8. Github 알아보기와 실습

- 1. Github Docs
- 2. Local Git 과 Github Repository 연동
- 3. PR 활용하기
  - PR 템플릿 만들기
  - PR 생성하기
- 4. Github Page
- 5. Github Release
- 6. Github Package

1. Github Docs

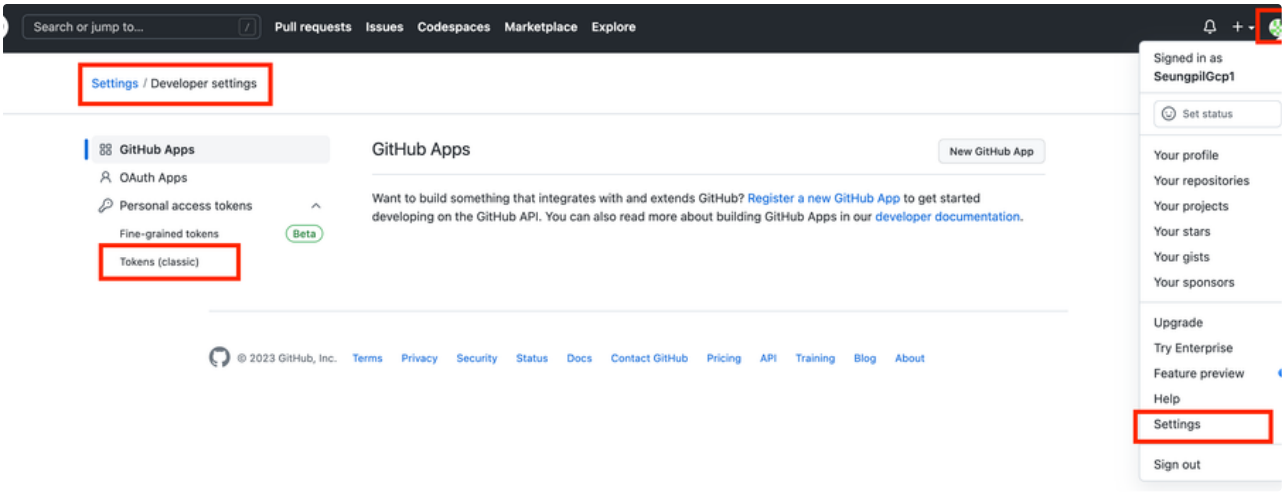
GitHub.com 도움말 문서

2. Local Git 과 Github Repository 연동

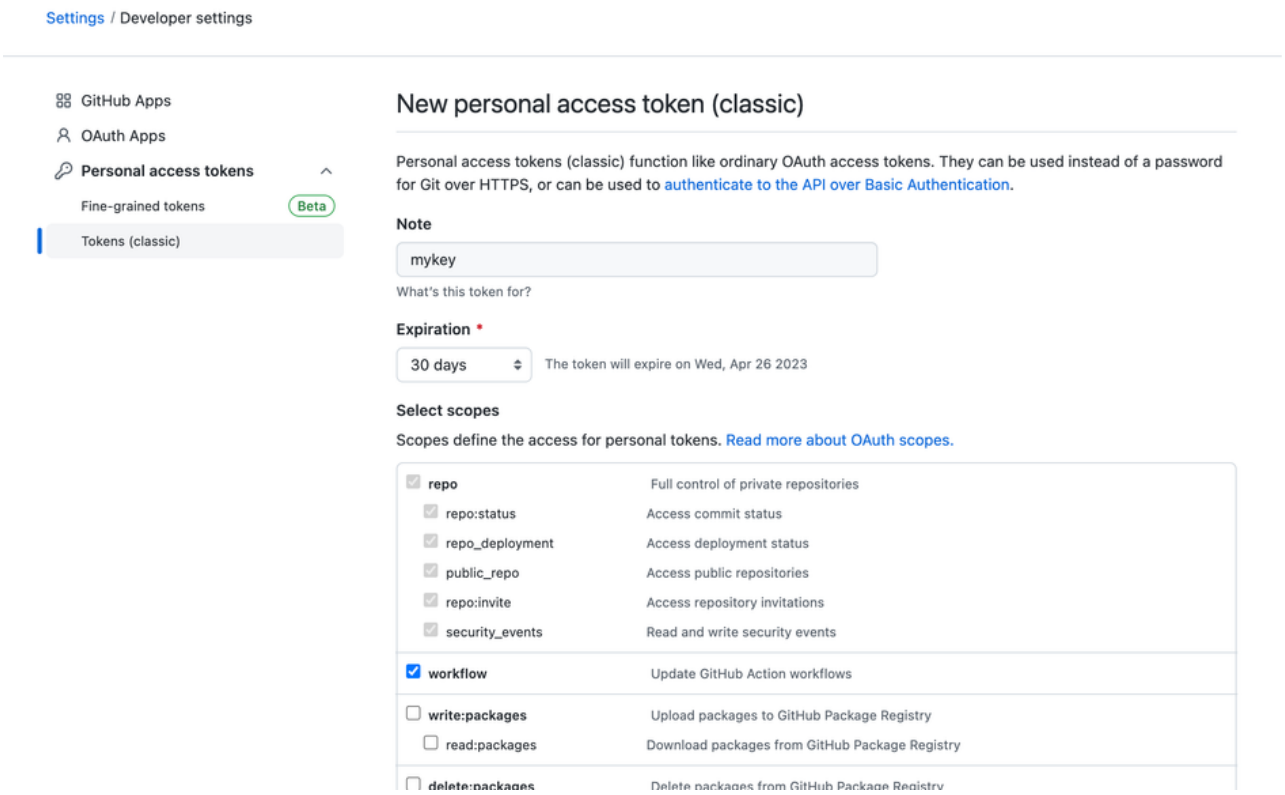
Github는 Git을 기반으로 한 코드 호스팅 플랫폼입니다. 이는 개발자들에게 코드 관리 및 협업을 쉽게 할 수 있는 기능을 제공합니다.

실습은 여전히 CodeSpace 기반으로 진행합니다. 실제로 Local Git 에서 Github Repository 를 Remote 저장소로 사용하고 싶으시다면 다음 절차로 Token 을 발급받아 진행할 수 있습니다.

