

37강. 배포란 무엇인가

Section 6. 생애 최초 배포 준비하기

1. 배포가 무엇인지 이해하고,
배포를 하기 위해 어떤 준비를 해야 하는지 알아본다.
2. 스프링 서버를 실행할 때 DB와 같은 설정들을 코드 변경 없이
제어하는 방법을 알아본다.

Section 6. 생애 최초 배포 준비하기

3. git과 github의 차이를 이해하고,
git에 대한 기초적인 사용법을 알아본다.
4. AWS의 EC2가 무엇인지 이해하고,
AWS를 통해 클라우드 컴퓨터를 빌려 본다.

배포

최종 사용자에게 SW를 전달하는 과정



weibo.com/leochenou

도서관리 애플리케이션의 현재 상황



우리 컴퓨터



실행 / 종료

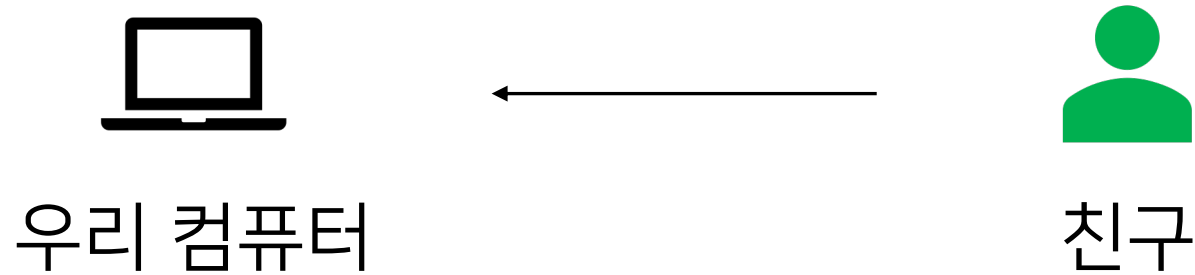
도서관리 애플리케이션의 현재 상황

우리 컴퓨터에 우리가 작성한 코드, 스프링, MySQL 등이 모두 있다!

도서관리 애플리케이션의 현재 상황

자 이제 우리 컴퓨터에 있는 서버에 친구가 접속한다고 해보죠!

친구 (최종 사용자)가 우리의 서버를 쓸 수 있는 방법

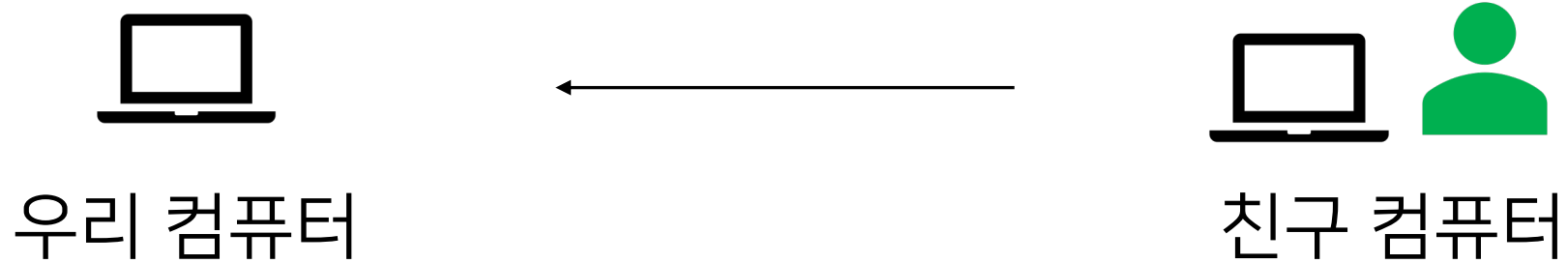


방법 1. 친구가 집으로 직접 놀러온다...!

친구 (최종 사용자)가 우리의 서버를 쓸 수 있는 방법

당연히 이상하고 불편합니다!!

친구 (최종 사용자)가 우리의 서버를 쓸 수 있는 방법

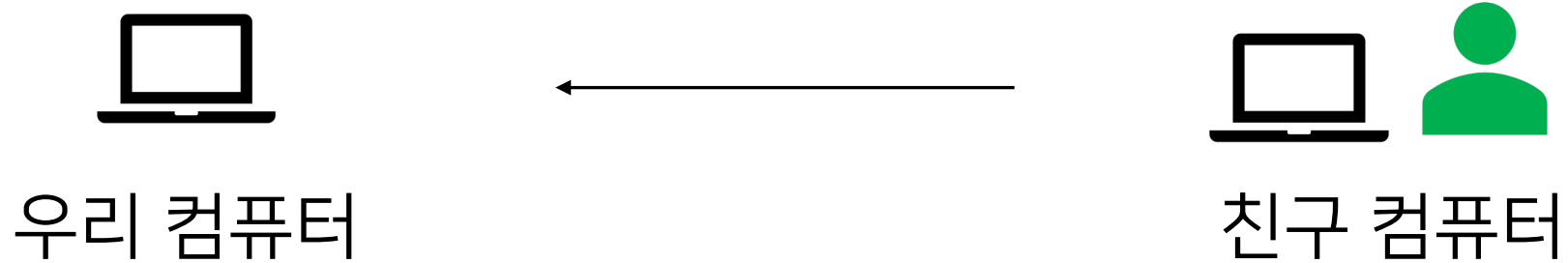


방법 2. 친구의 컴퓨터를 이용해 우리 컴퓨터에 접속한다...!

친구 (최종 사용자)가 우리의 서버를 쓸 수 있는 방법

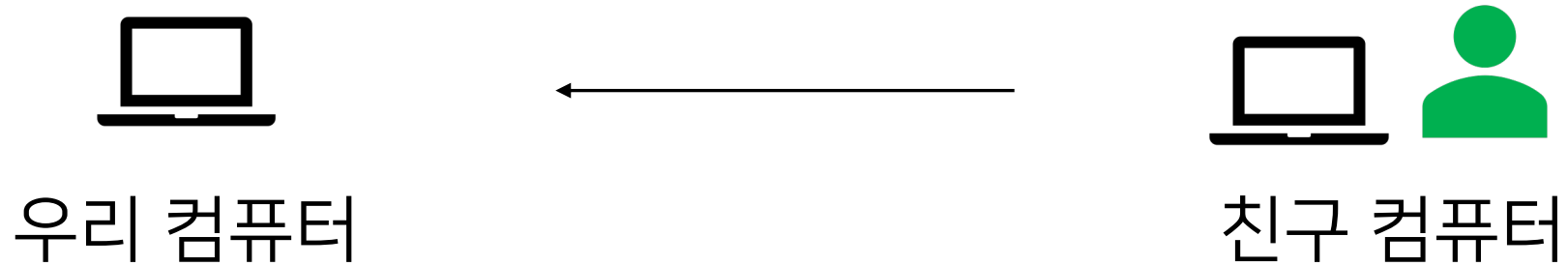
방법 1에 비해 발전했지만, 문제가 있다.

친구 (최종 사용자)가 우리의 서버를 쓸 수 있는 방법



우리 컴퓨터는 24시간 켜져 있지 않다!

친구 (최종 사용자)가 우리의 서버를 쓸 수 있는 방법

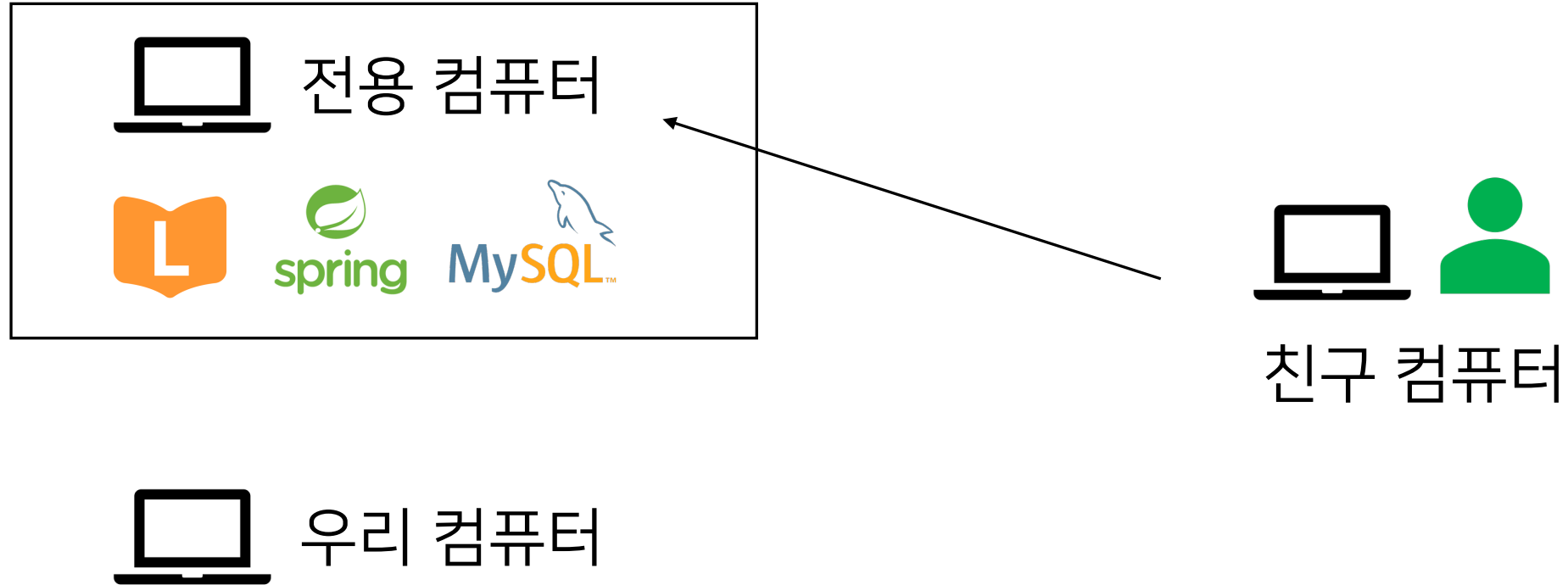


우리 컴퓨터는 스프링, MySQL 외에도
문서 편집기, 게임, 비디오 재생 등을 처리한다!

