

43강. EC2에 접속해 리눅스 명령어 다뤄보기

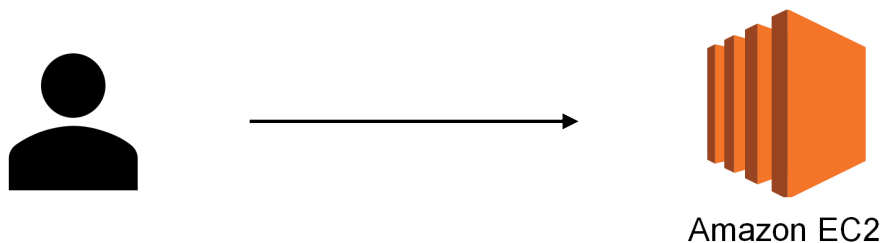
Section 7. 생애 최초 배포하기

1. EC2에 접속하는 방법을 알아보고,
EC2에 접속해 리눅스 명령어를 다뤄본다.
2. 개발한 서버의 배포를 위해 환경 셋팅을 리눅스에서 진행하고,
실제 배포를 진행한다.

Section 7. 생애 최초 배포하기

3. foreground와 background의 차이를 이해하고 background 서버를 제어한다.
4. 도메인 이름을 사용해 사용자가 IP 대신 이름으로 접속할 수 있도록 한다.

이제 지난 시간에 빌린 컴퓨터에 접속해보자!



- 1) 다운로드 받은 키 페어 (pem키) 를 이용하는 방법
- 2) AWS 콘솔을 활용해 접속하는 방법

[1] 다운로드 받은 키 페어를 이용하는 방법

- 1) 우리가 접속하려는 EC2의 IP 주소
- 2) 이전 시간에 다운로드 받았던 키 페어
- 3) 접속하기 위한 프로그램 (git CLI 혹은 Mac terminal)

[1] 다운로드 받은 키 페어를 이용하는 방법

```
ssh -i 경로/키페어이름.pem ec2-user@IP
```

[1] 다운로드 받은 키 페어를 이용하는 방법

```
ssh -i /Users/lannstark/studying-developer.pem  
ec2-user@43.201.84.217
```

[1] 다운로드 받은 키 페어를 이용하는 방법

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@          WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0777 for '/Users/lannstark/Desktop/studying-developer.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "/Users/lannstark/Desktop/studying-developer.pem": bad permissions
ec2-user@43.201.84.217: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
```


[1] 다운로드 받은 키 페어를 이용하는 방법

chmod 400 경로/키페어이름.pem

[2] AWS 콘솔을 활용하는 방법

인스턴스 (1/1) 정보

Find 인스턴스 by attribute or tag (case-sensitive)

인스턴스 상태 = running X 필터 지우기

<input checked="" type="checkbox"/>	Name	인스턴스 ID	인스턴스 상태	인스턴스 유형	상태 검사
<input checked="" type="checkbox"/>	Library App Se...	i-01a01f13449d94919	실행 중	t2.micro	2/2개 검사 통과

EC2 > 인스턴스 > i-01a01f13449d94919 > 인스턴스에 연결

인스턴스에 연결 정보

다음 옵션 중 하나를 사용하여 인스턴스 i-01a01f13449d94919 (Library App Server)에 연결

EC2 인스턴스 연결 | Session Manager | SSH 클라이언트 | EC2 직렬 콘솔

인스턴스 ID
i-01a01f13449d94919 (Library App Server)

퍼블릭 IP 주소
43.201.84.217

사용자 이름
ec2-user

사용자 지정 사용자 이름을 사용하여 연결하거나 인스턴스 시작에 사용한 AMI의 기본 사용자 이름 ec2-user을(를) 사용합니다.

참고: 대부분의 경우 추정된 사용자 이름은 정확합니다. 하지만 AMI 사용 지침을 읽고 AMI 소유자가 기본 AMI 사용자 이름을 변경했는지 확인하십시오.

취소 연결

가장 기본적인 5가지 리눅스 명령어를 알아보자!

mkdir : 폴더를 만드는 명령어

```
mkdir folder1
```

가장 기본적인 5가지 리눅스 명령어를 알아보자!

ls : 현재 위치에서 폴더나 파일을 확인하는 명령어

가장 기본적인 5가지 리눅스 명령어를 알아보자!

ls -l : 조금 더 자세한 정보를 확인할 수 있다!

가장 기본적인 5가지 리눅스 명령어를 알아보자!

```
[ec2-user@ip-172-31-4-234 folder1]$ ls -l  
합계 0  
drwxrwxr-x 2 ec2-user ec2-user 6 12월 22 12:13 folder1
```

drwxrwxr-x



가장 기본적인 5가지 리눅스 명령어를 알아보자!

```
[ec2-user@ip-172-31-4-234 folder1]$ ls -l
합계 0
drwxrwxr-x 2 ec2-user ec2-user 6 12월 22 12:13 folder1
```

drwxrwxr-x : folder1은 폴더이다.

가장 기본적인 5가지 리눅스 명령어를 알아보자!

```
[ec2-user@ip-172-31-4-234 folder1]$ ls -l  
합계 0  
drwxrwxr-x 2 ec2-user ec2-user 6 12월 22 12:13 folder1
```

d**rw**xrwxr-x

r : 읽을 수 있는 권한 / w : 쓸 수 있는 권한 / x : 실행할 수 있는 권한

가장 기본적인 5가지 리눅스 명령어를 알아보자!

