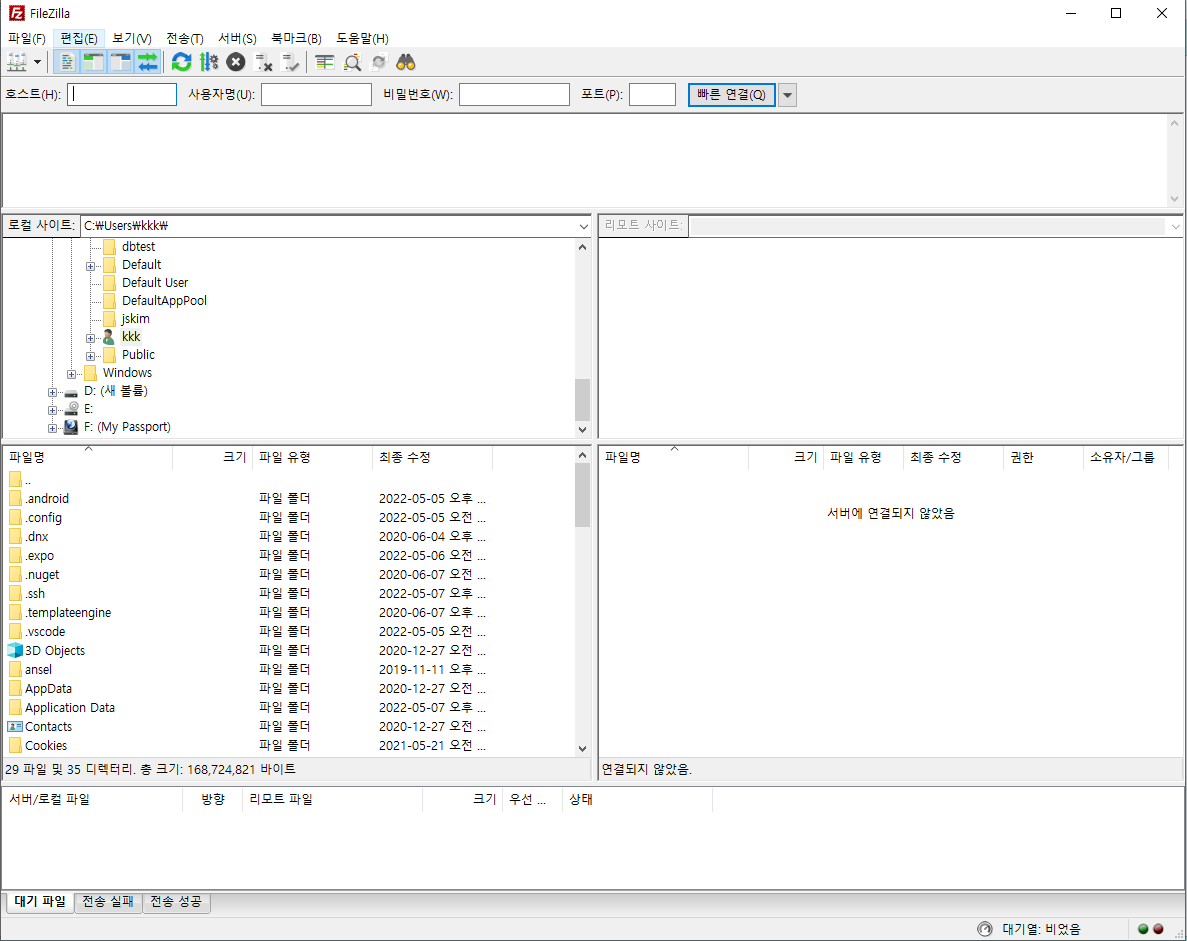
|  |
| --- |
| # python3 -> python  sudo update-alternatives --install /usr/bin/python python /usr/bin/python3 10  # pip3 -> pip  sudo apt-get update  sudo apt-get install -y python3-pip  sudo update-alternatives --install /usr/bin/pip pip /usr/bin/pip3 1  # port forwarding  sudo iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j REDIRECT --to-port 5000 |