친구 (최종 사용자)가 우리의 서버를 쓸 수 있는 방법

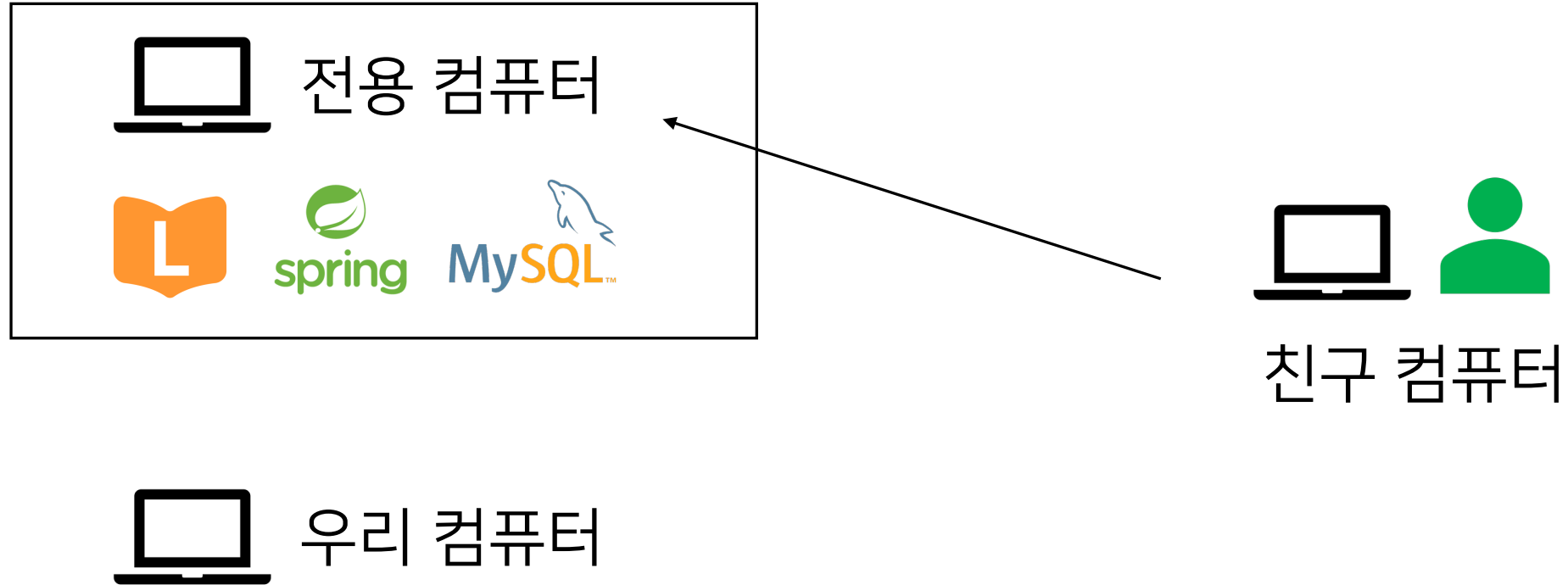
그래서 방법 30이 있다!

친구 (최종 사용자)가 우리의 서버를 쓸 수 있는 방법



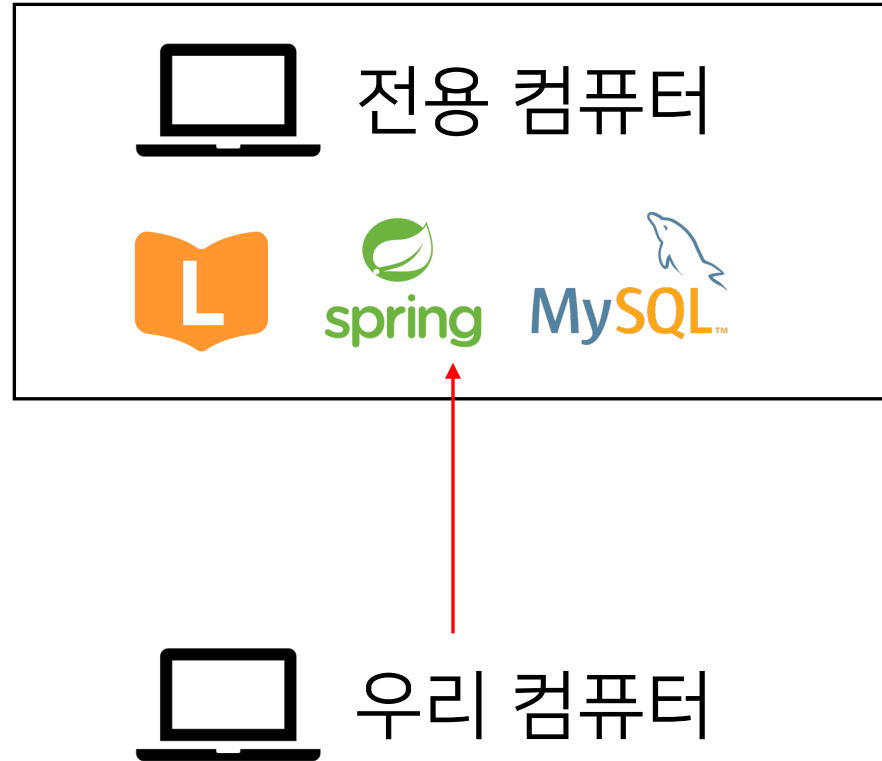
방법 3. 전용 컴퓨터에 코드를 옮기고
스프링, MySQL 등을 설치해 친구가 접속하게 한다.

친구 (최종 사용자)가 우리의 서버를 쓸 수 있는 방법

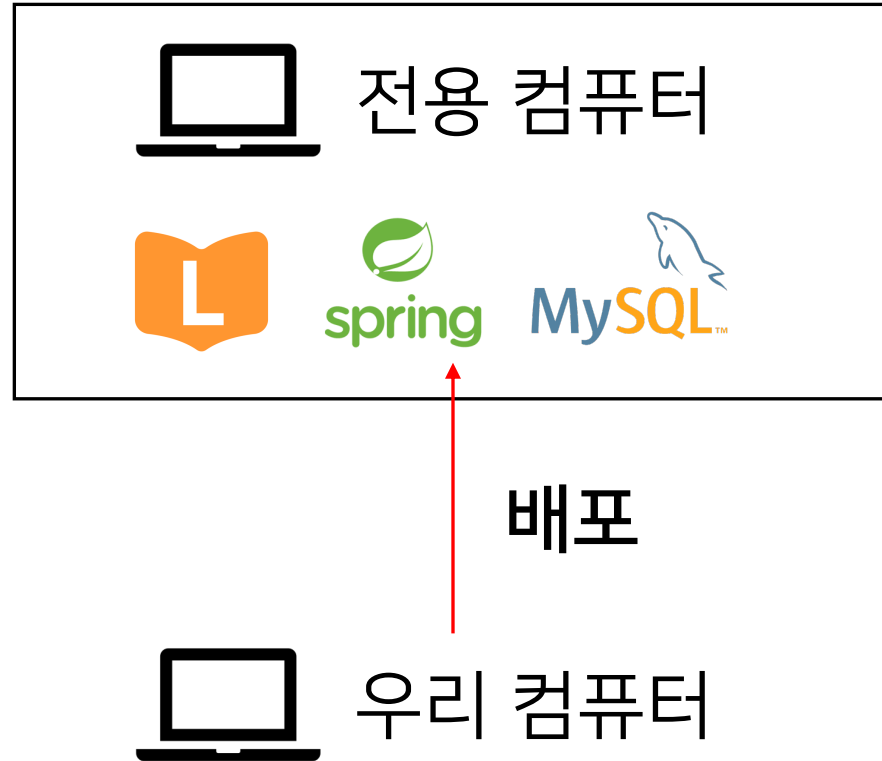


방법 3. 전용 컴퓨터에 코드를 옮기고
스프링, MySQL 등을 설치해 친구가 접속하게 한다.

친구 (최종 사용자)가 우리의 서버를 쓸 수 있는 방법



친구 (최종 사용자)가 우리의 서버를 쓸 수 있는 방법



배포

최종 사용자에게 SW를 전달하는 과정
= 전용 컴퓨터에 우리의 서버를 옮겨 실행시키는 것

우리는 전용 컴퓨터가 없는데요?



미국 쇼핑몰 아마존이 운영하는
AWS (Amazon Web Service)

AWS에서 무료로 전용 컴퓨터를 빌리자!



전용 컴퓨터



배포



우리 컴퓨터

AWS에서 컴퓨터를 빌릴 때 한 가지 알아두어야 할 점!

우리가 컴퓨터를 살 때 **운영체제(OS)**도 같이 선택하죠!



1985



1992-1994



1995-2000



2001-2005



2006-2008



2009-2012



current

Linux™



서버용 컴퓨터는 보통 리눅스를 사용합니다!



1985



1992-1994



1995-2000



2001-2005



2006-2008



2009-2012

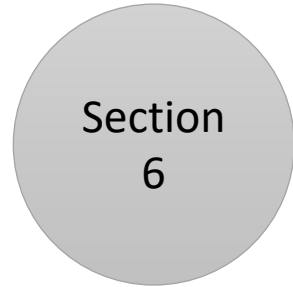


current

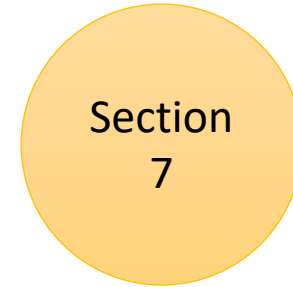
Linux™



AWS를 이용한 간단 배포



배포 준비



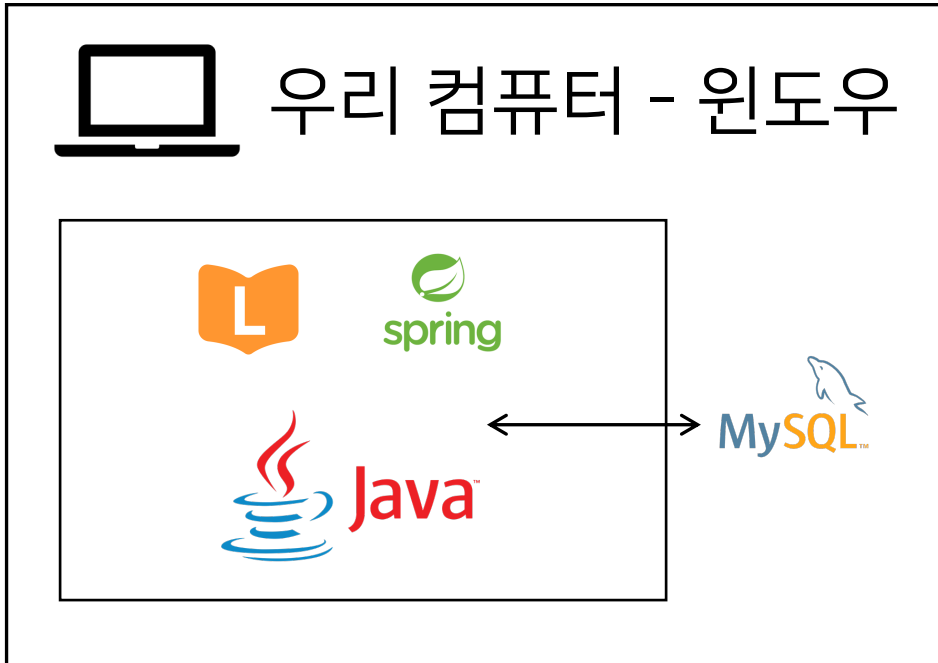
배포 후 자랑

38강. profile과 H2 DB

지난 시간에 '배포'에 대해 알아보았다!



