스파르타코딩클럽 8기 7주차

🔥 **매 주차 강의자료 시작에 PDF파일과 영상 링크를 올려두었어요!**

▼ PDF 강의자료 다운받기

> https://s3-us-west-2.amazonaws.com/secure.notion-static.com/9c5b031c-f784-4b8f-8745-fa2749928781/scc-8-1.pdf

▼ 영상강의 참고하기

- 아직 제공되지 않습니다!

[수업 목표]

1. 리눅스에 mongoDB를 설치할 수 있다.
2. EC2 서버에서 1~5주차 완성본을 실행할 수 있다.
3. SSH 접속을 끊어도 서버가 계속 돌게 한다.
4. URL뒤에 붙는 포트 번호(5000)을 없앨 수 있다.

전반 3시간

[시작] : 체크인 & 출석체크

🔥 **튜터님은 체크인과 함께 출석 체크(링크)를 진행해주세요!**

▼ "15초 체크인"을 진행합니다.

- 튜터님은 타이머를 띄워주세요! (링크)
- 본인의 감정상태와 오늘 있었던 소소한 일을 공유하는 시간입니다.

[2시간]: 수업

▼ 1) EC2에 접속하기 (이제 익숙하시죠?)

```
ssh -i 받은키페어를끌어다놓기 ubuntu@AWS에적힌내아이피
```

▼ 2) 1~5주차 완성본을 리눅스에서 실행하기 전에, mongoDB를 설치합니다.

> 👉 mongoDB를 설치하고,
> → 실행해보고,
> → 계정(접속 아이디/비밀번호)을 만들고,
> → 외부에 열어놓는 것까지 해볼게요!

▼ 1. mongoDB 설치 코드

👉 mongoDB 공식자료를 참고해서 정리했습니다. (링크)
한 줄 한 줄이 무슨 뜻인지는 굳이 알 필요 없어요!

*** 한 줄 씩 복사해서 붙여 넣어주세요**

```
wget -qO - https://www.mongodb.org/static/pgp/server-4.2.asc | sudo apt-key add -

echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu bionic/mongodb-org/4.2 multiverse" | sudo tee /etc/a

sudo apt-get update

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

▼ 2. mongoDB 실행하기

```
# 실행. 아무 반응이 없으면, 잘 실행된 것!
# 리눅스는 보통 잘 되면 아무것도 안나와요!^^;
sudo service mongod start

# 아래 명령어를 입력해서 127.0.0.1:27017 이 있는지 확인하기
# 있다면 mongoDB가 정상 작동 중이라는 뜻입니다.
netstat -tnlp
```

▼ 3. mongoDB 접속 계정 생성하기

👉 우리가 만든 mongoDB를 외부에 열어주기 전에, 접속에 필요한 아이디와 비밀번호를 세팅해봅시다! (설정 안하면 누구나 DB정보를 볼 수 있다는..!)

```
mongo
```

👉 좌측에 '>' 표시가 나오면 성공적으로 MongoDB에 접속한 것입니다! 다음 명령어를 순차적으로 입력해주세요.

눈치 채셨겠지만,
test, test 자리에 내가 넣고 싶은 아이디/비밀번호를 넣으면 됩니다. (영어로..!)

```
# admin으로 계정 바꾸기
use admin;

# 계정 생성하기
db.createUser({user: "test", pwd: "test", roles:["root"]});
```

👉 아래와 같은 화면을 보면 완성!

```
> use admin;
switched to db admin
> db.createUser({user: "test", pwd: "test", roles:["root"]});
Successfully added user: { "user" : "test", "roles" : [ "root" ] }
```

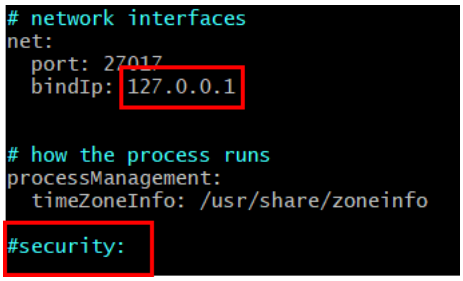
```
# 나오기
exit