```
[ec2-user@ip-172-31-4-234 folder1]$ ls -l
합계 0
drwxrwxr-x 2 ec2-user ec2-user 6 12월 22 12:13 folder1
```

drwxrwxr-x

폴더 소유자의 권한 / 폴더 소유그룹의 권한 / 아무나 접근했을 때의 권한

가장 기본적인 5가지 리눅스 명령어를 알아보자!

```
[ec2-user@ip-172-31-4-234 folder1]$ ls -l  
합계 0  
drwxrwxr-x 2 ec2-user ec2-user 6 12월 22 12:13 folder1
```



2 : 폴더에 걸려 있는 바로가기 개수

가장 기본적인 5가지 리눅스 명령어를 알아보자!

```
[ec2-user@ip-172-31-4-234 folder1]$ ls -l  
합계 0  
drwxrwxr-x 2 ec2-user ec2-user 6 12월 22 12:13 folder1
```

ec2-user : 이 폴더의 소유주 (주인) 이름

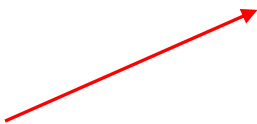
가장 기본적인 5가지 리눅스 명령어를 알아보자!

```
[ec2-user@ip-172-31-4-234 folder1]$ ls -l
합계 0
drwxrwxr-x 2 ec2-user ec2-user 6 12월 22 12:13 folder1
```

ec2-user : 이 폴더의 소유그룹 이름

가장 기본적인 5가지 리눅스 명령어를 알아보자!

```
[ec2-user@ip-172-31-4-234 folder1]$ ls -l  
합계 0  
drwxrwxr-x 2 ec2-user ec2-user 6 12월 22 12:13 folder1
```



6 : 이 폴더(파일)의 크기, byte 단위

가장 기본적인 5가지 리눅스 명령어를 알아보자!

```
[ec2-user@ip-172-31-4-234 folder1]$ ls -l  
합계 0  
drwxrwxr-x 2 ec2-user ec2-user 6 12월 22 12:13 folder1
```

12월 22 12:13 : 파일의 최종 변경 시각



가장 기본적인 5가지 리눅스 명령어를 알아보자!

folder2 / folder3 2개의 폴더를 추가로 만들고

folder2 안으로 들어가자!

가장 기본적인 5가지 리눅스 명령어를 알아보자!

cd : 폴더 안으로 들어가는 명령어