하나씩 차례대로 Git Bash창에 복붙하며 실행한다

|  |
| --- |
| 1. 실행  sudo update-alternatives --install /usr/bin/python python /usr/bin/python3 10 |
| <Git Bash 터미널 실행 결과>  ubuntu@ip-xxx:~$ sudo update-alternatives --install /usr/bin/python python  update-alternatives: using /usr/bin/python3 to provide /usr/bin/python (python) in a  ubuntu@ip-xxx:~$ |
| 2. 실행  sudo apt-get update |
| <Git Bash 터미널 실행 결과> (한참 실행한 후 다음 결과 나타남)  Get:34 http://security.ubuntu.com/ubuntu focal-security/restricted Translation-en [126 kB]  Get:35 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [702 kB]  Get:36 http://security.ubuntu.com/ubuntu focal-security/universe Translation-en [125 kB]  Get:37 http://security.ubuntu.com/ubuntu focal-security/universe amd64 c-n-f Metadata [14.4 kB]  Get:38 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 Packages [20.7 kB]  Get:39 http://security.ubuntu.com/ubuntu focal-security/multiverse Translation-en [5196 B]  Get:40 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 c-n-f Metadata [500 B]  Fetched 22.6 MB in 9s (2629 kB/s)  Reading package lists... Done  ubuntu@ip-xxx:~$ |
| 3. 실행  sudo apt-get install -y python3-pip |
| <Git Bash 터미널 실행 결과> (한참 실행한 후 다음 결과 나타남)  Setting up gcc-9 (9.4.0-1ubuntu1~20.04.1) ...  Setting up libpython3-dev:amd64 (3.8.2-0ubuntu2) ...  Setting up libstdc++-9-dev:amd64 (9.4.0-1ubuntu1~20.04.1) ...  Setting up gcc (4:9.3.0-1ubuntu2) ...  Setting up g++-9 (9.4.0-1ubuntu1~20.04.1) ...  Setting up python3.8-dev (3.8.10-0ubuntu1~20.04.4) ...  Setting up g++ (4:9.3.0-1ubuntu2) ...  update-alternatives: using /usr/bin/g++ to provide /usr/bin/c++ (c++) in auto mode  Setting up build-essential (12.8ubuntu1.1) ...  Setting up python3-dev (3.8.2-0ubuntu2) ...  Processing triggers for man-db (2.9.1-1) ...  Processing triggers for libc-bin (2.31-0ubuntu9.7) ...  ubuntu@ip-xxx:~$ |
| 4. 실행  sudo update-alternatives --install /usr/bin/pip pip /usr/bin/pip3 1 |
| <Git Bash 터미널 실행 결과>(바로 나타남)  ubuntu@ip-xxx:~$ sudo update-alternatives --install /usr/bin/pip pip /usr/bin/pip3 1  update-alternatives: warning: forcing reinstallation of alternative /usr/bin/pip3 because link group pip is broken  update-alternatives: warning: not replacing /usr/bin/pip with a link  ubuntu@ip-xxx:~$ |
| 5. 실행  sudo iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j REDIRECT --to-port 5000 |
| <Git Bash 터미널 실행 결과> (바로 나타남)  ubuntu@ip-xxx:~$ sudo iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j REDIRECT --to-port 5000  ubuntu@ip-xxx:~$ |

