



스파르타코딩클럽 7기 7주차 - 학습자료

[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.

🔊: 실습 시간

* 강의 상황에 따라, 시간은 유동적일 수 있습니다.

전반 3시간

[2시간]: 수업

▼ 1) EC2에 접속하기 (이제 익숙하시죠?)

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

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

- 참고: mongoDB 공식자료를 참고해서 정리했습니다. ([링크](#))
- 아래 코드를 차례대로 붙여 넣어서 설치한 뒤,

```
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 9DA31620334BD75D9DCB49F368818C72E52529D4
echo "deb [ arch=amd64 ] https://repo.mongodb.org/apt/ubuntu bionic/mongodb-org/4.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-4.0.list
sudo apt-get update
sudo apt-get install -y mongodb-org
```

- 아래 코드로 실행합니다.

```
// 실행
sudo service mongod start

// 확인(mongoDB 콘솔에 접속해보기)
mongo

// 나오기
exit

// 참고: 스탑
sudo service mongod stop
```

▼ 계정 생성하기 & IP bind 설정하기



우리의 컴퓨터에 각자 설치한 Robo3T를 이용해서, AWS의 MongoDB에 접근하기 위해서는 다음 단계의 작업이 필요합니다.

조금 어려울 수 있으니 튜터 님과 함께 진행해주세요!

▼ 1) MongoDB 계정 만들기

```
// MongoDB 시작하기
sudo service mongod start

// MongoDB 로 들어가기
mongo
```

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

```
// admin으로 계정 바꾸기
use admin;

// 계정 생성하기
db.createUser({user: "여기에 아이디를 입력해주세요", pwd: "여기에 비밀번호를 입력해주세요", roles:["root"]});
```

예를 들어 ID와 비밀번호가 모두 test인 계정의 생성이 성공적으로 마무리되었다면, 다음과 같은 화면을 보실 수 있습니다.

```
> 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
```



앞으로 mongo 명령어 실행하실 때는, 다음과 같은 옵션을 주시면 됩니다.

mongo -u 생성한아이디 -p 생성한비밀번호

▼ 2) IP bind 설정하기



이건 무엇을 위한 것인가요?

- 외부에서 MongoDB 접근이 가능하도록 MongoDB 설정의 잠금을 풀어주는 것입니다.

```
sudo vi /etc/mongod.conf
```

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

```
# network interfaces
net:
  port: 27017
  bindIp: 127.0.0.1

# how the process runs
processManagement:
  timeZoneInfo: /usr/share/zoneinfo

#security:
```

```
// 입력 모드 전환
i
```

위 붉은 박스의 내용을 아래와 같이 바꿔주세요!

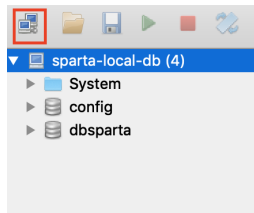
```
# network interfaces
net:
  port: 27017
  bindIp: 0.0.0.0

# how the process runs
processManagement:
  timeZoneInfo: /usr/share/zoneinfo

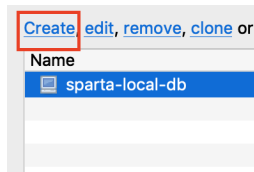
security:
  authorization: 'enabled'
```

```
// 내용 저장하고 에디터 종료하기
:wq
```

▼ 3) Robo3T로 연결 확인하기



좌측 상단 빨간 상자 내 아이콘을 클릭합니다.



'Create' 버튼을 클릭합니다.

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

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

- 코드가 없는 분들을 위한 완성본

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/775453b8-f8d4-4544-8532-e353685c924f/sparta-w4.zip>

▼ 4) 가상환경을 켜고, 완성본을 실행해봅니다.

- 실행하려고 시도하기

```
// home 디렉토리로 이동
cd ~

// 가상환경을 켜다 (myenv -> 본인이 설정한 가상환경의 폴더명을 입력)
. myenv/bin/activate

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

- 에러가 나죠? 패키지들을 설치하지 않았기 때문입니다.

```
// 설치하기
pip3 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

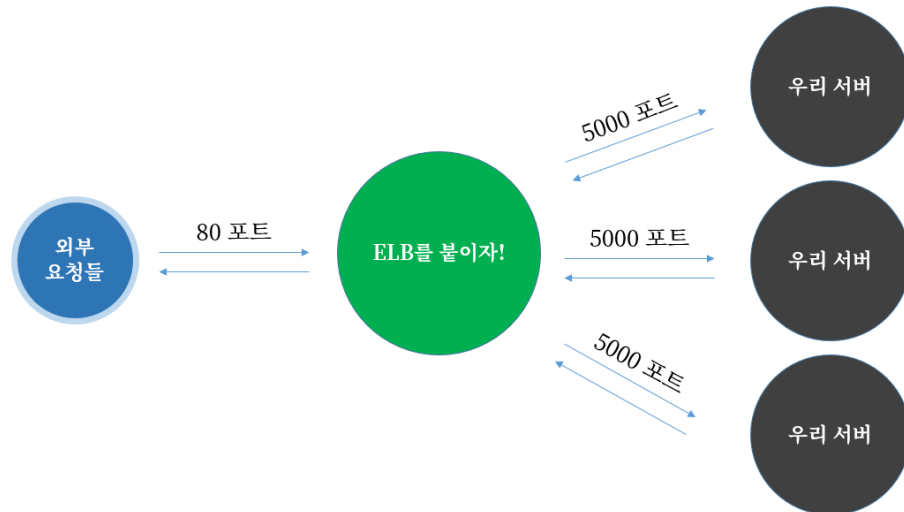
- 지금은 5000포트로 웹서비스가 돌아가고 있습니다. 그래서 매번 :5000을 뒤에 붙여줘야 하죠. 포트번호를 없애려면 어떻게 해야할까요?
- http 요청을 할 때, 포트를 80으로 설정해두면, 기본 포트이기 때문에 굳이 :80을 붙이지 않아도 됩니다.
- 결국, 1) 서버를 5000이 아니라 80으로 돌리거나, 2) 아니면 앞에 연결 통로를 붙여야 합니다. 그런데, 1)번 방법은 보안 문제로 애초에 불가능합니다.

