

# ILE3-031 Branch 관리

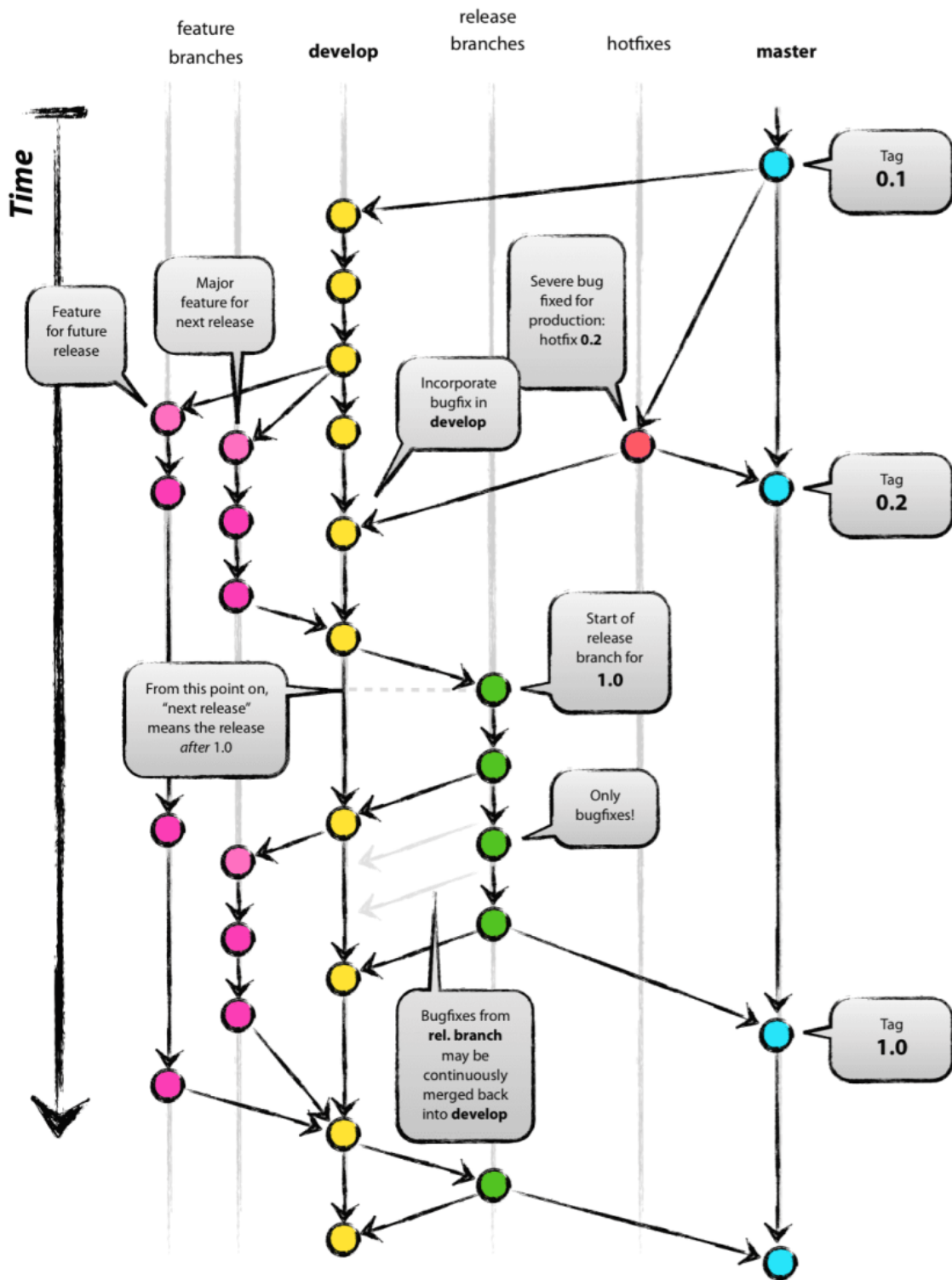
소프트웨어 개발 주기를 git의 Branch를 이용하여 관리하는 방법을 학습