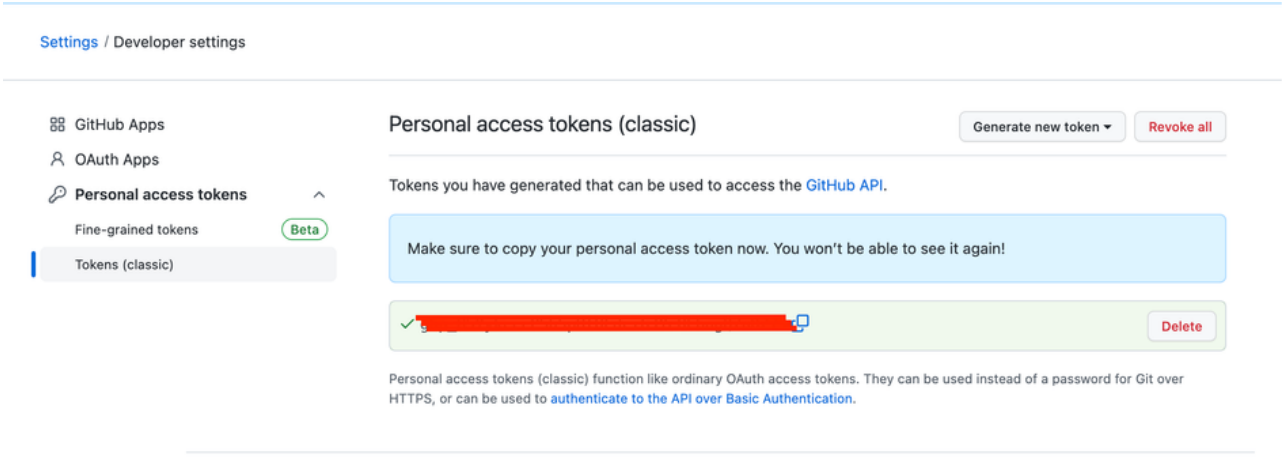
- 우측 상단 프로필 > Settings > Developer settings > Tokens