```
cd folder2
```


가장 기본적인 5가지 리눅스 명령어를 알아보자!

pwd : 현재 위치를 확인하는 명령어

가장 기본적인 5가지 리눅스 명령어를 알아보자!

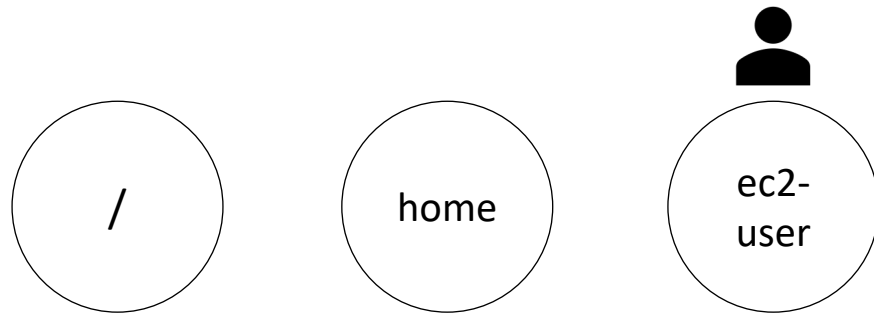
cd .. : 상위 폴더로 올라가는 명령어

가장 기본적인 5가지 리눅스 명령어를 알아보자!

rmmdir : 비어 있는 폴더(디렉토리)를 제거하는 명령어

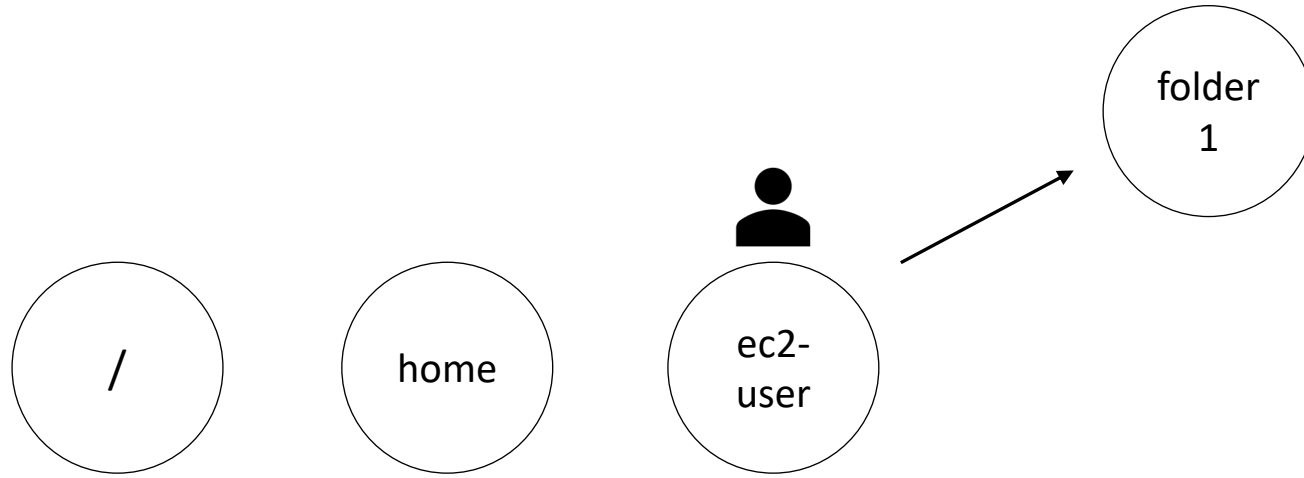
```
rmmdir folder3
```

리눅스 명령어 57가지 정리!



ec2-user로 접속하면 시작하는 위치

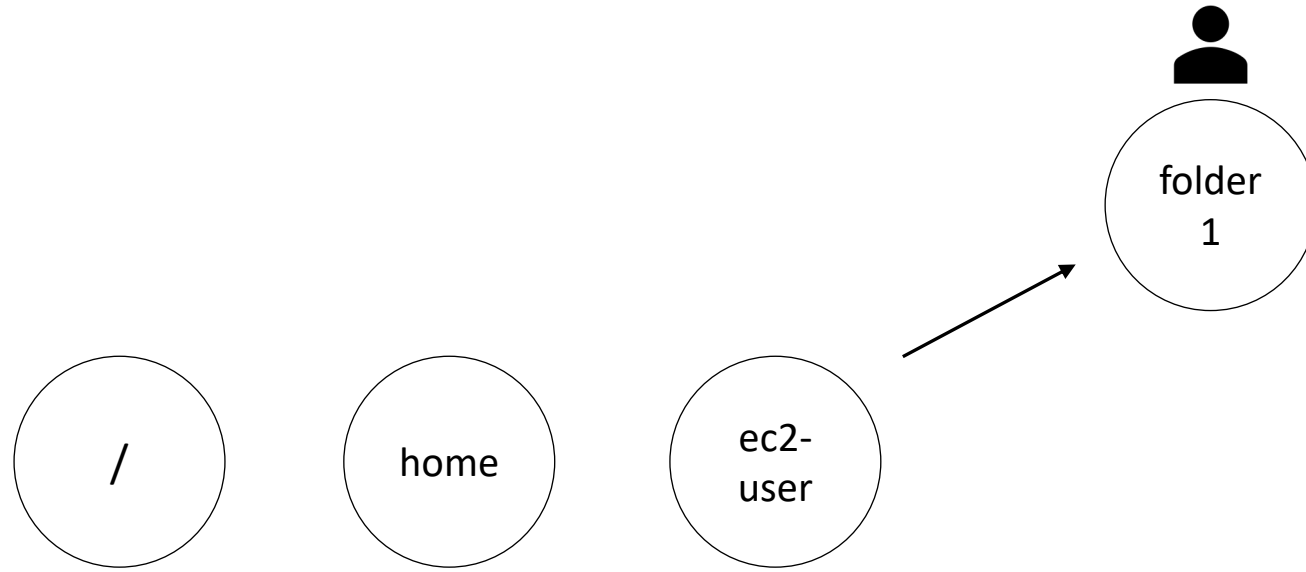
리눅스 명령어 57가지 정리!



`mkdir folder1`

folder1 폴더를 만든다.

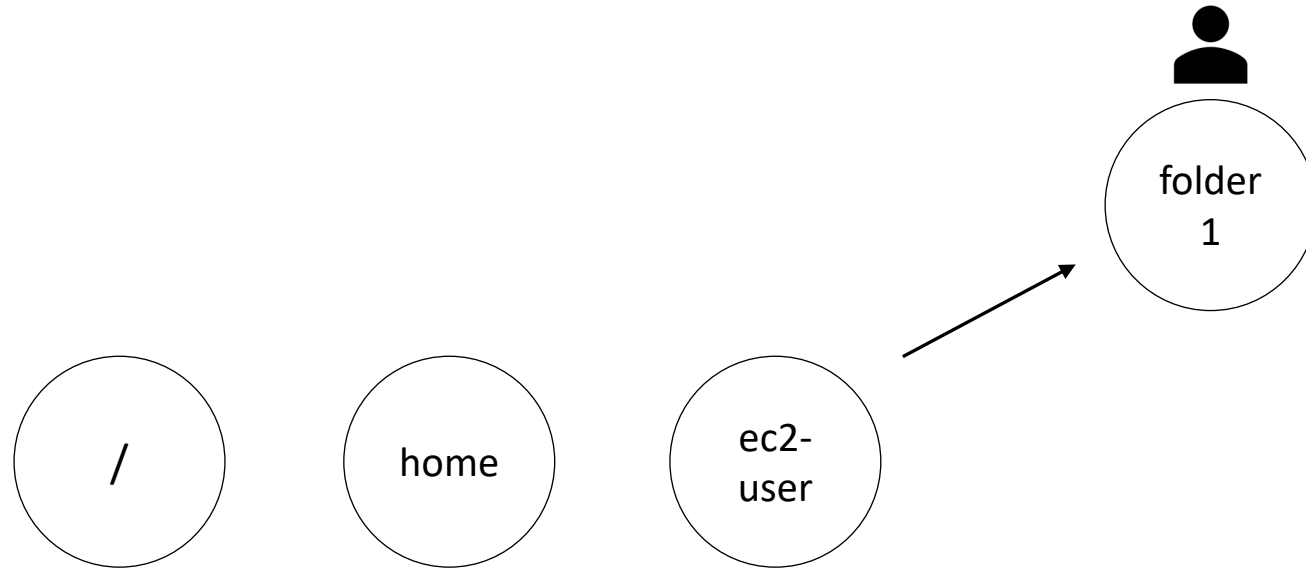
리눅스 명령어 57가지 정리!



`cd folder1`

