Gitflow Workflow

중앙 remote

인터넷이나 네트워크 어딘가에 있는 저장소

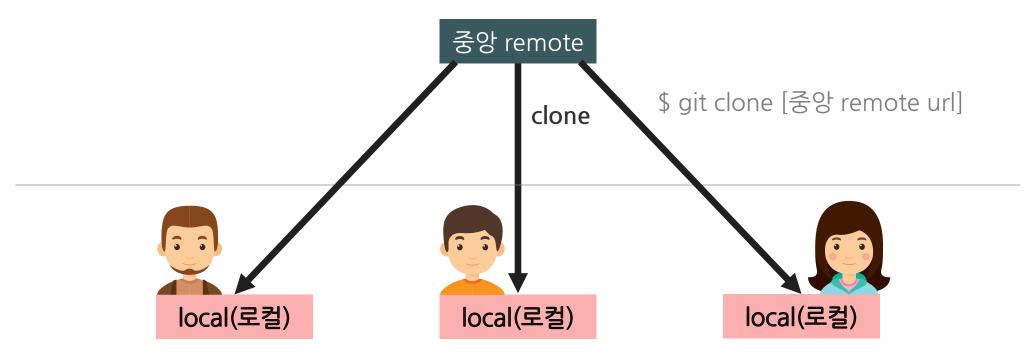
내 PC에 파일이 저장되는 개인 전용 저장소

local(로컬)

중앙 remote

Organization으로 저장소 생성

local(로컬)



Clone 명령어를 통해 (아래의 명령어를 포함한 작업이 수행된다.)

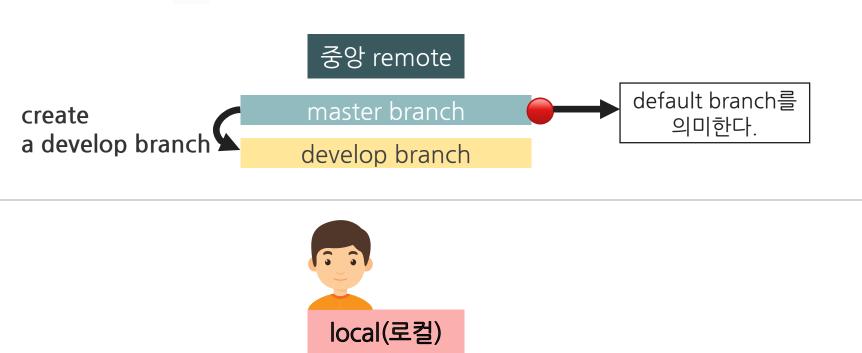
- 1. 디렉토리를 만들고
- 2. 디렉토리로 들어가고 나서 git init 명령으로 빈 Git 저장소를 만든다.
- 3. 입력한 URL을 origin이라는(기본값) 이름의 리모트로 추가하고(git remote add)
- 4. git fetch 명령으로 리모트 저장소에서 데이터를 가져온다.

중앙 remote origin

\$ git remote -v // 설정한 remote 목록을 확인



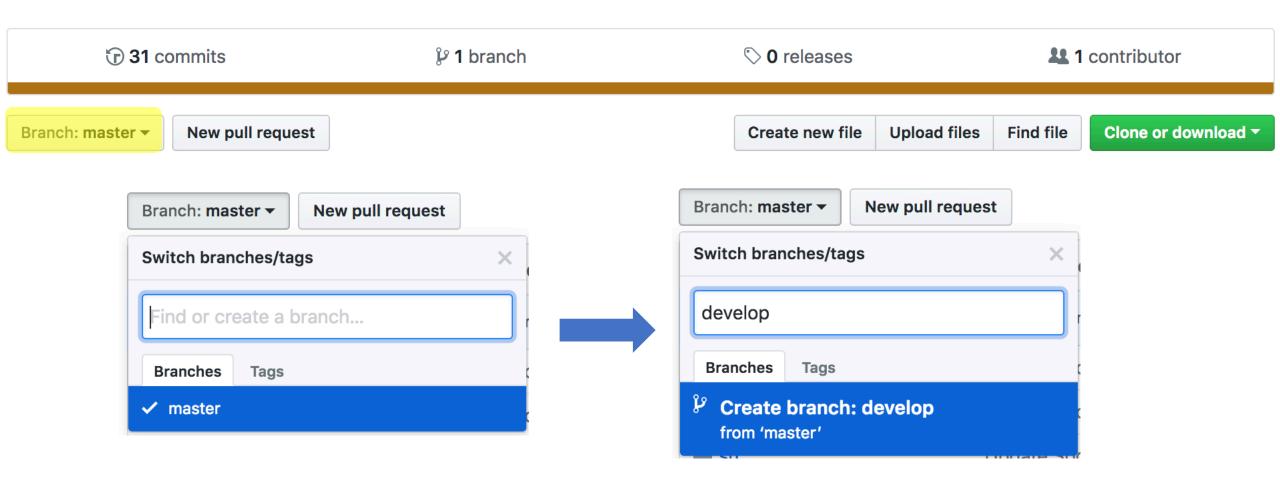
- * Clone 을 통해 기본적으로 clone한 URL(여기서는 중앙 remote URL)이 origin이라는(기본값) 이름의 리모트로 추가되어 있다.
- * 즉, \$ git clone [중앙 remote url] 명령어에는 \$ git remote add origin [중앙 remote url] 명령어가 포함되어 있다.