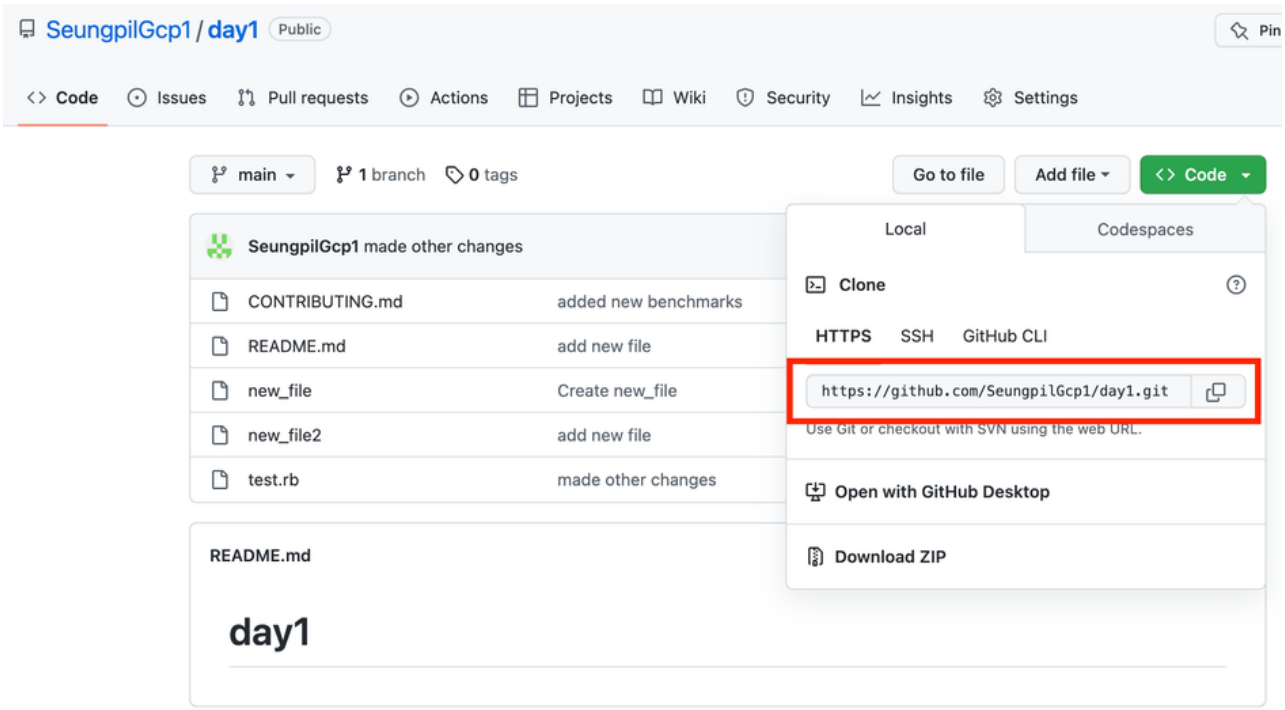
- 토큰 이름과 권한 설정



- 생성된 토큰 저장



- Repository > Code > Https 주소 git clone



- Repository local git 과 연동

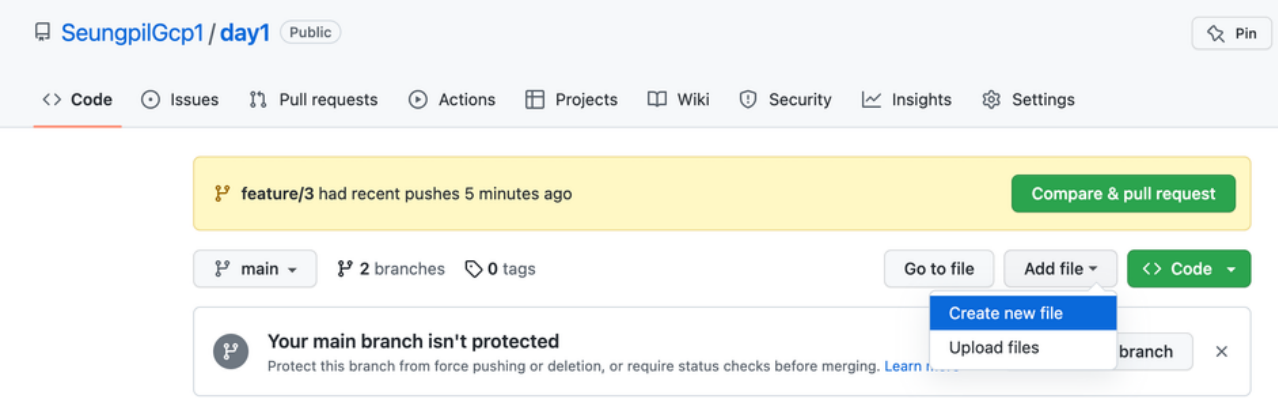
```
# 사용자 세팅
$ git config --global user.name "SeungpilGcp1"
$ git config --global user.email "seungpilgcp1@gmail.com"

# Repository 내려받고 token 설정하기
$ git clone https://github.com/SeungpilGcp1/day1
$ git remote remove origin
$ git remote add origin https://<token>@github.com/SeungpilGcp1/day1

$ git push --set-upstream origin main
Enumerating objects: 5, done.
```

```
Counting objects: 100% (5/5), done.
Delta compression using up to 10 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 282 bytes | 282.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/SeungpilGcp1/day1
   c64cc2e..65c5792  main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

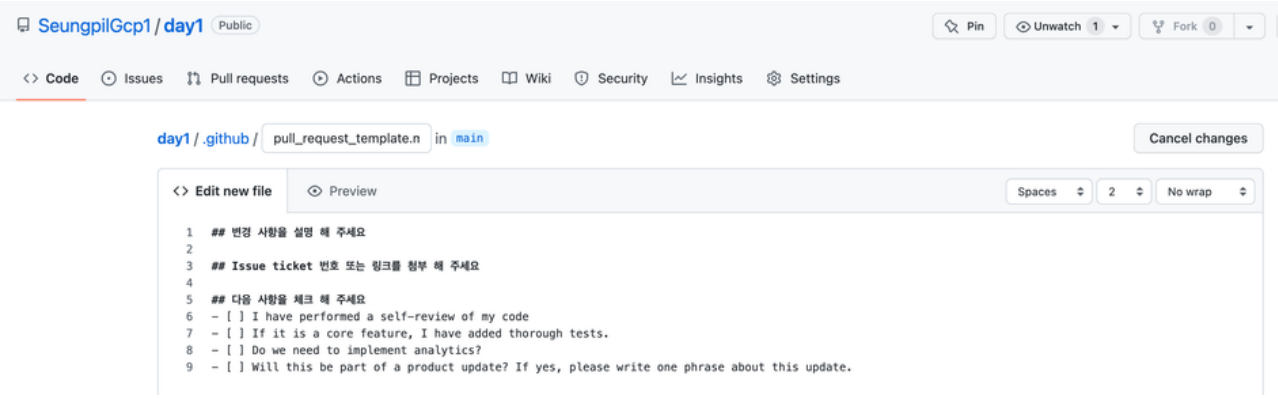
3. PR 활용하기

PR 템플릿 만들기

- Code > create new file



- .github/**pull_request_template.md** 파일 생성



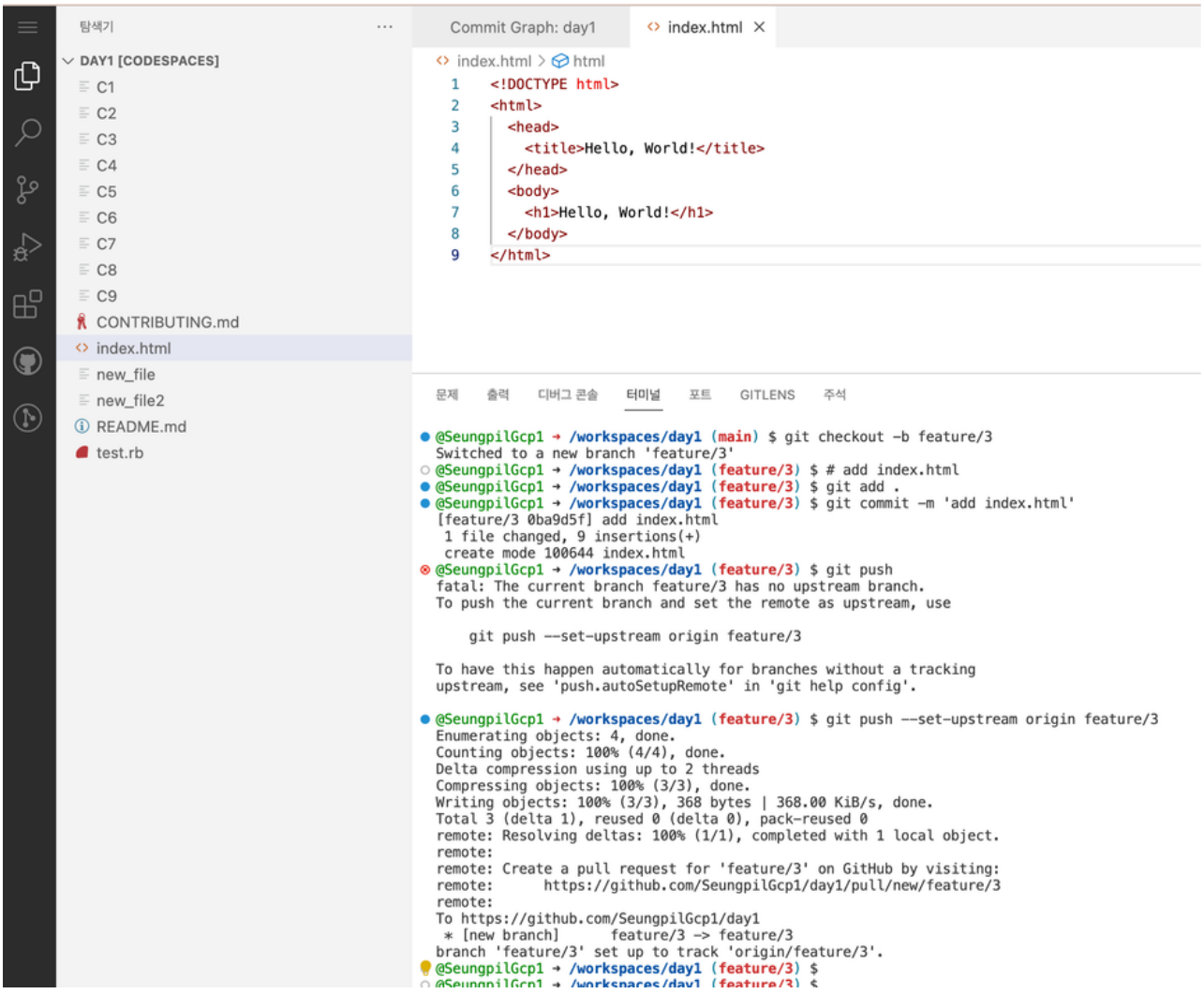

```
## 변경 사항을 설명 해 주세요

## Issue ticket 번호 또는 링크를 첨부 해 주세요

## 다음 사항을 체크 해 주세요
- [ ] I have performed a self-review of my code
- [ ] If it is a core feature, I have added thorough tests.
```