folder1 폴더로 들어간다.

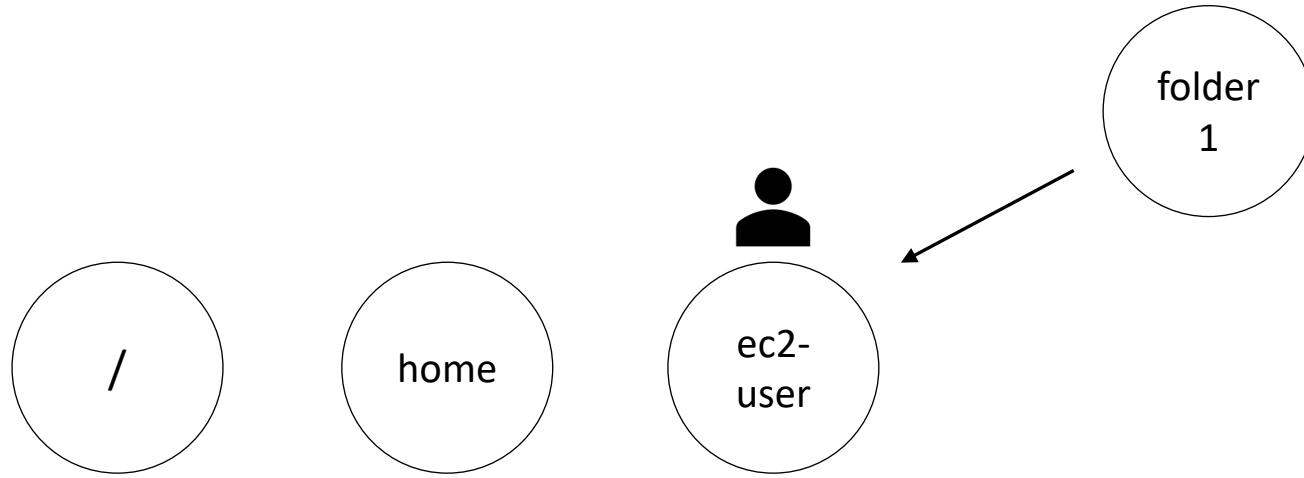
리눅스 명령어 57가지 정리!



pwd

현재 위치를 출력한다. /home/ec2-user/folder1

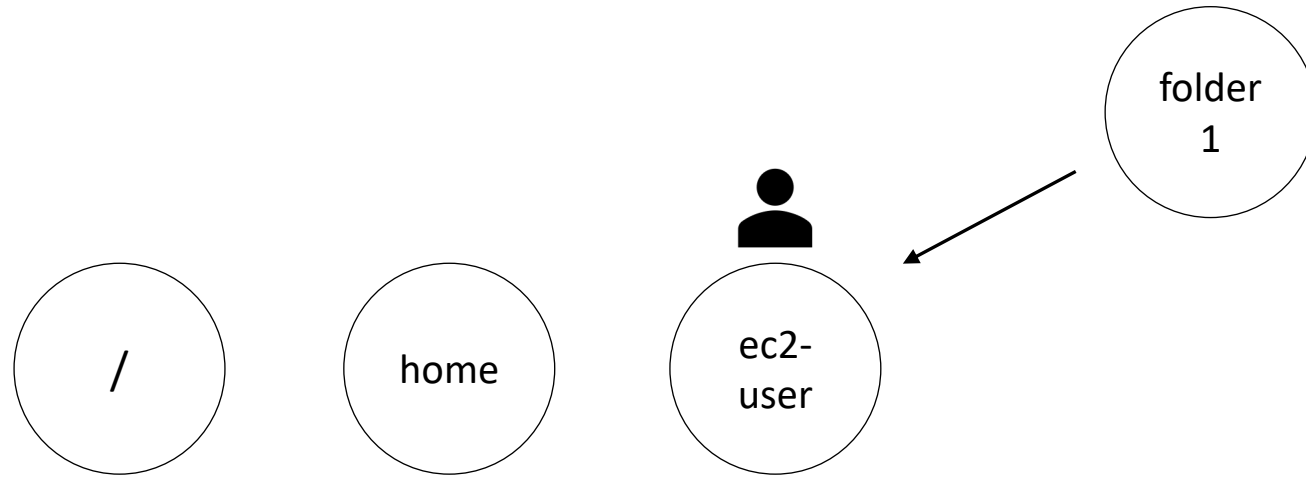
리눅스 명령어 57가지 정리!



`cd ..`

상위 폴더로 이동한다.

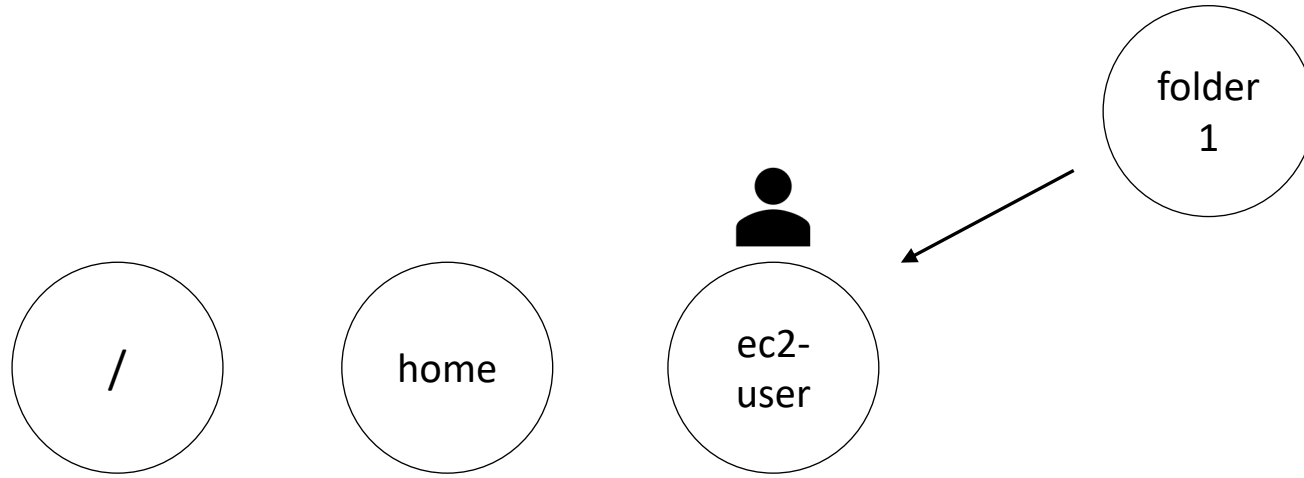
리눅스 명령어 5가지 정리!



ls 또는 **ls -l**

현재 위치에 있는 폴더 및 파일 목록을 출력한다.

리눅스 명령어 57가지 정리!



`rmdir folder1`

folder1을 제거한다.

44강. 배포를 위한 프로그램 설치하기

우리가 설치할 프로그램 목록

- 1) 코드를 가져오기 위한 git
- 2) 우리가 만든 서버를 구동할 java
- 3) 데이터베이스의 역할을 할 mysql

리눅스에서 스프링 서버 배포를 위한 프로그램을 설치하자!