# MongoDB 재시작
sudo service mongod restart
```

▼ 4. mongoDB를 외부에 열어주기

> 👉 mongoDB는 디폴트로 내부에서만 접속을 허용하고 있습니다. 이 작업은 외부에서 접근이 가능하도록 잠금을 풀어 주는 것입니다.
>
> 리눅스 자체 에디터(고급 메모장 정도로 생각!)인 Vim이 등장하는데요,
> 조금 생소할 수 있으니, 튜터님과 함께 해보아요!

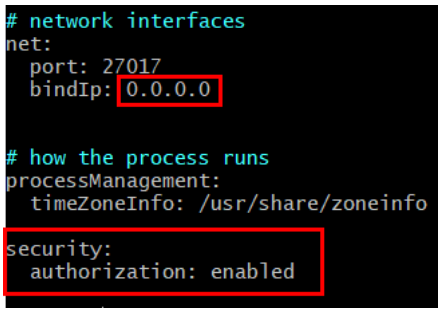
```
sudo vi /etc/mongod.conf

# sudo: 관리자(SuperUser) 권한으로 다음을 실행
# vi: 리눅스에서 VS Code 대신 사용하는 텍스트에디터인 Vim을 실행
# => "관리자 권한으로 /etc 폴더 아래 mongod.conf 파일을 Vim으로 켜줘!"라는 뜻입니다
```

위 명령어를 실행하신 후, 아래 방향 화살 키를 누르시면 다음과 같은 내용이 보입니다.




```
# 입력 모드 전환
i
```

위 붉은 박스의 내용을 아래와 같이 바꿔주세요!

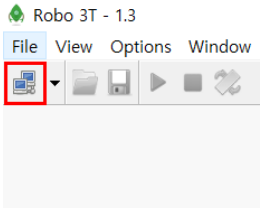