Filezilla 작업

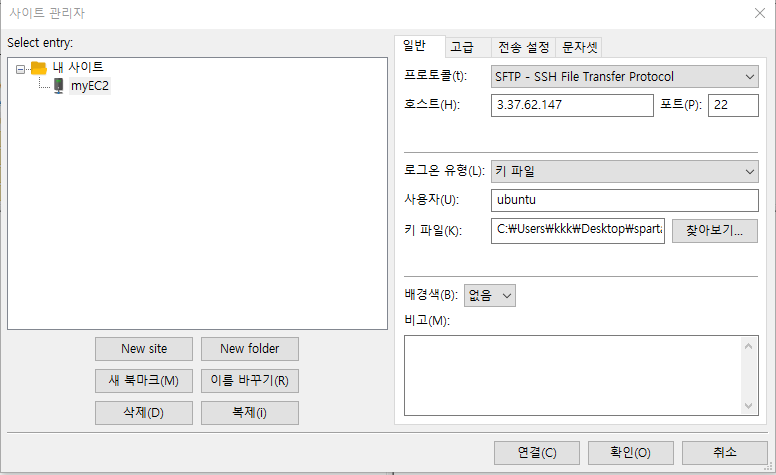
>Filezilla 구동



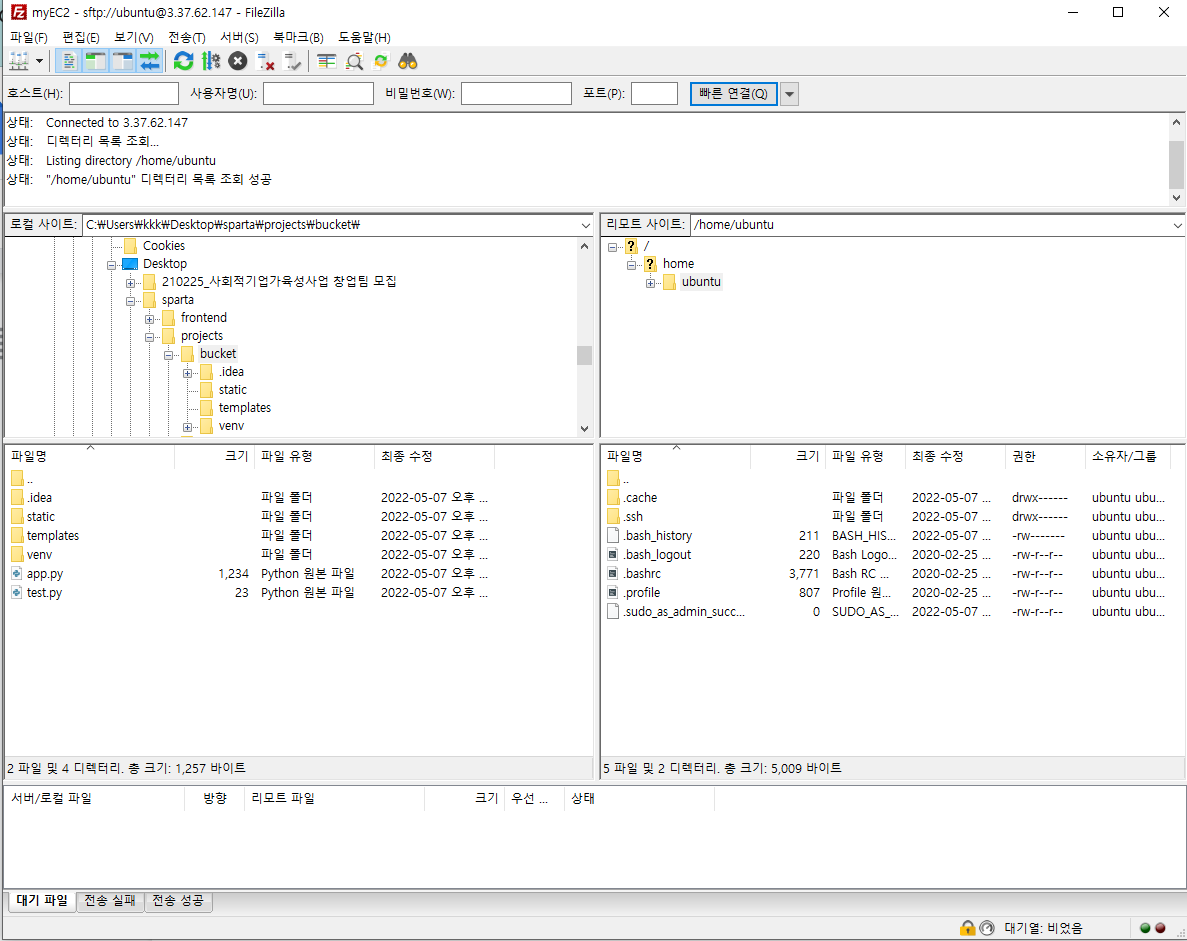
>왼쪽이 로컬, 오른쪽이 원격(서버)