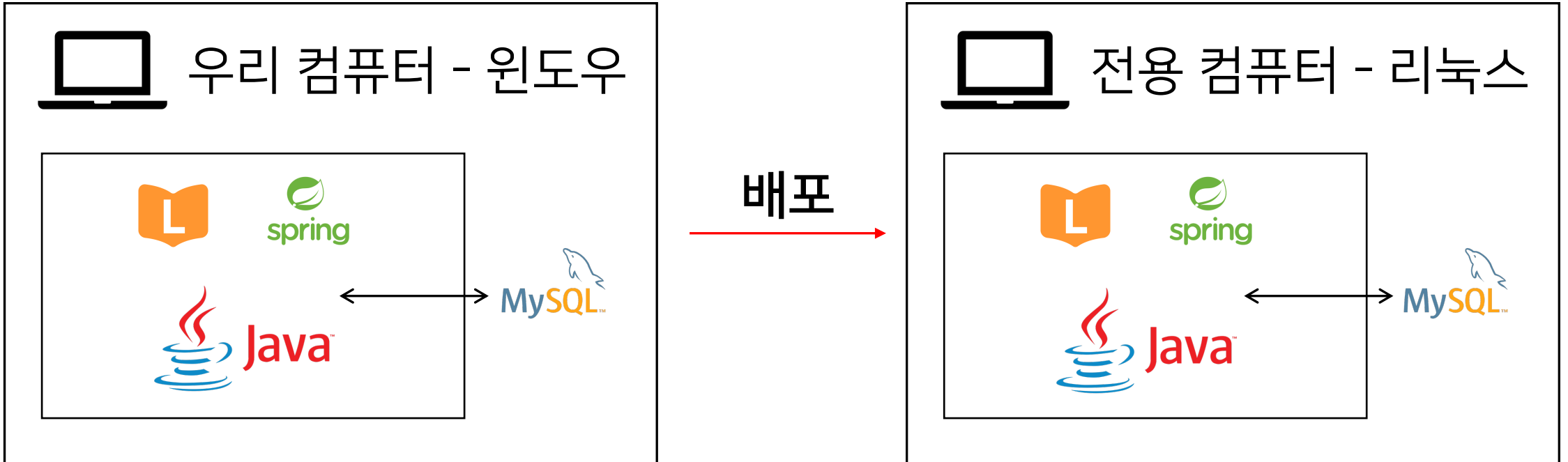
이걸 자세히 보면...



우리 컴퓨터(윈도우)에 설치한
java / mysql을 실행

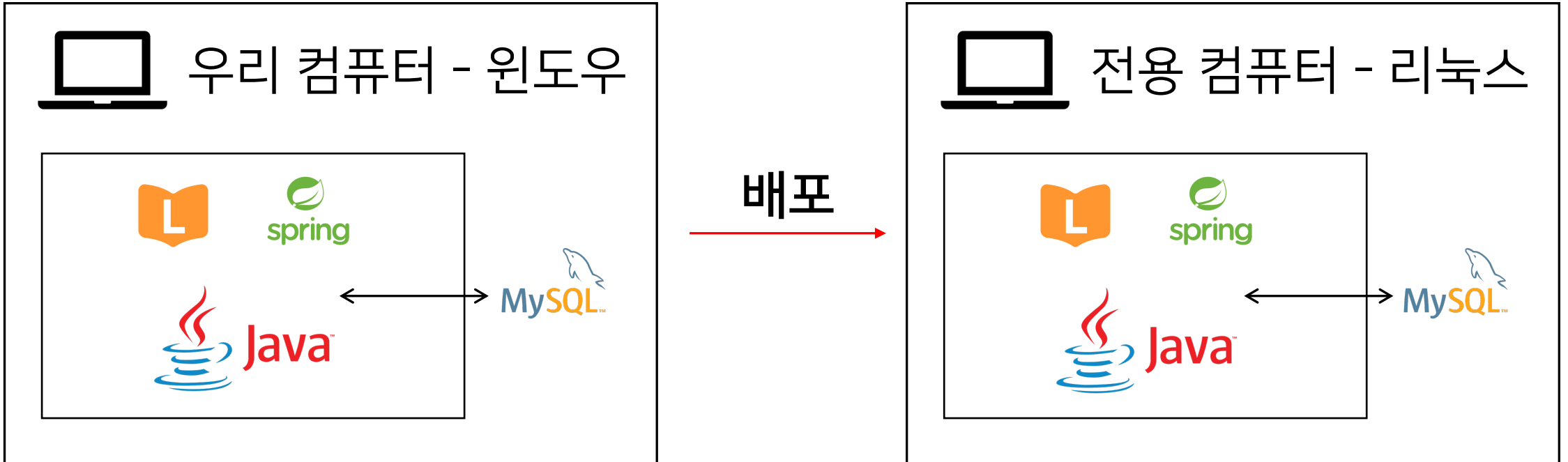
→ 우리 컴퓨터에 있는
mysql을 사용

이걸 자세히 보면...



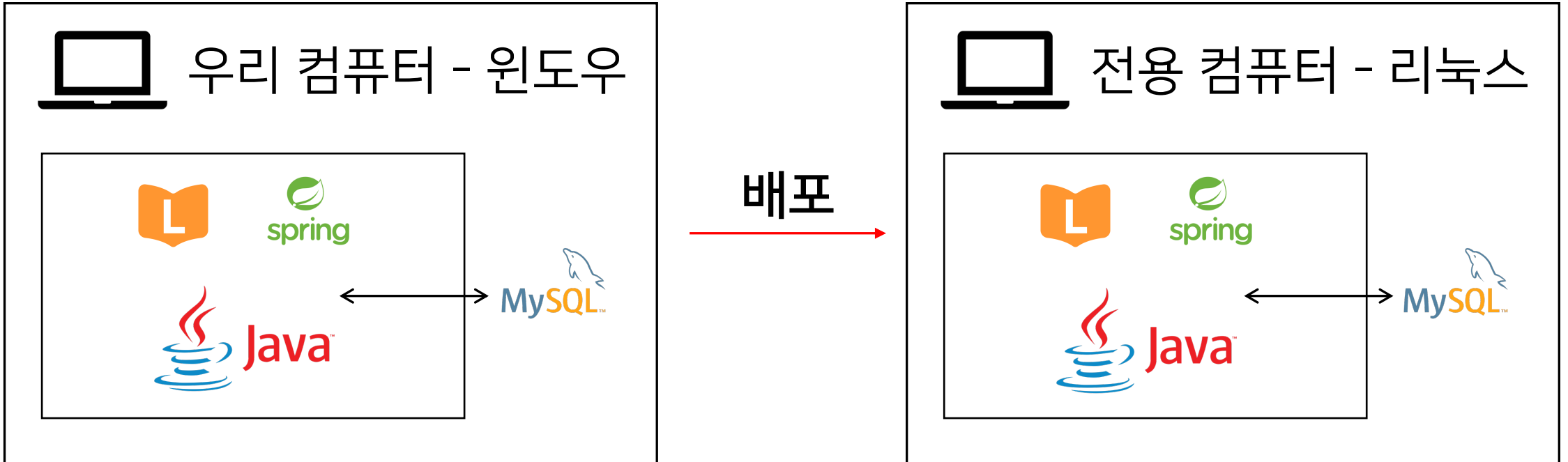
전용 컴퓨터(리눅스)에 설치한 java / mysql을 실행

이걸 자세히 보면...



전용 컴퓨터에서 실행된 서버는 전용 컴퓨터에 있는 mysql을 사용

이걸 자세히 보면...



우리 컴퓨터에서 사용할 때는 우리 컴퓨터의 MySQL
전용 컴퓨터에서 사용할 때는 전용 컴퓨터의 MySQL

똑같은 서버 코드를 실행시키지만,
우리 컴퓨터에서 실행할 때는 우리 컴퓨터의 MySQL
전용 컴퓨터에서 실행할 때는 전용 컴퓨터의 MySQL

여기서 profile이라는 개념이 등장합니다!

똑같은 서버 코드를 실행시키지만,
실행될 때 **설정**을 다르게 하고 싶다!

여기서 profile이라는 개념이 등장합니다!

똑같은 서버 코드를 실행시키지만,
실행될 때 **설정**을 다르게 하고 싶다!

지금은 DB이지만, 다양한 종류가 있다!

지금도 이 profile이란 것을 사용하고 있었다!

Profile을 적용해보자!

똑같은 서버 코드를 실행시키지만,

local 이라는 profile을 입력하면, **H2 DB**를 사용하고

dev 라는 profile을 입력하면 MySQL DB를 사용하게 바꾸자.

H2 DB란?!

경량 Database로, 개발 단계에서 많이 사용하며
디스크가 아닌 메모리에 데이터를 저장할 수 있다.

메모리에 데이터를 저장하면 휘발되지 않나요?!

네 맞습니다!

메모리에 데이터를 저장하면 휘발되지 않나요?!

그래서 개발단계에서만 사용한다!

메모리에 데이터를 저장하면 휘발되지 않나요?!

또한, 개발 단계에서는 테이블이 계속 변경되는데

어차피 데이터가 휘발되기 때문에 ddl-auto 옵션을 create로 주면
테이블을 신경쓰지 않고 코드에만 집중할 수 있다!

개발 단계에서는 데이터가 휘발되는 H2가 오히려 장점!

