

git과 github에 대해서 알아보자.

김우정

git 분산형 버전관리 시스템

분산형 버전관리 시스템


원가 한국말로 해도 어렵다.

일단 버전이란 무엇인가

일단 버전이란 무엇인가

version 

★ Top 1000 frequently used words

Pronunciation: /'və:ʃ(ə)n/ 

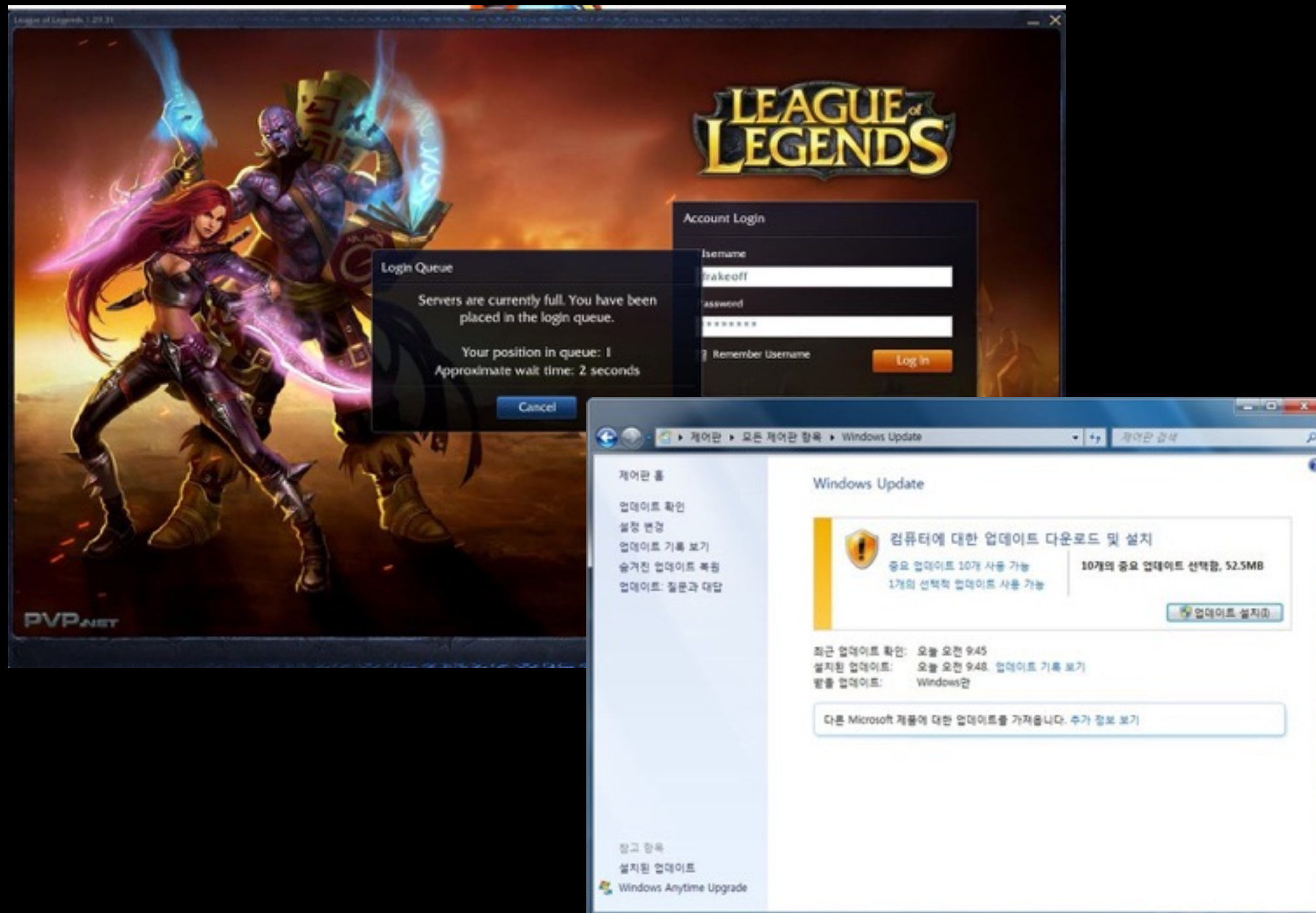
NOUN

- 1 A particular form of something differing in certain respects from an earlier form or other forms of the same type of thing:

[version](#) 미국식 [ˈvɜːrʒn]  영국식 [ˈvɜːʃn; -ʒn]  ★★

1. (이전의 것·비슷한 종류의 다른 것들과 약간 다른) -판

일단 버전이란 무엇인가



그럼 버전은 개발자들의 용어?

LL

그럼 버전은 개발자들의 용어?

버전
어떤 작업을 하고 있는데
수정된 새로운 판

그럼 버전관리란 무엇인가

그럼 버전관리란 무엇인가

일정한 규칙으로 수정된
복사본을 관리하는것
우리 이미 많이하고 있다.

그럼 버전관리란 무엇인가



그럼 버전관리란 무엇인가

수정본이 많아지고
다른 사람이 수정하고
실수가 많아짐
버전관리의 실패 ㅠㅠ

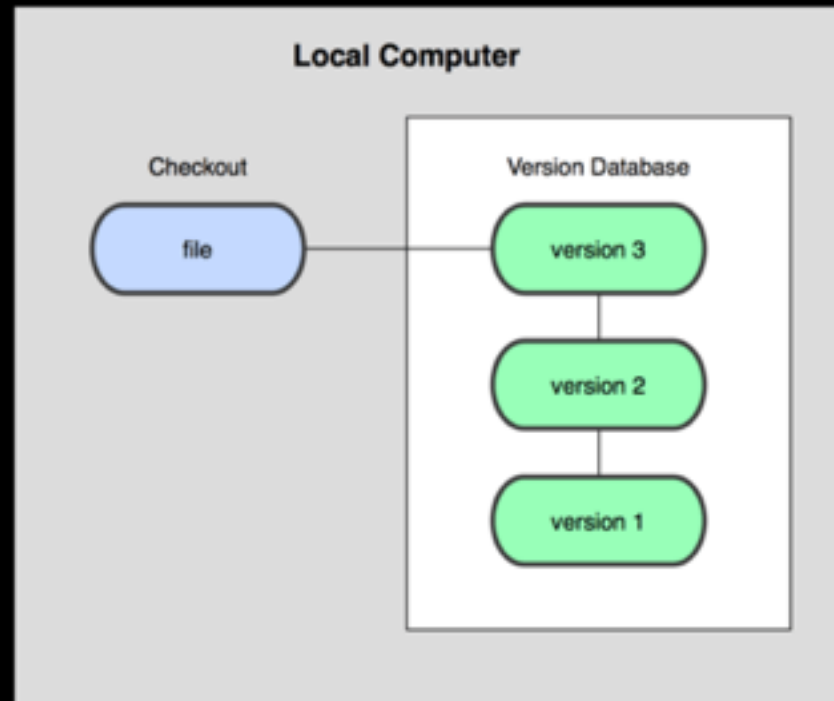
그럼 버전관리란 무엇인가

그래서 등장한게
버전관리 시스템

ㅇㅋ 이제 어느정도 이해가
된다 그럼 분산형은 뭔데?