>왼쪽 폴더를 >바탕화면>sparta>projects 폴더로 이동

>FTP사이트 등록



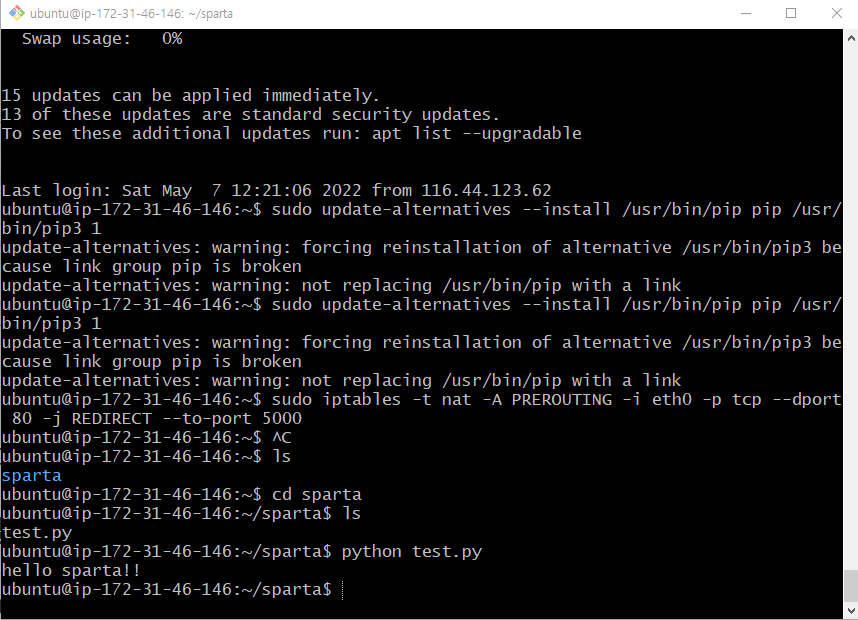
>FTP 접속



>원격 폴더. sparta로 이동 후 test.py 이동(복사) → 복사됨

**test.py 실행 테스트해보기**

>Git Bash 터미널 실행



결과인 hello sparta!! 정상 출력됨. 테스트 성공

**Flask 서버를 실행해 보기**

>homework 폴더의 app.py, templates, static 폴더를 원격 /sparta/ 폴더로 복사

**Flask 설치**

>pip install flask

Successfully installed Jinja2-3.1.2 MarkupSafe-2.1.1 Werkzeug-2.1.2 click-8.1.3 flask-2.1.2 importlib-metadata-4.11.3 itsdangerous-2.1.2

ubuntu@ip-172-31-46-146:~/sparta$

**pymongo 설치**

>pip install pymongo

ubuntu@ip-172-31-46-146:~/sparta$ pip install pymongo

Collecting pymongo

Downloading pymongo-4.1.1-cp38-cp38-manylinux2014\_x86\_64.whl (498 kB)

|████████████████████████████████| 498 kB 16.1 MB/s

Installing collected packages: pymongo

Successfully installed pymongo-4.1.1

ubuntu@ip-172-31-46-146:~/sparta$

**dnspython 설치**

>pip install dnspython

ubuntu@ip-172-31-46-146:~/sparta$ pip install dnspython

Collecting dnspython

Downloading dnspython-2.2.1-py3-none-any.whl (269 kB)

|████████████████████████████████| 269 kB 16.2 MB/s

Installing collected packages: dnspython

Successfully installed dnspython-2.2.1

ubuntu@ip-172-31-46-146:~/sparta$

**python app.py 실행(정상동작중)**

ubuntu@ip-172-31-46-146:~/sparta$ python app.py

\* Serving Flask app 'app' (lazy loading)

\* Environment: production

WARNING: This is a development server. Do not use it in a production deployment.

Use a production WSGI server instead.

\* Debug mode: on

\* Running on all addresses (0.0.0.0)

WARNING: This is a development server. Do not use it in a production deployment.