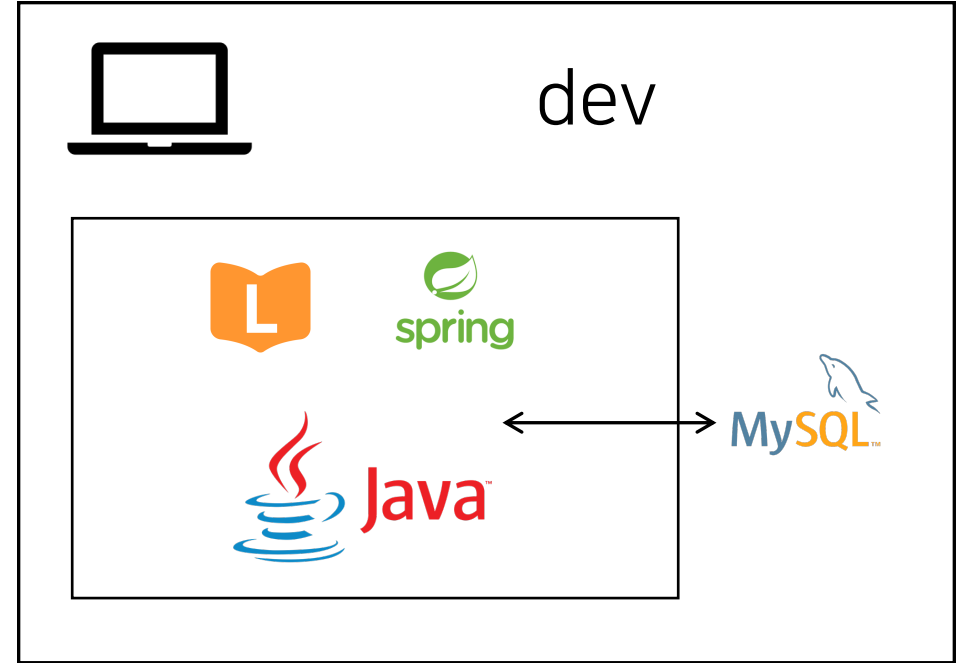
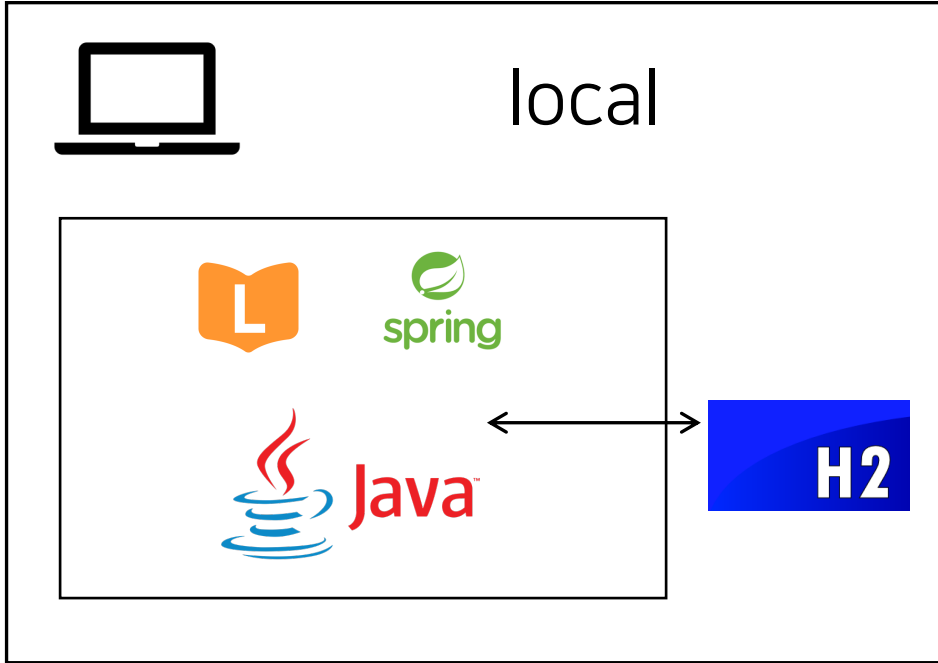
Profile을 적용해보자!

똑같은 서버 코드를 실행시키지만,

local 이라는 profile을 입력하면, **H2 DB**를 사용하고

dev 라는 profile을 입력하면 MySQL DB를 사용하게 바꾸자.

Profile을 적용해보자!



39강. git과 github이란 무엇인가?!

개발 공부를 시작해 이런저런 검색을 하다 보면...

git?

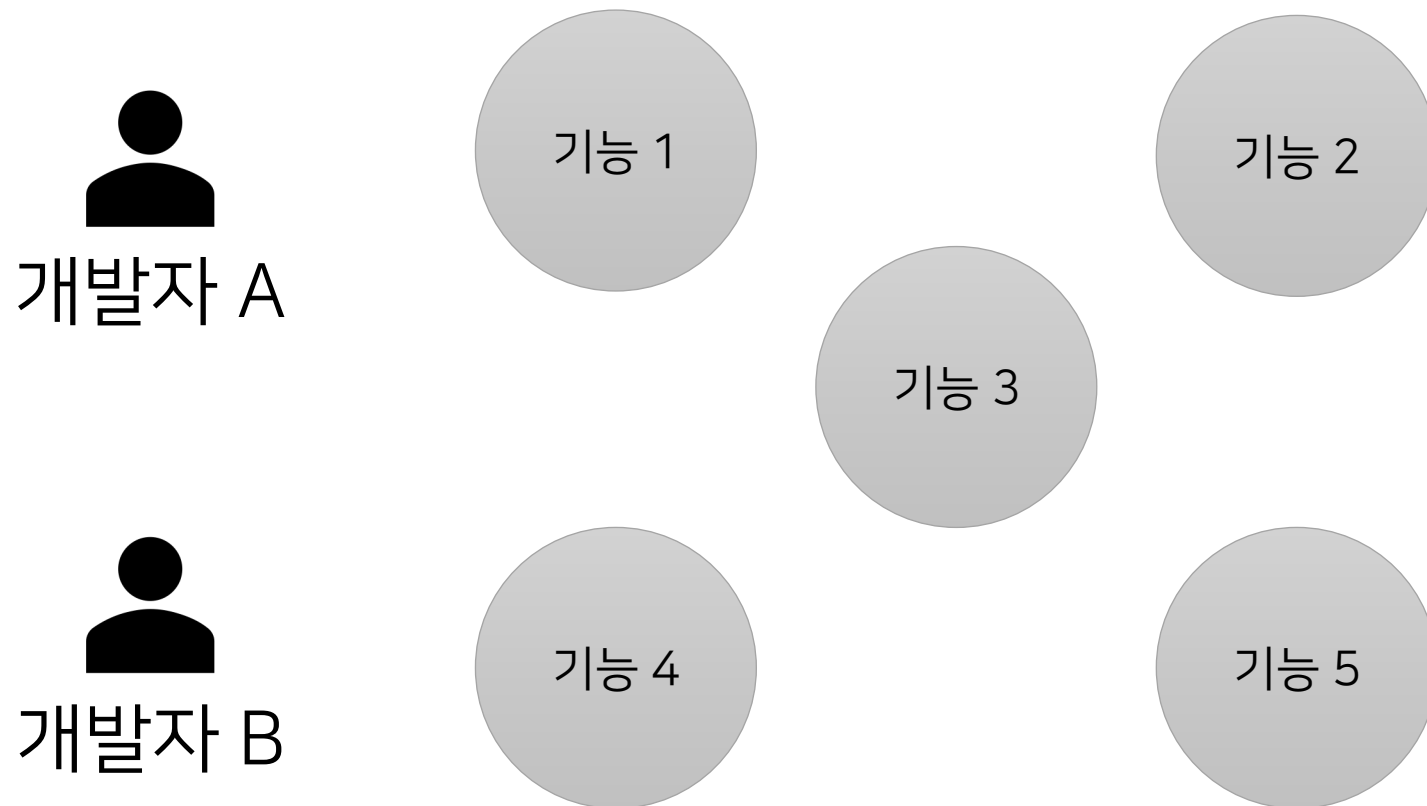


github?

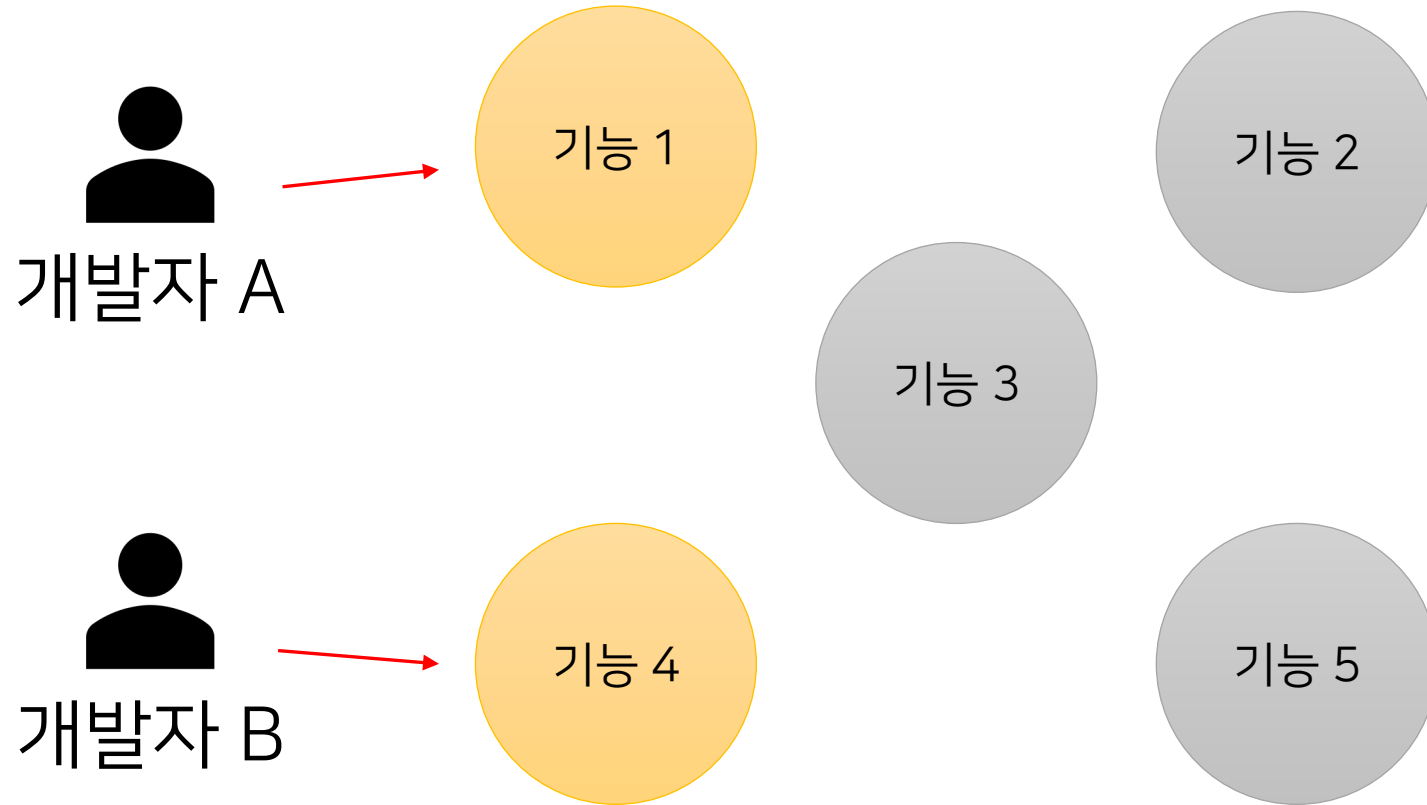
git이란 무엇인가?!

코드를 쉽게 관리할 수 있도록 해주는 버전 관리 프로그램

다음과 같은 상황을 생각해보자!