잠시 노잼 이야기

VCS



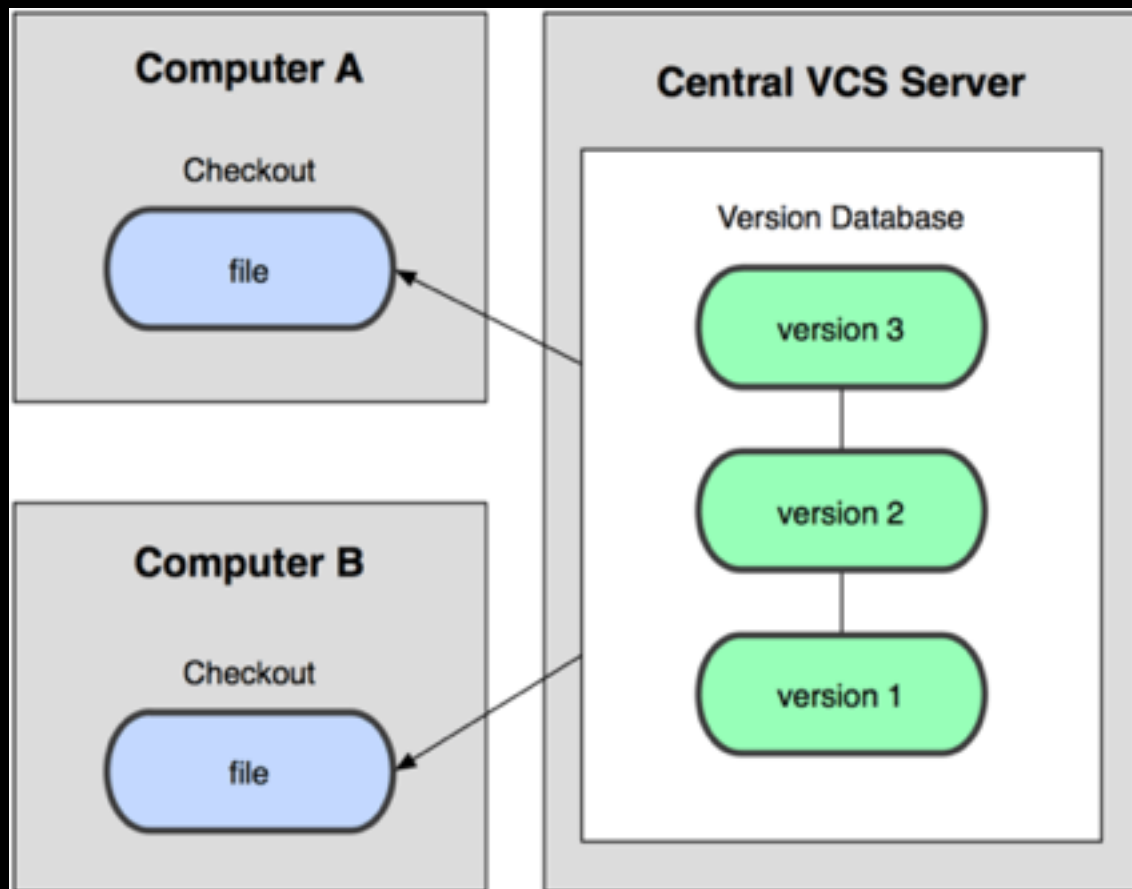
요즘도 많이 쓰이는 버전관리 시스템

장점 : 파일의 수정내역과 해당 버전으로 되돌리기가 용이함

단점: 다른 사람들과 작업을 못함

왜? 내컴퓨터에서만 저장되니까.

CVCS



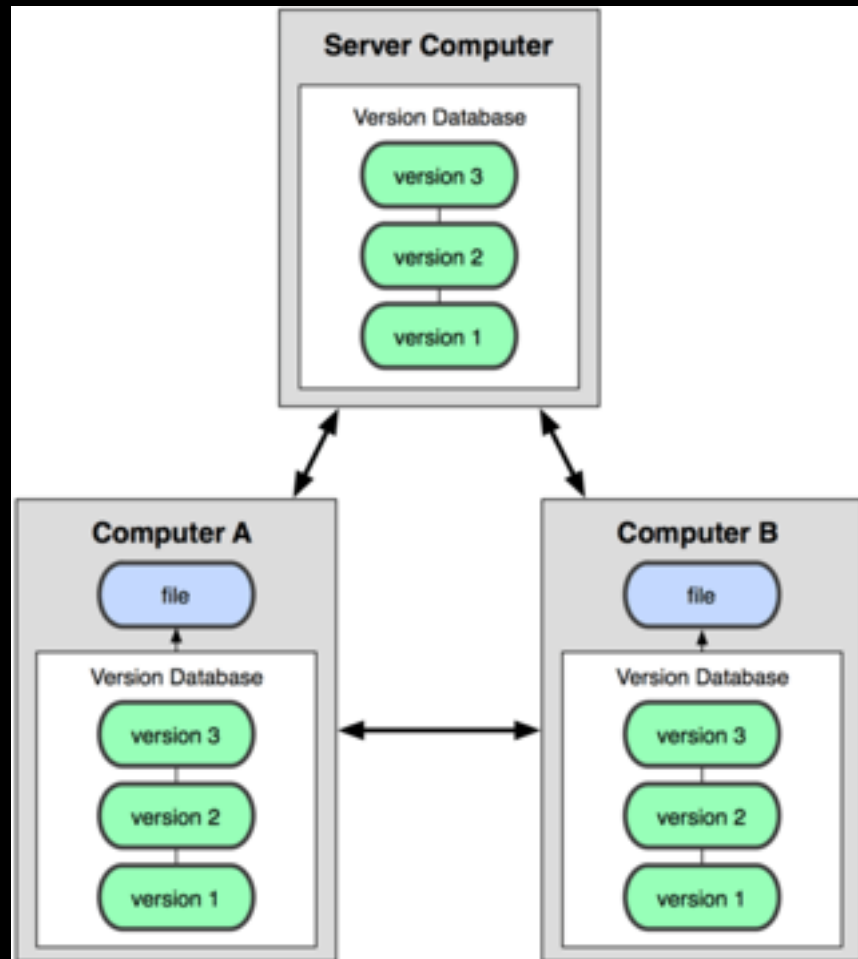
subversion이 좋은 예

장점 : 다른 사람들이랑 작업
할 수 있고 관리자가 버전관
리를 용이하게 할 수 있음

단점 : 중앙 시스템이 망하면
모두가 망함

그래서 등장한 것이

Distributed Version Control System; DVCS

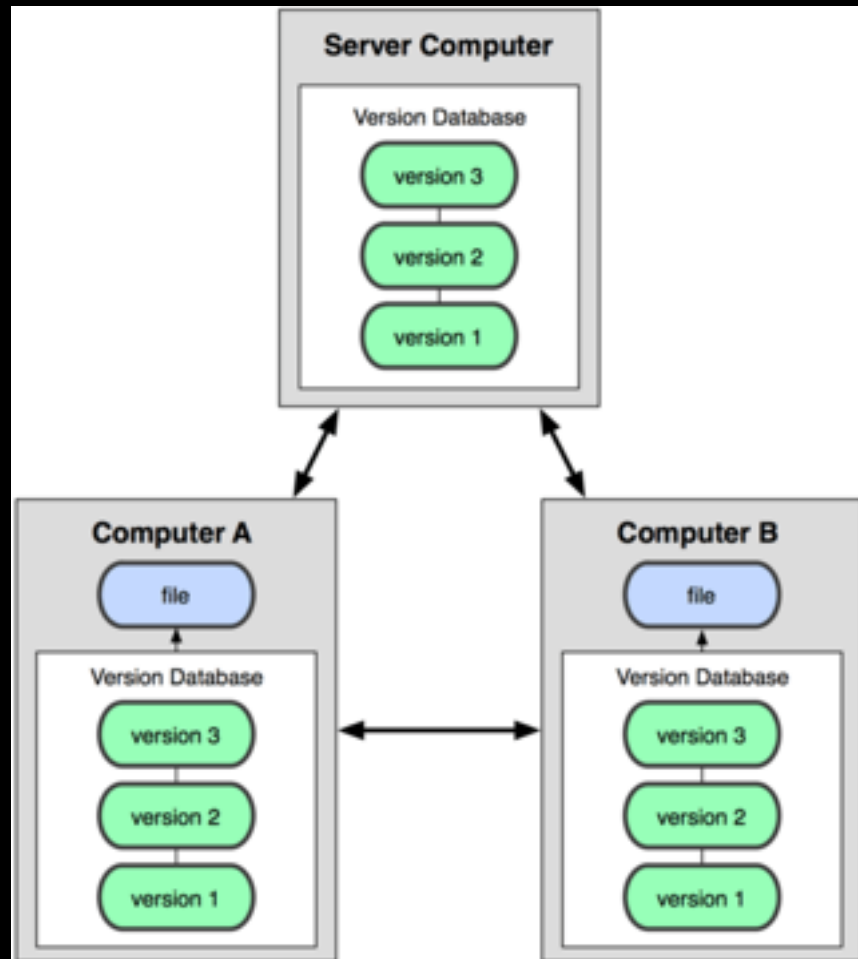


분산형 버전관리 시스템

변경 내역을 서버에도 저장하고
내컴퓨터에도 저장하자

그럼 어디에 불이 나도 작업
내용을 어디서 가져오기만 하면
복구 가능!