\* Running on http://127.0.0.1:5000

\* Running on http://172.31.46.146:5000 (Press CTRL+C to quit)

\* Restarting with stat

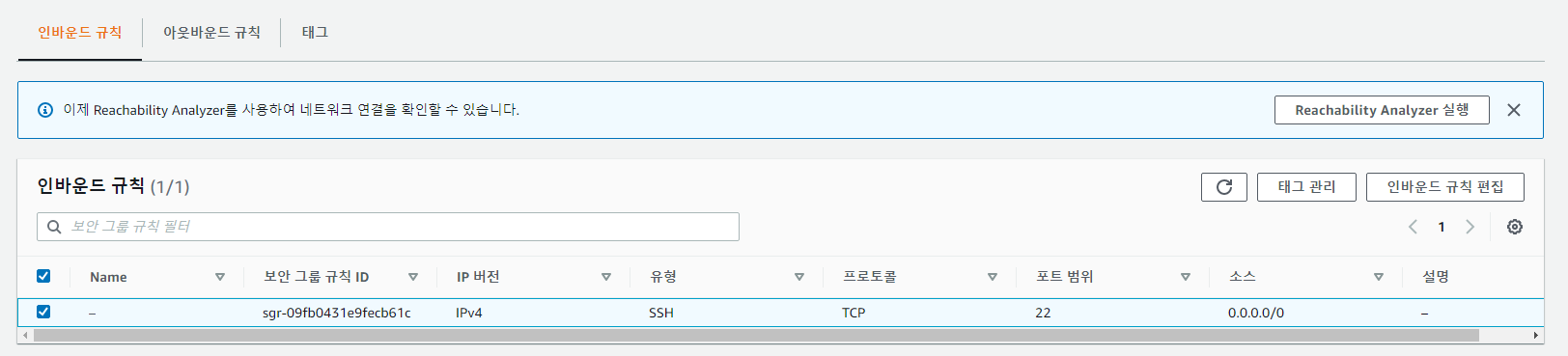
\* Debugger is active!