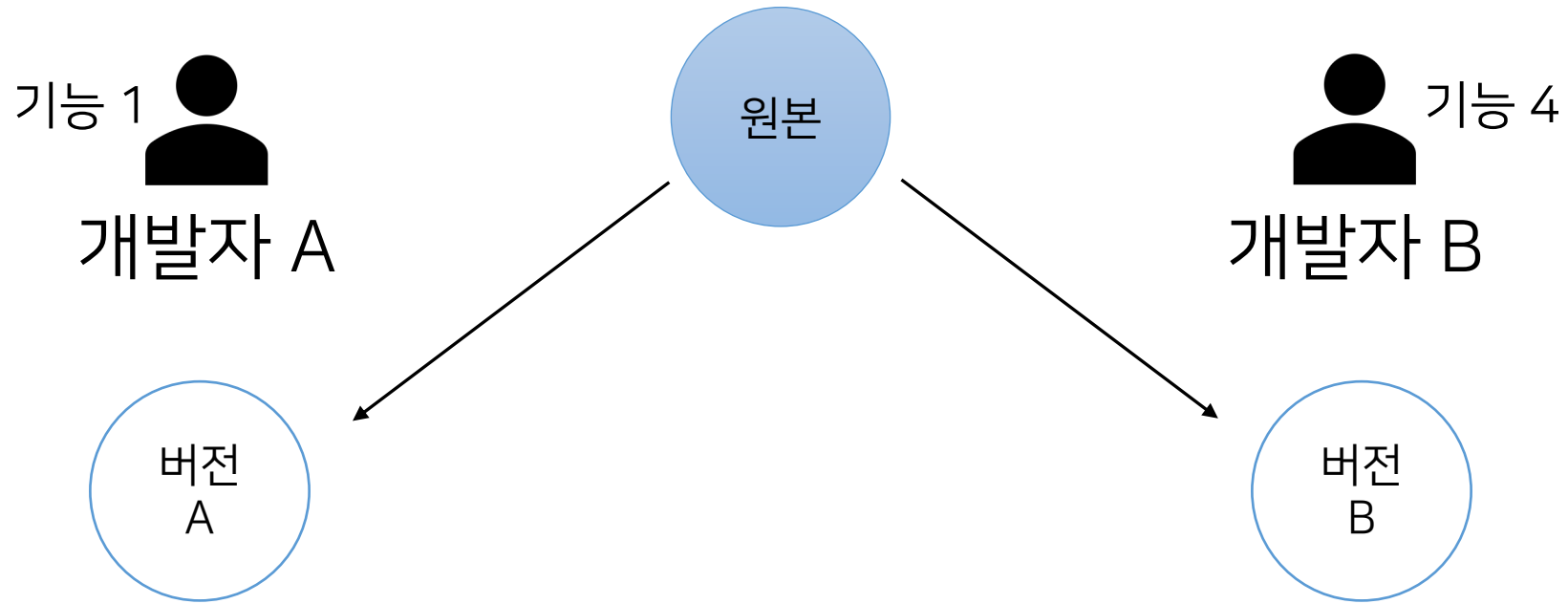


다음과 같은 상황을 생각해보자!

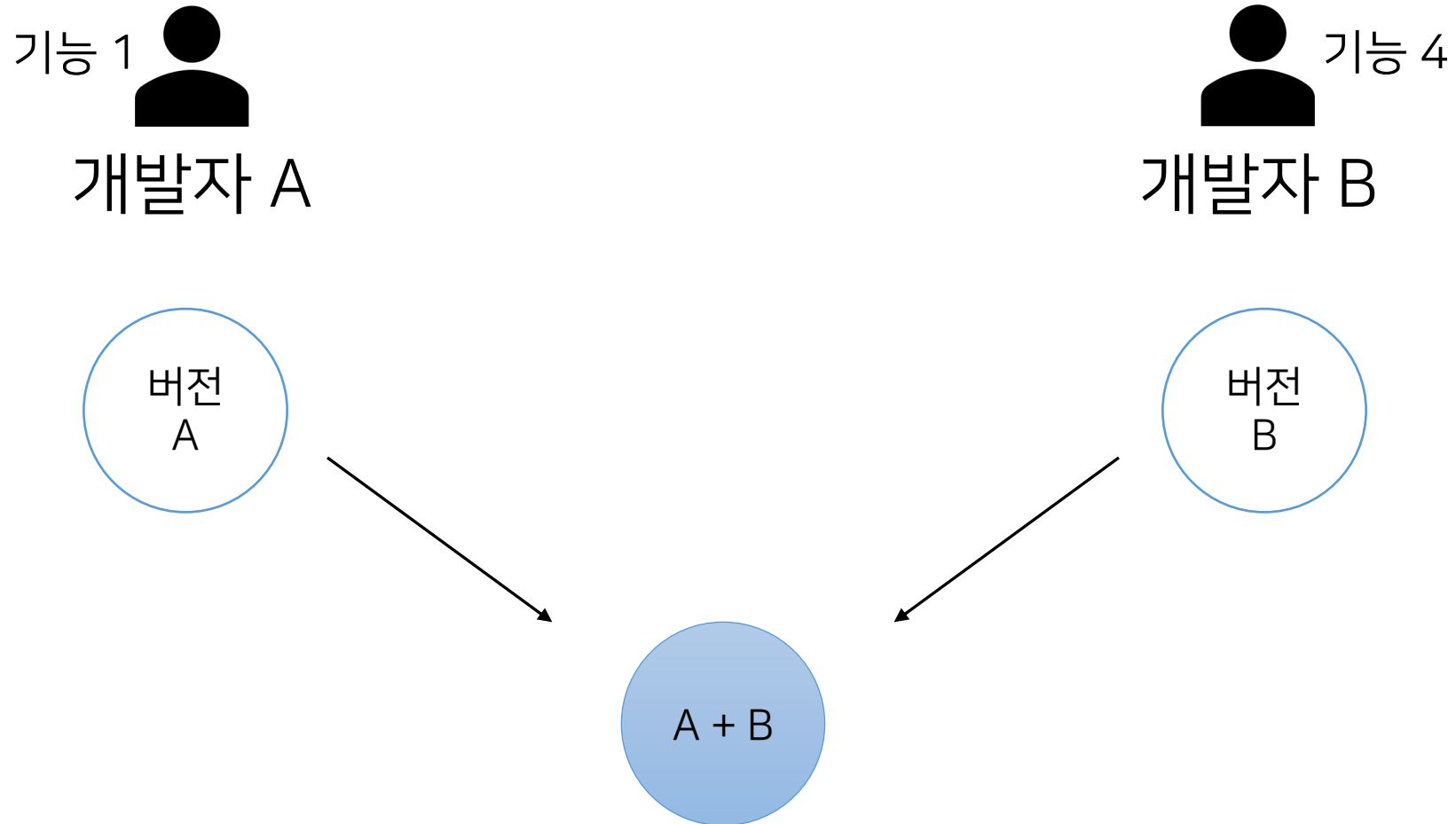


기능 1과 기능 4를 동시에 개발하자!

각자 개발하는 것까지는 좋다~



결국 하나로 합쳐야 하는데..?!



문제는 이제 생긴다...!

어떻게 합쳐야 하는가? - 서로 작업한 것만 복붙 할 수도 없고...

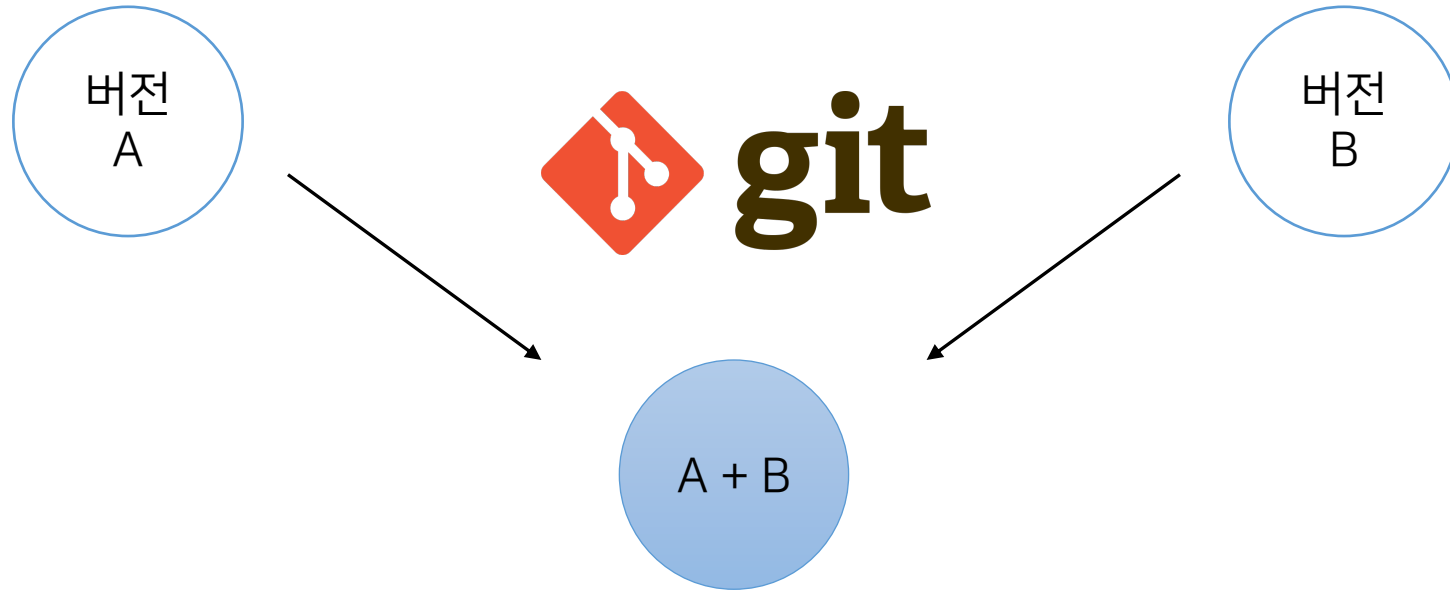
문제는 이제 생긴다...!

만약에 같은 파일, 같은 코드를 변경했다면?!