```
sudo yum update
```

리눅스에서 스프링 서버 배포를 위한 프로그램을 설치하자!

```
sudo yum update
```

관리자의 권한으로 실행한다.

리눅스에서 스프링 서버 배포를 위한 프로그램을 설치하자!

```
sudo yum update
```

리눅스 패키지 관리 프로그램 (gradle과 비슷한 역할)

리눅스에서 스프링 서버 배포를 위한 프로그램을 설치하자!

```
sudo yum update
```

현재 설치된 여러 프로그램들을 최신화한다.

[1] git 설치

```
sudo yum install git
```

[1] git 설치

```
sudo yum install git
```

yum을 이용해 프로그램을 다운로드 한다.

[1] git 설치

```
sudo yum install git -y
```

중간에 yes or no를 물어보지 않고 바로 설치한다.

[2] java 설치

```
sudo yum install java-11-amazon-corretto -y
```

[2] java 설치

java -version

[3] mysql 설치

sudo **systemctl** status mysqld

현재 보이지 않는 프로그램을 관리하는 명령어

[3] mysql 설치

```
sudo systemctl status mysqld
```

mysqld 프로그램의 상태를 보여줘!

[3] mysql 설치

sudo systemctl **restart mysql**

mysqld 프로그램을 재시작해줘!

[3] mysql 설치

```
sudo cat /var/log/mysqld.log | grep "A temporary password"
```

mysql8의 임시 비밀번호를 확인하는 명령어

[3] mysql 설치

mysql -u root -p

MySQL에 접속하는 명령어

Mysql 설치가 끝났으니, 데이터베이스 및 테이블을 설정하자!

임시 비밀번호를 변경하자!

```
ALTER user 'root'@'localhost'  
IDENTIFIED WITH mysql_native_password BY "비밀번호";
```

임시 비밀번호를 변경하자!