## git flow

Vincent Driessen이 제시한 브랜치 관리 방법

아래의 5가지의 branch를 사용

- **master** : 기준이 되는 브랜치로 제품을 배포하는 브랜치
- **develop** : 개발 브랜치로 개발자들이 이 브랜치를 기준으로 각자 작업한 기능들을 합병(Merge)
- **feature** : 단위 기능을 개발하는 브랜치, 개발이 완료되면 develop 브랜치에 합병
- **release** : 배포를 위해 master 브랜치, QA(품질검사)를 하기위한 브랜치
- **hotfix** : master 브랜치로 배포를 했는데 버그가 생겼을 때 긴급 수정하는 브랜치



## 시나리오 [🔗](#)

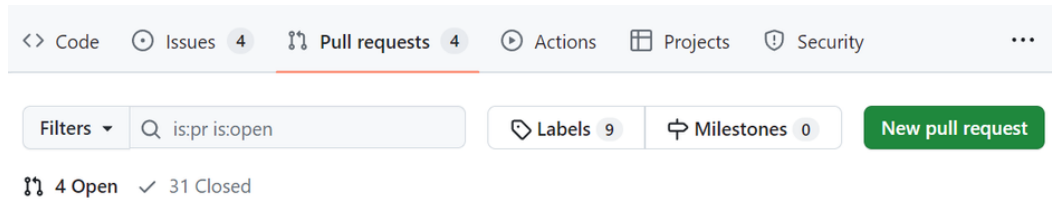
1. **master** 브랜치에서 시작
2. 동일한 commit에서 **develop** 브랜치를 생성, 개발자들은 이 develop 브랜치에서 개발을 진행
3. 회원가입, 장바구니 등의 기능 구현이 필요할 경우
  - a. A개발자는 develop 브랜치에서 **feature** 브랜치를 하나 생성해서 회원가입 기능을 구현
  - b. B개발자는 develop 브랜치에서 **feature** 브랜치를 하나 생성해서 장바구니 기능을 구현
4. 완료된 feature 브랜치는 검토를 거쳐 다시 **develop** 브랜치에 병합(Merge)

- 모든 기능이 완료되면 develop 브랜치를 **release** 브랜치에 병합  
그리고 QA(품질검사)를 하면서 보완점을 보완하고 버그를 수정
- 모든 것이 완료되면 이제 release 브랜치를 **master** 브랜치와 **develop** 브랜치로 병합  
**master** 브랜치에서 버전추가를 위해 태그를 하나 생성하고 배포
- 배포를 했는데 미처 발견하지 못한 버그가 있을 경우 **hotfixes** 브랜치를 만들어 긴급 수정 후 태그를 생성하고 바로 수정 배포

## Pull Request

develop, release, master 브랜치와 같이 작업이 종료되어 코드가 병합이 되거나 다른 브랜치로 병합을 하기 위하여 Project 담당자에게 요청을 하는 작업

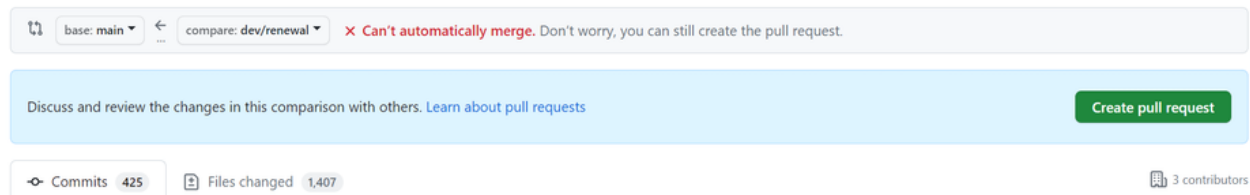
- Repository 페이지에서 Merge Request 클릭 후 초록색 버튼(New pull request) 클릭



- 병합할 브랜치와 병합 될 브랜치를 설정 후 초록색 버튼 (Create pull request) 클릭

### Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).

