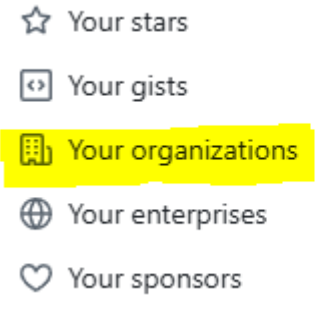
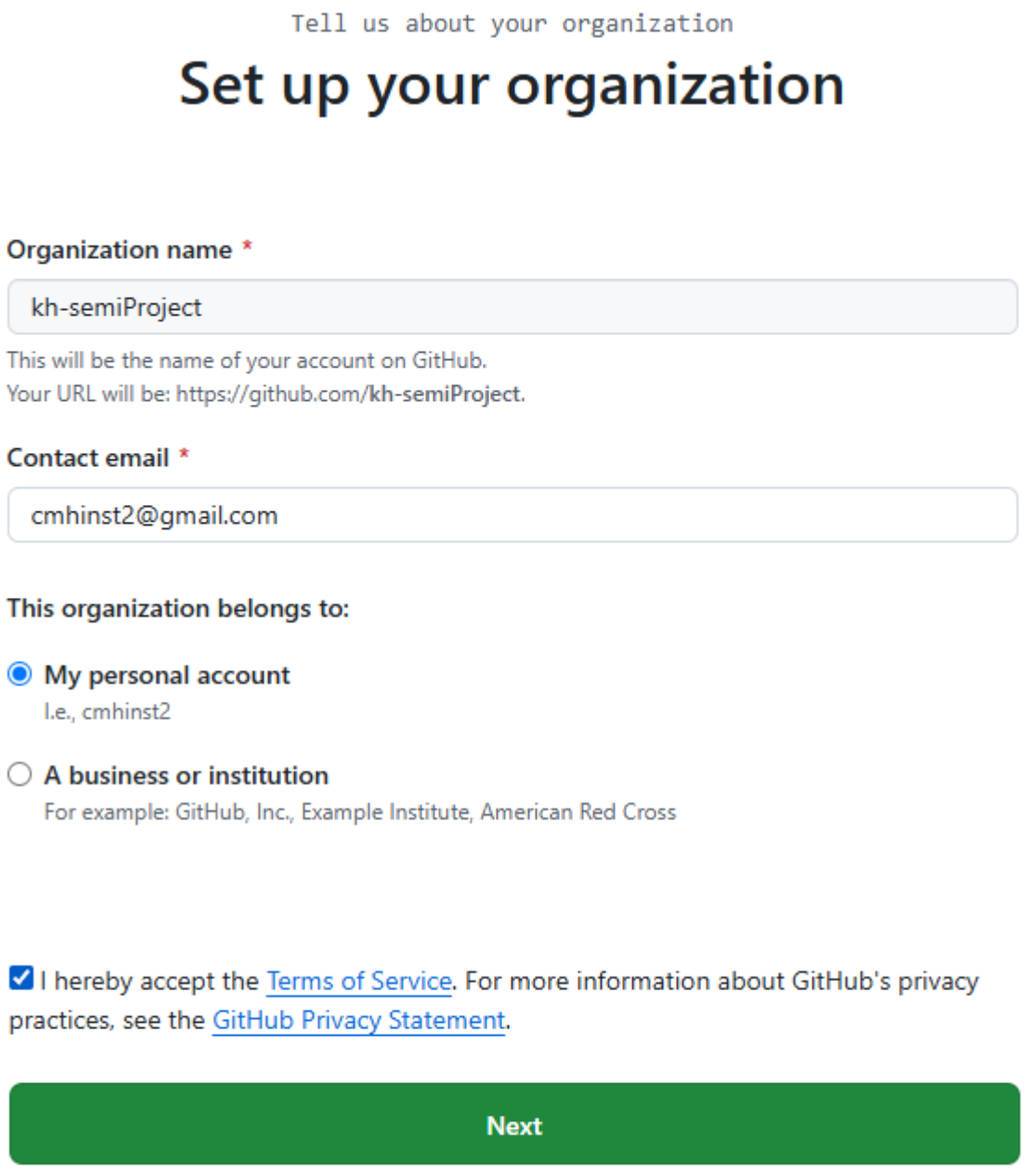
GitHub에서 Organizations 설정과 팀 Git 사용법