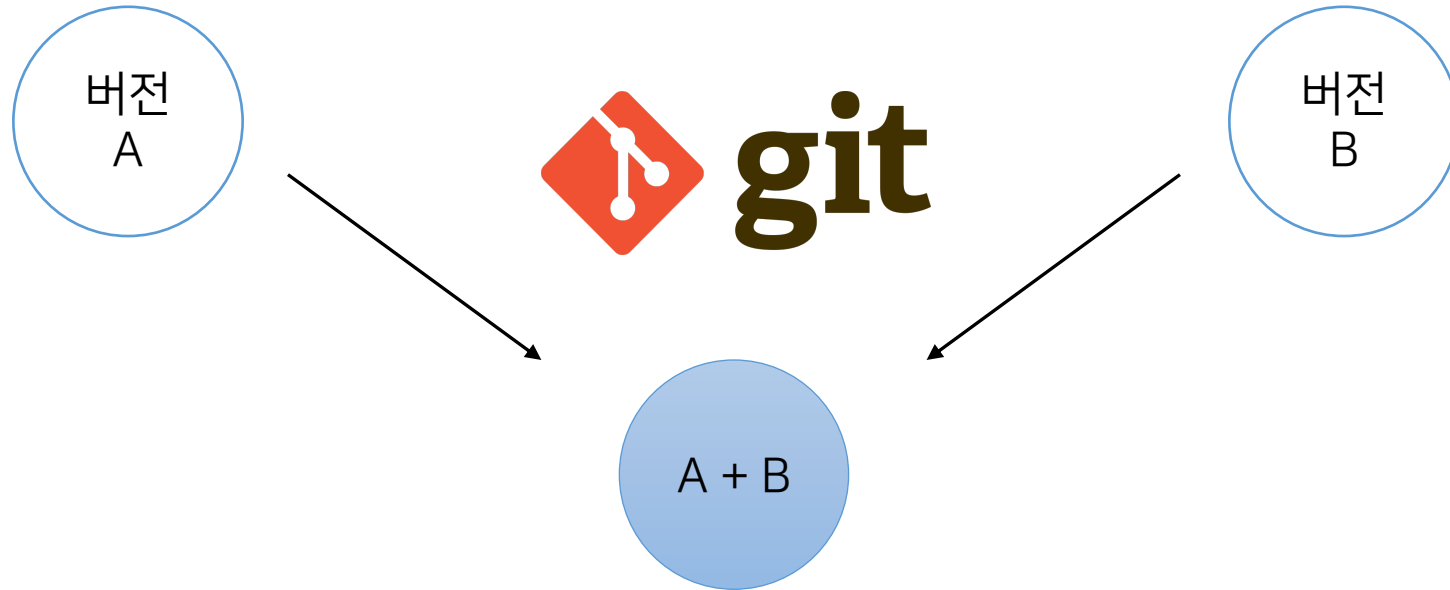
git 등장~

코드를 쉽게 관리할 수 있도록 해주는 버전 관리 프로그램

git 등장~



git 등장~



손쉽게 버전 A, 버전 B를 만들 수 있게 해주고~

git 등장~

2023기획서_컨펌

2023기획서_최종

2023기획서_최최종

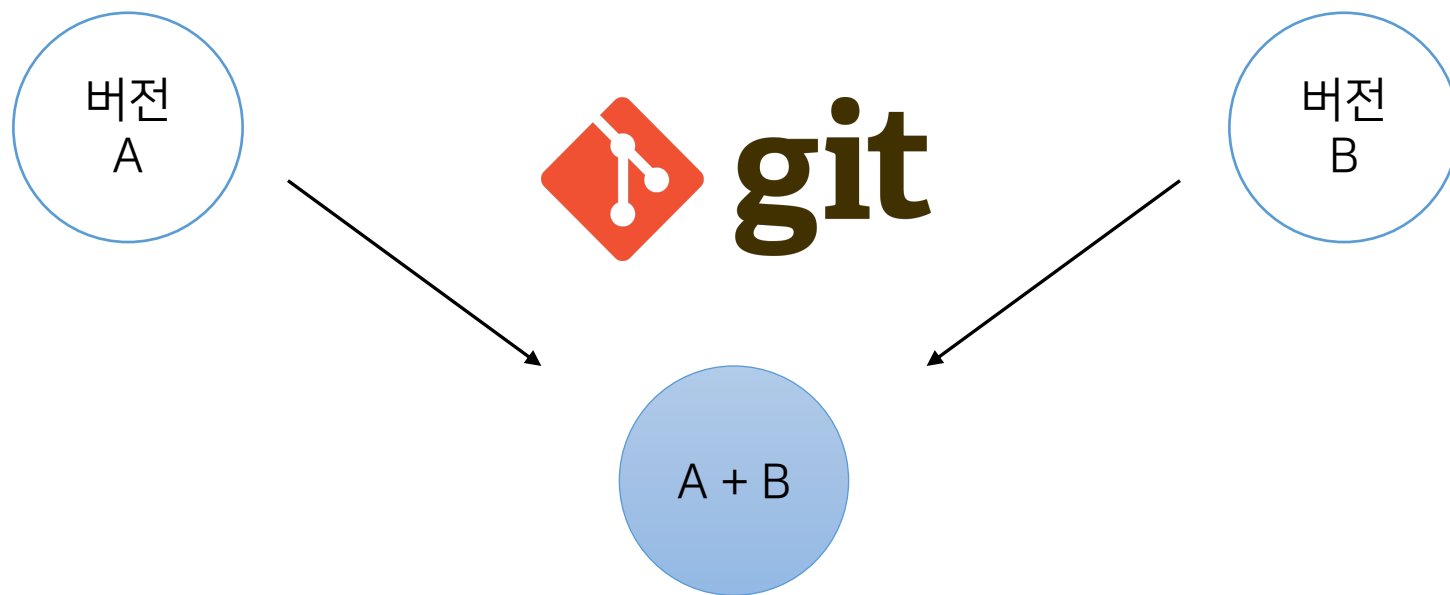
2023기획서_최최최종

2023기획서_진짜최종

2023기획서_정말정말최종

2023기획서_살려줘..

git 등장~



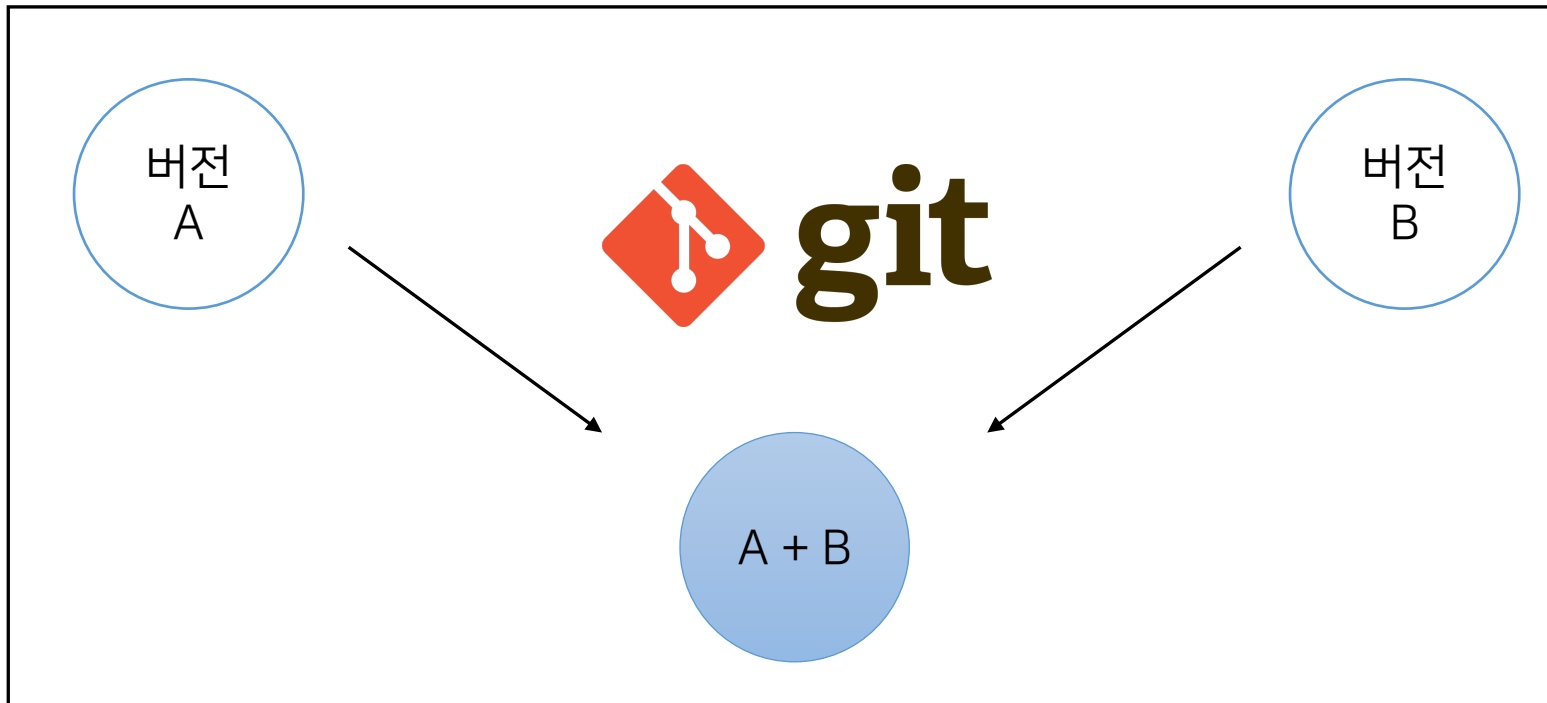
여러 버전을 손쉽게 합쳐준다!

github이란 무엇인가?!

git으로 관리되는 프로젝트의 코드가 저장되는 저장소

git이 관리 하는 프로젝트를 github에 저장할 수 있다!

서버 프로젝트



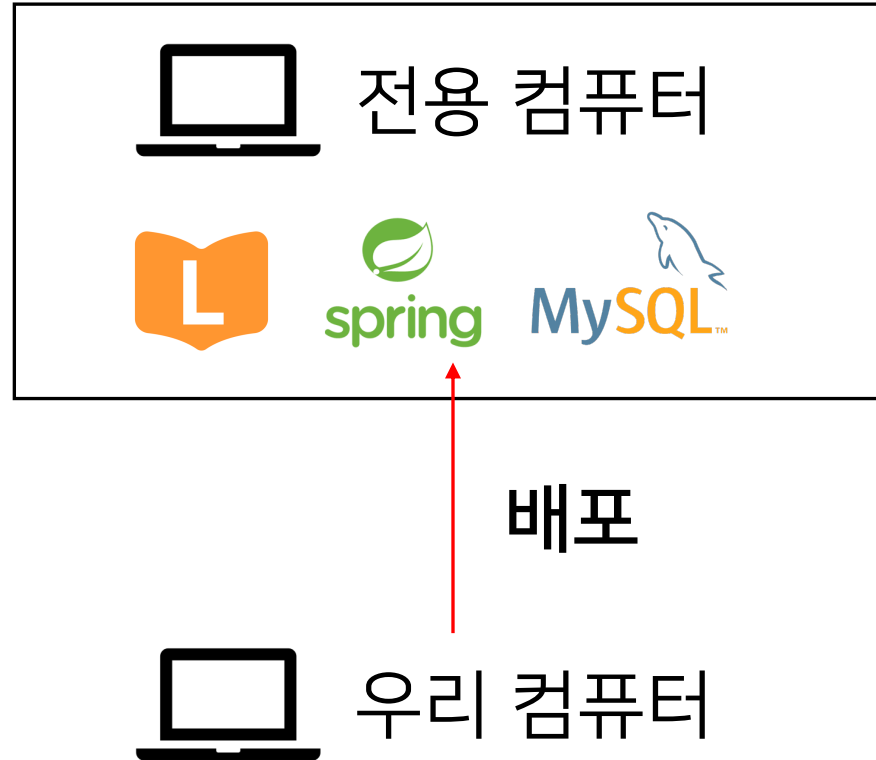
왜 github에 코드를 저장하는걸까?!

내 컴퓨터에 있는 코드는 모종의 이유로 소실 될 수 있다!

