

GitHub

1. GitHub 란?

1) GitHub 란?

형상관리 툴인 Git 레포지토리를 지원하는 웹서비스

깃허브란 전세계적으로 널리 알려진 형상관리툴인 깃 저장소를 지원하는 웹서비스로 현재 많은 오픈소스가 깃허브를 통해서 관리되고 있고, 기업도 깃허브 엔터프라이즈를 통해서 많은 소프트웨어를 관리하고 있다. 이는 리포지토리를 생성해서 사용할 수 있다.

2) GitHub 특징

(1) Fork

오픈소스 툴을 개발하고 있는데 이에 기여하기 위해서는 여기에 commit 하고 push 할 수 있는 권한이 있어야한다. 이때 누군지 모르는 사람에게 commit, push 의 권한을 주기 어렵기에 remote repository 를 복제해서 해당 repository 에서 commit, push 를 할 수 있도록 한다. 이에 따라 remote repository 에 대한 권한이 없어도 commit 과 push 가 가능하게 한다.

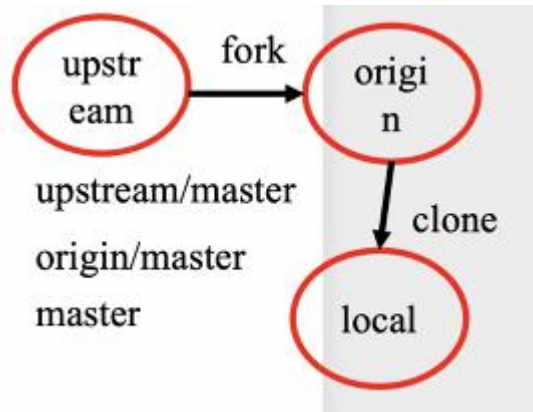
악의의 목적을 가지고 original repository 에 대해 접근하는 경우에도 original repository 에 영향을 미칠 수 없기에 오픈소스에서 매우 중요한 요소이다. Fork 는 upstream 이라는 것을 별도로 생성해서 original repository 가 업데이트되는 것을 반영할 수 있다.

ex)

upstream/master: original repository 에 대한 branch

origin/master: remote repository 에 있는 branch

master: local repository 에 있는 branch



(2) Pull request

original repository 에서 Fork 한 뒤, remote repository 에서 새롭게 작업된 commit 을 꼭 가져오는 역할을 한다. Pull request 는 original repository 에서 remote repository 에서 작업한 내용을 확인할 수 있고, 이를 가져올 때 사용한다. 직접 original repository 에 영향을 미치는 것이 아니라 remote repository 에서 작업한 뒤 pull 할 수 있기에 오픈소스에서 중요한 영향을 미친다.

2. Git Glance

(1) README.md

해당 프로젝트가 무엇인지, 어떻게 프로젝트를 사용할 수 있을지, 어떻게 기여할 수 있는지, 어떻게 설치할 수 있는지 등 전반적인 내용을 담고 있다. 협업 시 이 내용을 보고 프로젝트 내용을 알 수 있기에 중요하다.

(2) commit history

커밋한 내용을 확인할 수 있다.

(3) pull request

기능제한을 하거나, 리뷰를 할 때 많이 사용한다.

(4) Issues

게시판의 용도로 사용할 수 있다.

(5) Projects

- 프로젝트를 관리할 수 있는 이슈별 진행사항을 체크한다.

(6) Insights

- 기여도, fork 등에 대해 알 수 있는 자료들이 있다.

3. GitHub Flow

git flow 와 유사한 면이 존재한다.

1) GitHub Flow

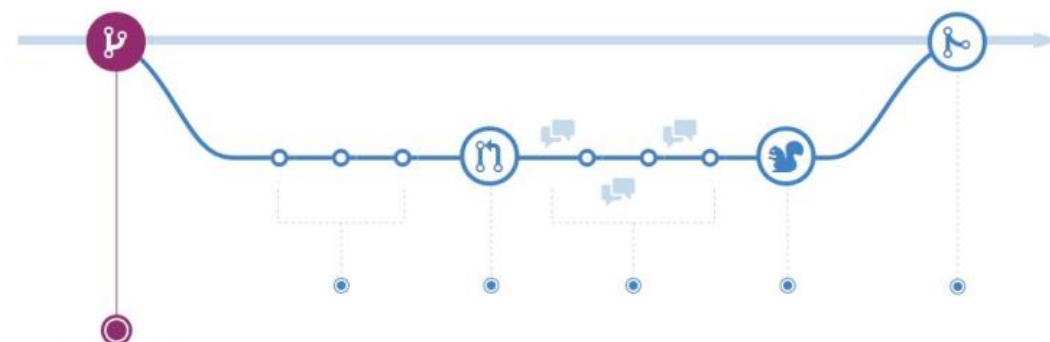


pull request 를 내 코드를 리뷰해줄 수 있는 사람들에게 보낸 뒤 통과되면 merge 된다.