```
# 내용 저장하고 에디터 종료하기. esc 누르고 다음 입력.
:wq

# 재시작
sudo service mongod restart

# 아래 명령어를 입력하여 0.0.0.0:27017 이 목록에 있음을 확인합니다.
# 만약 127.0.0.1:27017 이 있다면 설정 파일이 제대로 수정되지 않은 것입니다.
netstat -tnlp
```

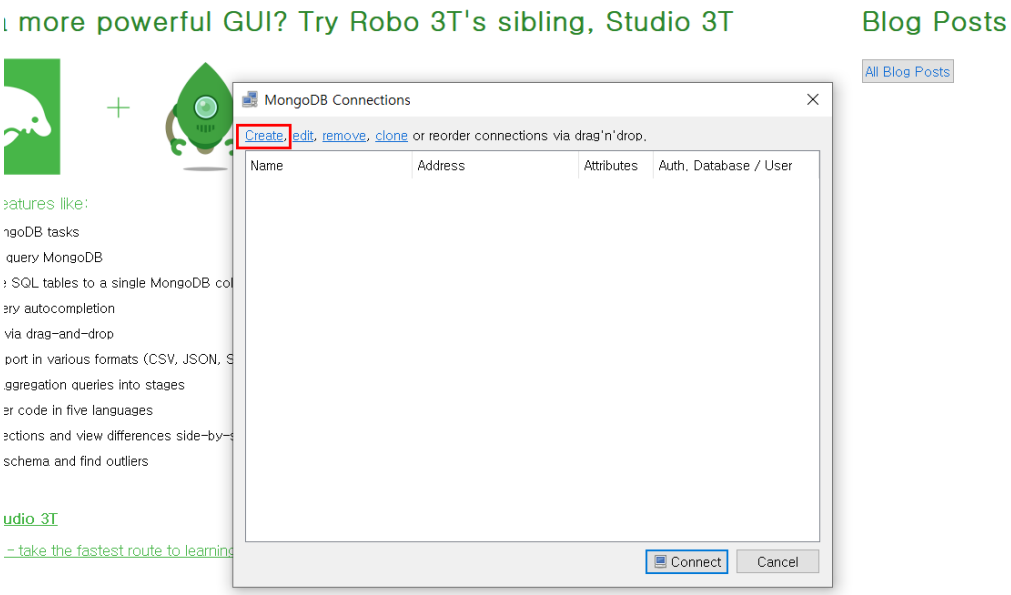
▼ 5. Robo3T를 이용해서, "내 컴퓨터에서"→"서버에 있는 mongoDB"에 접속하기

▼ (1) id/pw를 넣고 접속해보기

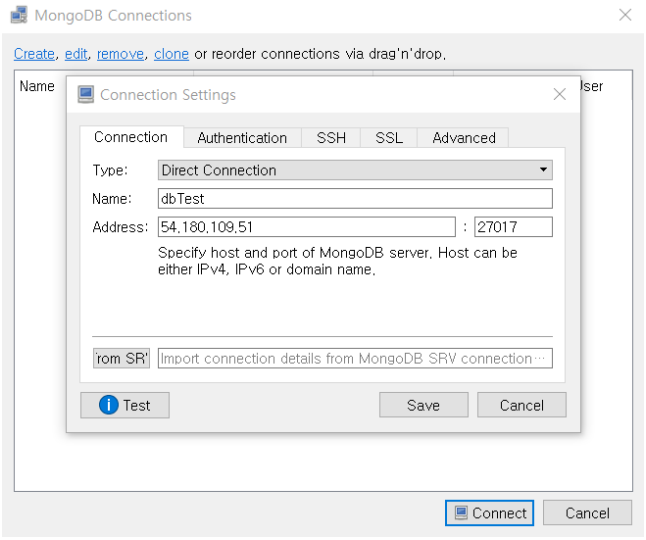
- 좌측 상단 빨간 상자 내 아이콘을 클릭합니다.



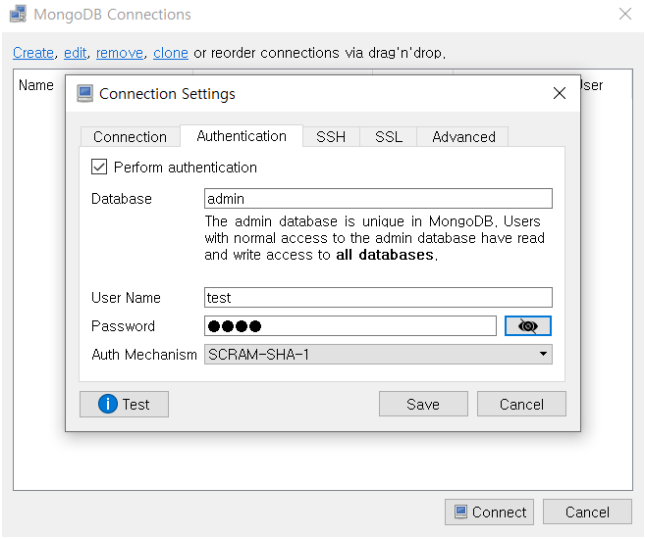
- Create 클릭!



- 접속 정보를 세팅합니다.



- 상단 Authentication 탭을 클릭합니다.
  1. Perform authentication 체크박스를 클릭합니다.
  2. 생성한 계정의 아이디와 비밀번호를 입력하고, 'save'를 클릭합니다.



▼ (2) 만약에 안된다면 다음 사항을 확인해보세요.

1. AWS에 접속합니다.

```
ssh -i 키페어 ubuntu@내AWS아이피
```

2. MongoDB 상태 확인

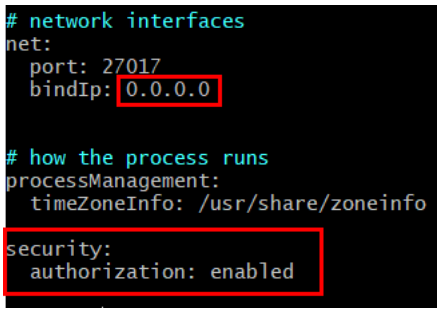
```
netstat -tnlp
```

아래처럼 "127.0.0.1:27017" 을 확인합니다.

```
ubuntu@ip-172-31-34-31:~$ 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 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:5000            0.0.0.0:*               LISTEN      13341/python
tcp        0      0 127.0.0.1:27017         0.0.0.0:*               LISTEN      -
tcp6       0      0 :::22                   :::*                    LISTEN      -
```

3. 설정 확인

```
sudo vi /etc/mongod.conf

위 명령어 입력 후 다음 내용이 제대로 입력되었는지 확인합니다.
net:
  port: 27017
  bindIp: 0.0.0.0

security:
  authorization: enabled
```



4. 재시작

```
sudo service mongod stop

netstat -tnlp
```

위 명령어를 차례대로 입력하여 27017 포트에 아무것도 떠있지 않음을 확인합니다.