```
ALTER user 'root'@'localhost'  
IDENTIFIED WITH mysql_native_password BY "Abcd1234!";
```

profile도 함께 변경해주어야 한다!

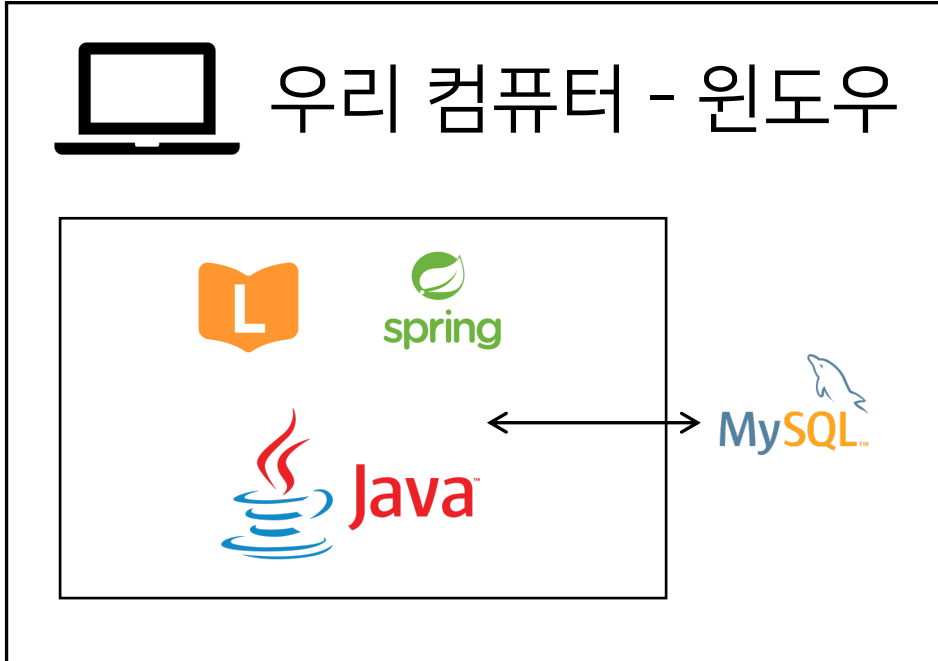
```
spring:
  config:
    activate:
      on-profile: dev
  datasource:
    url: "jdbc:mysql://localhost/library"
    username: "root"
    password: "Abcd1234!"
    driver-class-name: com.mysql.cj.jdbc.Driver
```



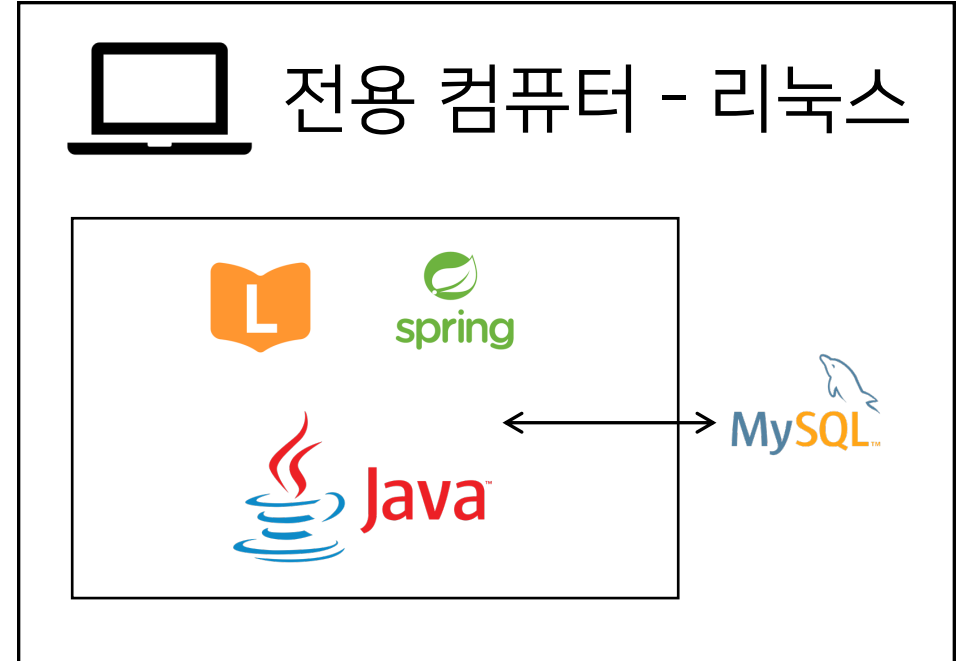
git을 다시 한 번 사용해 원격 저장소를 최신화 해주자!

45강. 빌드와 실행, 그리고 접속

가장 먼저 코드를 가져와야 합니다!



배포
→



main

1 branch

0 tags

Go to file

Add file

<> Code

studying-developer 첫 번째 커밋

gradle/wrapper	첫 번째 커밋
src	첫 번째 커밋
.gitignore	첫 번째 커밋
build.gradle	첫 번째 커밋
gradlew	첫 번째 커밋
gradlew.bat	첫 번째 커밋
settings.gradle	첫 번째 커밋

Local

Codespaces New

Clone

HTTPS SSH GitHub CLI New

https://github.com/studying-developer/library

Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

Download ZIP

Help people interested in this repository understand your project by adding a README.

Add a README

코드를 가져오자!

git clone [github 저장소 주소]

git clone <https://github.com/studying-developer/library-app.git>

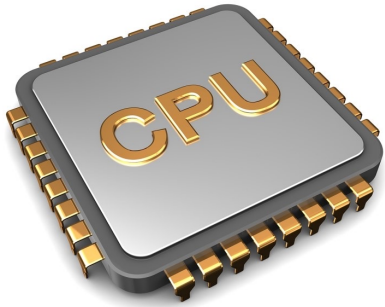
이제 빌드를 하고 실행을 할건데..!

현재 우리가 사용하고 있는 컴퓨터는 성능이 그렇게 좋지 않아,
빌드나 실행시 렉이 많이 걸릴 수 있다.

이제 빌드를 하고 실행을 할건데..!

따라서, 메모리가 부족한 경우 디스크를 사용할 수 있도록 설정해주자!

Swap 설정



CPU (연산담당)



RAM (메모리, 단기 기억)



DISK (장기 기록)

원래 RAM을 사용해야 하지만, 일부 DISK를 사용하게 해준다.

Swap 설정

```
# swap 메모리를 할당한다 (128M * 16 = 2GB)
```

```
sudo dd if=/dev/zero of=/swapfile bs=128M count=16
```

```
# 스왑 파일에 대한 권한 업데이트
```

```
sudo chmod 600 /swapfile
```

```
# swap 영역 설정
```

```
sudo mkswap /swapfile
```

```
# swap 파일을 사용할 수 있도록 만든다.
```

```
sudo swapon /swapfile
```

```
# swap 성공 확인
```

```
sudo swapon -s
```

빌드를 준비하자!