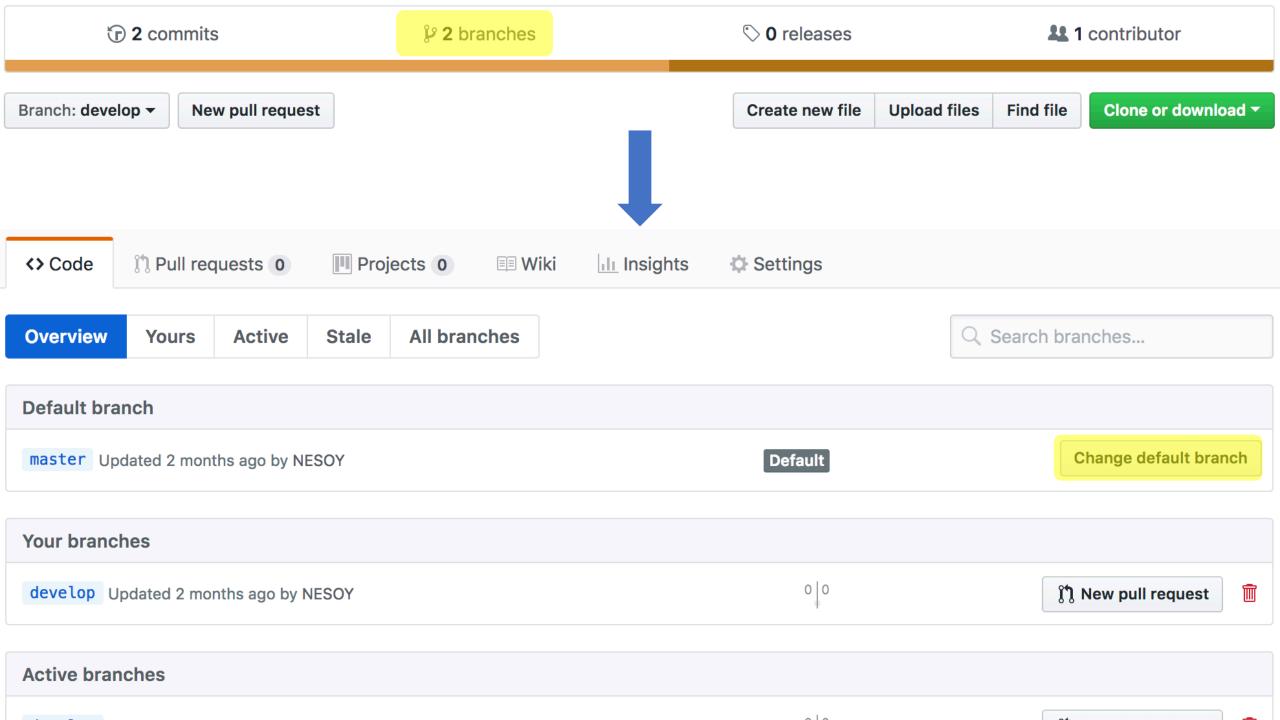
1. GitHub 페이지에서 "Create branch" 버튼을 이용하면, master branch 외의 새로운 branch를 생성할 수 있다.

master branch

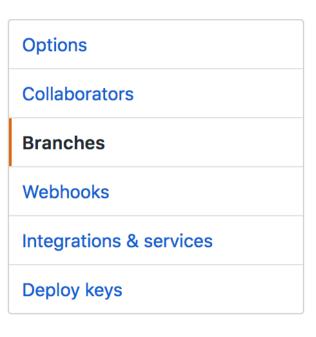
2. 새로 생성한 'develop' branch를 default branch로 설정한다.



* 평소에는 'develop' branch를 기반으로 개발을 진행하기 때문에 \$ git push origin some-feature(내 로컬 저장소의 some-feature branch를 중앙 원격 저장소로 올리는 명령)를 할때, 중앙 원격 저장소의 default로 설정되어 있는 'develop'으로 올라가도록 설정한다.