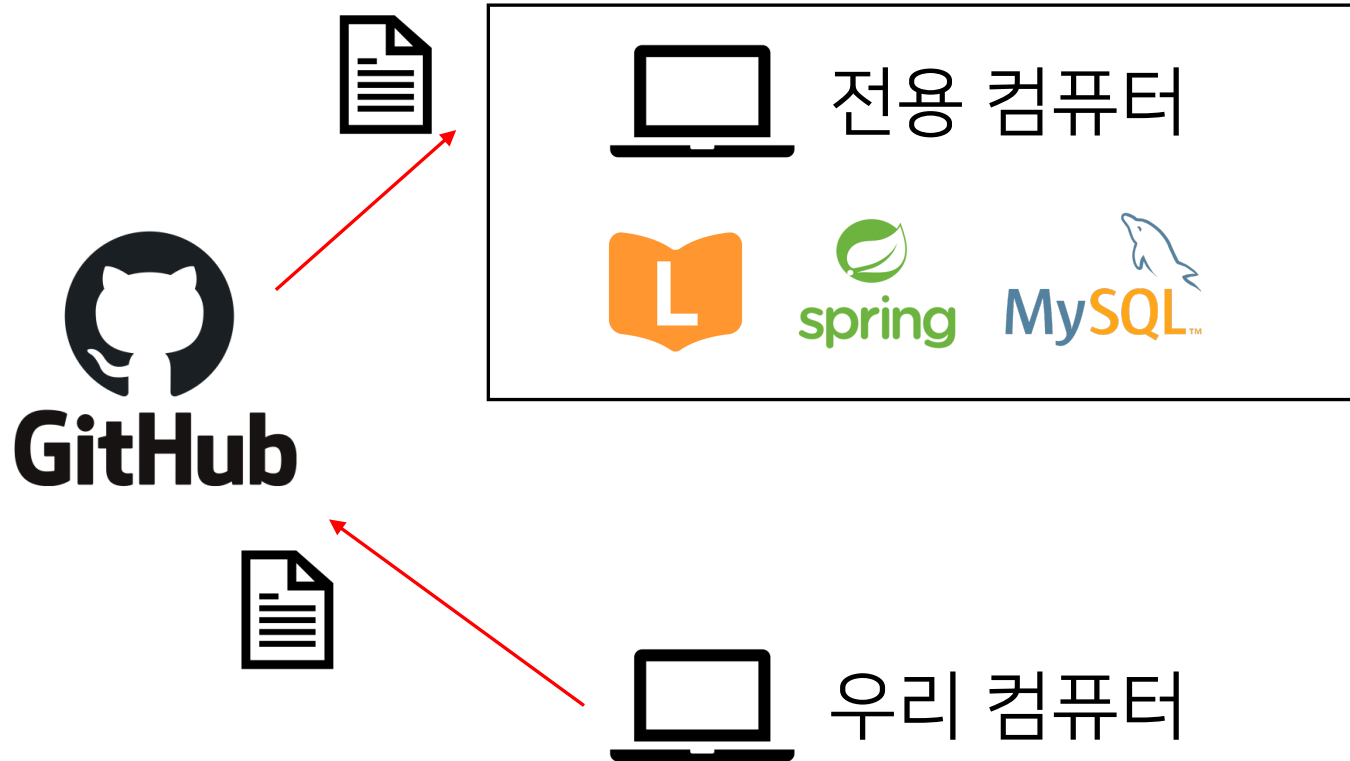
왜 github에 코드를 저장하는걸까?!

배포를 할 때 활용할 수 있다!

왜 github에 코드를 저장하는걸까?!



왜 github에 코드를 저장하는걸까?!



정리해보면

git : 코드를 쉽게 관리할 수 있도록 해주는 버전 관리 프로그램

github : git으로 관리되는 프로젝트의 코드가 저장되는 저장소

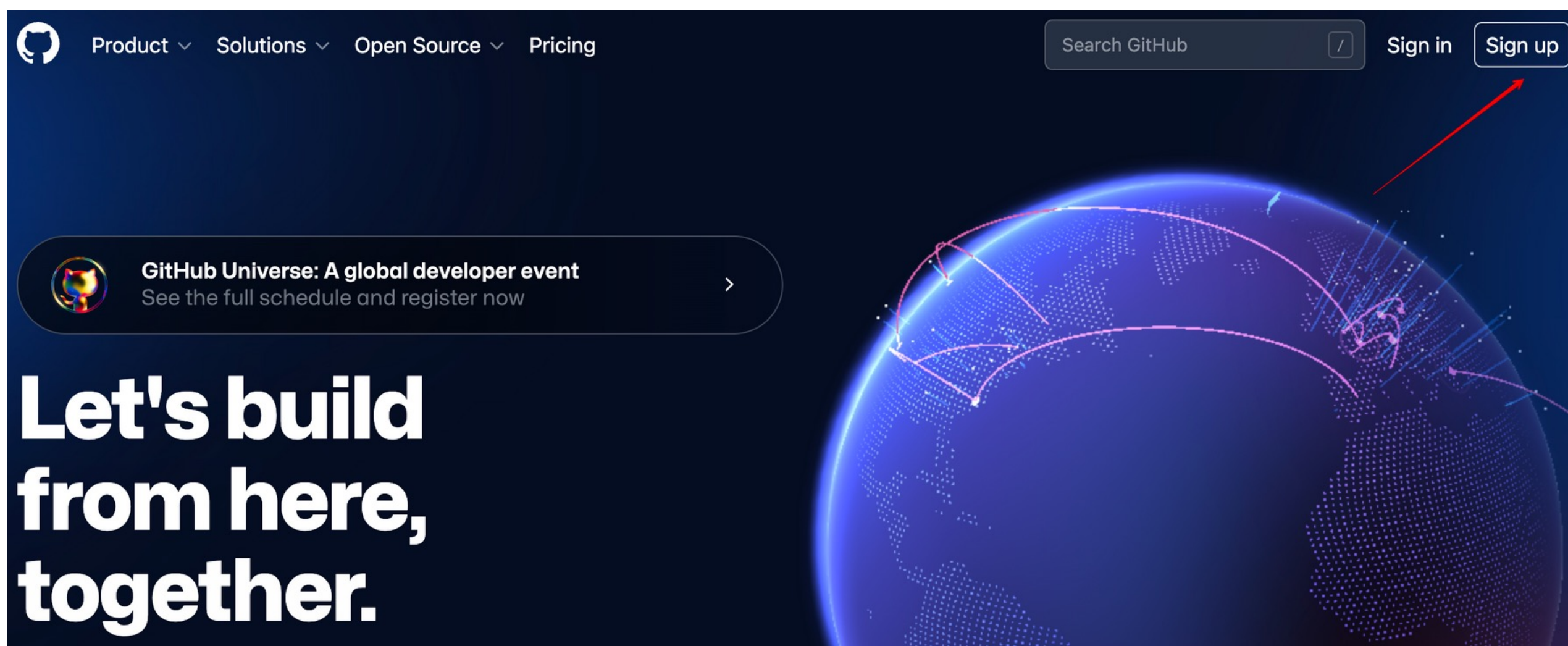
다음 시간에...

git을 이용해 github에 우리가 개발한

<도서 관리 애플리케이션>을 업로드 해보겠습니다!

1. github 사이트에 들어간다.

2. 우측 상단에 Sign Up 버튼을 누른다.



3. 이메일과 비밀번호, 별명을 입력한다.

Welcome to GitHub!
Let's begin the adventure

Enter your email

✓ studyingdeveloper@gmail.com

Create a password

✓ ••••••••

Enter a username

→ studying-developer

Continue

4. 로봇이 아닙니다 인증을 하고 계정을 생성한다!! 메일로 인증 번호가 오게 된다.

You're almost done!

We sent a launch code to `studyingdeveloper@gmail.com`

→ Enter code

Didn't get your email? [Resend the code](#) or [update your email address](#).

5. 메일에 온 숫자 코드를 입력하면 회원가입 완료!

40강. git 기초 사용법

git을 활용해 github에 우리 프로젝트를 업로드 하자!

github에 다들 가입하셨죠?!

자 그러면 먼저 github 저장소부터 만들어보겠습니다!

이제 github 저장소는 다 만들었구요~

IntelliJ Terminal을 이용해 git 명령어를 쓰겠습니다!

git 프로젝트 시작하기

git init

git 프로젝트 시작하기

git init : 이 프로젝트를 이제 git이 관리하겠다.

git 프로젝트의 github 저장소 설정하기

```
git remote add origin [각자의 주소]
```

이제 기초적인 셋팅은 모두 끝났습니다!

**이제 코드를 github에 올릴건데요,
택배를 보내는 것과 비슷합니다!**

택배 보내는 방법

1. 택배 상자를 가져와 물건을 담는다.
2. 택배 상자를 포장하고 송장을 붙인다.
이때 하고 싶은 말도 적을 수 있다.
3. 택배 상자를 우체국에 가서 부친다.


[1] 코드들을 텍배 상자에 담는 명령어

```
git add .
```

[1] 코드들을 텍배 상자에 담는 명령어

git add .

모든 파일



[1] 코드들을 텍배 상자에 담는 명령어

git add 파일이름

현재 상황 확인하기

git status

[2] 택배 상자에 송장을 붙이는 명령어

```
git commit -m "메시지"
```

혹시 다음과 같은 에러가 나온다면...!

```
*** Please tell me who you are.
```

Run

```
git config --global user.email "you@example.com"  
git config --global user.name "Your Name"
```


이메일과 이름을 설정해주자!!

```
git config --global user.email "studyingdeveloper@gmail.com"  
git config --global user.name "studying-developer"
```