2) 구체적인 flow

Without Affecting others

- Implement new features and ideas with a new branch

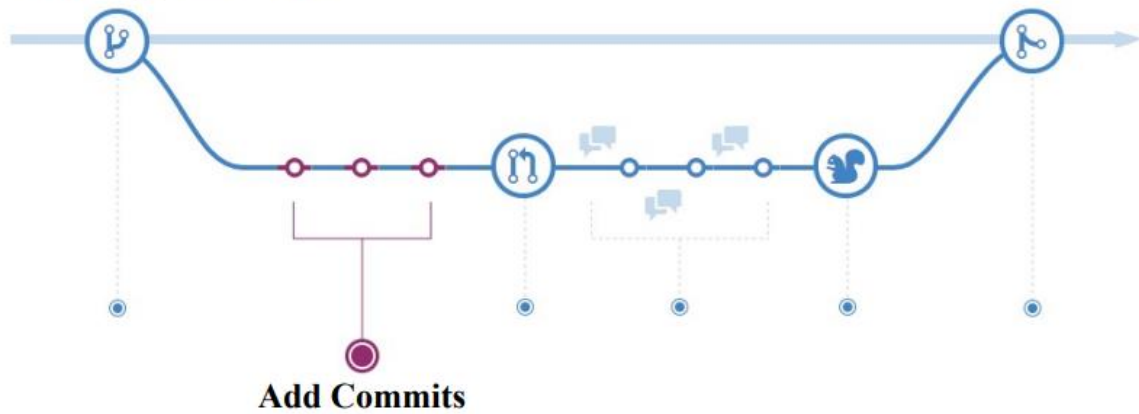


Create Branch

- A new branch must be created from master branch
- Your changes should be applied on master branch

다른 사람들과의 충돌을 방지하기 위해 branch 를 생성한다.

Making Changes

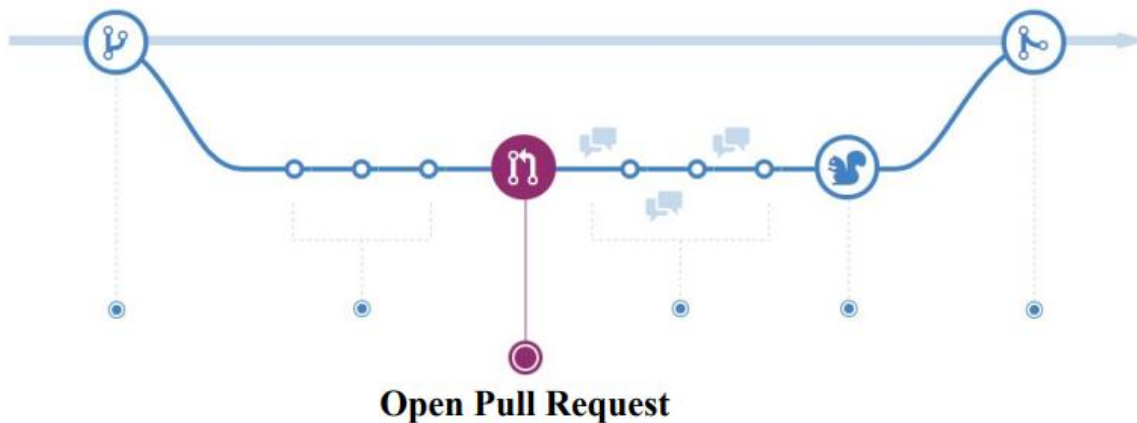


- When you add/edit/delete a file, you must make a commit
This commit is added to your branch

Commit Message

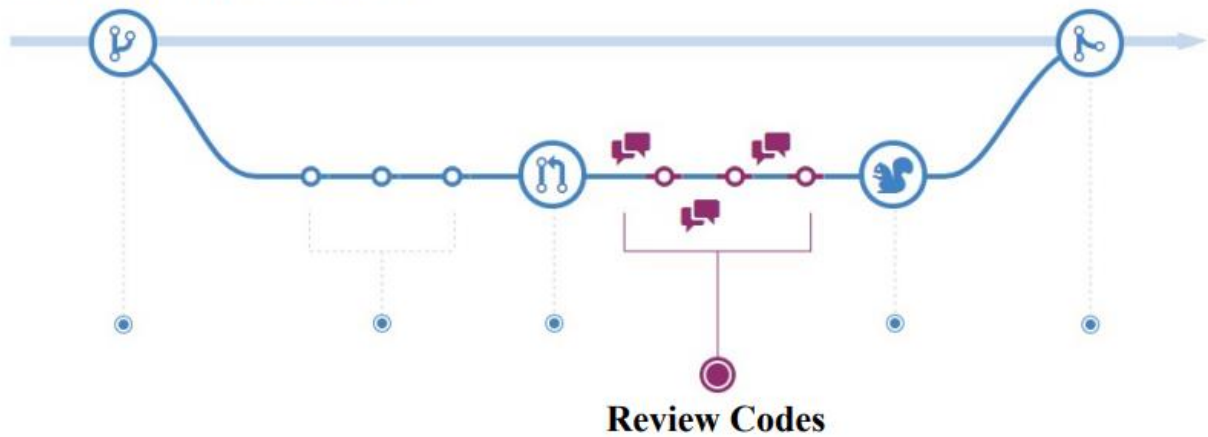
- Writing clear commit message (Important!)