```
- [ ] Do we need to implement analytics?
- [ ] Will this be part of a product update?
```

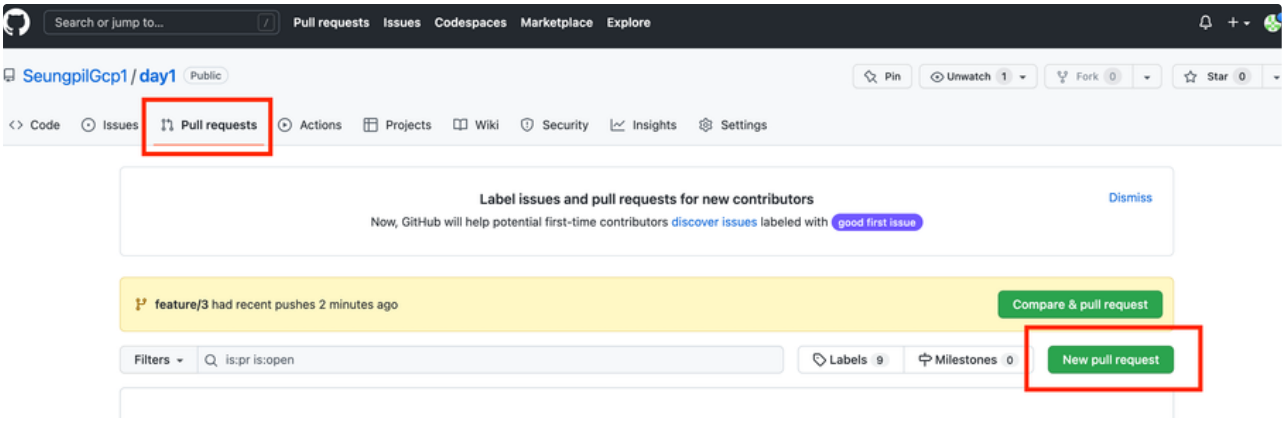
PR 생성하기

- 그림처럼 feature/3 브랜치를 생성후, index.html 을 만들어 origin 에 push

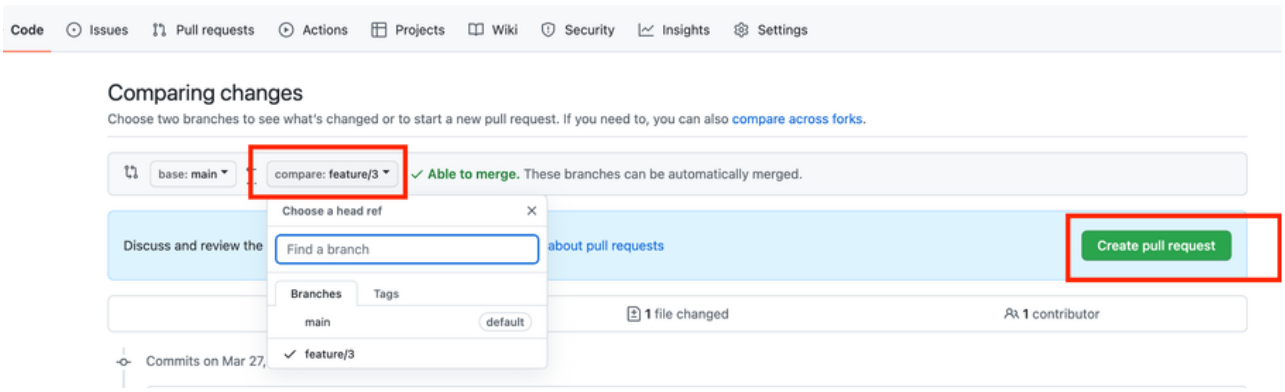
```
<!DOCTYPE html>
<html>
 <head>
   <title>Hello, World!</title>
 </head>
 <body>
   <h1>Hello, World!</h1>
 </body>
</html>
```



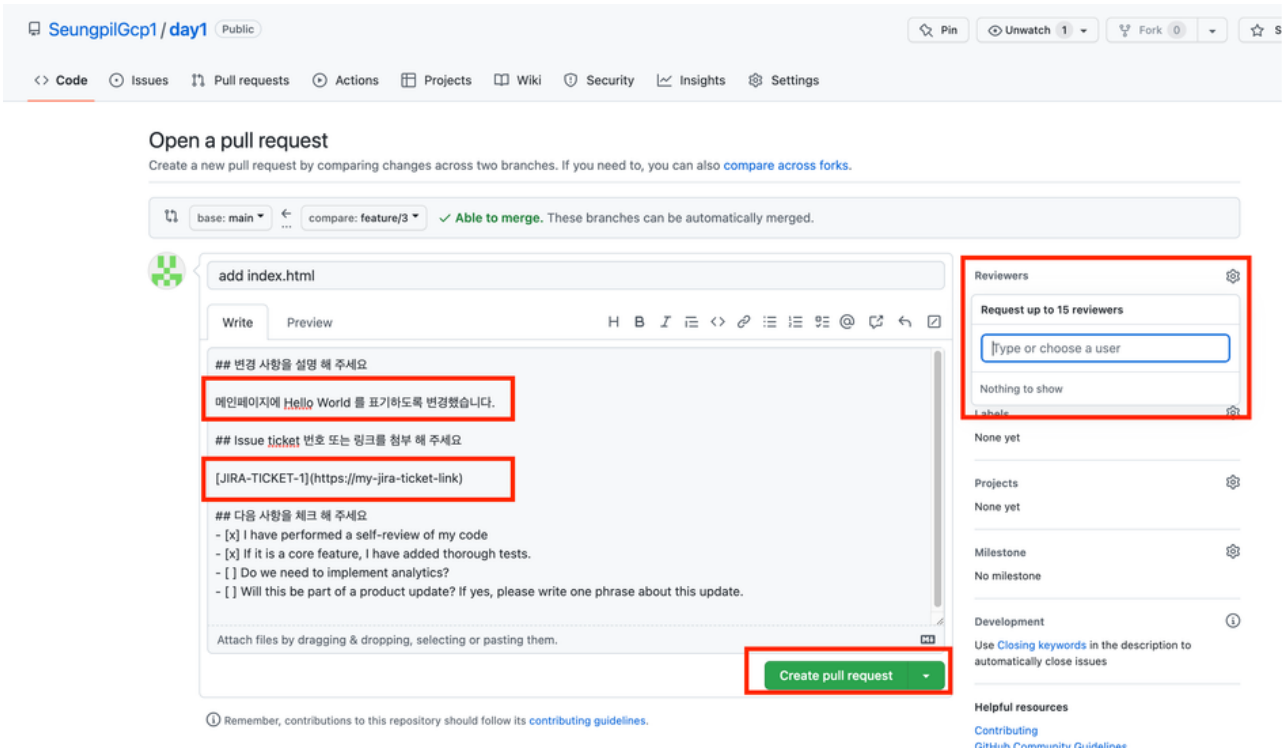
- Github > Pull requests > 신규 PR 생성



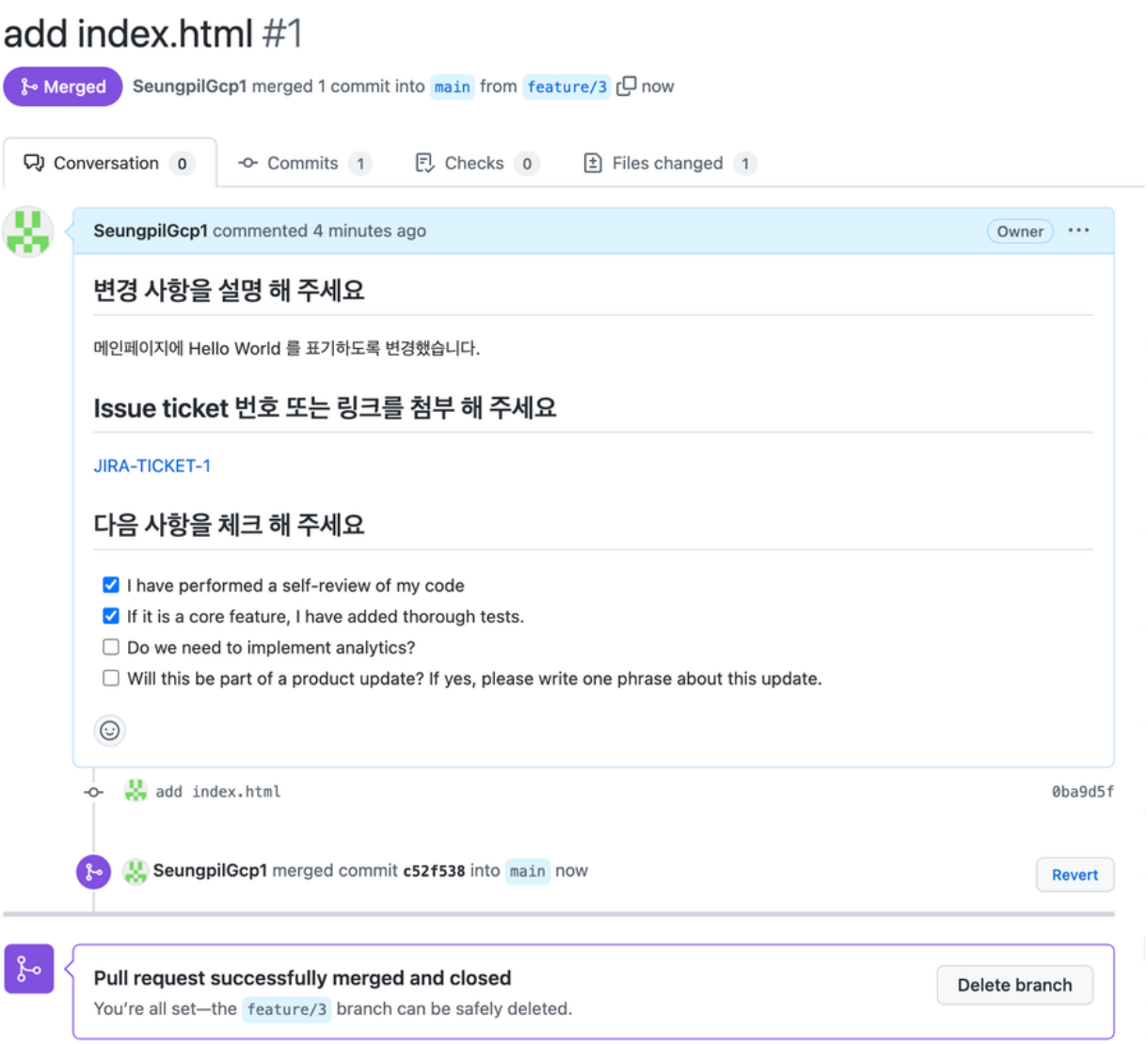
- merge 될 브랜치 선택



- 템플릿이 나타나게 되고, 텍스트 작성
- 코드 리뷰를 받을 인원 추가

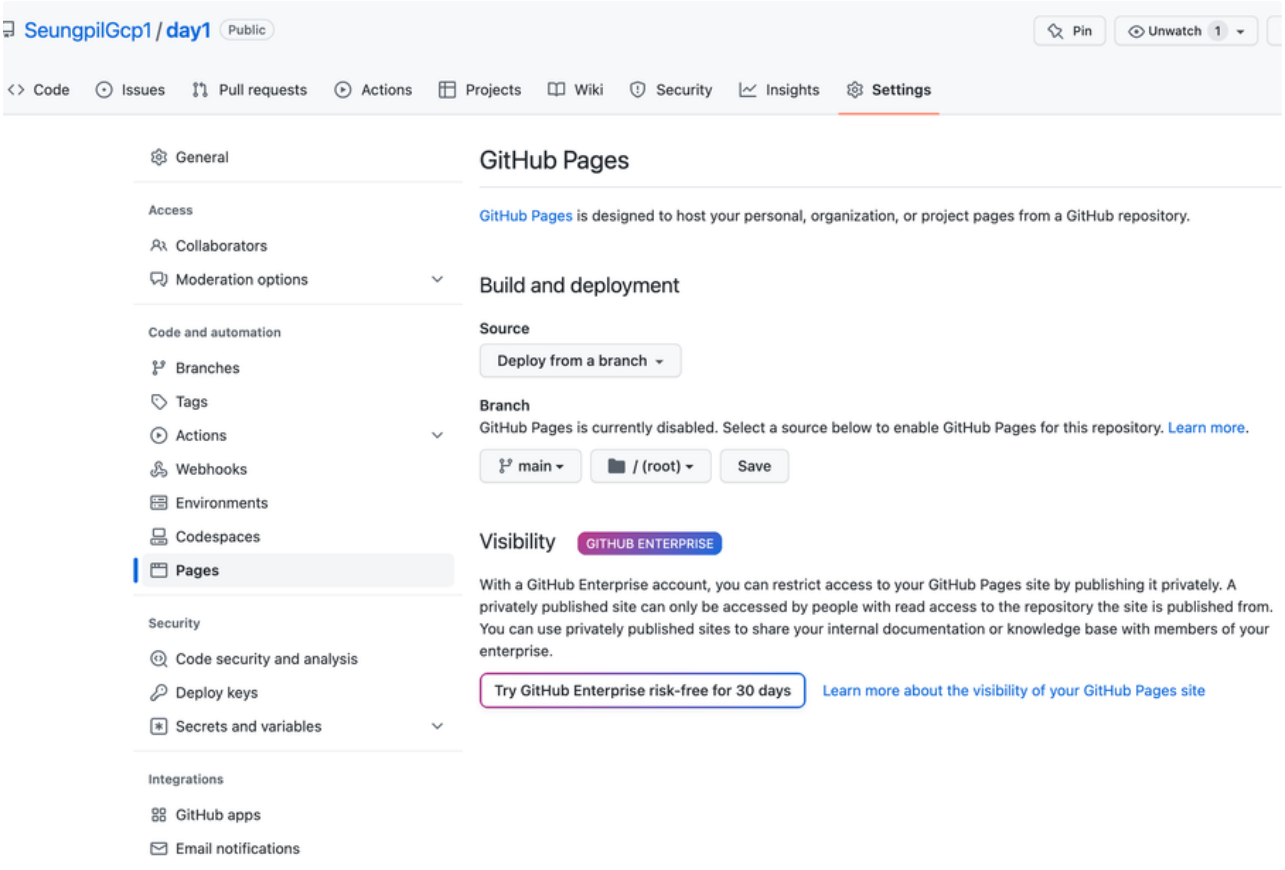


- Confirm 시 Merge 되면서 PR 은 종료상태가 됨.

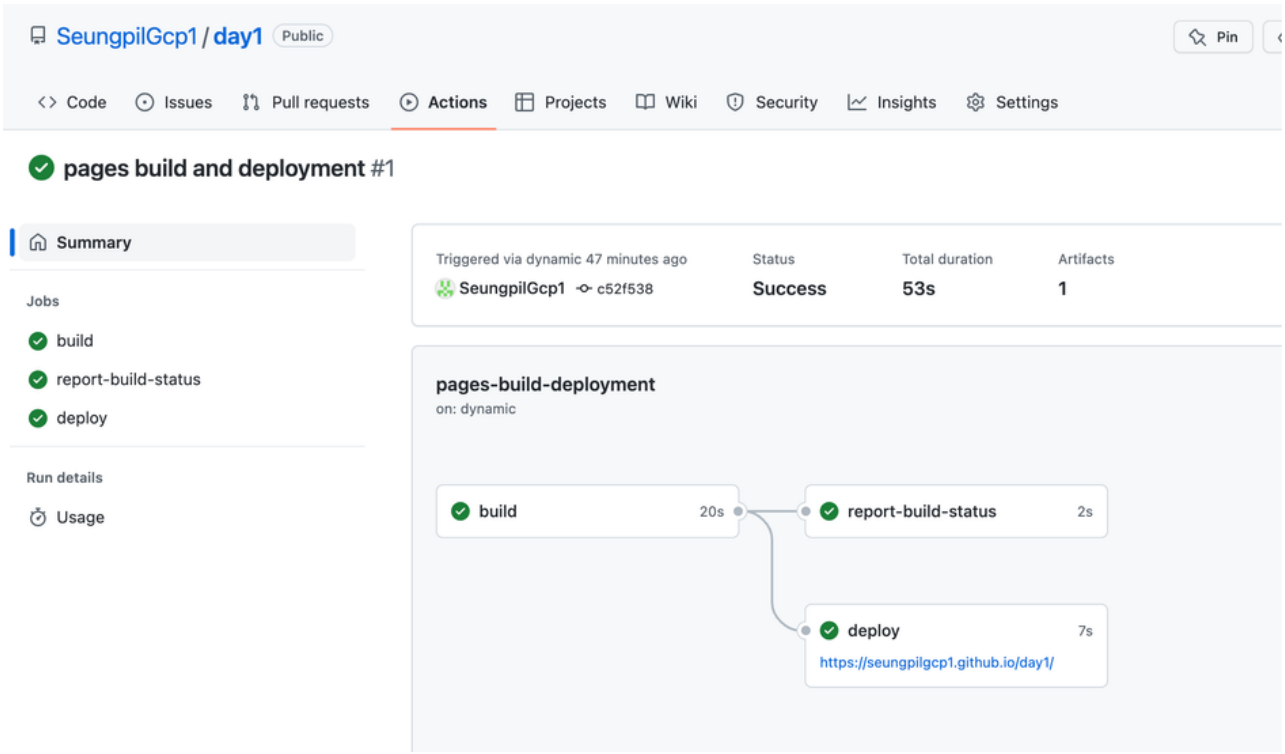


4. Github Page

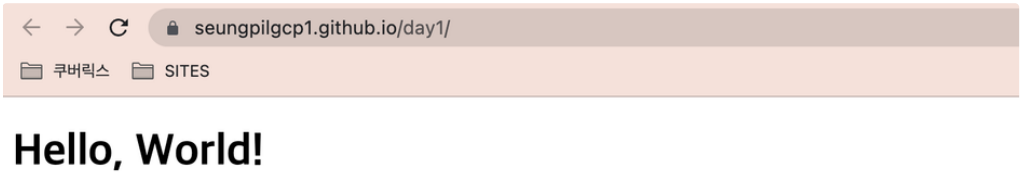
- 이전 PR 에서 index.html 을 main 에 병합 후