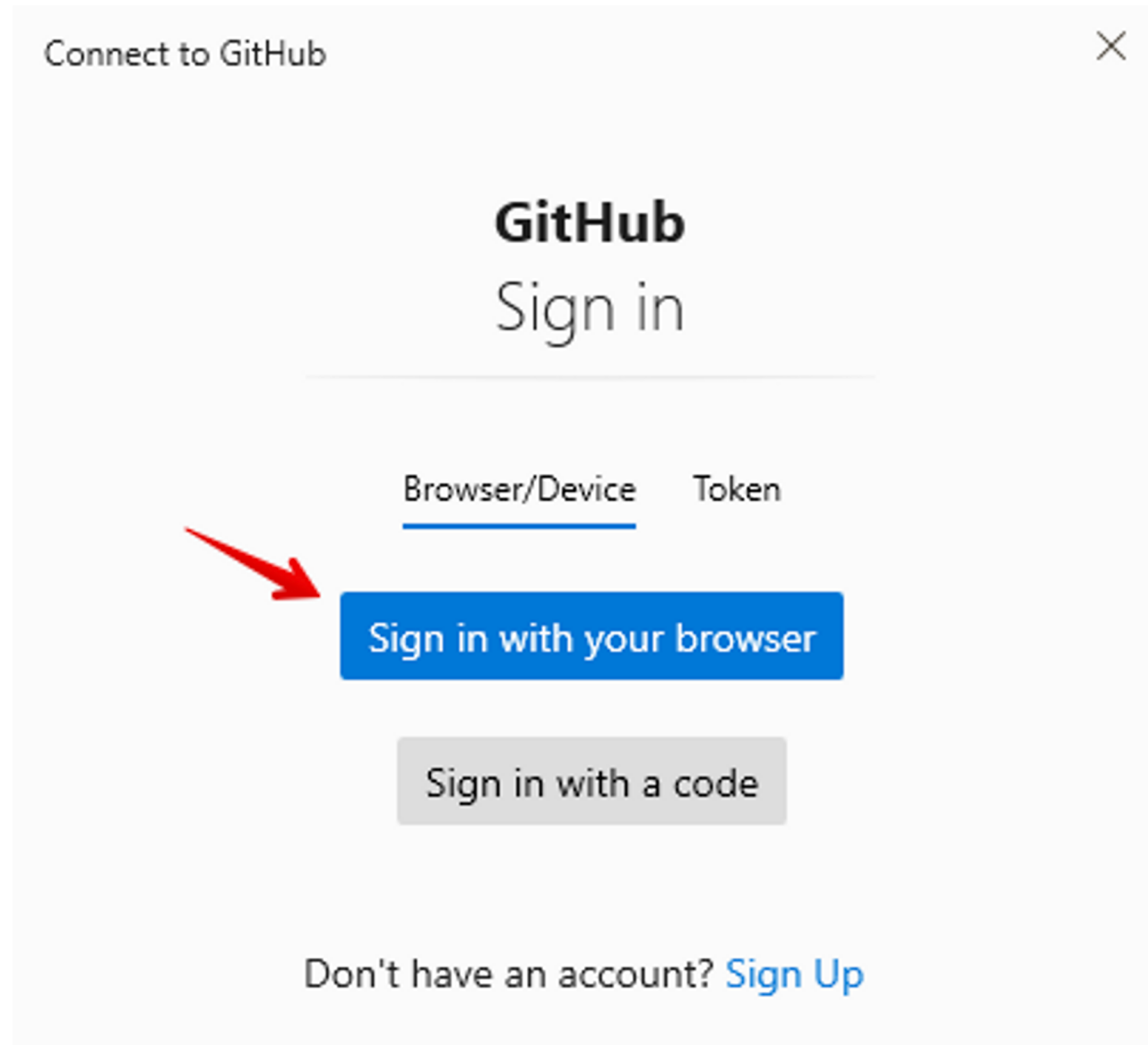
[3] 택배 상자를 github에 보내기

git push

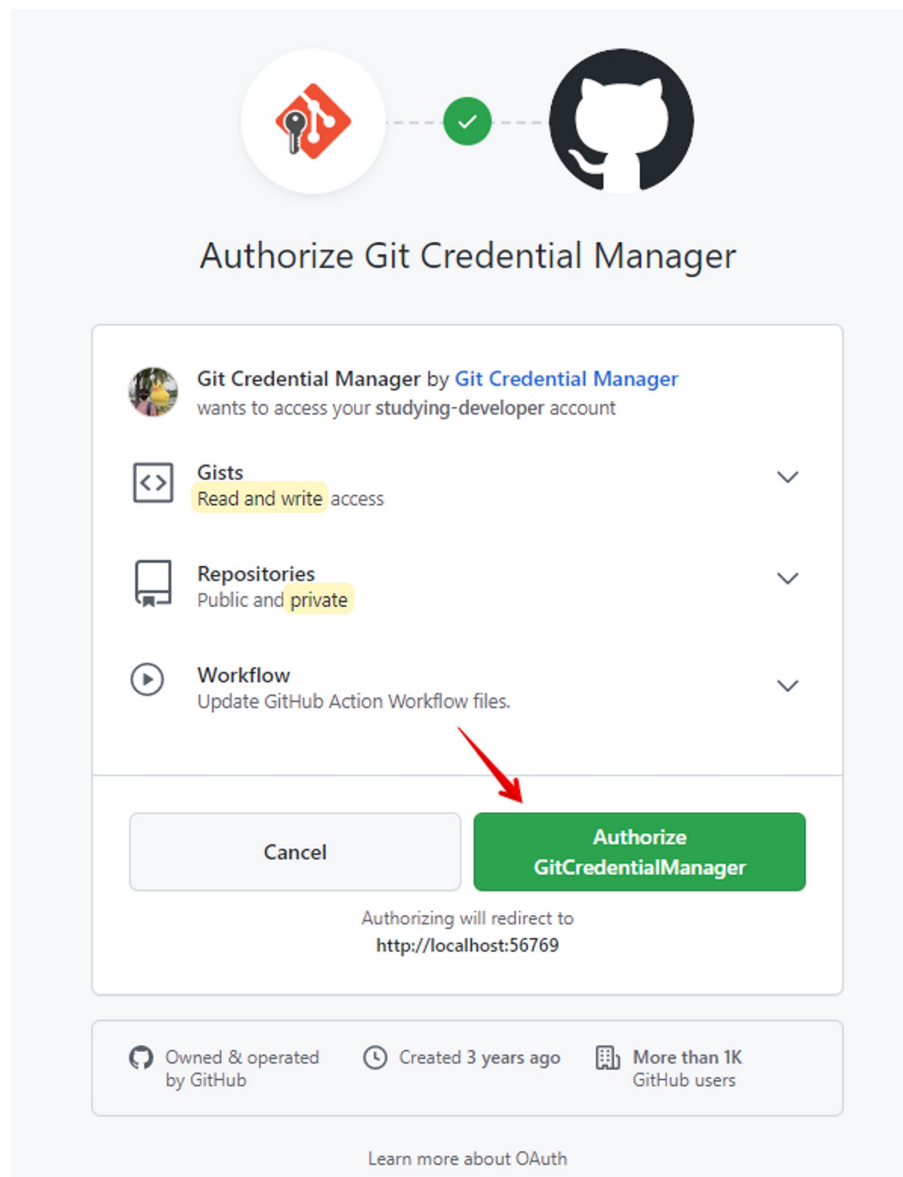
[3]택배 상자를 github에 보내기 (최초 1회)

```
git push -set-upstream origin master
```

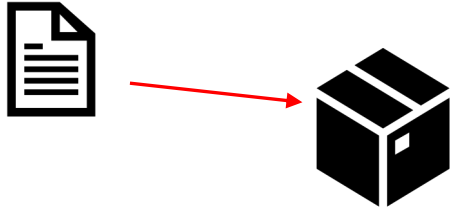
최초 push를 할 때에는 권한 허가가 필요하다.



최초 push를 할 때에는 권한 허가가 필요하다.

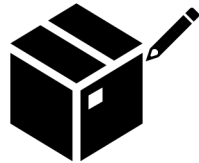


git 기초 명령어 정리



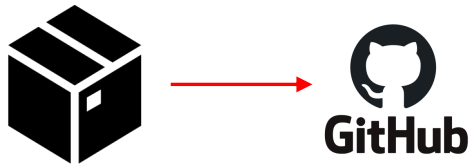
코드를 택배 상자에 담기

`git add .`



택배에 송장 붙이기

`git commit -m "적고 싶은 메시지"`



택배 상자 github에 보내기

`git push`

41강. AWS의 EC2 사용하기

AWS 가입 과정

AWS에 가입

연락처 정보

AWS를 어떻게 사용할 계획이신가요?

- ☐ 비즈니스 – 업무, 학교 또는 조직의 경우
- ☐ 개인 – 자체 프로젝트의 경우

이 계정에 대해 누구에게 문의해야 하나요?

전체 이름

전화 번호

 +1 ▼

222-333-4444

국가 또는 리전

미국 ▼

주소

아파트, 동, 호수, 빌딩, 층 등

시

시, 도 또는 리전

우편 번호

☐ [AWS 이용약관](#) 을 읽었으며 이에 동의합니다.

계속(2/5단계)

AWS 가입 과정

결제 정보

신용카드 번호



AWS는 현지에서 발급된 대부분의 신용카드를 허용합니다. 결제 옵션에 대해 자세히 알아보려면 [FAQ](#)를 참조하세요.

만료 날짜

카드 소유자 이름

청구지 주소

☒ 내 연락처 주소 사용

☐ 새 주소 사용

이메일 주소

이메일 주소는 AWS와의 거래를 위해 VAT 영수증을 발송하는데 사용됩니다.

확인 및 계속(3/5단계)

확인 요금을 승인하기 위해 은행의 웹 사이트로 리디렉션될 수 있습니다.

AWS 가입 과정

자격 증명 확인

AWS 계정을 사용하려면 먼저 전화번호를 확인해야 합니다. 계속하면 AWS 자동 시스템이 확인 코드 전송을 위해 연락합니다.

확인 코드를 어떻게 보내 드릴까요?

☒ 문자 메시지(SMS)


☐ 음성 통화

국가 또는 리전 코드

미국 (+1) ▼

휴대전화 번호

보안 검사

| | |
|--|---|
|  | 🔊 |
| | ↺ |

위에 보이는 문자를 입력하세요.

SMS 전송(4/5단계)

AWS 가입 과정

Support 플랜 선택

비즈니스 또는 개인 계정에 대한 Support 플랜을 선택합니다. [플랜 및 요금 예시를 비교](#) 해 보세요. 언제든지 AWS Management Console에서 플랜을 변경할 수 있습니다.

☒ 기본 지원 - 무료

- AWS를 처음 시작하는 신규 사용자에게 권장
- AWS 리소스에 대한 연중 무휴 24시간 셀프 서비스 액세스
- 계정 및 청구 문제 전용
- Personal Health Dashboard 및 Trusted Advisor에 대한 액세스



☐ 개발자 지원 - 시작가는 29 USD/월

- AWS를 체험해보는 개발자에게 권장
- 업무 시간 중 AWS Support에 대한 이메일 액세스
- 12시간(업무 시간 기준) 이내의 응답 시간



☐ 비즈니스 지원 - 시작가는 100 USD/월

- AWS 기반 프로덕션 워크로드 실행에 추천
- 이메일, 전화 및 채팅을 통한 연중무휴 24시간 기술 지원
- 1시간 이내의 응답 시간
- Trusted Advisor 모범 사례 권장 사항 전체 세트



엔터프라이즈 수준의 지원이 필요하신가요?

최저 월 15,000 USD로 15분 이내에 응답을 받을 수 있으며 기술 지원 관리자가 배정된 컨시어지 스타일의 서비스를 이용할 수 있습니다. [자세히 알아보기](#)

가입 완료

AWS 가입이 완료되었다면, 로그인해 들어가자!

이제 AWS로부터 우리의 컴퓨터를 빌릴 것이다!

42강. Section 6 정리. 다음으로!

Section 6. 생애 최초 배포 준비하기

1. 배포가 무엇인지 이해하고, 배포를 하기 위해 어떤 준비를 해야 하는지 알아 본다.
2. 스프링 서버를 실행할 때 DB와 같은 설정들을 코드 변경 없이 제어할 수 있는 방법을 알아본다.

Section 6. 생애 최초 배포 준비하기

3. git과 github의 차이를 이해하고
git에 대한 기초적인 사용법을 알아본다.
4. AWS의 EC2가 무엇인지 이해하고,
AWS를 통해 클라우드 컴퓨터를 빌려본다.

감사합니다