Distributed Version Control System; DVCS



분산형 버전관리 시스템

변경 내역을 서버에도 저장하고
내컴퓨터에도 저장하자

그럼 어디에 불이 나도 작업
내용을 어디서 가져오기만 하면
복구 가능!

In case of fire



1. git commit
2. git push
3. leave building

어떻게 동작하는지 알아보자

Working
Directory

Staging
Area

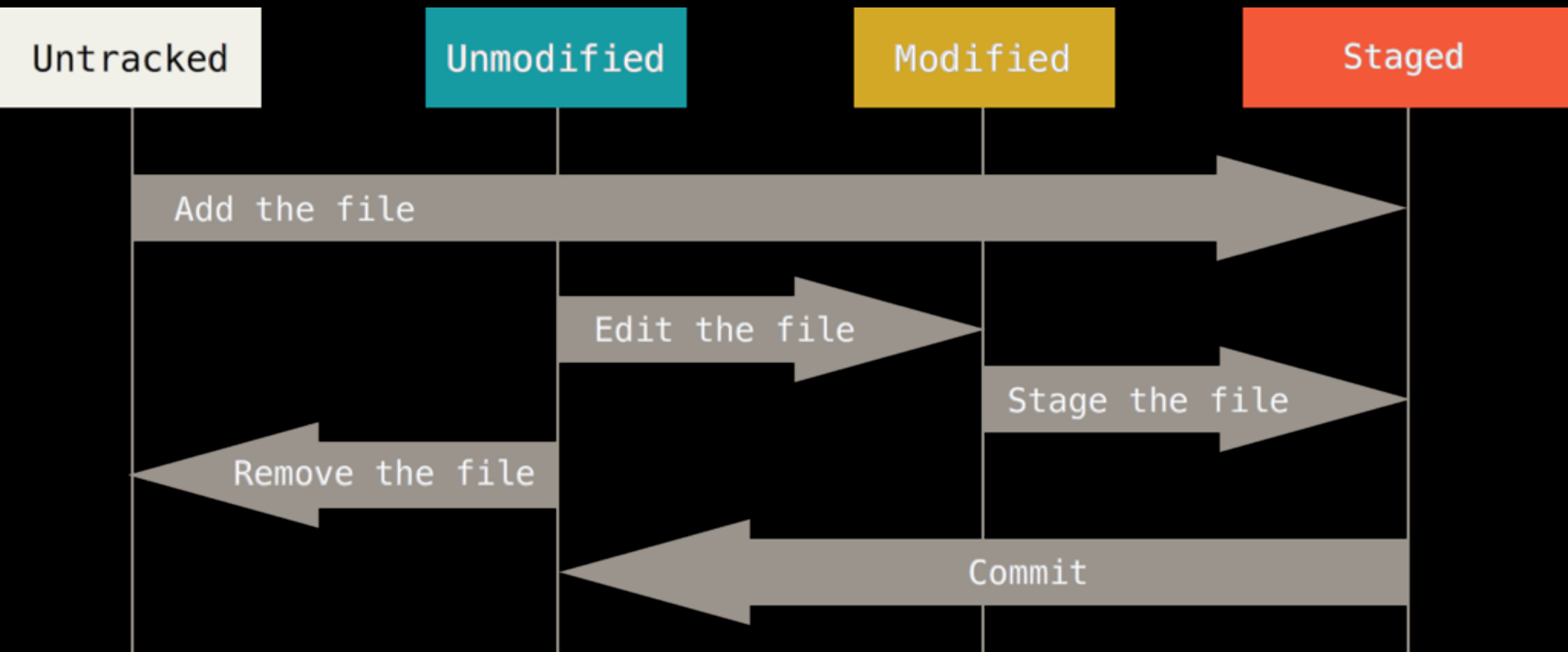
.git directory
(Repository)

Checkout the project

Stage Fixes

Commit

<https://git-scm.com/book/ko/v2/시작하기-git-기초>



<https://git-scm.com/book/ko/v2/Git의-기초-수정하고-저장소에-저장하기>

시작해보자

시작하기 설치

[https://git-scm.com/book/ko/
v2/시작하기-Git-설치](https://git-scm.com/book/ko/v2/시작하기-Git-설치)

시작하기 설정

```
$ git config --global user.name "John Doe"  
$ git config --global user.email  
 johndoe@example.com
```

시작하기 도움말

```
$ git help <verb>  
$ git <verb> --help
```

도움말 을 볼 수 있다.
<verb>는 해당 액션에 대한

본격적으로 시작해보자
우선 방법1 저장소를 만들어서 버전관리하기

```
$ git init
```

이제부터 이 폴더에서 버전관리를 하겠다.
.git폴더가 생성됨

```
$ git add <filename>
```

버전관리할 파일 추가

```
$ git add commit -m 'msg'
```

커밋

방법 2

기존 저장소를 복사하기

```
$ git clone <url>
```

해당 주소의 저장소를 복사한다.

**이제 수정하고 저장소에
저장해보자.**

```
$ git status
```

```
$ git diff
```

```
$ git add <filename>
```

```
$ git rm <filename>
```

```
$ git log
```

리모트 저장소를 제공하는 곳
많다. 혹은 직접 서버를 구성해서 해도 됨

권장

github.com

bitbucket.org

github에서 프로젝트 관리

Quick setup — if you've done this kind of thing before

 Set up in Desktop or **HTTPS** **SSH** <https://github.com/gnujoow/test.git> 

We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).


...or create a new repository on the command line

```
echo "# test" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/gnujoow/test.git
git push -u origin master
```



...or push an existing repository from the command line

```
git remote add origin https://github.com/gnujoow/test.git
git push -u origin master
```