```
chmod +x ./gradlew
```

gradlew를 사용하기 위해 실행할 수 있도록 설정한다.

빌드

```
./gradlew build -x test
```

gradle을 이용해 우리 프로젝트를 빌드한다. 이때 테스트를 돌리지 않는다.

빌드

`./gradlew build`

gradle을 이용해 우리 프로젝트를 빌드한다. 테스트도 돌린다.

빌드된 프로젝트 실행하기

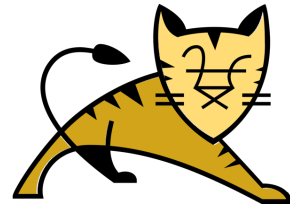
```
java -jar build/libs/library-app-0.0.1-SNAPSHOT.jar  
--spring.profiles.active=dev
```

빌드된 우리의 프로젝트를 실행시킨다!

Jar 파일을 실행만 했는데 서버가 동작한다!

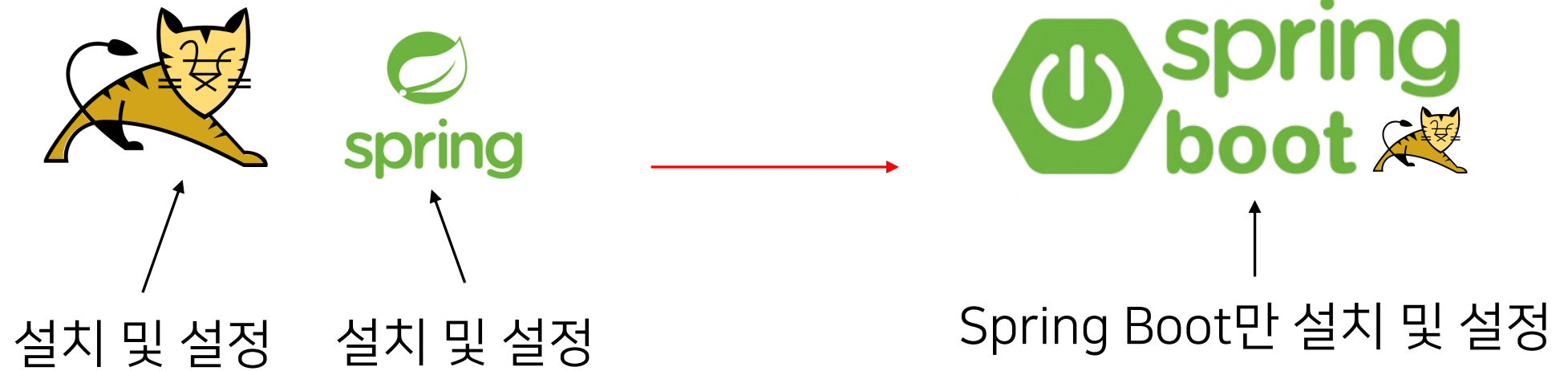
Spring Boot에서는
톰캣이 내장되어 있어 가능하다!

툰캣



웹 애플리케이션 서버 (Web Application Server, WAS)의 한 종류
요청이 들어오면 그 요청을 약속된 형식에 맞추어 스프링에 전달해준다!

톰캣



이제 서버가 실행되었으니 웹 UI에 들어가보자!

실행중인 서버 중단하기

ctrl + c

무언가를 중단하는 신호

빌드된 프로젝트 제거하기

`./gradlew clean`

현재 빌드되어 있는 결과물을 제거한다.

46강. 종료되지 않는 실행

AWS EC2 접속을 종료해도 서버는 실행되어야 한다!

foreground vs background

foreground

우리가 보고 있는 프로그램

문서를 보고 있다면, PDF 프로그램이 foreground 프로그램

인프런 강의를 보고 있다면, 브라우저가 foreground 프로그램

background

우리가 보고 있지 않는데 실행중인 프로그램

컴퓨터에서 조용히 돌아가고 있는 백신이 background 프로그램

음악을 틀고, 게임을 한다면

음악 재생 프로그램이 background 프로그램이다.

background로 동작하게 만드는 명령어

nohup [명령어] &

파일을 제거하는 명령어

rm : 파일을 제거하는 명령어

```
rm nohup.out
```


그렇다면 background로 동작하는 서버를 어떻게 끝까?

작업관리자 명령어

ps aux : 현재 실행중인 프로그램 목록을 확인할 수 있다.

작업관리자 명령어

```
ps aux | grep java
```

현재 실행중인 프로그램 중 java가 들어가는 프로그램을 확인한다.

종료 명령어

kill -9 프로그램번호 : 해당 프로그램을 종료시킨다.

파일의 내용물을 확인해보자!

- 1) 파일에 직접 들어가 내용물을 읽는 방법
- 2) 파일에 들어가지 않고,
현재 접속한 터미널에서 내용물을 확인하는 방법

파일에 직접 들어가는 명령어

vi : 리눅스 편집기인 vim을 사용하여 파일을 연다.

```
vi nohup.out
```

파일의 전체 내용을 확인하는 방법

cat : 파일에 있는 내용을 모두 출력하는 명령어

```
cat nohup.out
```

파일의 전체 내용물을 확인하는 방법

cat

파일 내용물의 양이 많지 않고,
실시간 업데이트가 잘 되지 않는 파일을 확인할 때

파일의 끝 부분을 확인하는 방법

tail : 현재 파일의 끝 부분을 출력하는 명령어

```
tail nohup.out
```

파일의 끝 부분을 확인하는 방법

tail **-f** : 현재 파일의 끝 부분을 실시간으로 출력해준다!