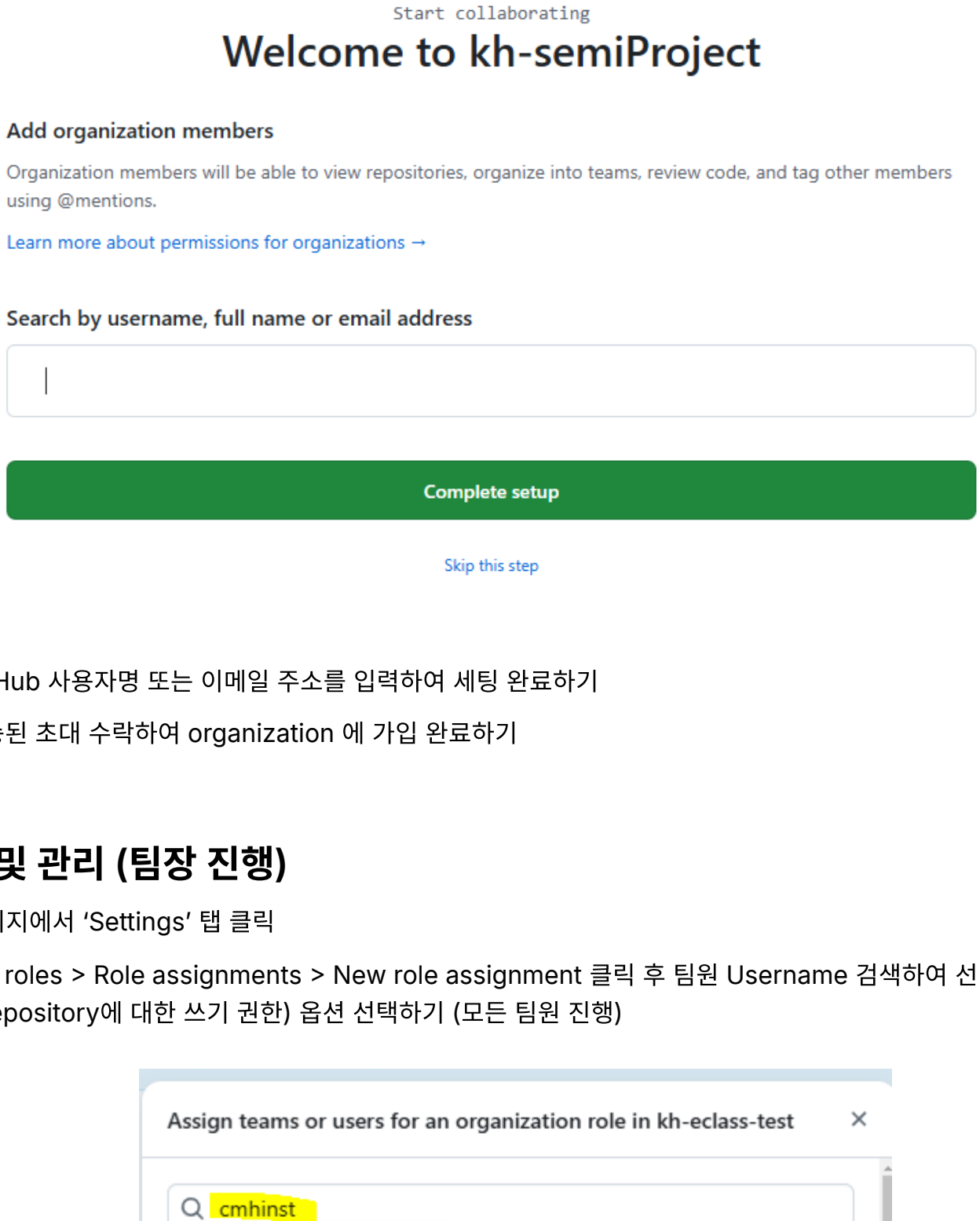
1. Organizations 생성하기(팀장 진행)