github에서 프로젝트 관리

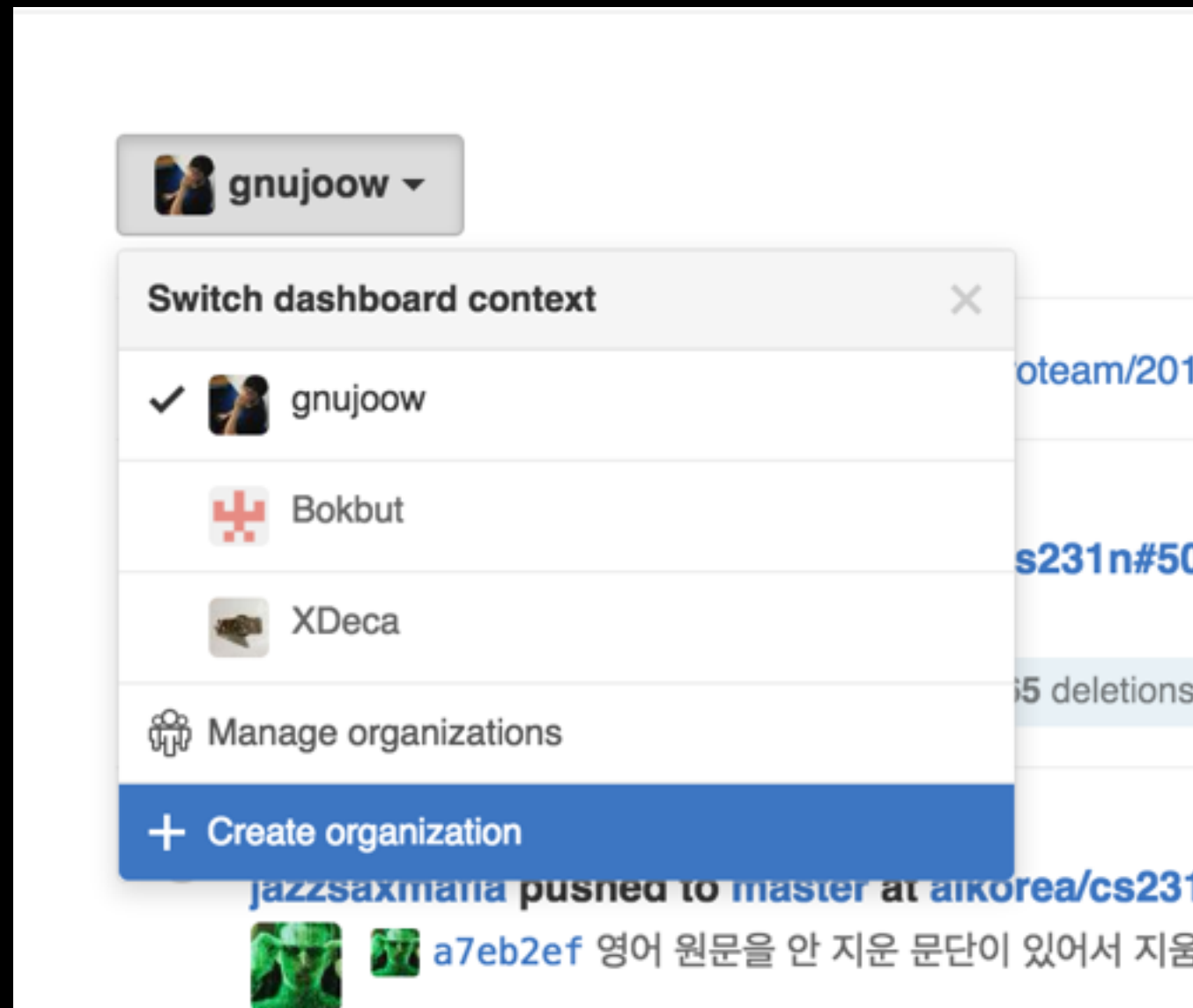
프로세스

init -> commit -> push
-> commit -> push

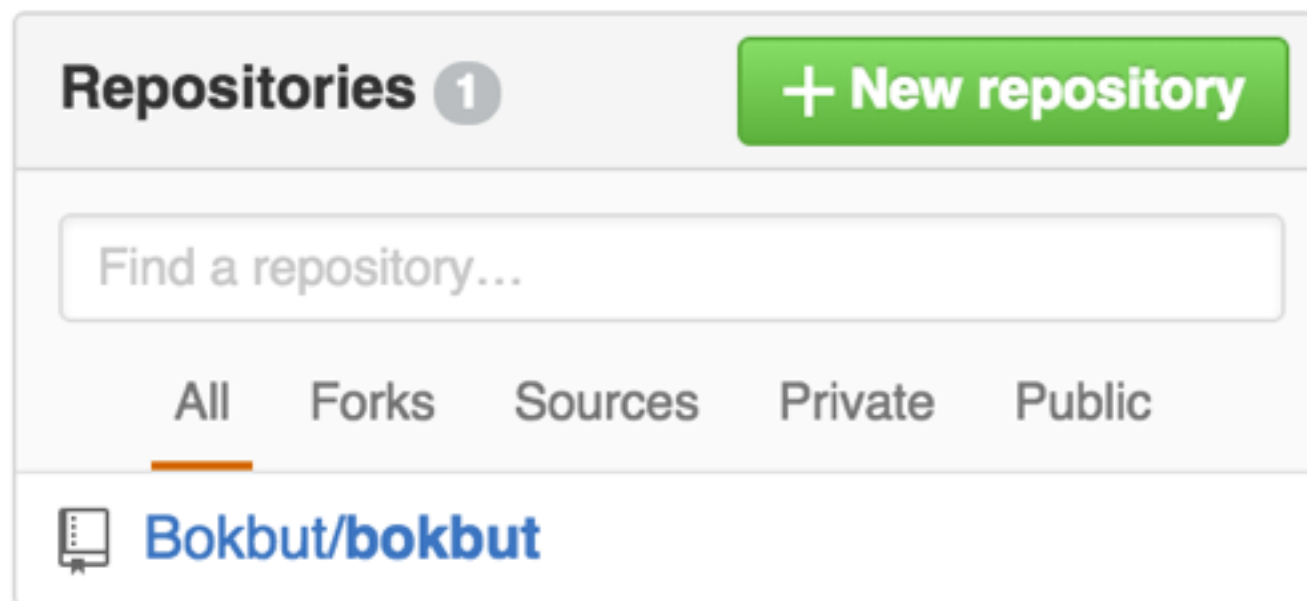
반복

그럼 다같이 할 때는 어떻게 해요?
organization을 활용합니다.