```
tail -f nohup.out
```

47강. 가비아를 이용한 도메인 구입, DNS 적용

우선 가비아에 회원가입을 해야 합니다!

메인 페이지 > My 가비아

최태현님

My 정보 관리

12월 결제 예상 금액

0원 >

결제 수단 등록하기



1% 더 할인 받으려면?
정기결제 신청하기



결제 관련 바로가기

미결제 주문서 **0개** 발급
결제하지 않은 주문서 및 가상 계좌

세금 계산서 **0개** 발급 가능
이번 달 기준 발급 가능한 세금 계산서

카트 **0개** 생성
구매할 서비스 목록이 담긴 주문서

예치금 **0원**
현금처럼 사용 가능한 예치금

이용 중인 서비스

대시보드 튜토리얼보기 >

현재 이용 중인 서비스를 확인하세요.

전체보기

이용 중인 서비스 관리 버튼입니다.



도메인

1건 >

IT환경의 최적화된 가비아
서비스를 이용해보세요.



호스팅



홈페이지



쇼핑몰



클라우드



하이웍스



IDC



보안

DNS 관리툴

소유권 이전

담당자 설정

도메인
통합 관리툴

• 관리 정보 입력 필요 서비스
0건 >

• 12월 연장 필요 서비스
0건 >

레코드 기록

타입 : A

이 이름을 주어지는 IP (43.201.84.217)로 연결해줘!

레코드 기록

호스트 : WWW

나중에 www.도메인이름:8080 으로 접속할 수 있게 된다!

레코드 기록

호스트 : www

나중에 www.도메인이름:8080 으로 접속할 수 있게 된다!

<http://www.studying-developer.shop:8080>

혹시 접속이 잘 안된다면, AWS 보안 그룹을 확인하자!

48강. Section 7 정리. 다음으로!

Section 7. 생애 최초 배포하기

1. EC2에 접속하는 방법을 알아보고,
EC2에 접속해 리눅스 명령어를 다뤄본다.
2. 개발한 서버의 배포를 위해 환경 셋팅을 리눅스에서 진행하고,
실제 배포를 진행한다.

Section 7. 생애 최초 배포하기

3. foreground와 background의 차이를 이해하고 background 서버를 제어한다.
4. 도메인 이름을 사용해 사용자가 IP 대신 이름으로 접속할 수 있도록 한다.

지금까지 나왔던 리눅스 명령어 총 정리!

mkdir : 폴더를 만드는 명령어

```
mkdir folder1
```

지금까지 나왔던 리눅스 명령어 총 정리!

ls : 현재 위치에서 폴더나 파일을 확인하는 명령어

지금까지 나왔던 리눅스 명령어 총 정리!

ls -l : 조금 더 자세한 정보를 확인할 수 있다!

지금까지 나왔던 리눅스 명령어 총 정리!

cd : 폴더 안으로 들어가는 명령어

cd folder2

지금까지 나왔던 리눅스 명령어 총 정리!

cd .. : 상위 폴더로 올라가는 명령어

지금까지 나왔던 리눅스 명령어 총 정리!

pwd : 현재 위치를 확인하는 명령어

지금까지 나왔던 리눅스 명령어 총 정리!

rmmdir : 비어 있는 폴더(디렉토리)를 제거하는 명령어

```
rmmdir folder3
```

지금까지 나왔던 리눅스 명령어 총 정리!

```
sudo yum update
```

관리자의 권한으로 설치되어 있는 여러 프로그램을 최신화한다.

지금까지 나왔던 리눅스 명령어 총 정리!

`sudo yum install 프로그램이름`

관리자의 권한으로 프로그램을 설치한다.

지금까지 나왔던 리눅스 명령어 총 정리!

`sudo systemctl status 프로그램` : 프로그램의 상태를 확인한다.

`sudo systemctl status mysqld`

지금까지 나왔던 리눅스 명령어 총 정리!

`sudo systemctl restart 프로그램` : 프로그램을 재시작한다.

`sudo systemctl restart mysqld`

지금까지 나왔던 리눅스 명령어 총 정리!

chmod : 파일이나 폴더의 권한을 변경한다.

지금까지 나왔던 리눅스 명령어 총 정리!

```
chmod +x ./gradlew
```

gradlew를 사용하기 위해 실행할 수 있도록 설정한다.

지금까지 나왔던 리눅스 명령어 총 정리!

ctrl + c

foreground로 실행중인 프로그램을 중단하는 신호

지금까지 나왔던 리눅스 명령어 총 정리!

nohup [명령어] &

명령어를 background로 실행시킨다.

지금까지 나왔던 리눅스 명령어 총 정리!

rm : 파일을 제거하는 명령어

```
rm nohup.out
```

지금까지 나왔던 리눅스 명령어 총 정리!

vi : 리눅스 편집기인 vim을 사용하여 파일을 연다.

```
vi nohup.out
```

지금까지 나왔던 리눅스 명령어 총 정리!

cat : 파일에 있는 내용물을 모두 출력하는 명령어

```
cat nohup.out
```

지금까지 나왔던 리눅스 명령어 총 정리!

tail : 현재 파일의 끝 부분을 출력하는 명령어

```
tail nohup.out
```

지금까지 나왔던 리눅스 명령어 총 정리!

tail **-f** : 현재 파일의 끝 부분을 실시간을 출력해준다!

```
tail -f nohup.out
```


지금까지 나왔던 리눅스 명령어 총 정리!

ps aux : 현재 실행중인 프로그램 목록을 확인할 수 있다.

지금까지 나왔던 리눅스 명령어 총 정리!

```
ps aux | grep java
```

현재 실행중인 프로그램 중 java가 들어가는 프로그램을 확인한다.

지금까지 나왔던 리눅스 명령어 총 정리!

kill -9 프로그램번호 : 해당 프로그램을 종료시킨다.

Gradle 관련 명령어

`./gradlew build` : 프로젝트를 빌드한다.

Gradle 관련 명령어

`./gradlew build -x test` : 프로젝트를 빌드하는데, 테스트는 생략한다.

Gradle 관련 명령어

`./gradlew clean` : 현재 빌드된 결과물을 제거한다.

감사합니다