- GitHub 계정에 로그인
- 오른쪽 상단의 프로필 아이콘을 클릭하고 'Your organizations'를 선택



- 'New organization'을 클릭
- 무료(Free) 플랜을 선택
- 조직 이름과 연락처 이메일을 입력

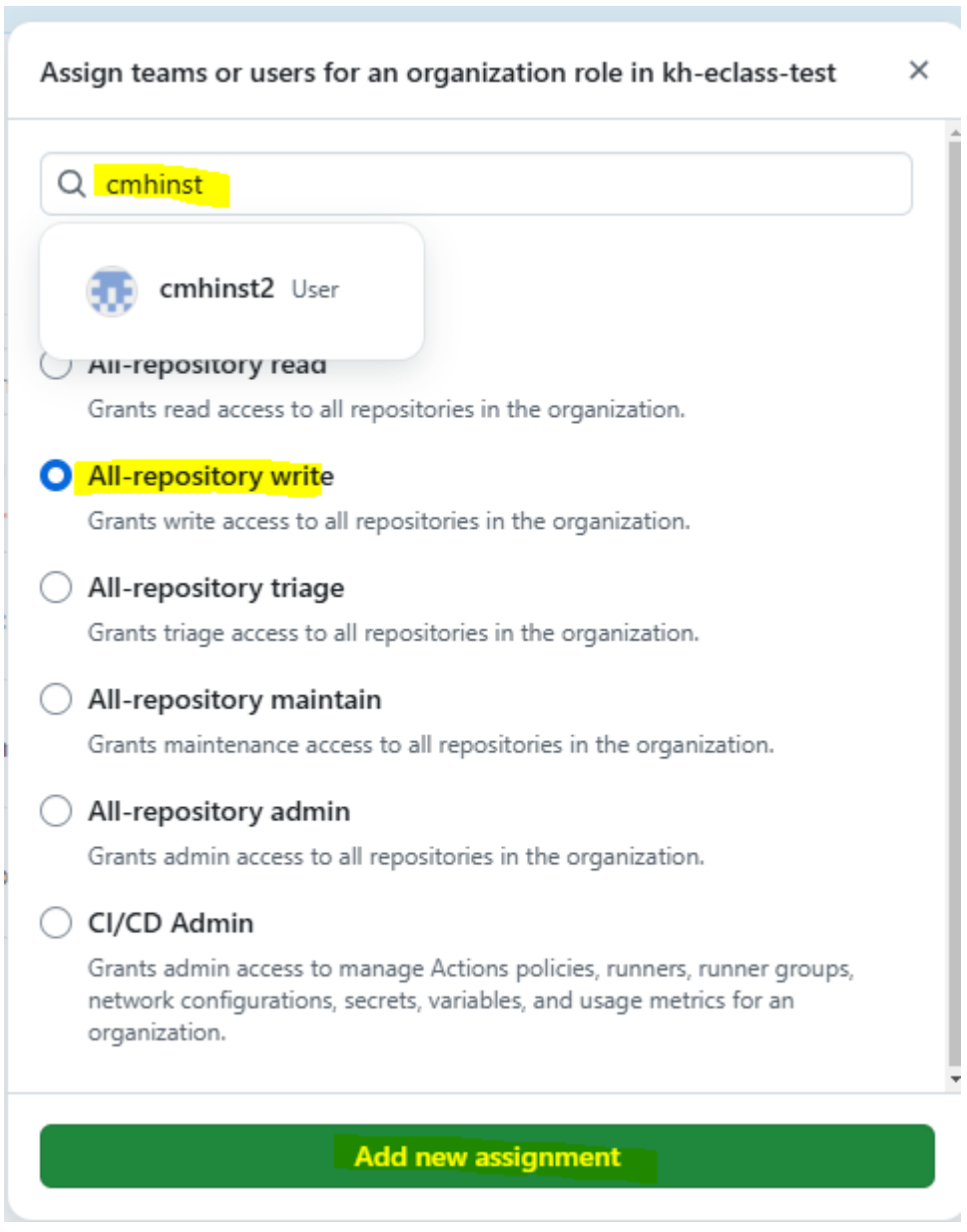




- 초대할 팀원의 GitHub 사용자명 또는 이메일 주소를 입력하여 세팅 완료하기
- 팀원들은 각자 전송된 초대 수락하여 organization 에 가입 완료하기

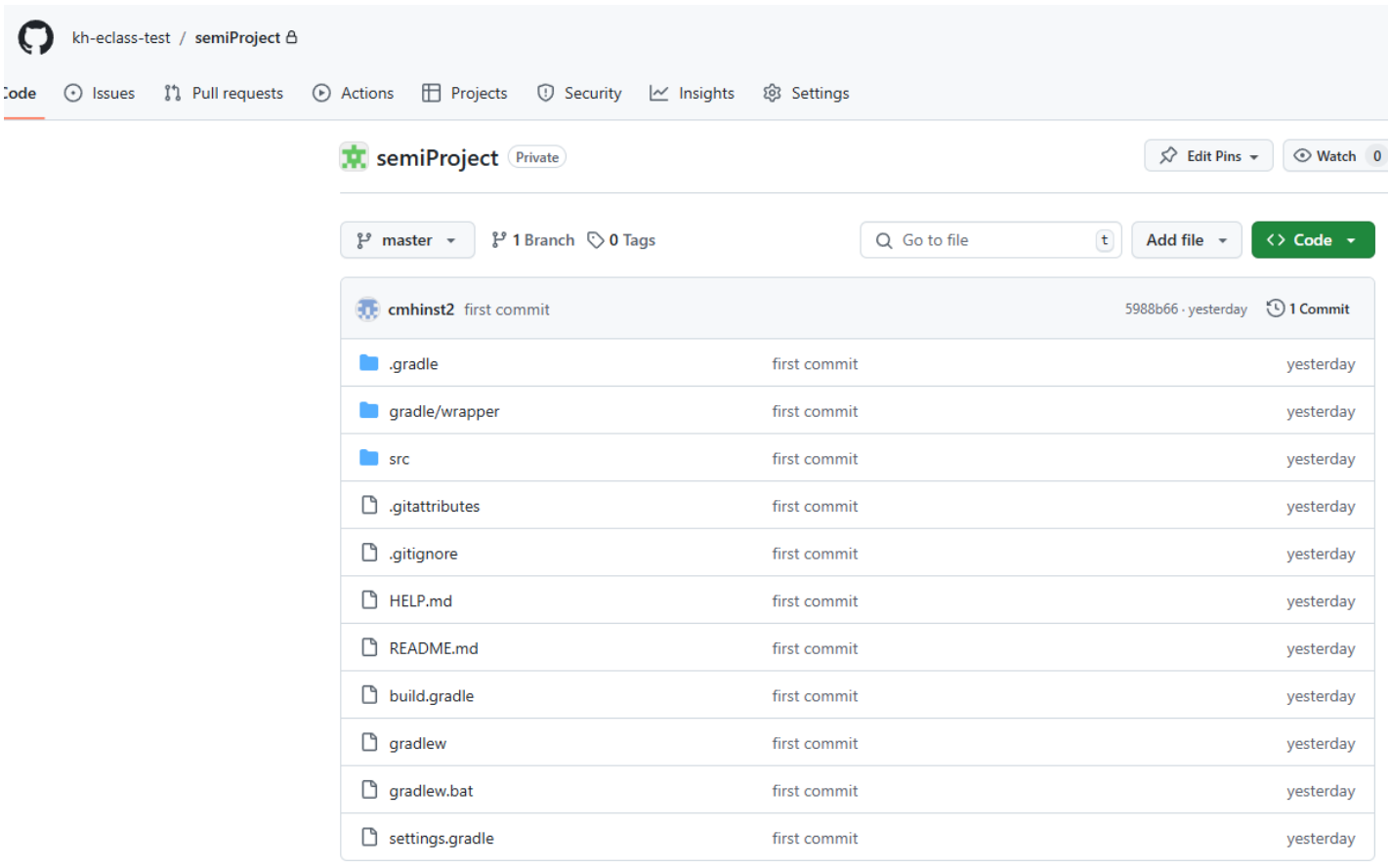
2. 저장소 생성 및 관리 (팀장 진행)