생성

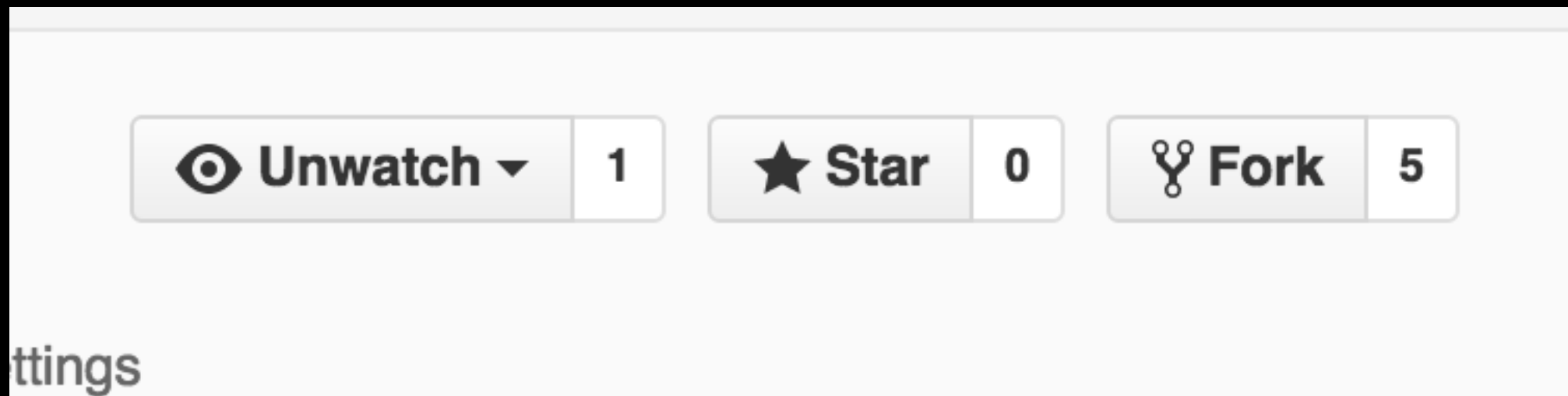


레포생성

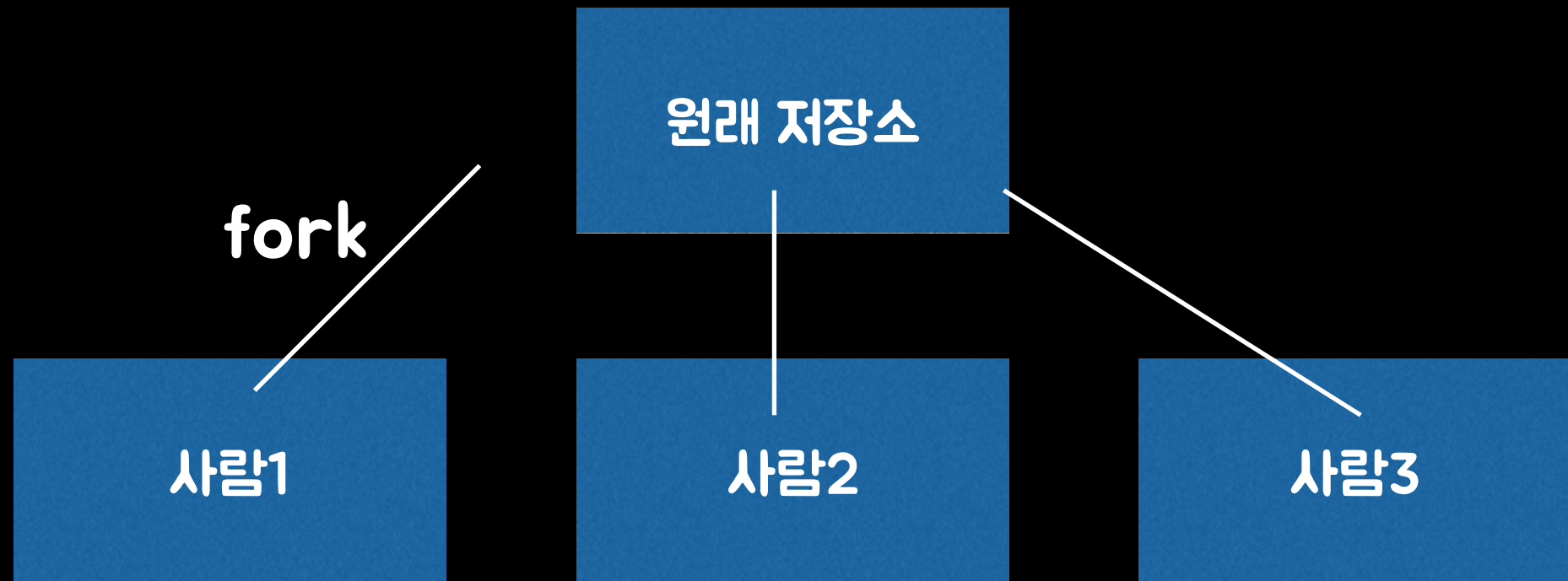


팀원들에게 fork하라고 해!

잠깐 여기서 fork란?
github에 존재하는 저장소를 내 저장소에
복사하는 것! clone이랑 다릅니다.



전체적인 구조



전체적인 작업구조



사람1

열심히 작업을 하고 이 수정을 수정하였다.

git commit

git push

전체적인 작업구조



사람1

그렇다면 반영이 어디로 될까?

전체적인 작업구조



사람1

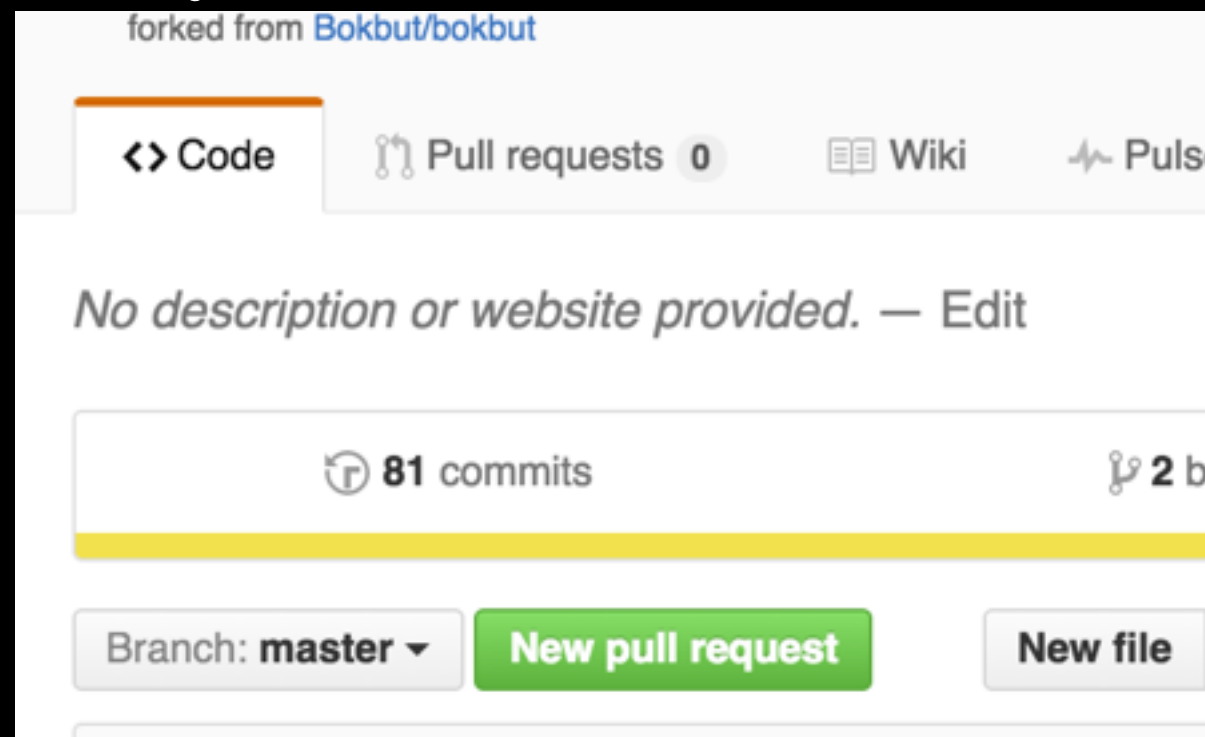
바로 사람1의 remote repository

전체적인 작업구조

사람1

원래 저장소에 변경
내용을 적용하고 싶어
요!

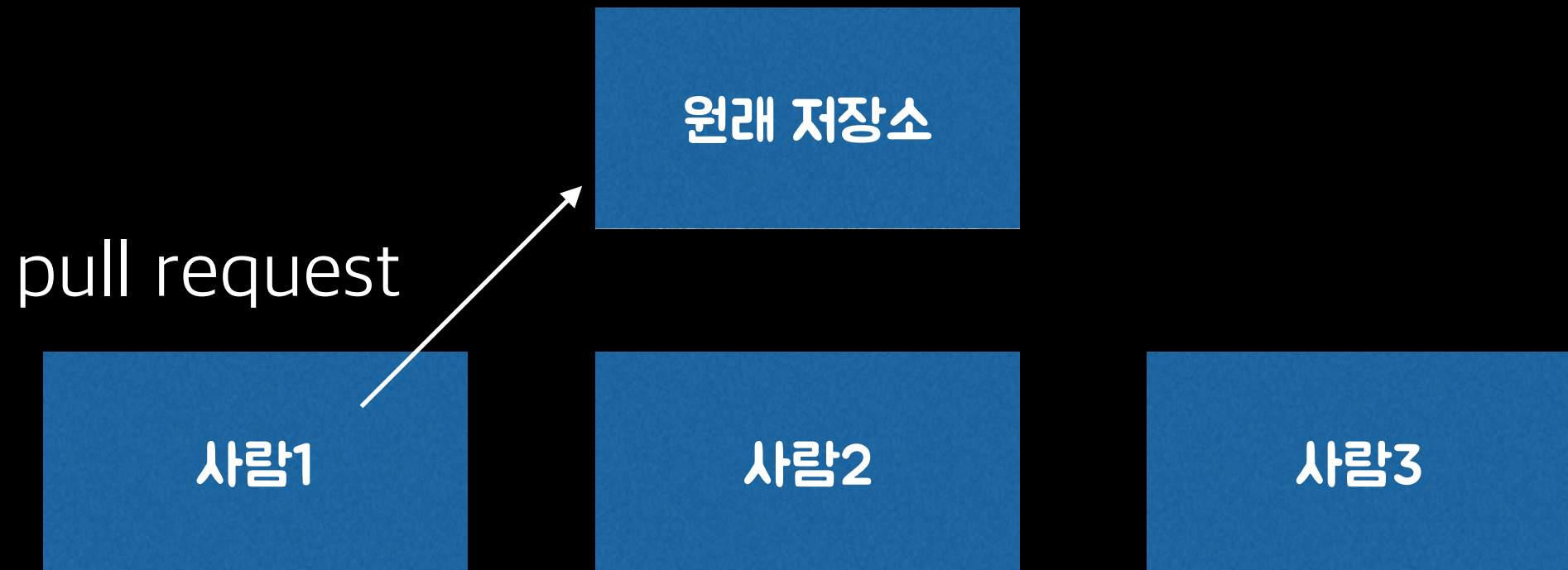
pull request
를 보내렴!