\* Debugger PIN: 507-492-971

5000 port

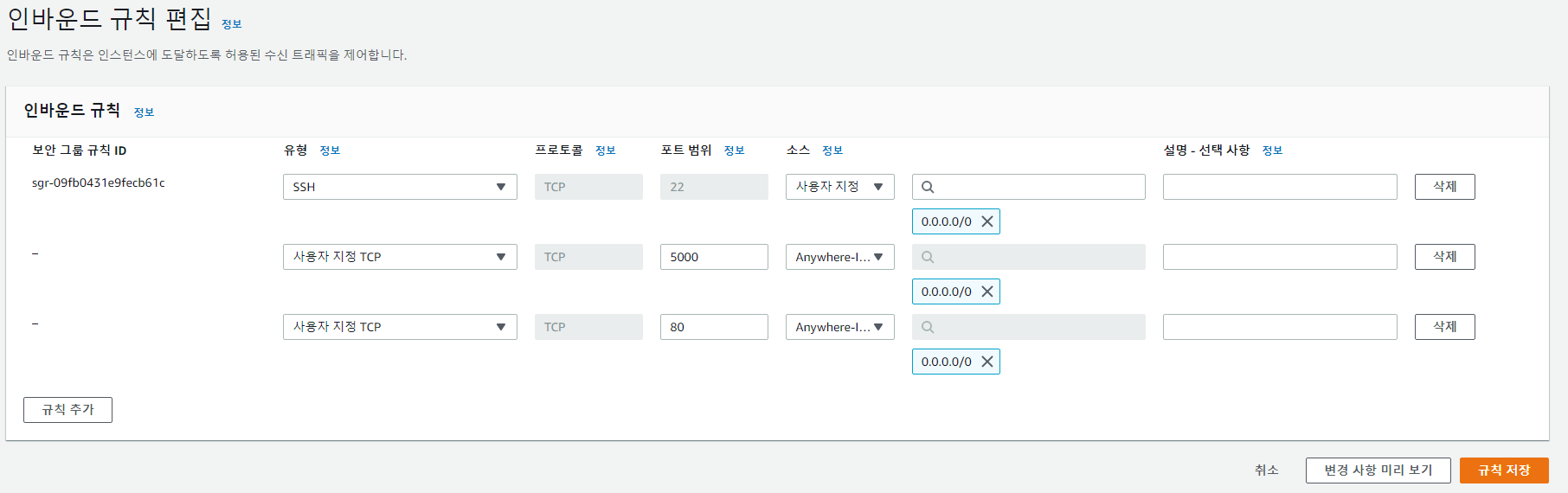
**EC2서버 port 5000 Open하기**

> 서버>보안 Tab>보안그룹 선택

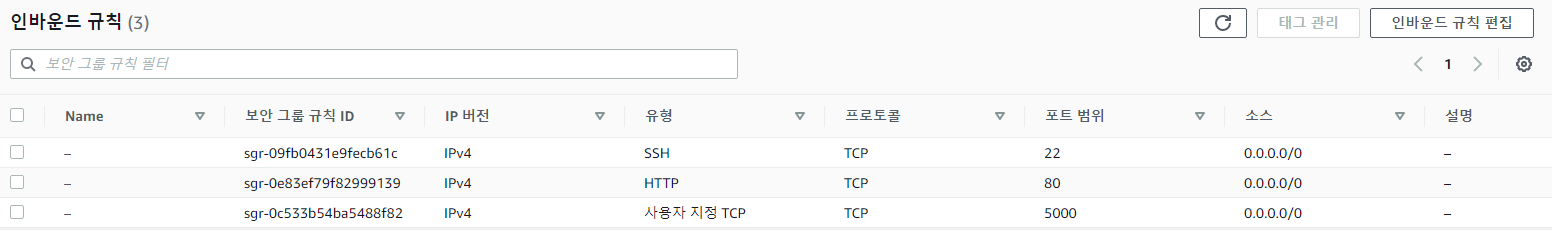


>인바운드 규칙>인바운드 규칙 편집 버튼 선택

>5000, 80 포트 규칙 추가

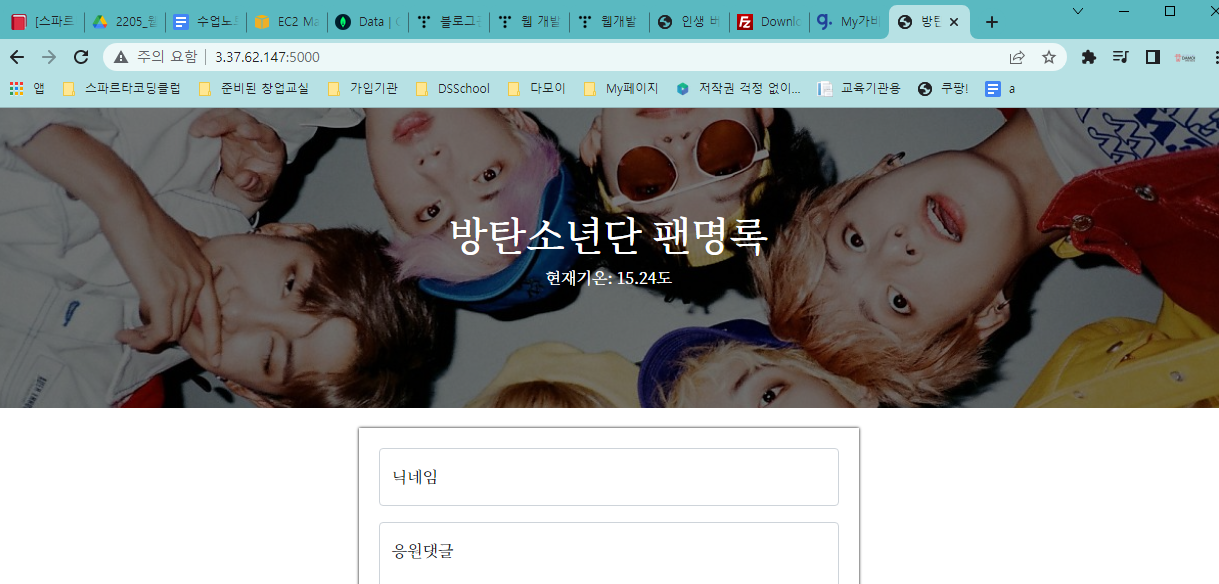


>인바운드 규칙 확인



**웹페이지 접속 재확인**

>웹브라우저로 해당 url 접속⇒정상 동작 확인

****

\* http 80 포트 접속시 http 5000 port로 포워딩시켰음

⇒(sudo iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j REDIRECT --to-port 5000)