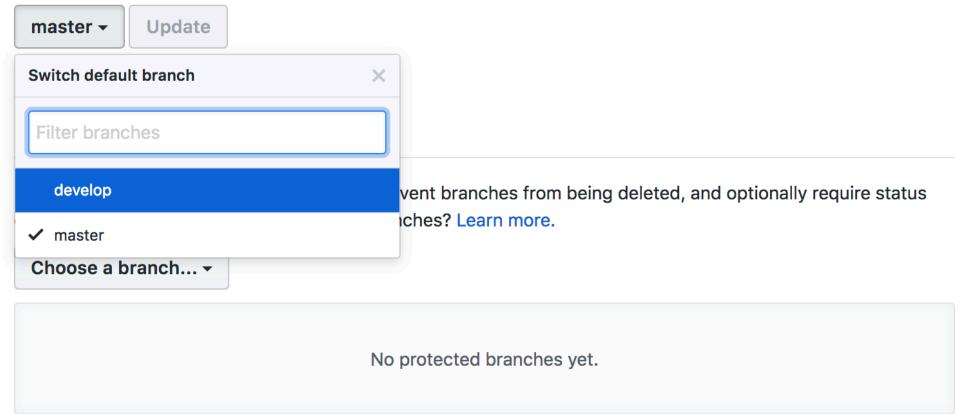


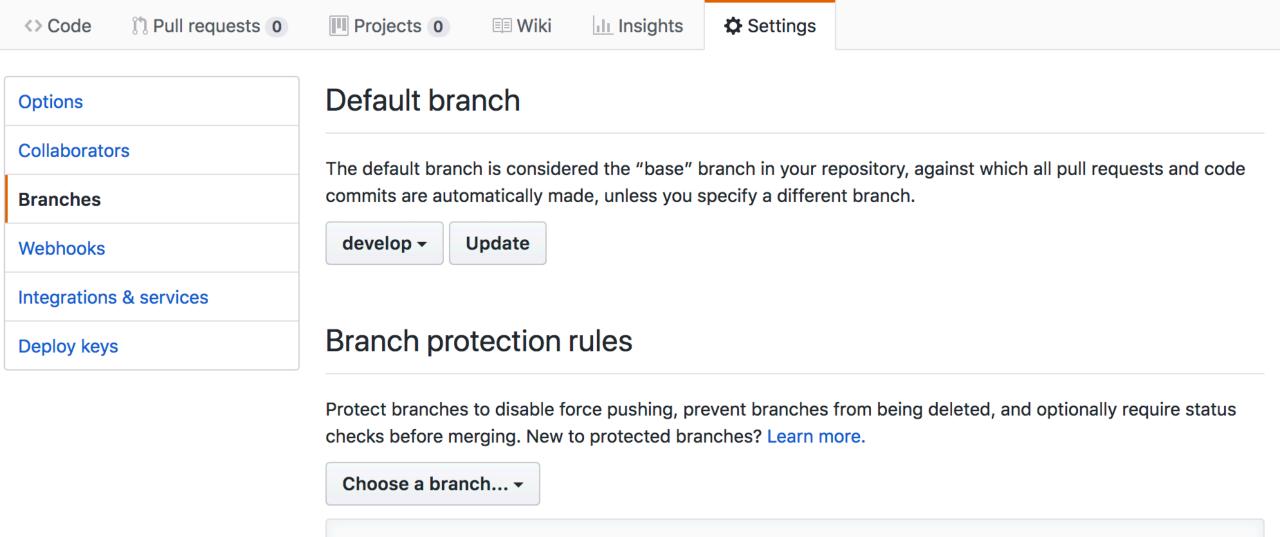




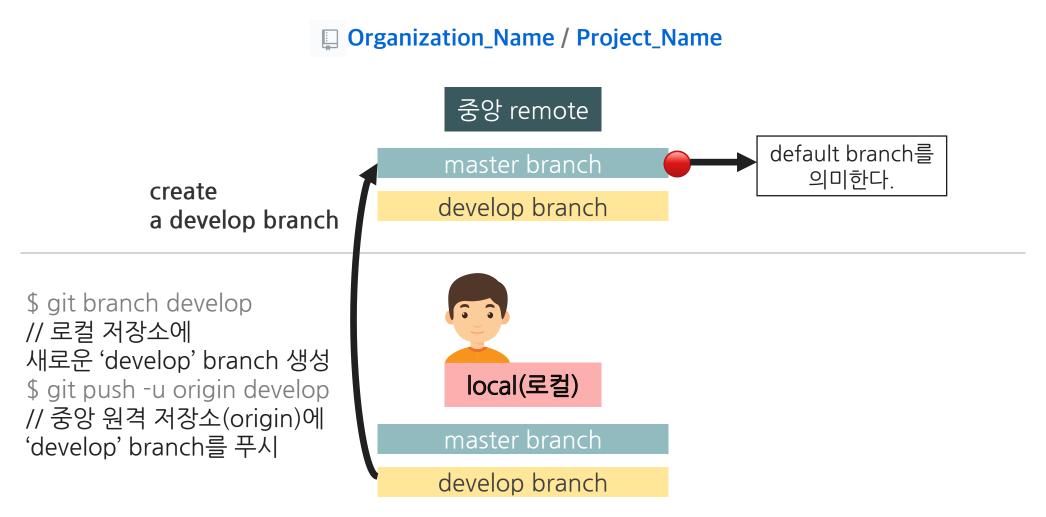
Default branch

The default branch is considered the "base" branch in your repository, against which all pull requests and code commits are automatically made, unless you specify a different branch.

