 kim hanwoong 내용 추가함.

400ad71 on 15 Jan

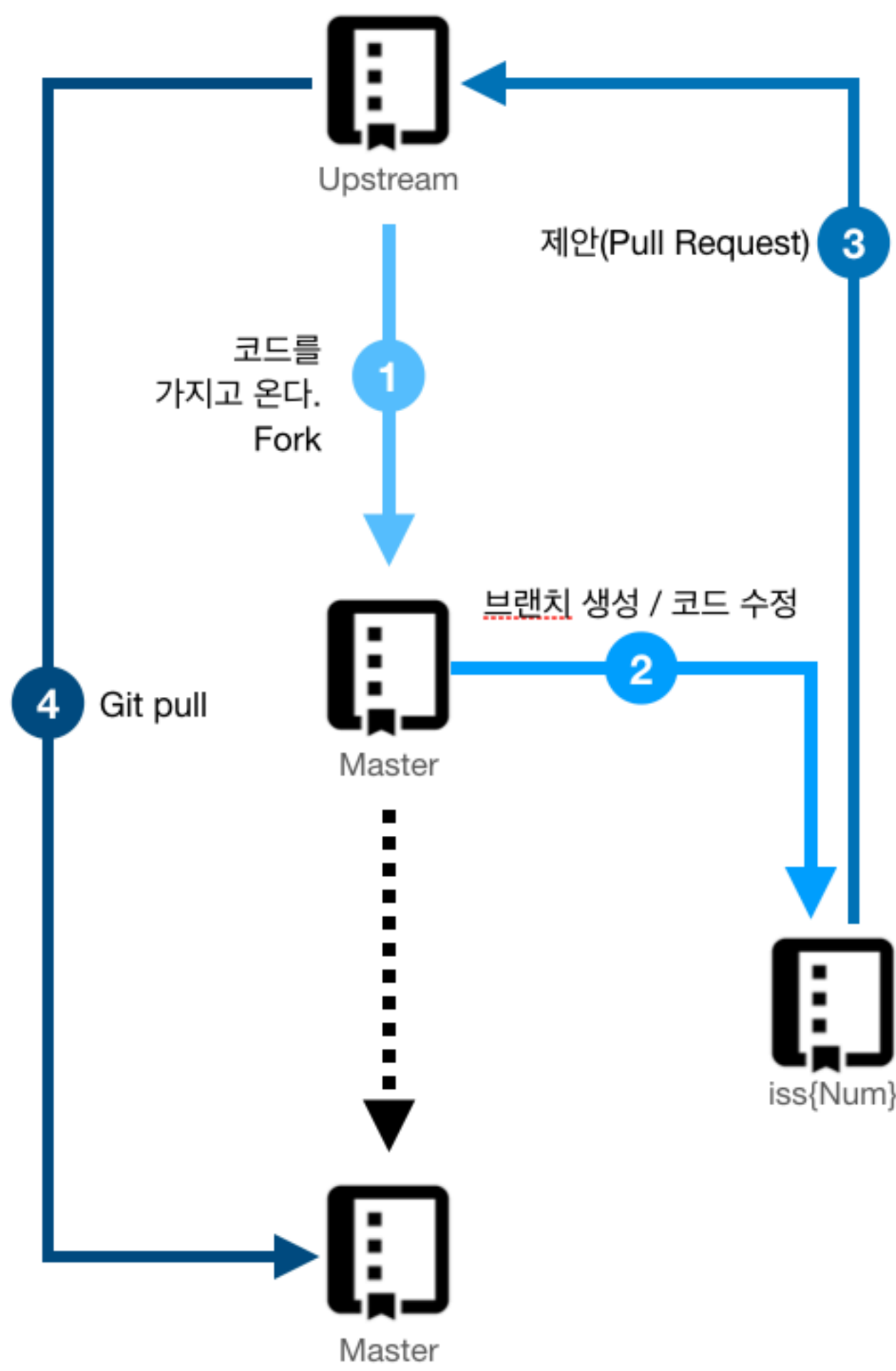
0 contributors

72 lines (59 sloc) | 2.74 KB

# Git Collaboration

회사에 다니다 보면 A개발자가 만든 기능, 함수, 코드더미를 B개발자가 다시 만드는 상황이 많이 발생합니다. 대부분 이 문제는 대화를 하지 않고 작업하거나 문제 해결에 대한 기록이 없어서인 경우가 가장 많습니다. 모든 프로젝트는 혼자 하는 일이 아니라면 최초부터 협업모델로 프로그래밍을 시작하는 것이 좋습니다.

이번 시간에는 여러 개발자가 하나의 리포지토리를 관리하기 위해 알아야 할 특징을 다루겠습니다.



## 협업방식

설명을 위해서 단체는 lazypic을 개인은 woong 이라는 예제로 작성되었습니다.

- 1. Github에 존재하는 프로젝트를 Fork 합니다.
- 2. Fork한 자신의 리포지토리를 생성합니다.

```
$ git clone https://github.com/woong/repo.git
```

3. git pull 한 리포지터리에서 up stream 설정을 합니다.

```
$ git remote add upstream https://github.com/lazypic/repo.git
```

4. upstream에서 최신 코드를 다운로드 받습니다. fetch는 오직 다운로드만 받는 명령어 입니다.

```
$ git fetch upstream
```

5. branch를 만들고 해당 이슈로 개발을 진행합니다.

```
$ git branch iss53
```

6. 코드를 작성하고 이슈로 만든 branch에 push 합니다.

```
$ git push origin iss53
```

7. 이슈와 함께 제안(Pull Request) 합니다.

8. 제안이 받아들여지면 master branch로 바꾸고 upstream에서 git fetch 하여 최신의 master branch 를 유지합니다.

```
$ git checkout origin master
$ git fetch upstream
```

9. branch가 많아지면 관리하기 힘드니 완료된 issue branch는 제거합니다.

```
$ git branch
$ git branch -d iss53
```

## 팁: Master 브랜치에 Push하는 것을 막는 방법

push 하기전에 master 브랜치인지 체크하기 위해서 아래 경로에 파일을 생성해줍니다.

```
.git/hooks/pre-push
```

pre-push 파일명으로 .git/hooks 경로에 스크립트를 생성하게 되면 git push 전에 해당 내용의 스크립트를 실행합니다.

```
#!/bin/bash

current_branch=$(git symbolic-ref --short HEAD)

if [ $current_branch == "master" ]
then
    echo "prevent master push."
    exit 1
fi

exit 0
```

자신이 master에서 자주 push 하는 습관이 있다면 이 스크립트를 활용해 보세요.

## Reference link

- <https://blog.ghost.org/prevent-master-push/>
- upstream 사용 예) : <https://opentimelineio.readthedocs.io/en/latest/tutorials/contributing.html>
- merge type : <https://help.github.com/articles/about-merge-methods-on-github/>