```
ubuntu@ip-172-31-34-31:~$ sudo service mongod stop
ubuntu@ip-172-31-34-31:~$ 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 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:5000            0.0.0.0:*               LISTEN      13341/python
tcp6       0      0 :::22                   :::*                    LISTEN      -
```

```
sudo service mongod start

netstat -tnlp
```

위 명령어를 차례대로 입력하여 0.0.0.0:27017 이 목록에 있음을 확인합니다.

```
ubuntu@ip-172-31-34-31:~$ sudo service mongod start
ubuntu@ip-172-31-34-31:~$ 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 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:5000            0.0.0.0:*               LISTEN      13341/python
tcp        0      0 0.0.0.0:27017           0.0.0.0:*               LISTEN      -
tcp6       0      0 :::22                   :::*                    LISTEN      -
```

다시 위로 돌아가서 (1) id/pw를 넣고 접속해보기 과정을 시도합니다.

▼ 3) 1~5주차 완성본을 filezilla로 EC2에 업로드해봅니다.

- 코드가 없는 분들을 위한 완성본

  https://s3-us-west-2.amazonaws.com/secure.notion-static.com/0a5ebaf2-1c4f-45a9-947b-0a1abaed909b/sparta-w4.zip

- MongoDB 설정을 확인해주세요!

  어떤 설정을 말하는 것인가요?

  AWS의 MongoDB에 아이디와 비밀번호를 추가했으니, 우리의 Pymongo에도 아이디와 비밀번호를 입력해주어야 합니다! 그래야 Pymongo가 올바르게 DB에 접근할 수 있습니다. 아래 예시 캡처를 참고해주세요.

```
# client = MongoClient('mongodb://sampleID:samplePW@12.34.56.78', 27017)
client = MongoClient('mongodb://설정한아이디:설정한비밀번호@내AWS아이피', 27017)
db = client.dbsparta
```

▼ 4) 가상환경을 켜고, 완성본을 실행해봅니다.

- 실행하려고 시도하기

```
# home 디렉토리로 이동
cd ~
```

```
# 해당 폴더로 이동해서 아래 코드를 실행합니다.
python app.py
```

- 에러가 나죠? 패키지들을 설치하지 않았기 때문입니다.

```
# 설치하기
pip install requests beautifulsoup4 pymongo
```

- 다시 실행해봅니다

```
python app.py
```

▼ 5) 접속해봅니다!

- 브라우저에서 접속하기

```
http://내AWS아이피:5000/
```

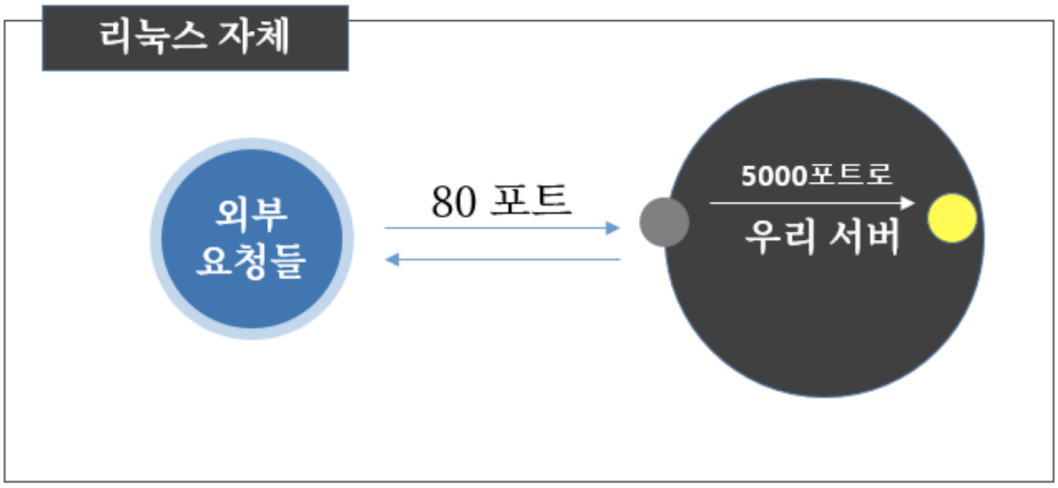
▼ 6) 포트 번호 없애기 (튜터의 배경 설명이 필요한 부분!)

▼ 개념

- 지금은 5000포트로 웹서비스가 돌아가고 있습니다. 그래서 매번 :5000을 뒤에 붙여줘야 하죠. 포트번호를 없애려면 어떻게 해야할까요?
- http 요청을 할 때, 포트를 80으로 설정해두면, 기본 포트이기 때문에 굳이 :80을 붙이지 않아도 됩니다.
- 결국, 1) 서버를 5000이 아니라 80으로 돌리거나, 2) 아니면 앞에 연결 통로를 붙여야 합니다. 그런데, 1)번 방법은 보안 문제로 애초에 불가능합니다.

▼ 우리가 쓸 방법

- 우리가 쓸 방법은 리눅스 자체의 'port forwarding'을 하는 것입니다.
- 그림으로 정리



▼ 리눅스 자체 포트포워딩을 작동시키기

- ssh / putty로 서버에 접속합니다.
- 포트포워딩 룰을 입력합니다.

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

- 접속해봅니다! :5000 없이 정상 접속이 됩니다.

▼ 7) 포트 번호를 떼고 접속해봅니다!

- 브라우저에서 접속하기