1. organization 페이지에서 ‘Settings’ 탭 클릭
    - organization roles > Role assignments > New role assignment 클릭 후 팀원 Username 검색하여 선택 후 All-repository write(모든 Repository에 대한 쓰기 권한) 옵션 선택하기 (모든 팀원 진행)



2. organization 페이지에서 'Repositories' 탭을 클릭

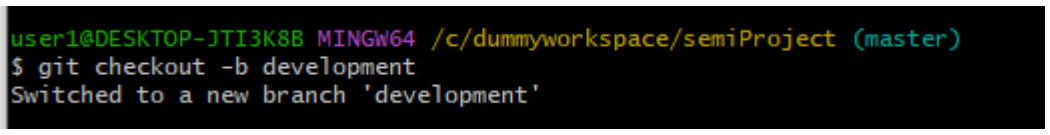
3. 'New repository' 버튼을 클릭하여 새 저장소를 생성
4. 로컬 레파지토리에 있는 팀 프로젝트 원격 레파지토리에 연결하기
5. first commit 까지 마치기



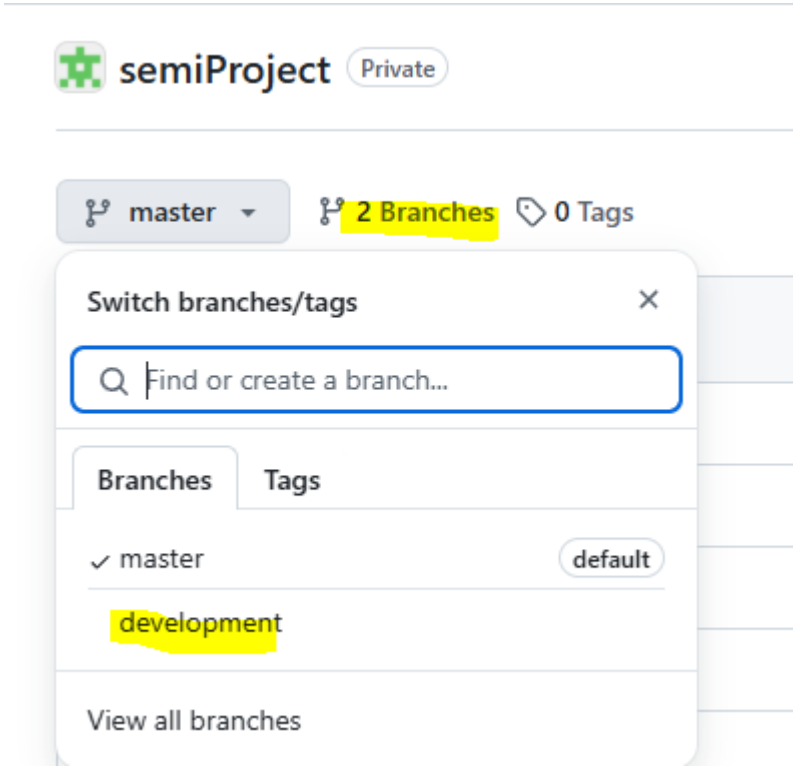
3. 브랜치 나누기

1. master 브랜치 : 최종 결과물을 병합할 브랜치로 최종 배포 시 기준이 되는 브랜치 (기본 생성 되어있음) - 팀장이 진행
2. development 브랜치 : 개발 시 사용되는 중간 병합 브랜치 (master → development 생성) - 팀장이 진행