**nohup 설정하기**

**nohub 실행**

(터미널을 종료해도 계속 실행하는 cmd)

>nohup python app.py &

>실행 후 터미널 강제 종료해도 웹 페이지 접속이 가능함

**강제종료하기**

**nohup로 기동중인 python 프로세스를 강제 종료**

>ps -ef | grep 'python app.py' | awk '{print $2}' | xargs kill

ubuntu@ip-172-31-46-146:~$ ps -ef | grep 'python app.py' | awk '{print $2}' | xargs kill

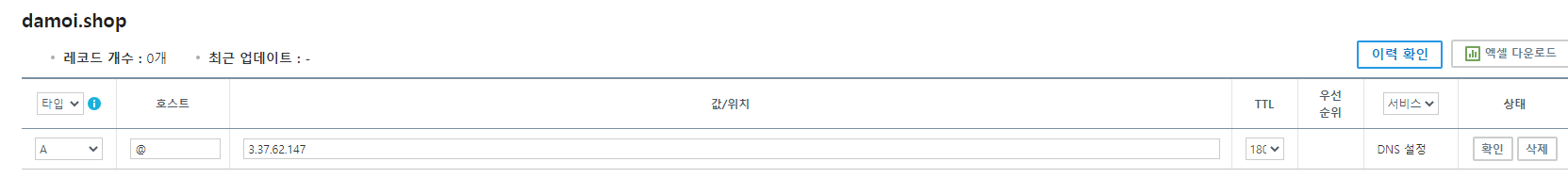
kill: (19192): No such process

ubuntu@ip-172-31-46-146:~$

도메인 연결하기

>가비아 접속

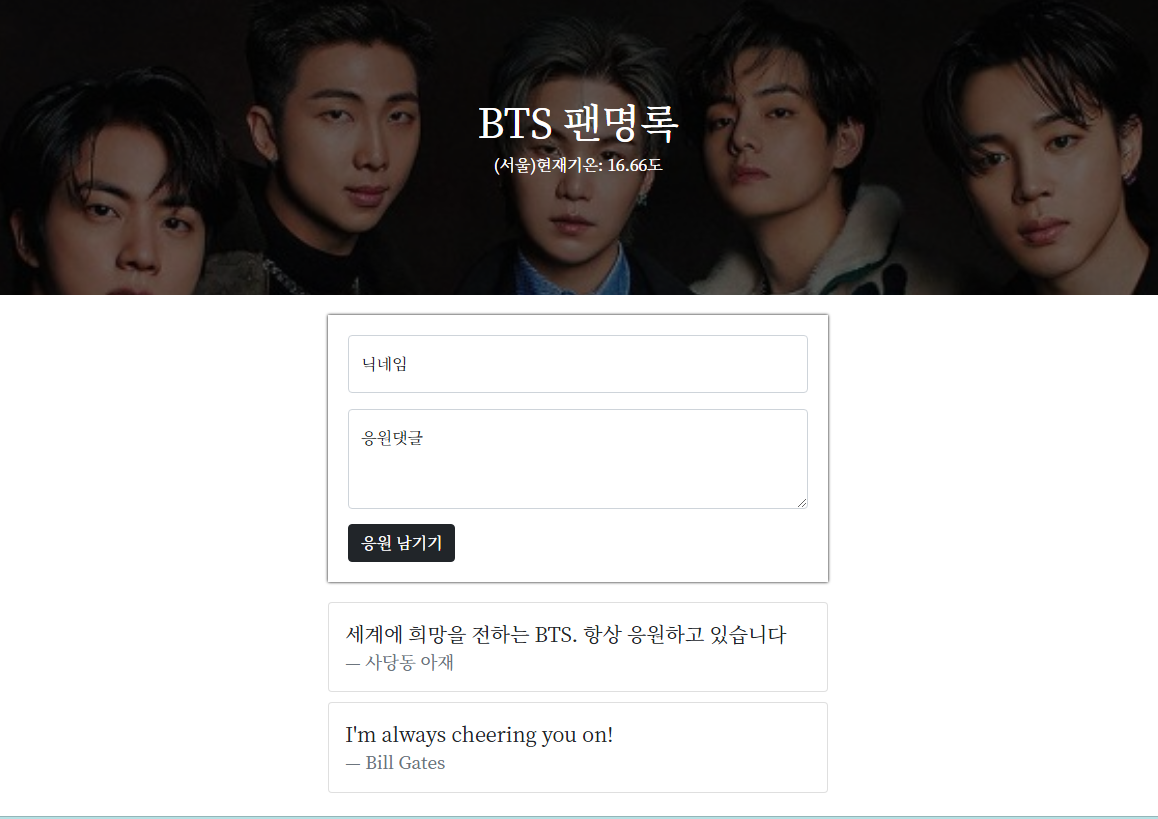
>DNS설정



>확인 후 저장

>도메인 테스트=>OK

>스마트폰 접속 테스트 => OK



**숙제: 팬명록 도메인 연결, 완성하기**

이제 마지막 숙제만 제출하면 완강이다.

**데이터가 안나온다?**

작전대로라면 방명록이 잘 나타나야 하는데, 왠걸? 하단 목록이 안나타난다.

로컬에서 해 본다. 잘 나온다. 그럼 환경문제가 확실하다.

분명 나같은 사람이 있었을 거야 하며 slack FAQ을 훑어보았다.

찾았다!

로컬에서 mongoDB가 잘 보이는데 AWS에서 잘 보이지 않는다는…

* AWS에 MongoDB용 포트 추가
* MongoDB에 접근 허용 IP 추가

지시하는 대로 2개의 문제를 차례로 처리하고 접속한 후 MongoDB 접속 테스트를 해 본다.

mongoDB 접속 페이지가 정상적으로 나타나지 않는다.

또 다른 삽질 시작!

AWS 인스턴스 재시작 후 접속해 보니 이번엔 또 웹페이지 접속이 되지 않는다.

80port, 5000 port 모두 안된다.

혹시?

다시 터미널 접속하여