```
http://내AWS아이피/
```

▼ 8) SSH 접속을 끊어도 서버가 계속 돌게 하기

- 현재 상황

  Git bash 또는 맥의 터미널을 종료하면 (=즉, SSH 접속을 끊으면) 프로세스가 종료되면서, 서버가 돌아가지 않고 있습니다. 그러나 우리가 원격접속을 끊어도, 서버는 계속 동작해야겠죠?

- 원격 접속을 종료하더라도 서버가 계속 돌아가게 하기

```
# 아래의 명령어로 실행하면 된다
nohup python app.py &
```

- 서버 종료하기 - 강제종료하는 방법

```
# 아래 명령어로 미리 pid 값(프로세스 번호)을 본다
ps -ef | grep 'python'

# 아래 명령어로 특정 프로세스를 죽인다
kill -9 [pid값]
```

- 다시 켜기

```
nohup python app.py &
```

▼ 9) SSH 접속을 종료한 뒤, 접속해봅니다!

- 브라우저에서 접속하기

```
http://내AWS아이피/
```

[1시간]: 프로젝트 코딩 & Todo 점검

▼ 10) 개별 면담

- 튜터와 5분 씩 개별 면담을 진행합니다.
- 프로젝트의 범위와 우선순위를 점검합니다.

후반 3시간

[월수/화목반] : 체크인 & 출석체크

> 🔥 **튜터님은 체크인과 함께 출석 체크(링크)를 진행해주세요!**

▼ "15초 체크인"을 진행합니다.

- 튜터는 타이머를 띄워주세요! (링크)
- 본인의 감정상태와 오늘 있었던 소소한 일을 공유하는 시간입니다.

[2시간]: 프로젝트 개발

▼ 11) 본격적으로 개발을 진행합니다.

- API 별로 차례차례 구현합니다.

[1시간]: 튜터 피드백 - Todo 리스트

▼ 12) 개별 면담을 통해 Todo 리스트를 튜터와 함께 선정합니다.

- 각자 프로젝트 코딩을 하고, 튜터가 5분 씩 개별 면담을 진행하여 Todo 리스트를 함께 선정합니다.

[끝]

▼ "15초 체크아웃"을 진행합니다.

- 튜터는 타이머를 띄워주세요! (링크)
- 다음 시간까지 해올 Todo 리스트에 대한 계획을 말해도 좋습니다!

[숙제] - 다음 수업까지 개발일지 써오기

- 튜터와 함께 정한 Todo 리스트 개발일지를 작성합니다.

> 👉 개발일지는, 본인을 위해 작성하는 것입니다! 꼭 일주일에 한 편씩만 작성하지 않아도 되며, 사실은 개발을 할 때마다 매일 작성하는 것이 가장 좋은 방법입니다.
>
> TIL (today I learned)를 검색하면, 많은 분들의 개발일지를 볼 수 있습니다.

1. 한 주 동안의 회고
2. 한 주 동안의 배운 것들
3. 이번주의 목표

[설치] - 다음 시간을 위해 미리 설치해와야 할 것들

1. 가비아 가입하기
   - 가입: https://www.gabia.com
   - 가비아에서 할인이벤트(500원/1년)를 진행하는 도메인을 구매해서 진행 할 예정입니다.