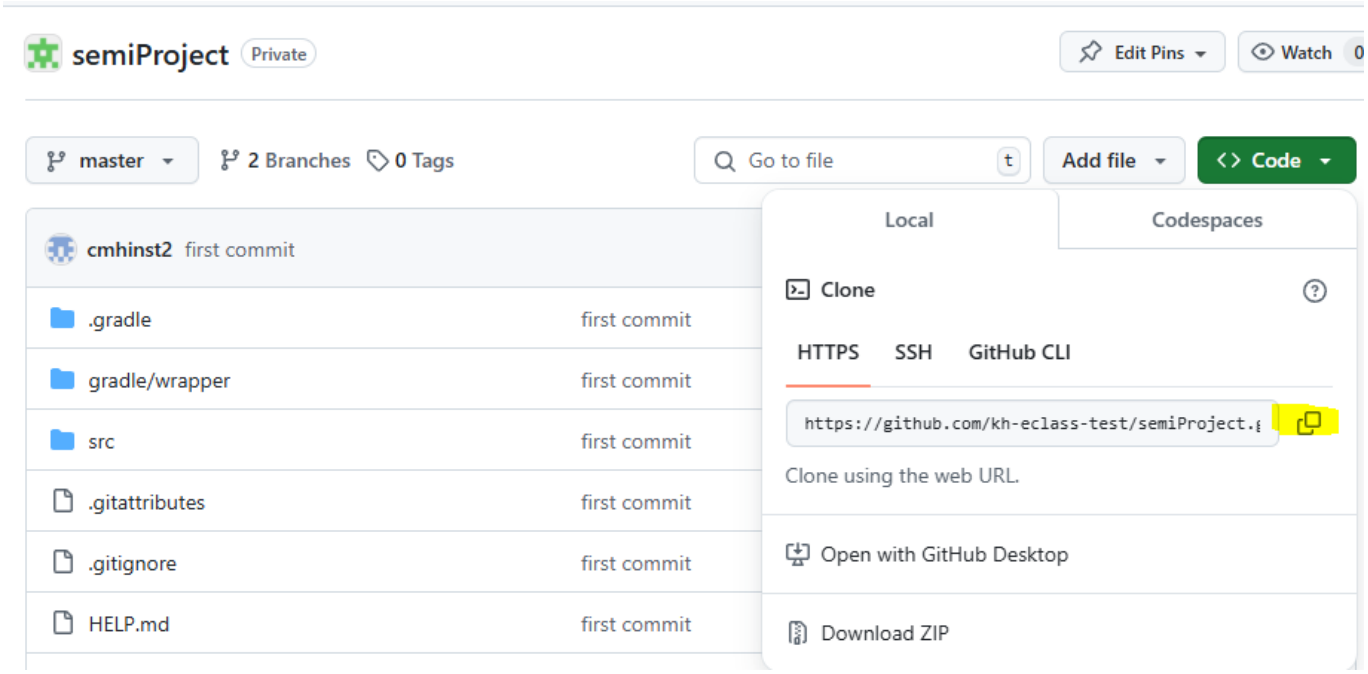
```
# master branch 에서 진행
$ git checkout -b development # development branch 생성 후 이동
```




```
# 로컬 저장소에 생성된 development branch를 원격으로 올림
$ git push --set-upstream origin development
```



3. 팀원은 모두 Github에서 Repository를 각자 컴퓨터에 Clone 하기




```
$ git clone 원격레파지토리주소
$ git remote update #원격레파지토리 업데이트(브랜치 변경사항까지)
$ git branch -a #로컬&원격레파지토리의 모든 브랜치 조회
$ git checkout -t remotes/origin/development  # 원격레파지토리에 있는 development 브랜치를 로컬에 생성 후 브랜치 이동
```