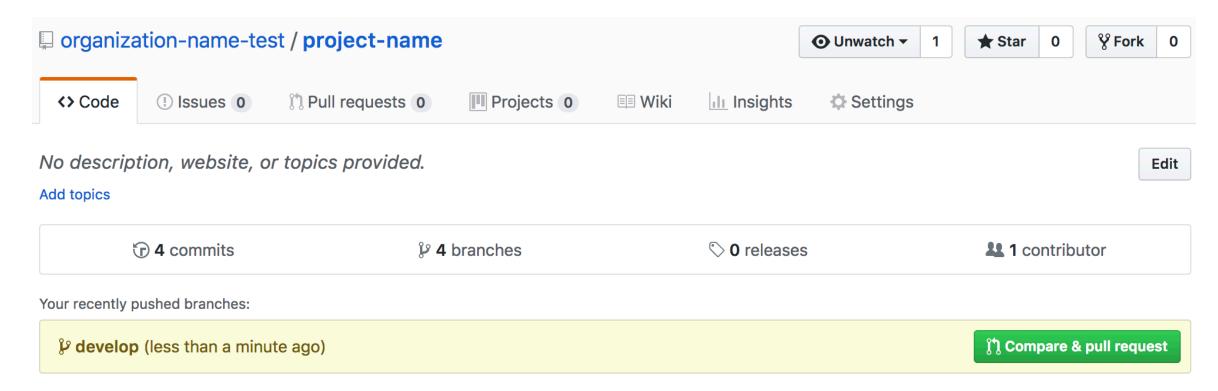


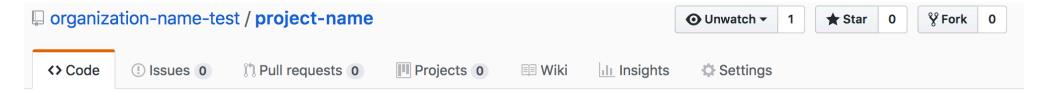


Copyright (c) Kwon Hee-Jeong. All rights reserved.



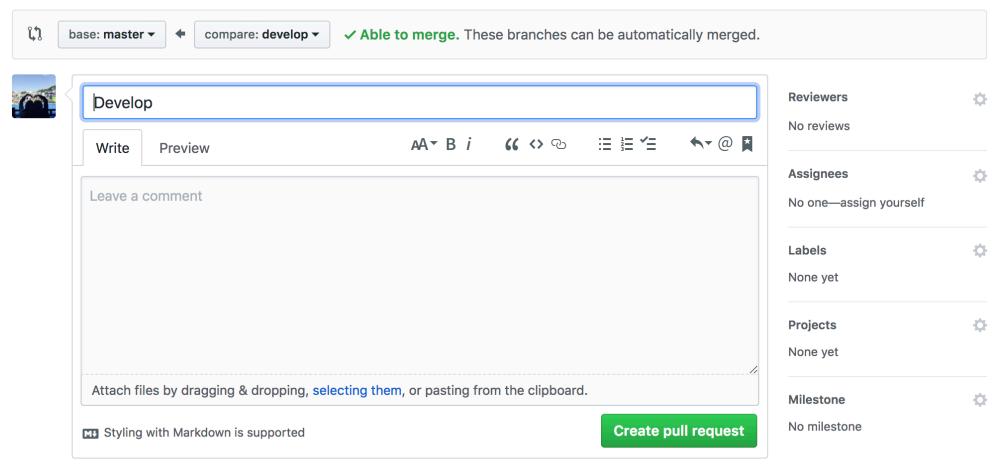
- 1. GitHub 페이지에서 자신이 push한 'develop' branch를 병합해달라는 pull request를 날린다.
- 2. 프로젝트 관리자는 해당 pull request를 merge하여 새로운 'develop' branch를 중앙 원격 저장소에 생성한다.
- 3. 방법 1 과 마찬가지로 새로 생성한 'develop' branch를 default branch로 생성한다. Hee-Jeong. All rights reserved.