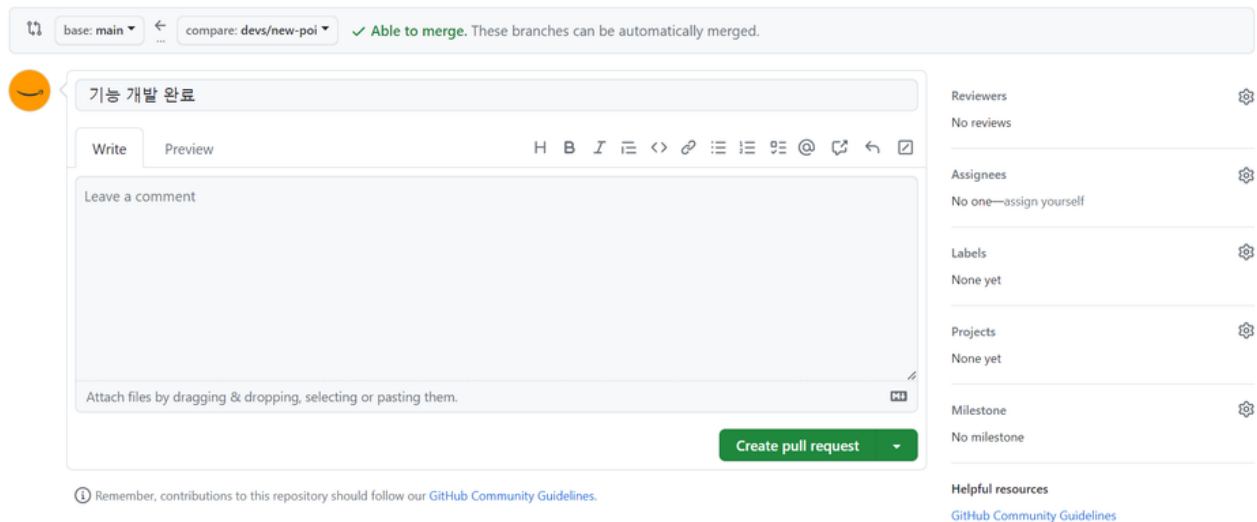


- Merge에 대한 정보 및 리뷰어 설정

Assigner에 merge request를 처리할 담당자를 지정  
그리고 하단의 초록색 버튼 클릭(Create pull request)

- 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](#).



## 5. Code Review 및 Branch 병합

Assigner가 생성된 pull request를 확인 및 검토 후 병합을 수행

### 기능 개발 완료 #1

The screenshot shows a GitHub Pull Request (PR) titled "기능 개발 완료 #1" (Feature Development Complete #1). The PR is created by "inavi-jjcheon" and aims to merge 7 commits from the "devs/new-poi" branch into the "test" branch. The interface includes a header with "Open" status and a link to open the PR. Below the header, there's a section for "Conversation" (0), "Commits" (7), "Checks" (1), and "Files changed" (27). A comment by "inavi-jjcheon" states "No description provided." Below the comment, a list of 7 commits is shown, including "POI API Service 추가", "필요 없는 Dockerfile 제거 및 Github Action 수정", "Fork 환경을 위한 Github Action 추가", "Github Action CI 수정", "POI DB 교체", "Provider 기능 확장", and "Delete 기능 추가". On the right side, there are sections for "Reviewers" (No reviews), "Assignees" (No one—assign yourself), "Labels" (None yet), "Projects" (None yet), "Milestone" (No milestone), and "Notifications" (Unsubscribe). At the bottom, a status bar indicates "All checks have failed" (1 failing check) for the "Deploy Main / build (push)" action. A green checkmark indicates "This branch has no conflicts with the base branch". A button "Merge pull request" is visible, along with a link to "GitHub Actions: Speed up build times with larger GitHub-hosted runners".

## 과제

이전 수업에서 사용한 Repository에 아래의 작업을 수행

- feature Branch 생성
- README.md 파일 생성
- 변경 사항 commit
- feature Branch를 main Branch로 병합 요청 (assigner는 자신)
- pull request를 처리하여 feature branch의 변경 사항을 main branch에 병합