4. development 브랜치에서 본인 이름(영어로) branch 새로 생성

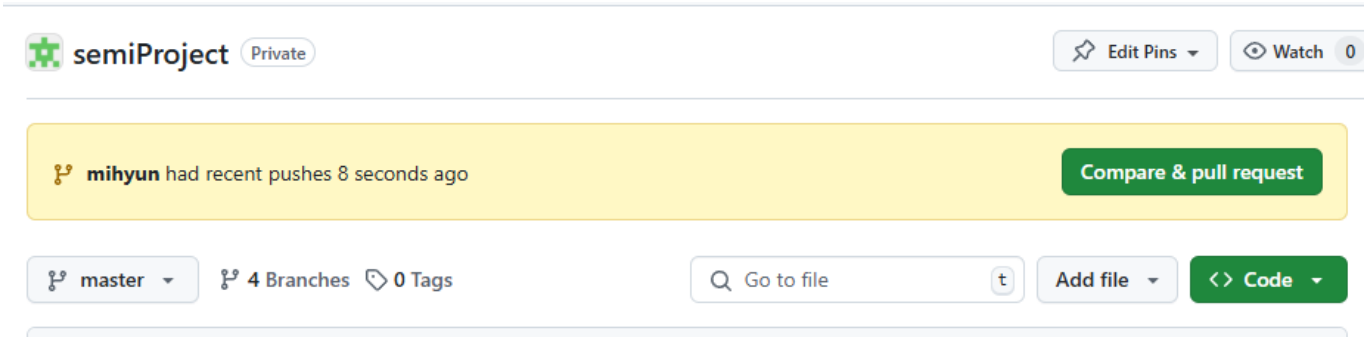
```
$ git checkout -b zzangu # 짱구라는 팀원이 본인이름의 브랜치를 로컬에 생성 후 브랜치 이동

$ git push --set-upstream origin zzangu # 원격에 해당 브랜치 push
```

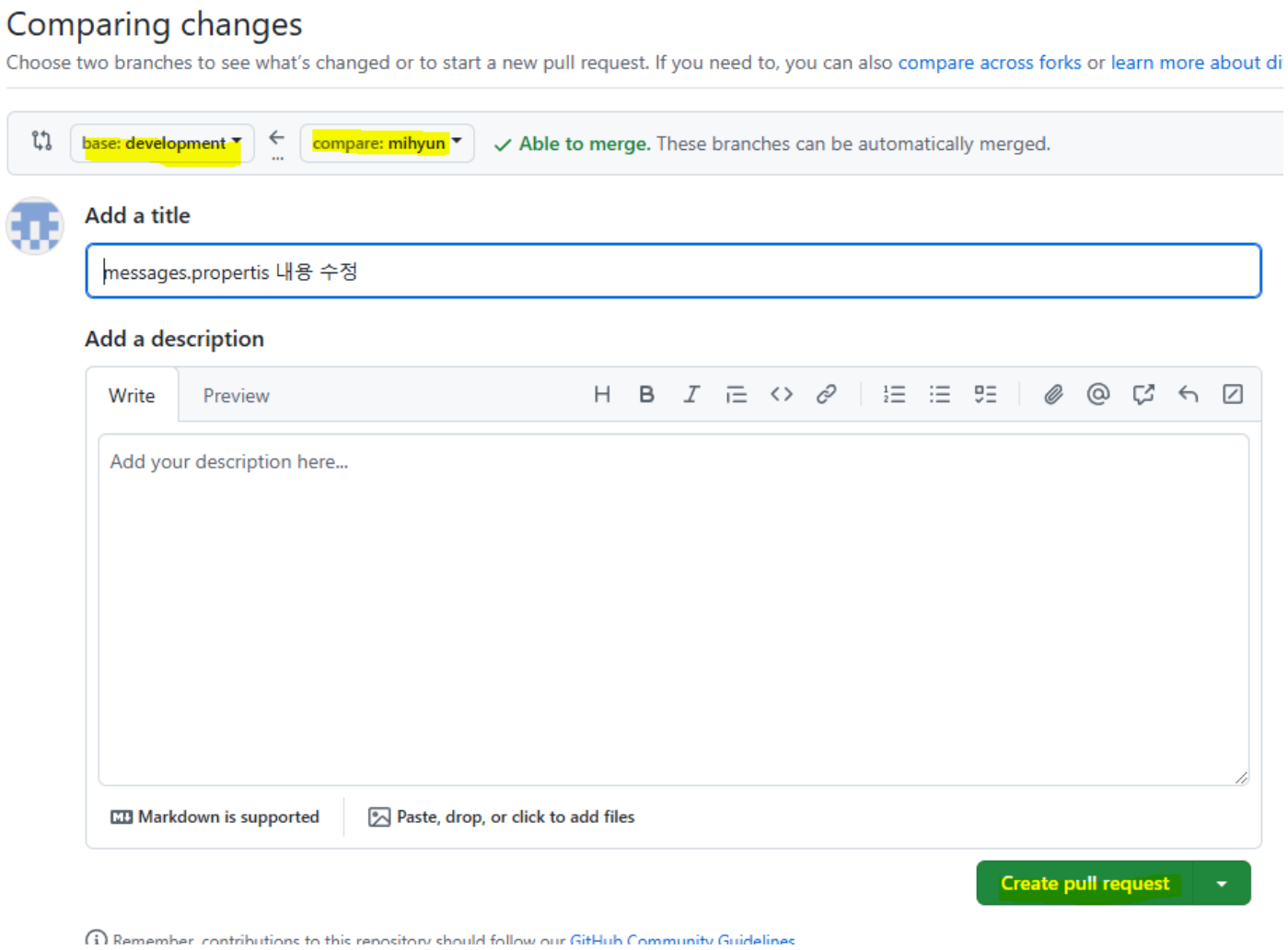
- **팀원은 모두 개인 브랜치에서 개별적으로 작업하기**
  - **pull → commit/push 평소에 하던것처럼 "본인이름브랜치"에서 작업 진행**