- Github Pages 를 통해 static web site 를 구현할 수 있다.
- Settings > Pages > branch 선택 후 save



- Github Actions 에서 build 수행 확인 > deploy 결과의 URL 클릭시



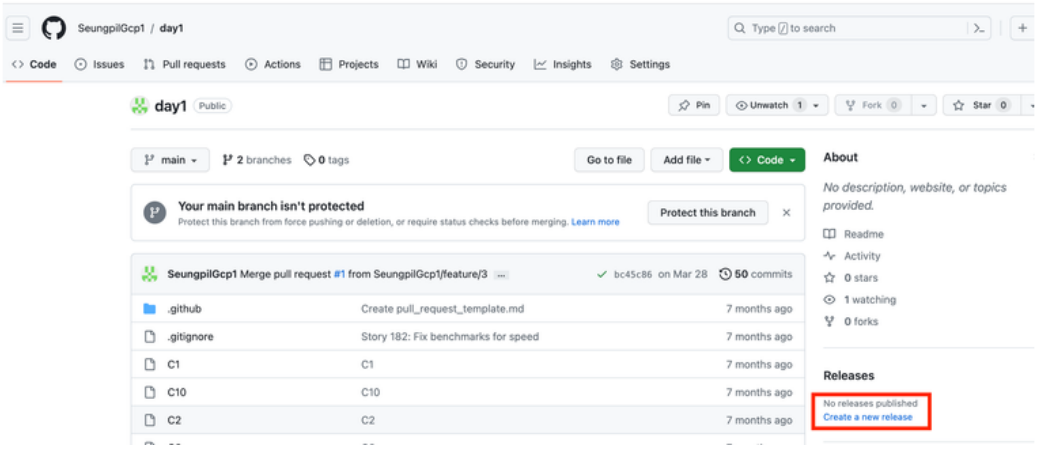
- 인터넷 상에서 접속가능한 페이지 확인



5. Github Release

github 에 Prjecct 를 진행하고 있었고 목표한 기능에 도달했다면 다음 절차에 따라 Release 를 할 수 있고 이때 빌드된 바이너리 파일도 같이 업로드 할 수 있다.

- **Create a new release** 버튼 클릭



---

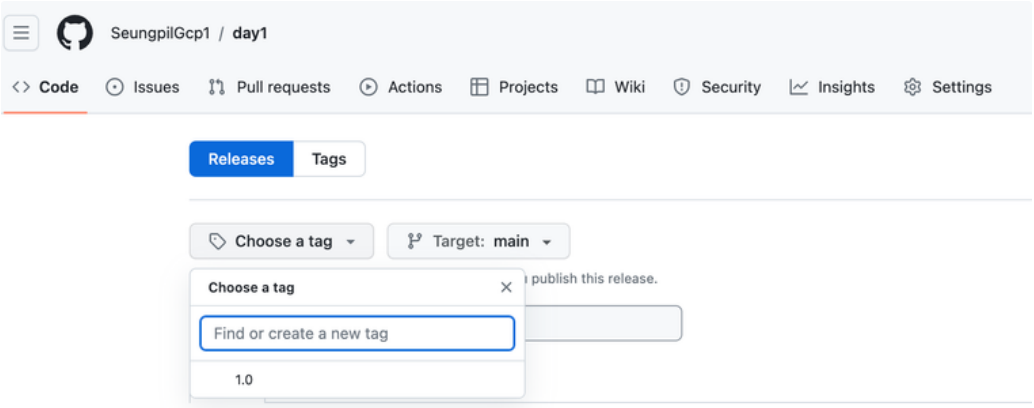
1. 태그를 설정해야 한다.

깃허브의 릴리즈는 저장소(Repository)의 특정 지점 기록을 표시하는 Git Tag 를 기반으로 작동하게 된다.

여기서 말하는 태그란 쉽게 말해, **릴리즈의 버전 번호**를 의미한다.

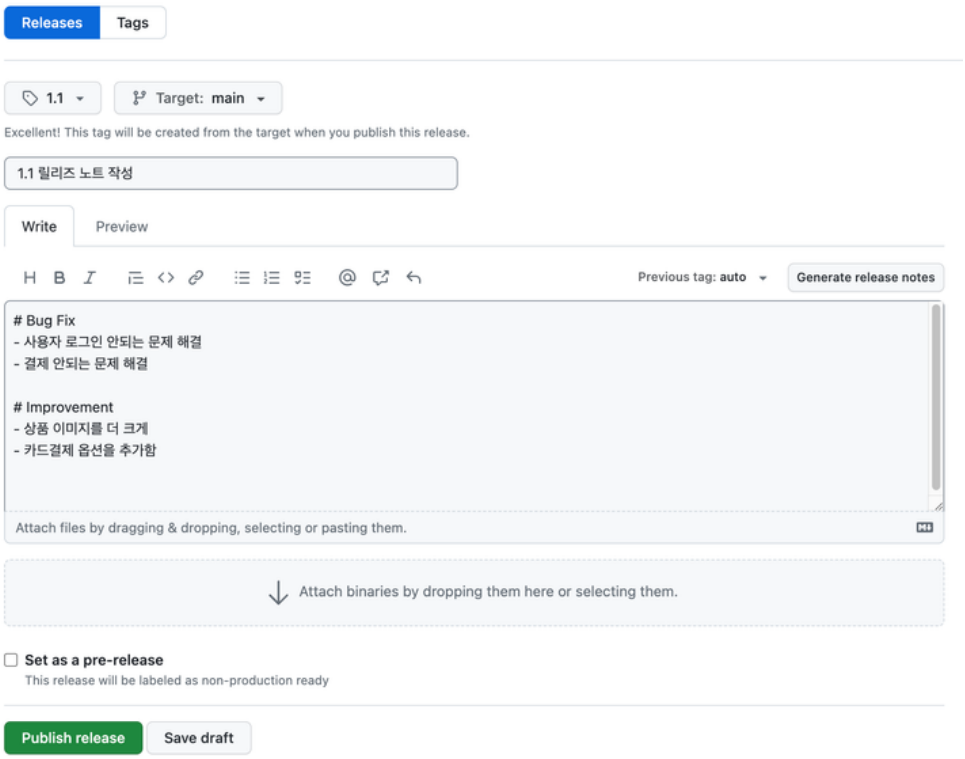
Branch를 만드는 것과 유사하게, 릴리즈할 버전을 입력하고 Create new tag를 눌러주면, tag가 생성된다.

이미 원하는 tag를 생성하면서 작업을 한 경우, tag를 선택하기만 하면 된다.



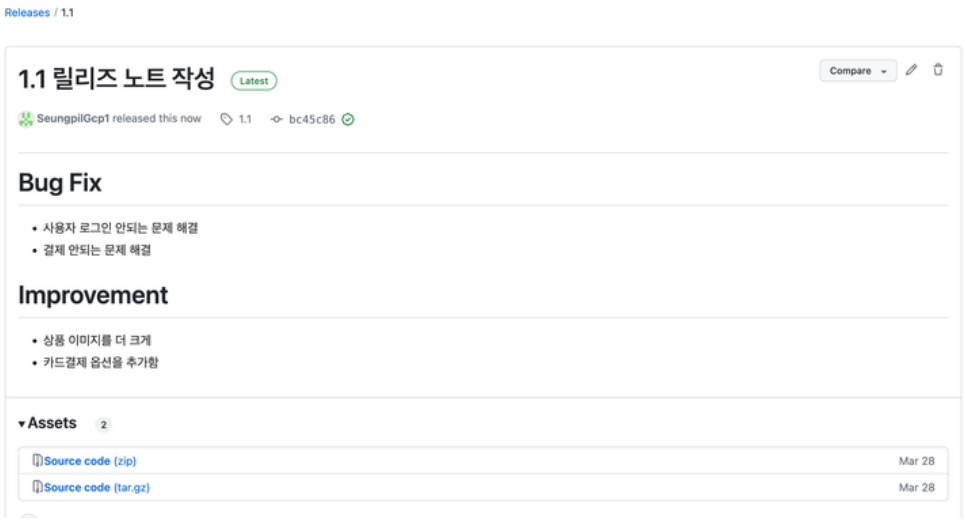
만약 새로운 태그를 생성해줬을 때는 릴리즈할 프로젝트가 담겨있는 branch를 선택해줘야 한다.

2. Release title과 Describe를 작성해주자