Initiate Discussion



- Open a Pull Request at any time, even without commit
- Receive feedback from team or specific people

Discuss and Review

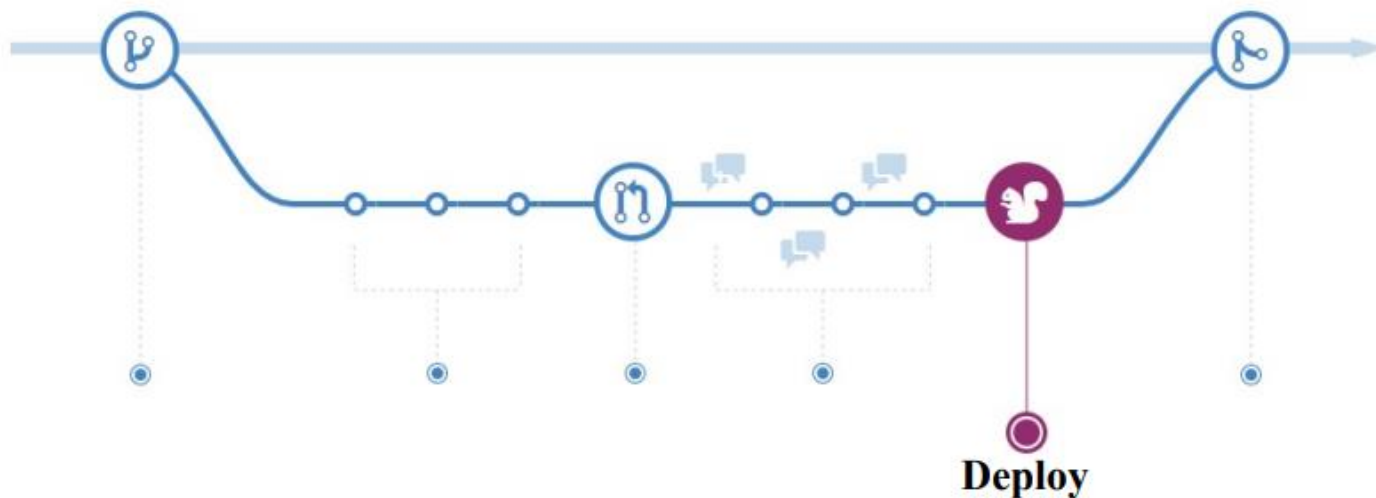


- Questions and comments about your changes

Further Changes

- You can add further commit for apply feedback (e.g. bug)

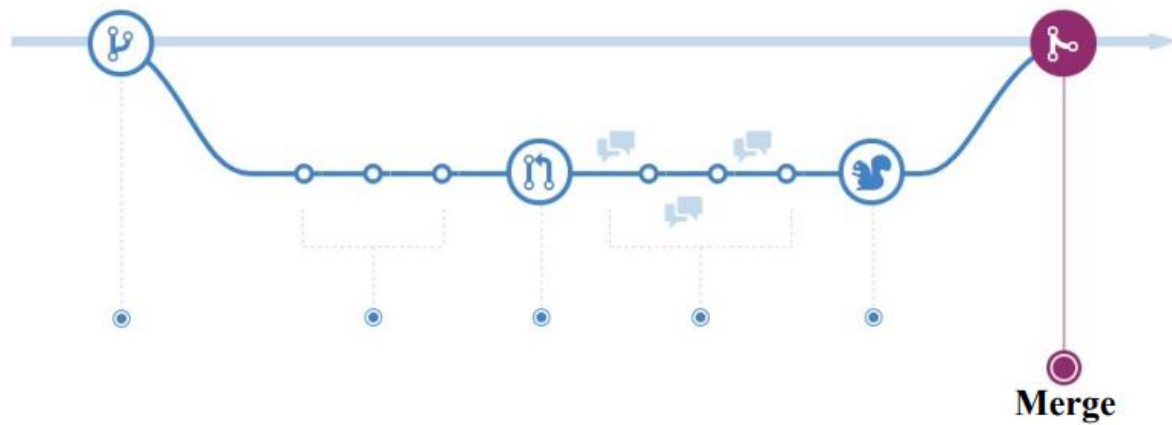
Final Testing



- After review, you deploy your changes to verify them in production

Deploy 는 테스트하는 것을 의미한다.

Merge Changes into Master Branch



- After merged, Pull Requests preserved a record of the historical changes to your code