* ip 포워딩 실행
* nuhup 실행

작업 후 접속해 보니 이젠 접속이 된다.

아하! 서버 재시작시엔 이 작업을 매번 반복해 주어야 하는구나…

**정렬 순서**

글을 등록해 보니 DB에 저장도 잘 되고 정상 작동이 된다.

근데 글이 밑으로 계속 추가된다.

그렇지 정렬 문제구나. 이건 간단한 거니 한번 해 볼까? 덤벼 보지 뭐! 구글링 시작!

mongoDB 레퍼런스를 뒤지며 이것 저것 해 본다.

|  |
| --- |
| old>  homework\_list = list(db.homework.find({}, {'\_id':False})) |
| New>  homework\_list = list(db.homework.find({},{'\_id':False}).sort({'\_id':-1})) |

“근데 안 된다. 삽질을 얼마나 더 해야 하나?

아니다. 이 정도로 그만하자. MongoDB 과정이 아니잖아?”

애써 창피한 변명으로 위안을 하며, 숙제를 종료하기로 결정했다.

slack에 질문을 남겨 둔 채… 이건 실전에서 분명 찾아낼 거야! 하며 마지막 숙제를 종료한다.

**5주차 수강 후기**

개발의 꽃, 배포단계!

클라우드 MongoDB와 클라우드 AWS 웹서버에 배포를 마지막으로 모든 과정을 마무리하는 단계다.

마지막으로 GET, POST를 한번 더 다시 복습하며 새로운 시각을 제시하고자 하는, 마지막 수업까지도 하나라도 더 가르치려는 튜터님의 흔적이 보인다.

배포하는 과정에서 AWS와 MongoDB 등 환경설정 문제에 부딪힐 수 밖에 없었다.

실전 환경에서 부딪히게 될 문제들을 마지막으로 제시하고 해결하기를 강요(?)하며 5주차 과정을 마무리하고 있다. 그리고 결국에는 해 냈다. 수고했다.

그리고 지금까지 달려오게 애써주신 튜터님께, 그리고 스파르타코딩클럽 모든 분들께 감사드리고 싶다.

감사합니다.

이 땅에 남아 있던 고집 센 꼰대 노땅 개발자 한 사람의 눈을 뜨게 해 주셔서 진심으로 감사합니다.