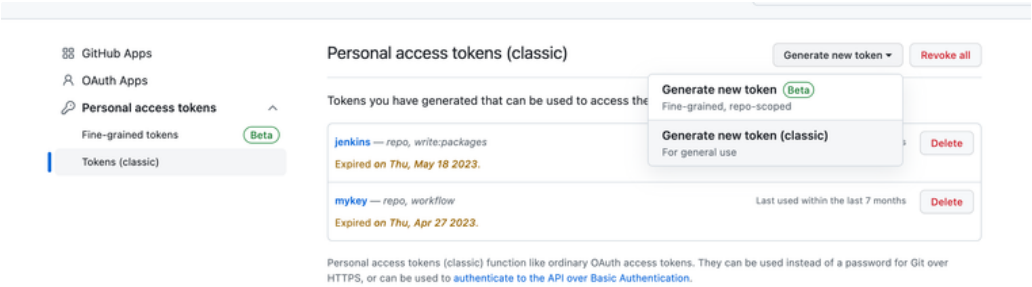
추가로, 릴리즈에 함께 빌드된 바이너리 파일을 포함하고 싶다면, 아래 공간에 Drag & Drop으로 놓아두자.

이 과정을 모두 다 마쳤다면, Publish relase 버튼을 눌러주면 된다.



6. Github Package

- 우측 상단 프로필 > Settings > Developer settings > Tokens → Personal access tokens → Generate new token (classic)



- 아래의 Github Package 권한으로 토큰 발급

Scope	Description	Required permission
`read:packages`	Download and install packages from GitHub Packages	read
`write:packages`	Upload and publish packages to GitHub Packages	write
`delete:packages`	Delete packages from GitHub Packages	admin
`repo`	Upload and delete packages (along with `write:packages`, or `delete:packages`)	write or admin

- 진행에 Docker 설치 필요

Install Docker Desktop on Windows 에서 알맞은 환경의 도커 데스크탑 설치

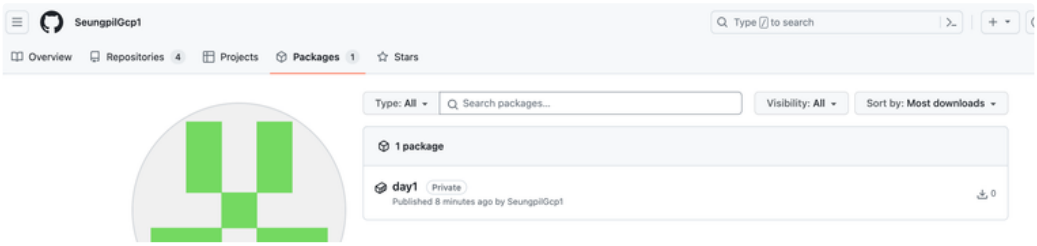
- 도커 로그인