전체적인 구조



전체적인 구조



pull request

browse paginate 기능 삽입했습니다. #41

Merged gnujoow merged 2 commits into Bokbut:master from likelion-minhyeok:master 41 minutes ago

Conversation 2 Commits 2 Files changed 6

likelion-minhyeok commented 3 hours ago

디자인이 아직 좀 별로지만 일단.

done paginate function yet design 9f0ef34

gnujoow commented 3 hours ago

Bokbut member

pull request

browse paginate 기능 삽입했습니다. #41

Merged gnujoow merged 2 commits into `Bokbut:master` from `likelion-minhyeok:master` 42 minutes ago

 Conversation 2

 Commits 2

 Files changed 6

 Commits on May 13, 2016



done paginate function yet design

likelion-minhyeok committed 3 hours ago



delete container

likelion-minhyeok committed 2 hours ago

pull request

browse paginate 기능 삽입했습니다. #41

Merged gnujoow merged 2 commits into `Bokbut:master` from `likelion-minhyeok:master` 42 minutes ago

Conversation 2 Commits 2 Files changed 6

Showing changes from all commits ▾ 6 changed files ▾

1 ■■■■■ Gemfile

... @@ -1,6 +1,7 @@

```
1 source 'https://rubygems.org'
2
3 # Gem for this project
4
5 gem 'omniauth-facebook' # facebook login
6 gem 'devise'             # manage users
7 gem 'rouge', '1.10.1'    # code highlight
```

```
1 source 'https://rubygems.org'
2
3 # Gem for this project
4 +gem 'will_paginate', '~> 3.1.0' # browse pag
5 gem 'omniauth-facebook' # facebook login
6 gem 'devise'             # manage user
7 gem 'rouge', '1.10.1'    # code highli
```



pull request

github상에서 변경내용을 확인 할 수 있다.
토론도 가능하다
변경내역을 정리해서 볼 수 있다!
다른 사람의 코드를 리뷰 할 수 있다.
몰라서 못했다는 변명은 naver

pull request

 Bokbut / bokbut

 Code

 Issues 7

 Pull requests 0

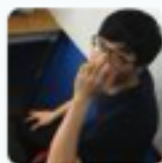
 Wiki

 Pulse

 Graphs

Branch: master ▾

 Commits on May 13, 2016



Merge pull request **#41** from likelion-minhyeok/master ...

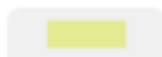
gnujoow committed 44 minutes ago



delete container

likelion-minhyeok committed 2 hours ago

master



done paginate function vet design

전체적인 구조

원래 저장소

사람1

사람2

사람3

우리도 최신 상태로 유지하고 싶어요!

전체적인 구조

no problem
나를 리모트 저장소로 지정해라!

원래 저장소

사람1

사람2

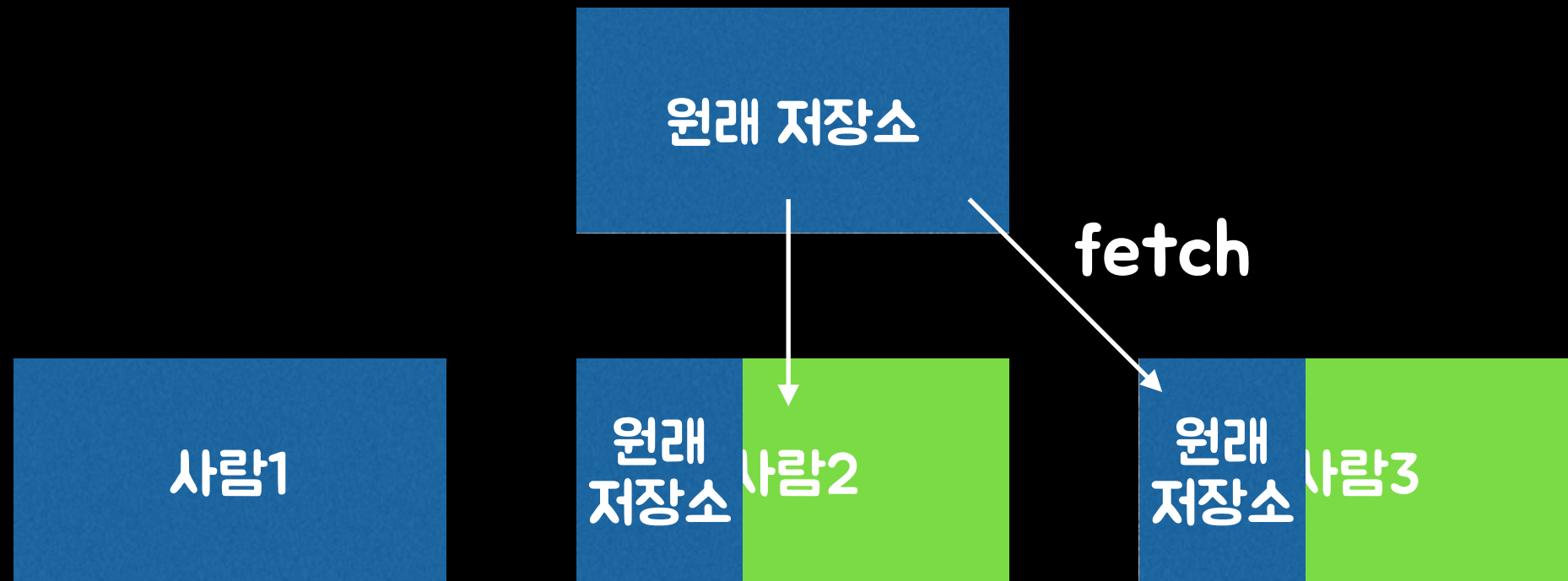
사람3

전체적인 구조

```
$ git remote add upstream <url>  
$ git fetch upstream  
$ git merge
```

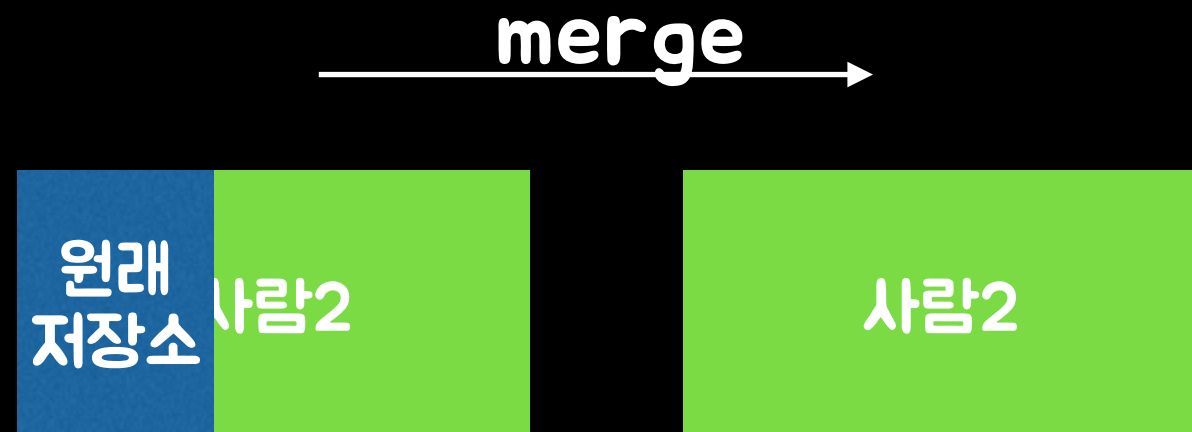
전체적인 구조

no problem
나를 리모트 저장소로 지정해라!



전체적인 구조

no problem
나를 리모트 저장소로 지정해라!