4. Pull Request 와 Merge

1. 팀원들은 각자 "로컬 레파지토리의 본인 브랜치"에서 개별 작업 후 push 를 통해 "원격 레파지토리의 본인 브랜치"에 commit 내역을 올림.
2. 아래 이미지와 같이 원격 레파지토리에 특정 브랜치에서 push 된 사항이 있다는 알림 확인 가능 *(간혹 알림이 안뜨는 경우도 있으나, 당황하지 말고 상단 탭 중 Pull requests 클릭 후 New pull request 클릭하면 됨)*

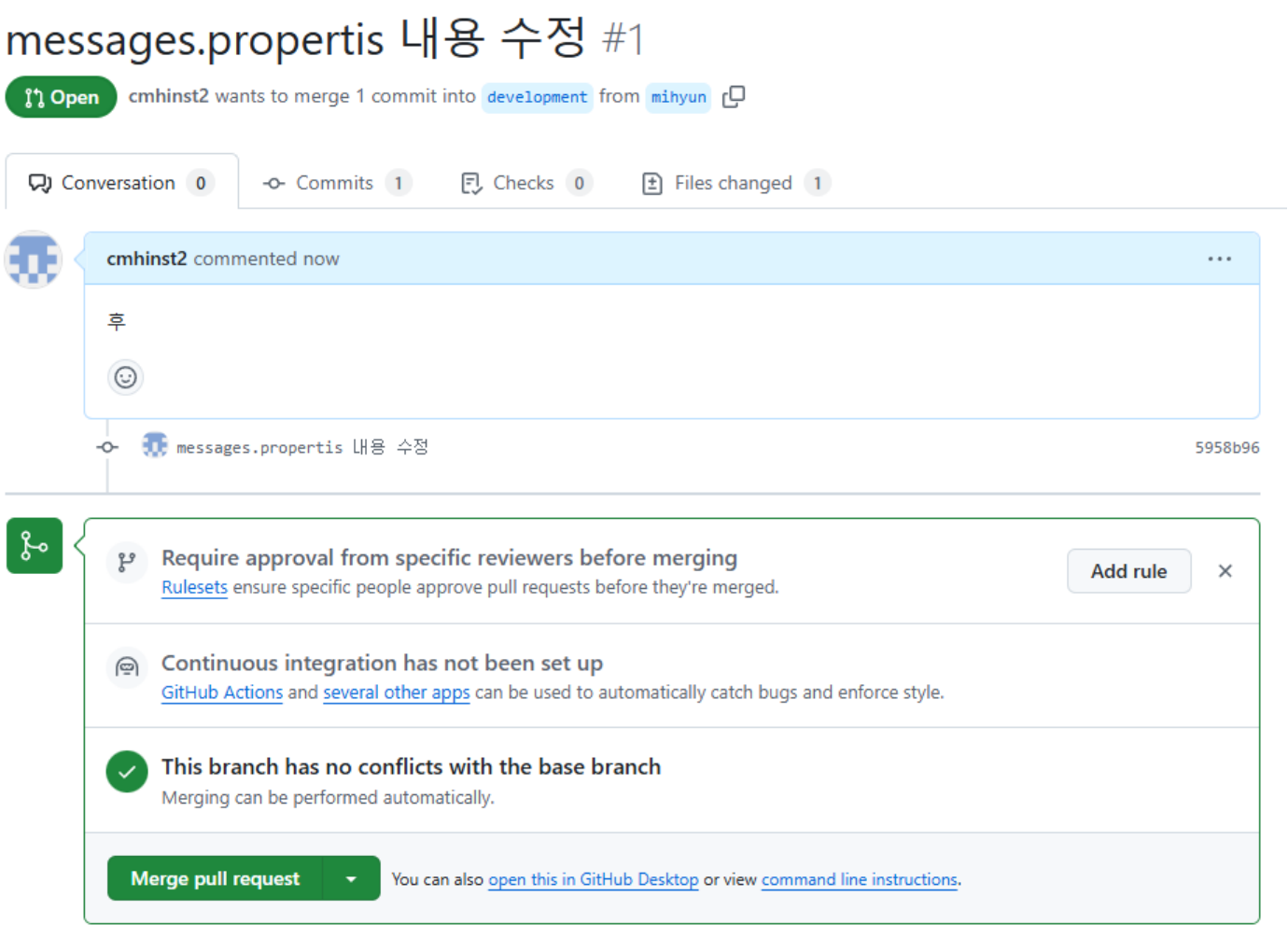


3. Compare & pull request 클릭 하면 아래와 같은 화면으로 이동

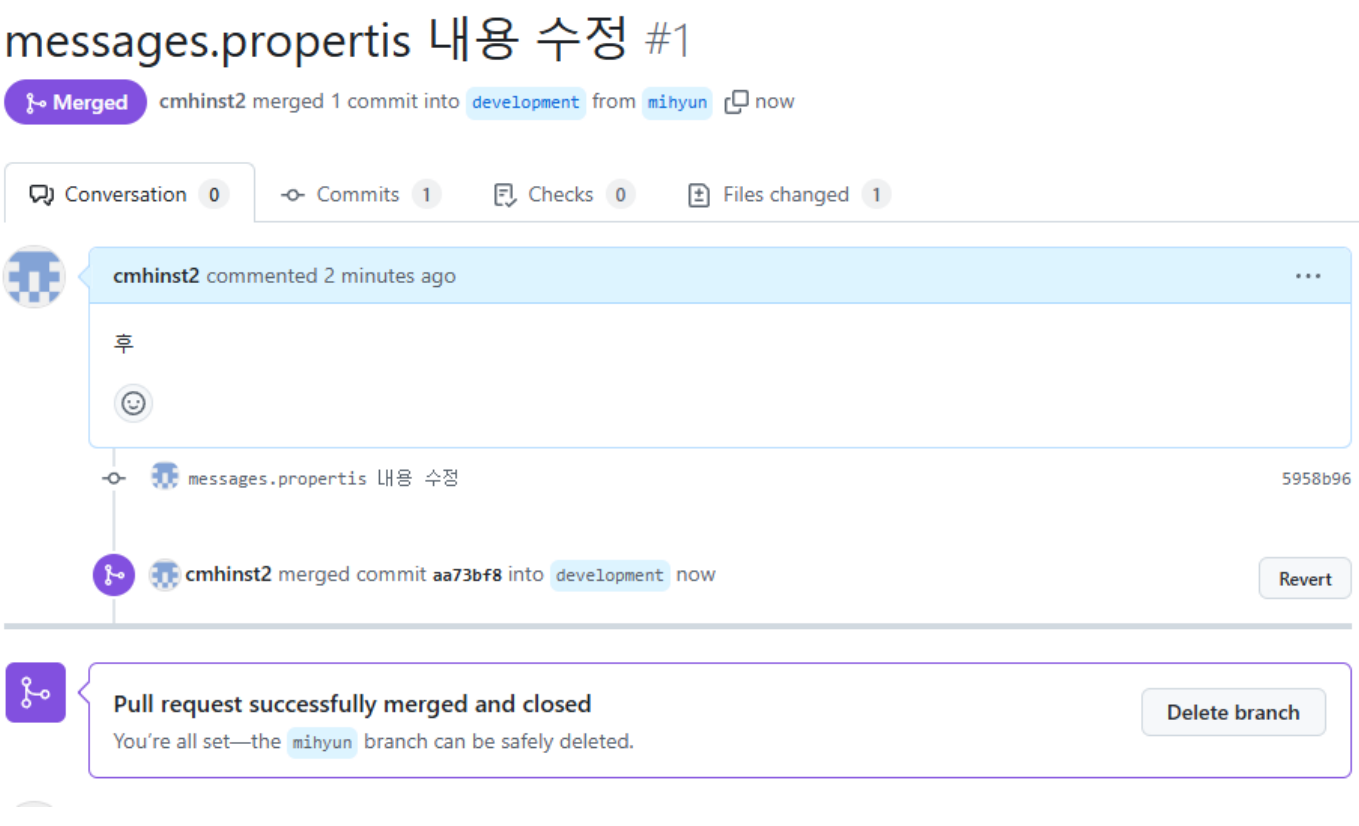


4. **이 화면에서 브랜치를 잘 선택해야함 (병합대상브랜치 ← 본인브랜치)**
    - development 브랜치에 모든 팀원이 각자 작업한 코드를 병합해야함. (병합대상이 됨)
    - pull Request 는 “A라는 브랜치에 있는 변경사항을 B에 병합해주세요” 라는 요청임.

5. Create pull request 까지 클릭하고 나면 아래와 같은 화면으로 이동함



6. 코드상에 충돌없이 병합이 가능한 상태라면 위 화면과 같이 Merge pull Request 라는 버튼이 활성화 됨.
7. 해당 버튼을 클릭하면 mihyun 브랜치에서 작업된 코드 내용이 development 브랜치에 병합이 완료됨.



8. development에 새로운 변경사항이 생겼으면 모든 팀원은 각자 컴퓨터에서 원격 development의 병합된 가장 최근 코드를 pull 받고 다시 각자 브랜치에서 작업 진행..

```
$ git pull origin development # 원격 레파지토리에 있는 development 브랜치의 모든 변경사항을 pull 받음
(각자 이름의 브랜치에서 진행할 것)
```