```
$ docker login ghcr.io
Username: <username>
Password: <token>
Login Succeeded
```

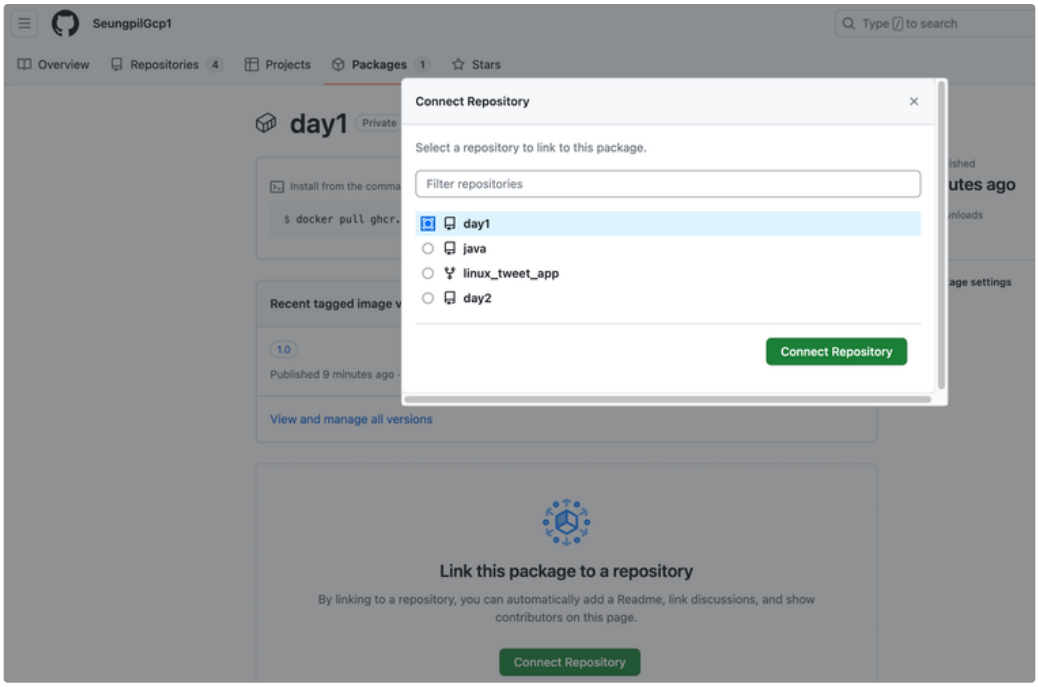
- 도커 이미지 업로드 진행