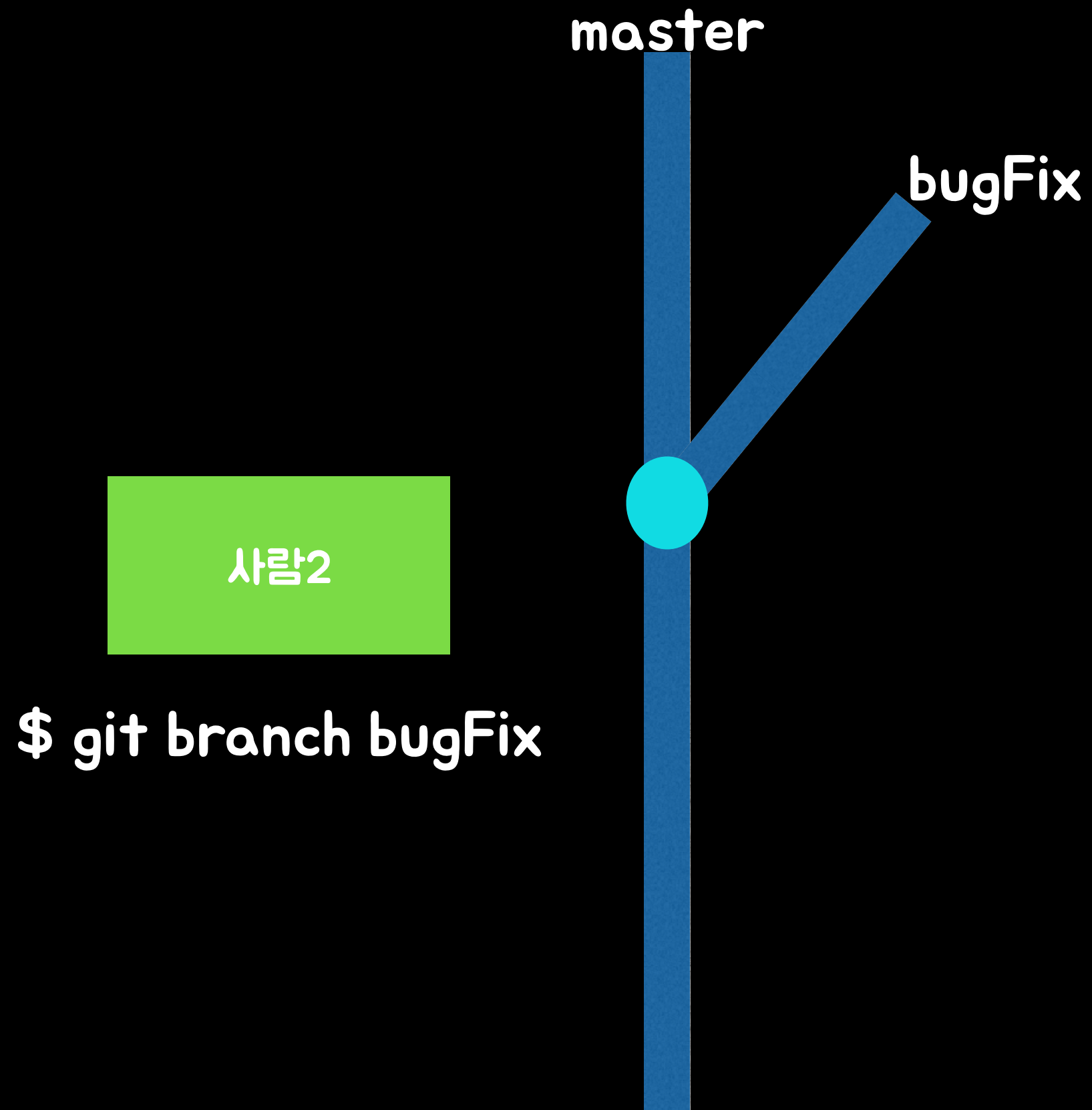
종종 충돌이 일어나지만 충돌된건 직접 수정해주면 된다.

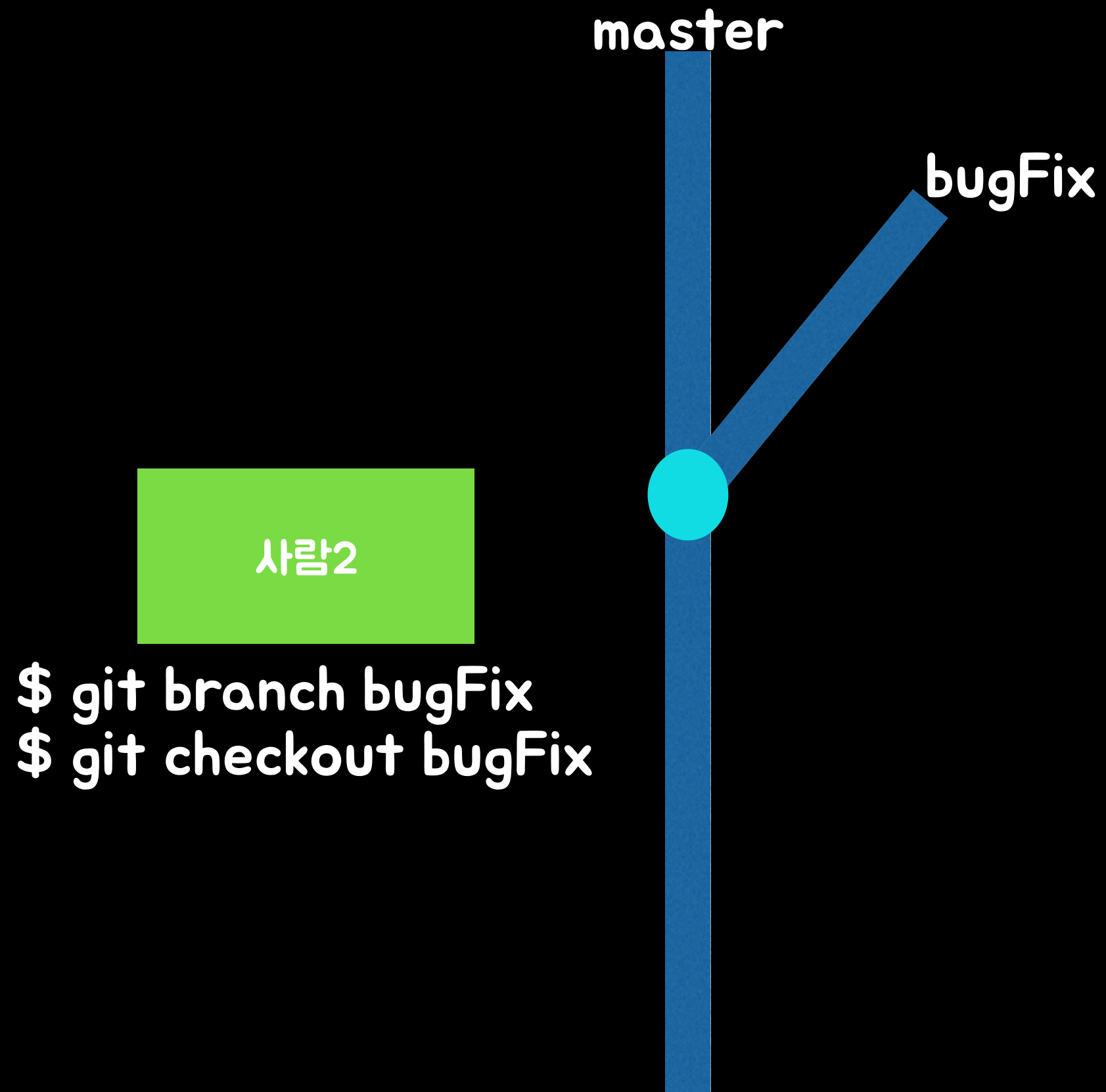
저는 제 저장소에서 작업을 좀 나눠서 하고 싶어요!