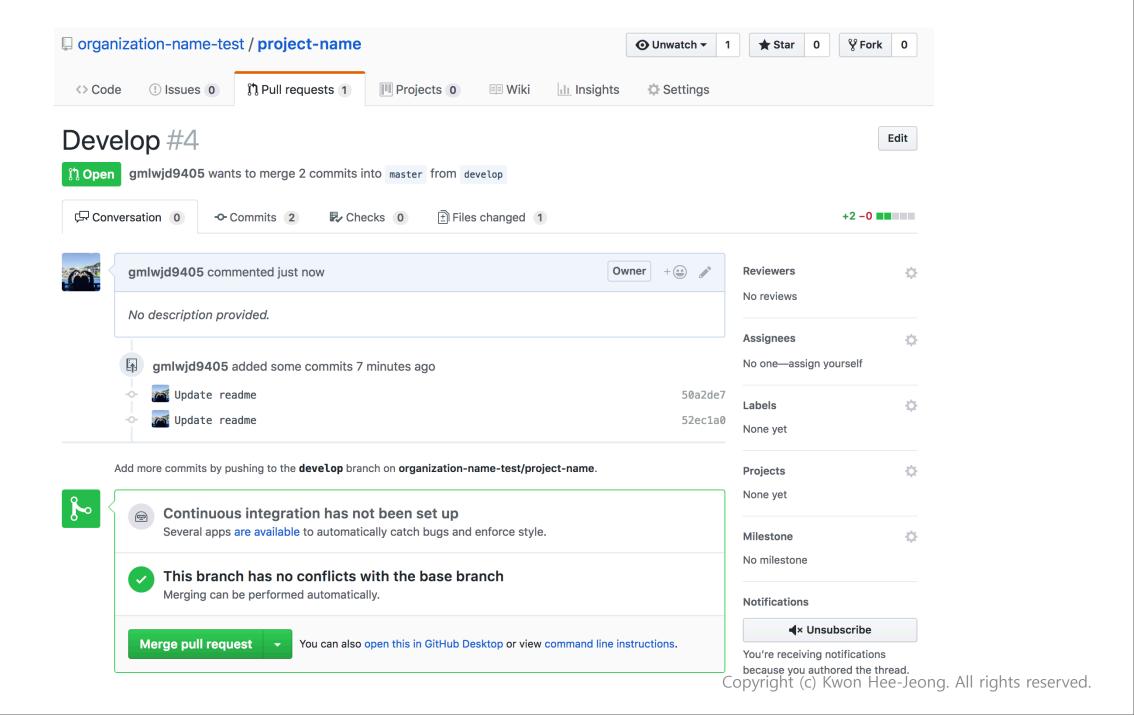


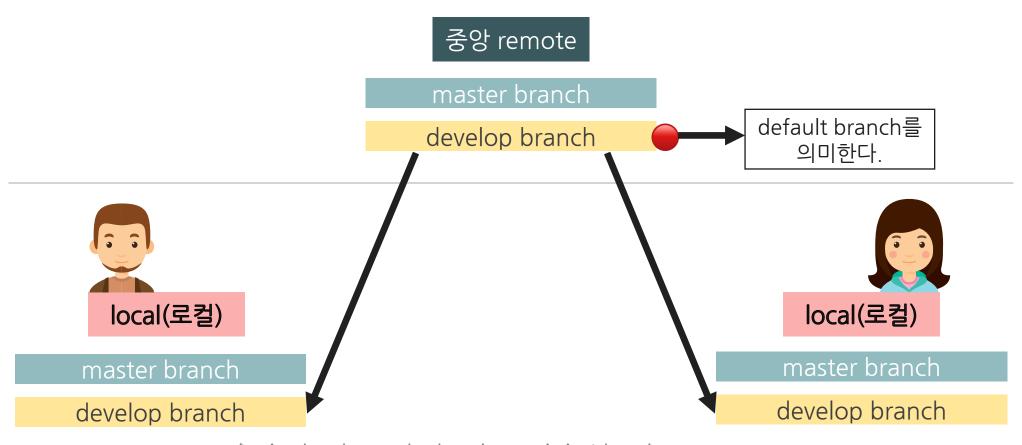


Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks.





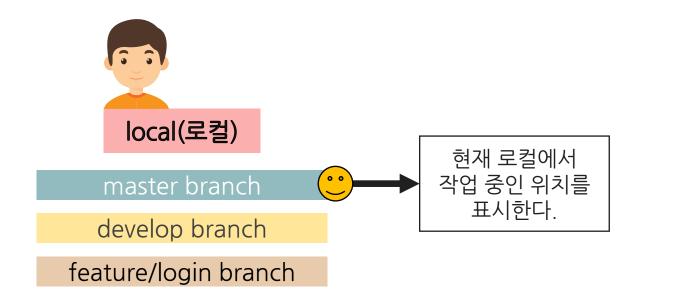


\$ git checkout -b develop origin/develop // 중앙 원격 저장소(origin)의 'develop' branch와 연결된 새로운 'develop' branch를 로컬 저장소에 생성 Copyright (c) Kwon Hee-Jeong. All rights reserved.