```
# 이미지 내려받기
$ docker pull alpine:latest
latest: Pulling from library/alpine
579b34f0a95b: Pull complete
Digest: sha256:eece025e432126ce23f223450a0326fbebde39cdf496a85d8c016293fc851978
Status: Downloaded newer image for alpine:latest
docker.io/library/alpine:latest

# 도커 이미지 이름을 Github 계정에 맞추어 변경
$ docker tag alpine:latest ghcr.io/<사용자계정_소문자로>/day1:1.0

# 도커 이미지 업로드
$ docker push ghcr.io/<사용자계정_소문자로>/day1:1.0
The push refers to repository [ghcr.io/seungpilgcp1/day1]
5f4d9fc4d98d: Pushed
1.0: digest: sha256:6ce9a9a256a3495ae60ab0059ed1c7aee5ee89450477f2223f6ea7f6296df555 size: 528
```

- https://github.com/<사용자계정>?tab=packages 접속
- package 에 도커 이미지 업로드 확인



- 소스코드 Repository 와 연계 설정


SeungpilGcp1
Type / to search
Overview
Repositories 4
Projects
Packages 1
Stars
day1
Private
Install from the comma
$ docker pull ghcr.
Recent tagged image v
1.0
Published 9 minutes ago
View and manage all versions
Link this package to a repository
By linking to a repository, you can automatically add a Readme, link discussions, and show contributors on this page.
Connect Repository
Connect Repository
Select a repository to link to this package.
Filter repositories
day1
java
linux_tweet_app
day2
Connect Repository