사람2

사람2

```
$ git branch bugFix
```





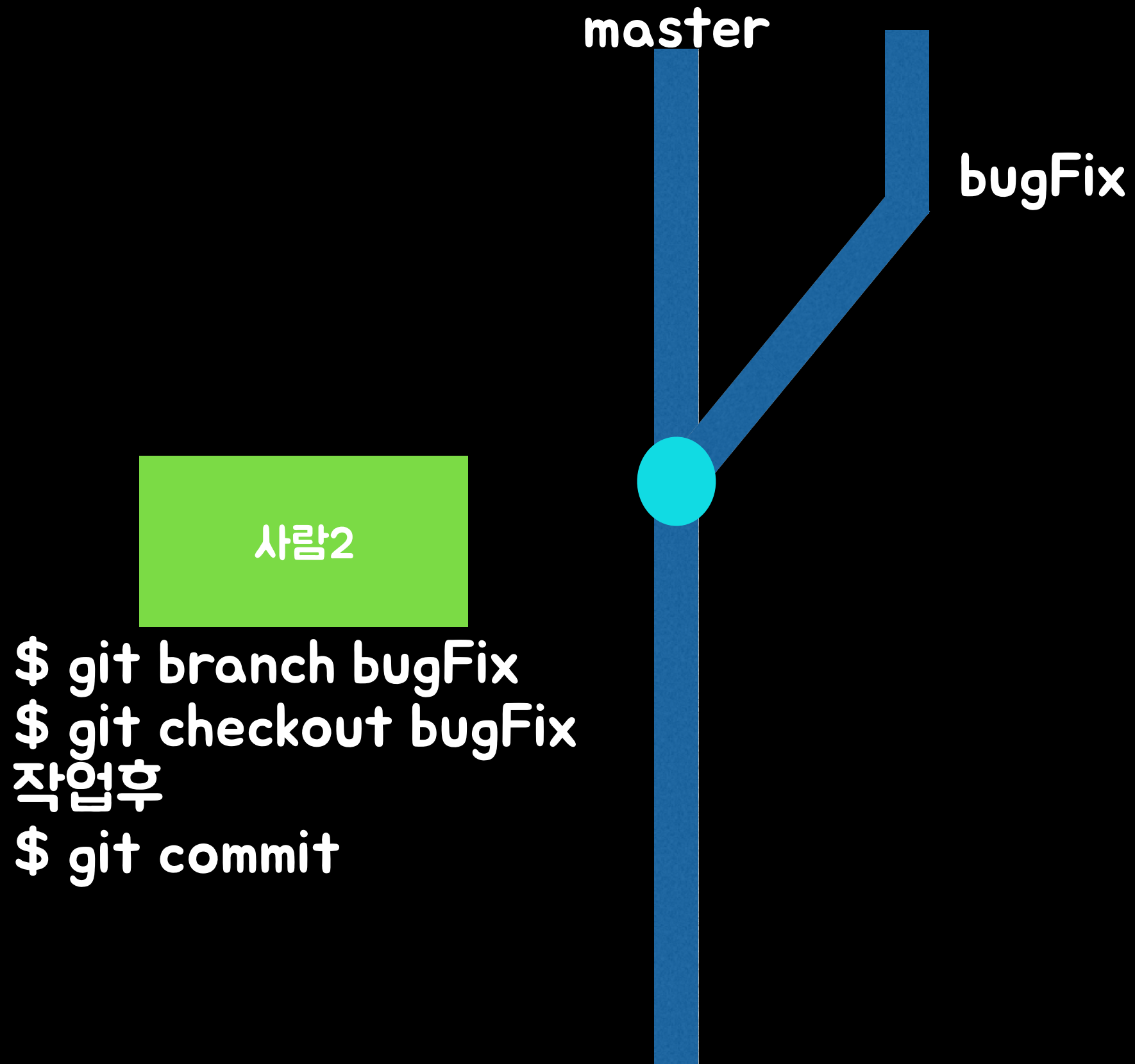
master

bugFix

사람2

```
$ git branch bugFix  
$ git checkout bugFix  
작업후  
$ git commit
```





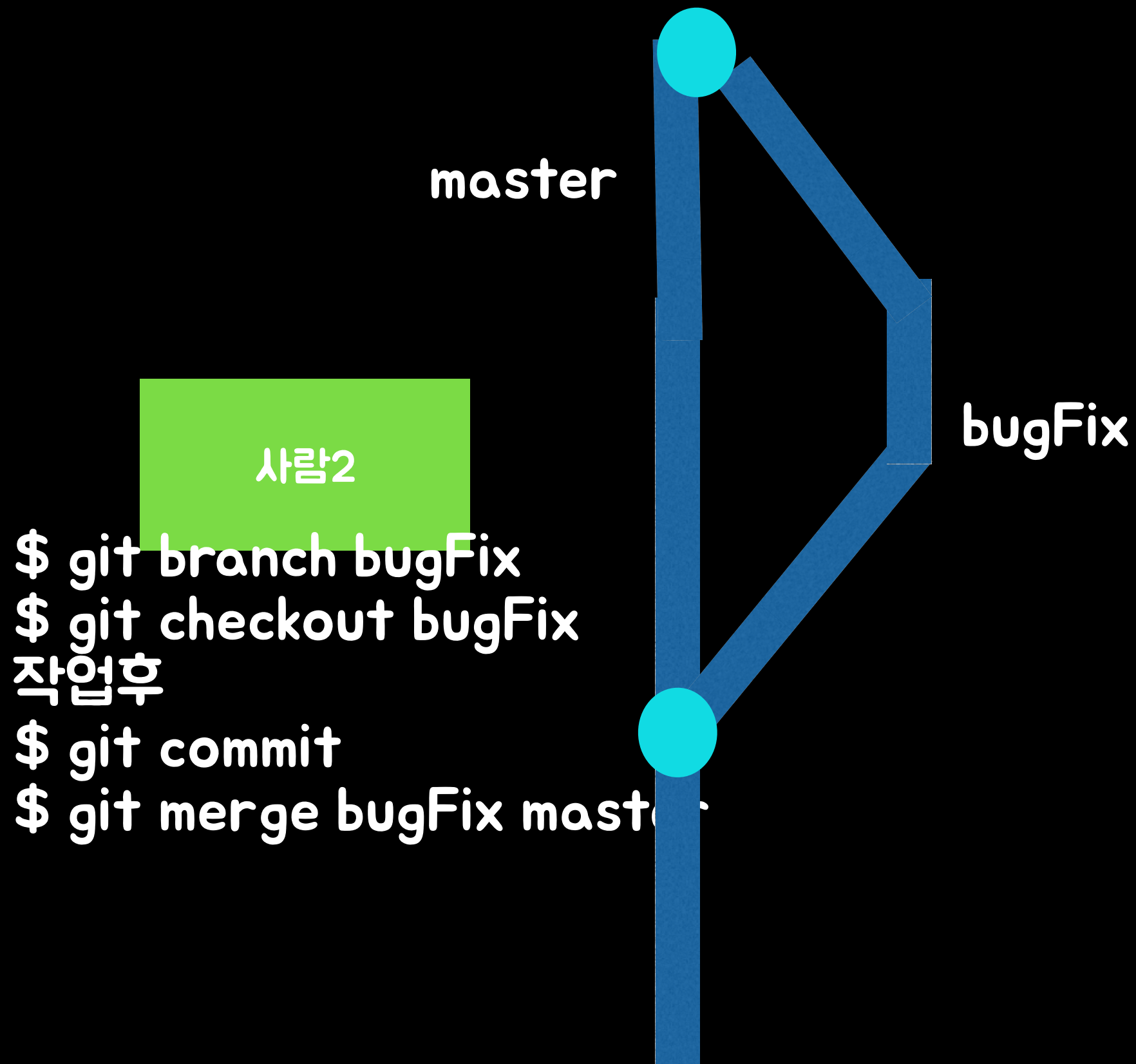
사람2

```
$ git branch bugFix  
$ git checkout bugFix  
작업후  
$ git commit
```

master

bugFix





master

사람2

bugFix

```
$ git branch bugFix  
$ git checkout bugFix  
작업후  
$ git commit  
$ git merge bugFix master
```

읽어주셨음 하는 문서들
사실 나보다 설명 더 잘되어 있음

저 앞에 설명하는 사람이 너무 못했다.
혹은 너무 복잡하다.
간단하게 알고싶다.

<http://rogerdudler.github.io/git-guide/index.ko.html>

처음부터 차근차근 해보고 싶다.

http://backlogtool.com/git-guide/kr/intro/intro1_1.html

git 이미 알고 있다. 깊게 알고 싶다.
<https://git-scm.com/book/ko/v2/>

협업하는 부분 좀 더 자세히 알려주세요

<https://dogfeet.github.io/articles/2012/how-to-github.html>

나는 누가 설명해주는게 좋아

<https://opentutorials.org/course/1492>

<https://opentutorials.org/course/307/2475>

브랜치에 대해서 직접 실습을 해보고싶다.
<http://learnbranch.urigit.com/>

감사합니다.