▼ 우리가 쓸 방법

- 2) 연결 통로를 붙이는 방법에도 여러가지가 있는데, 우리가 쓸 방법은 리눅스 자체의 'port forwarding'을 하는 것입니다.
- 다른 방법으로는,
 1. AWS를 충분히 활용하는 방법
 - AWS에서 제공하는 Elastic load balancer와, route53을 사용해 80포트로 들어오는 것을 5000포트로 보내주게 함

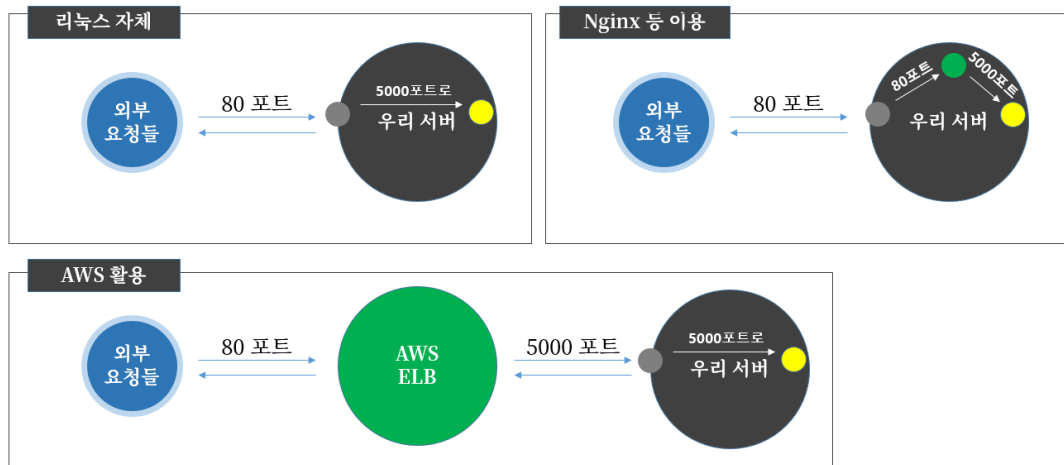
▼ ELB란?

- Elastic load balancer란, 본래 아래와 같이 웹서버의 트래픽을 분산하고자 쓰는 프로그램입니다. 로드밸런싱 프로그램만 설치된 서버를 하나 사셨다고 생각하면 됩니다.
- 기본적으로 웹서버 보다 앞에 붙어서 외부 요청을 forwarding해주는 역할을 하기 때문에, 서버가 한 대일 때에도 포트 번호를 바꿔주는 데 쓰기도 용이합니다.



2. 직접 라우팅 프로그램을 설치한다.

- 굳이 AWS에서 제공해주는 걸 쓰지 않고, 기본적으로 서버 내에 밸런싱해주는 프로그램을 설치해서 쓰면 됩니다!
 - apache, nginx와 같은 프로그램 이야기를 들어보셨나요? 이들이 하는 역할입니다.
- 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.

설명	인바운드	아웃바운드	태그
편집			
유형 <small>i</small>	프로토콜 <small>i</small>	포트 범위 <small>i</small>	소스 <small>i</small>
HTTP 	TCP	80	0.0.0.0/0
HTTP	TCP	80	::/0
SSH	TCP	22	0.0.0.0/0
사용자 지정 TCP 규칙	TCP	5000	0.0.0.0/0
사용자 지정 TCP 규칙	TCP	5000	::/0

▼ 리눅스 자체 포트포워딩을 해제시키기

- 모든 룰을 봅니다.

```
sudo iptables -t nat -L --line-numbers
```

- 삭제 하고 싶은 룰을 삭제합니다. (방금 정한 것은 아마 1번째 것일 거예요)

```
sudo iptables -t nat -D PREROUTING 1
```

▼ 7) 포트 번호를 떼고 접속해봅니다!

- 브라우저에서 접속하기

```
http://내AWS아이피/
```

▼ 8) SSH 접속을 끊어도 서버가 계속 돌게 하기

- 현재 상황

putty 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 개요 & 목표 공유

- ▼ 각자의 프로젝트 진행 상황 / 목표를 공유합니다.



"개발일지" 작성해오셨나요?

채팅방에 각자의 개발일지를 공유하고, 돌아가며 발표합니다.

- 발표에 꼭! 들어가야 하는 내용들

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

- ▼ github repository의 주소를 톡방에 공유합니다.

- 튜터는 진행상황을 체크하고, 코드리뷰를 준비합니다

휴식시간 (30분)

후반 3시간

[3시간]: 개별 코딩 & 코드리뷰

- 집중 코딩을 합니다.
- 궁금한 것은 튜터에게 물어봅니다.
- 30분~1시간에 한번씩 튜터가 돌며 코드 리뷰를 합니다.

[숙제] - 다음 수업까지 개발일지 써오기

- Medium에 아래 내용이 포함된 개발일지를 작성해옵니다.



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.