중앙 remote

master branch

develop branch



중앙 remote

master branch

develop branch

내 로컬에서 새로운 작업을 하는 경우 develop branch에서 그 작업에 대한 브랜치를 생성 후 이동한다.

> branch checkout



master branch

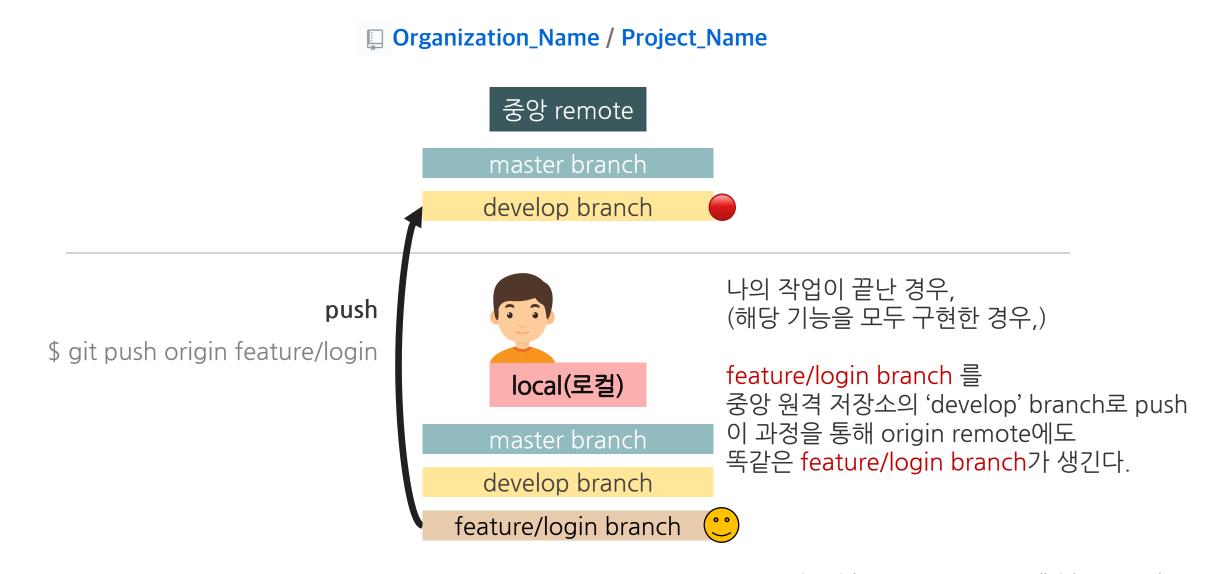
develop branch

feature/login branch

\$ git branch feature/login develop
// 새로운 branch 생성
\$ git checkout feature/login
// 해당 branch로 작업 위치 이동

\$ git checkout -b feature/login **develop** // 위의 두 명령어를 합한 것

팀이 풀 리퀘스트를 이용하기로 했을 때, 해당 기능의 feature branch를 그대로 푸시



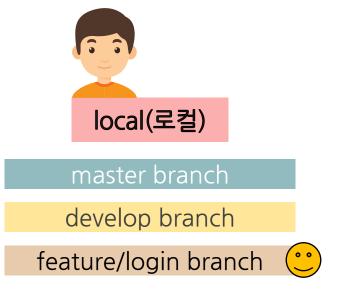
중앙 remote

master branch

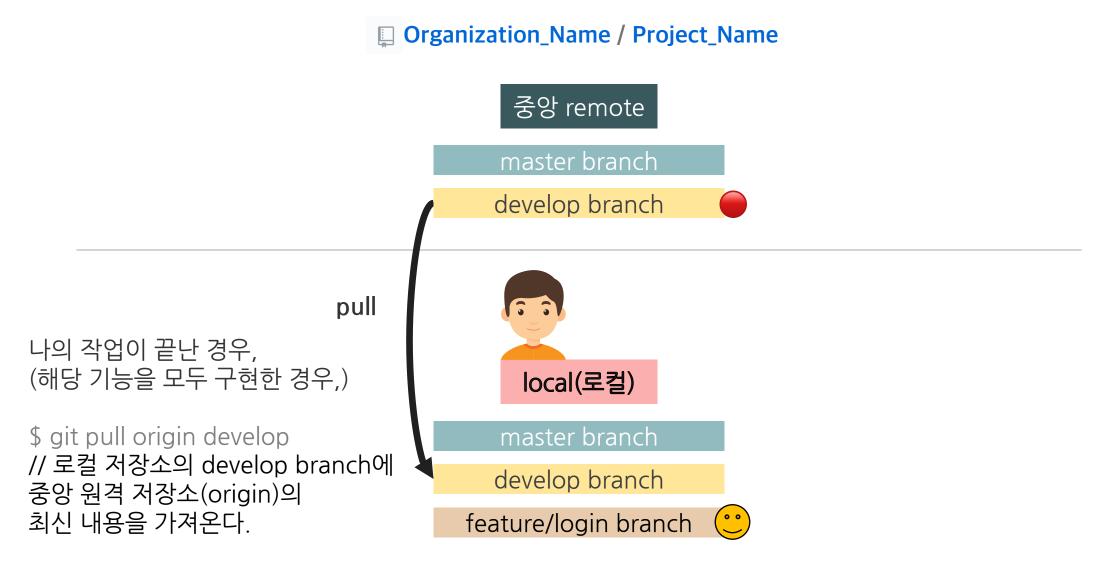
develop branch

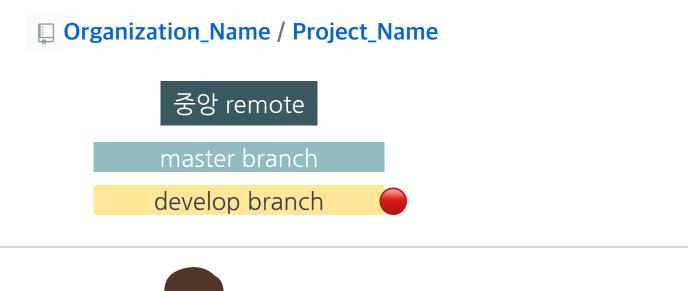
feature/login branch

변경 내용을 merge 팀원이 모두 확인한 후 (충돌이 일어났다면, 해결!) develop branch에 merge한다.

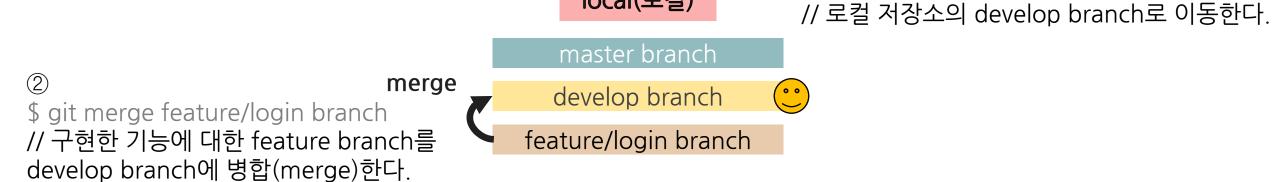


물 리퀘스트를 이용하지 않기로 했을 때, 자신이 직접 병합(merge) 후 중앙 원격 저장소에 푸시





local(로컬)



(현재 작업 중인 위치는 develop branch여야 한다.)

Copyright (c) Kwon Hee-Jeong. All rights reserved.

\$ git checkout develop

Organization_Name / Project_Name 중앙 remote master branch develop branch push \$ git push // 로컬 저장소의 develop branch를 local(로컬) 중앙 원격 저장소에 push한다. master branch develop branch feature/login branch \$ git branch -d feature/login branch // 기능이 완성된 feature branch를 삭제한다.

중앙 remote master branch

중앙 원격 저장소의 develop branch에는 새로운 내용이 갱신되어 있는 상태

develop branch



develop으로 이동한 후에 develop에 새로운 내용(중앙에 갱신된 내용)을 받아와야 한다.

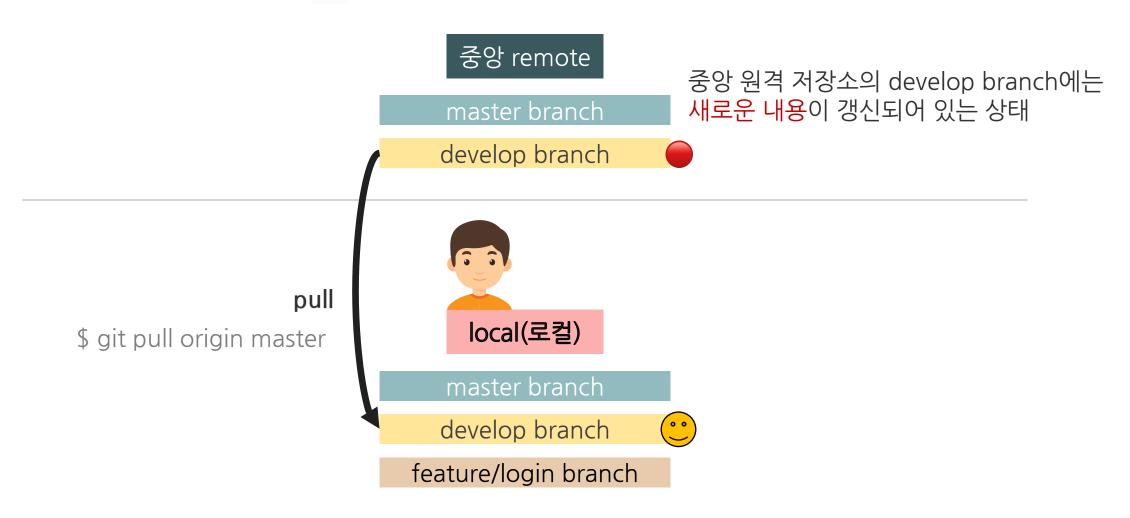
\$ git checkout develop

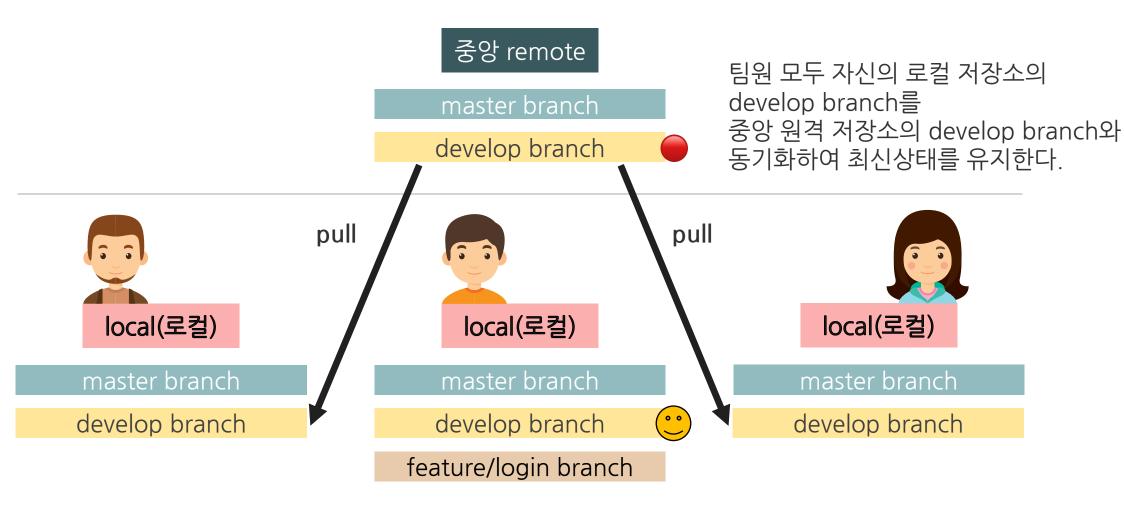
checkout

master branch

develop branch

feature/login branch







중앙 remote

master branch

develop branch



master branch

develop branch

feature/signup branch

\$ git branch -d feature/login

local에서 완성한 이전 작업 브랜치를 삭제한다.

다시 새로운 작업을 하는 경우 develop branch에서 그 자연에 대하 브래치를 새성?

그 작업에 대한 브랜치를 생성하여 작업한다.

branch



Gitflow Workflow의 배포하기와 버그 수정하기

중앙 remote

master branch

develop branch

'develop' branch에서 버전 1.2에 대한 기능이 모두 구현이 되었으면 develop branch에서 그 작업에 대한 'release' branch를 생성 후 이동한다.



\$ git branch release-1.2 develop // 새로운 branch 생성 \$ git checkout release-1.2 // 해당 branch로 작업 위치 이동

\$ git checkout -b release-1.2 develop // 위의 두 명령어를 합한 것

develop branch branch

release-1.2 branch

master branch



중앙 remote

master branch

develop branch



master branch

develop branch

release-1.2 branch

① 'release' branch에서 배포를 위한 최종적인 버그 수정, 문서 추가 등의 작업을 진행한다.



③ 'develop' branch도 마찬가지로 release-1.2 브랜치 내용을 병합(merge) 한 후 중앙 원격 저장소에 올린다. (배포를 준비하는 동안 release 브랜치가 변경되었을 수 있으므로)

중앙 remote

master branch

develop branch

배포한 버전에 긴급하게 수정을 해야 할 필요가 있을 경우, 'master' branch에서 직접 branch('hotfix' 브랜치)를 만들어 필요 한 부분만을 수정한다.

branch

checkout



master branch

develop branch

hotfix-1.2.1 branch

\$ git branch hotfix-1.2.1 master // 새로운 branch 생성 \$ git checkout hotfix-1.2.1 // 해당 branch로 작업 위치 이동

\$ git checkout -b hotfix-1.2.1 master // 위의 두 명령어를 합한 것

중앙 remote

master branch

develop branch



master branch

develop branch

hotfix-1.2.1 branch

- ① 'hotfix' branch에서 긴급한 버그를 빠르게 수정한다.
- ② "배포하기" 과정에서와 마찬가지로 'hotfix' branch를 master branch와 develop branch에 병합(merge)한 후 중앙 원격 저장소로 올린다.