SAMSUNG SW ACADEMY FOR YOUTH

1차 Job Fair

IT 포트폴리오 작성

멀티캠퍼스

II

블로그 운영

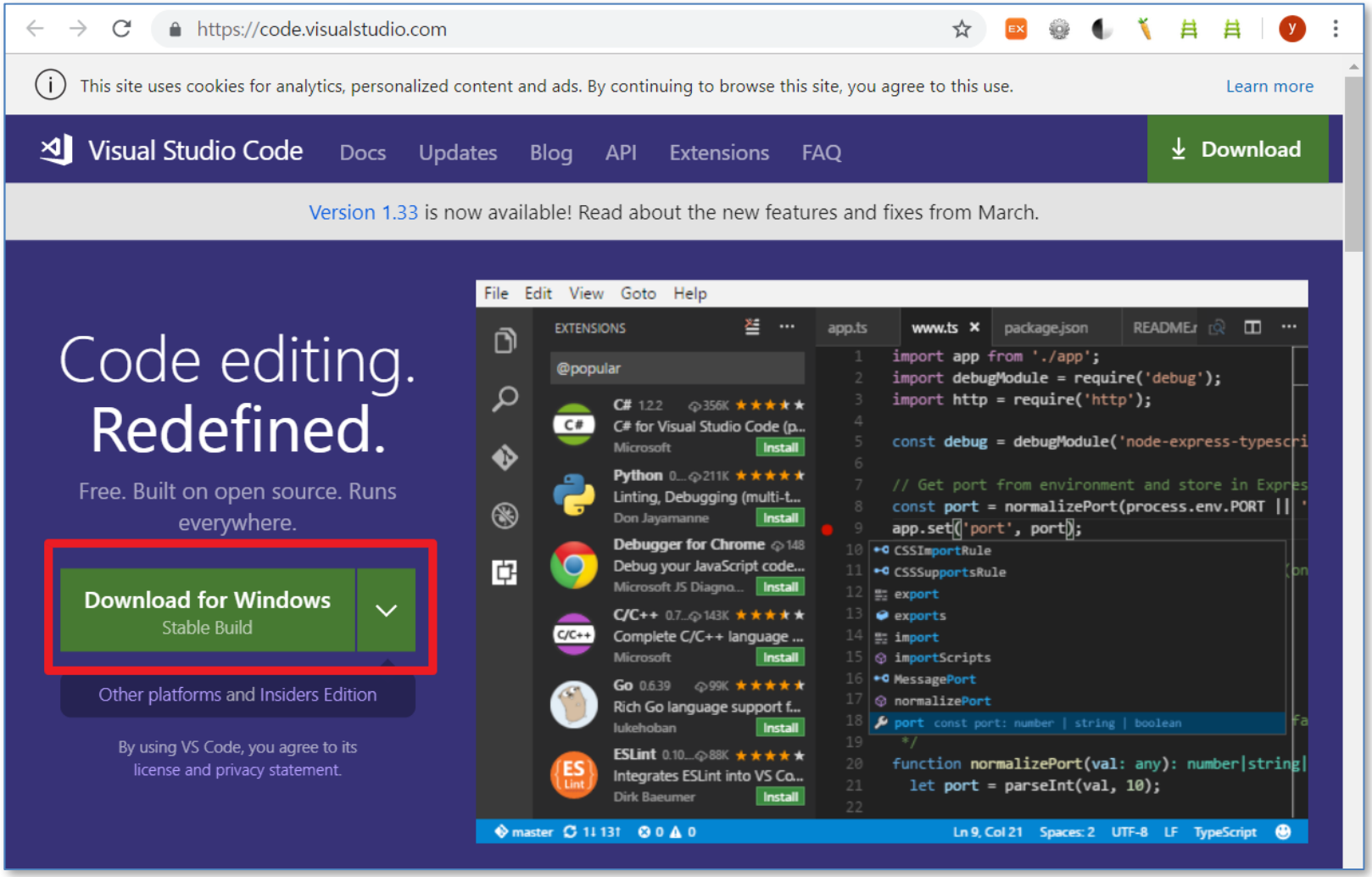
멀티캠퍼스

블로그 운영을 위한 준비

- visual studio code
  - MS에서 제공하는 크로스 플랫폼 에디터로 다양한 언어를 지원하며, IntelliSense와 git , 그리고 풍부한 확장 기능(extension)을 제공.
- node.js
  - node.js는 chrome V8 javascript 엔진으로 빌드된 javascript 런타임으로 확장성 있는 네트워크 애플리케이션(특히 서버 사이드) 개발에 사용되는 소프트웨어 플랫폼임.

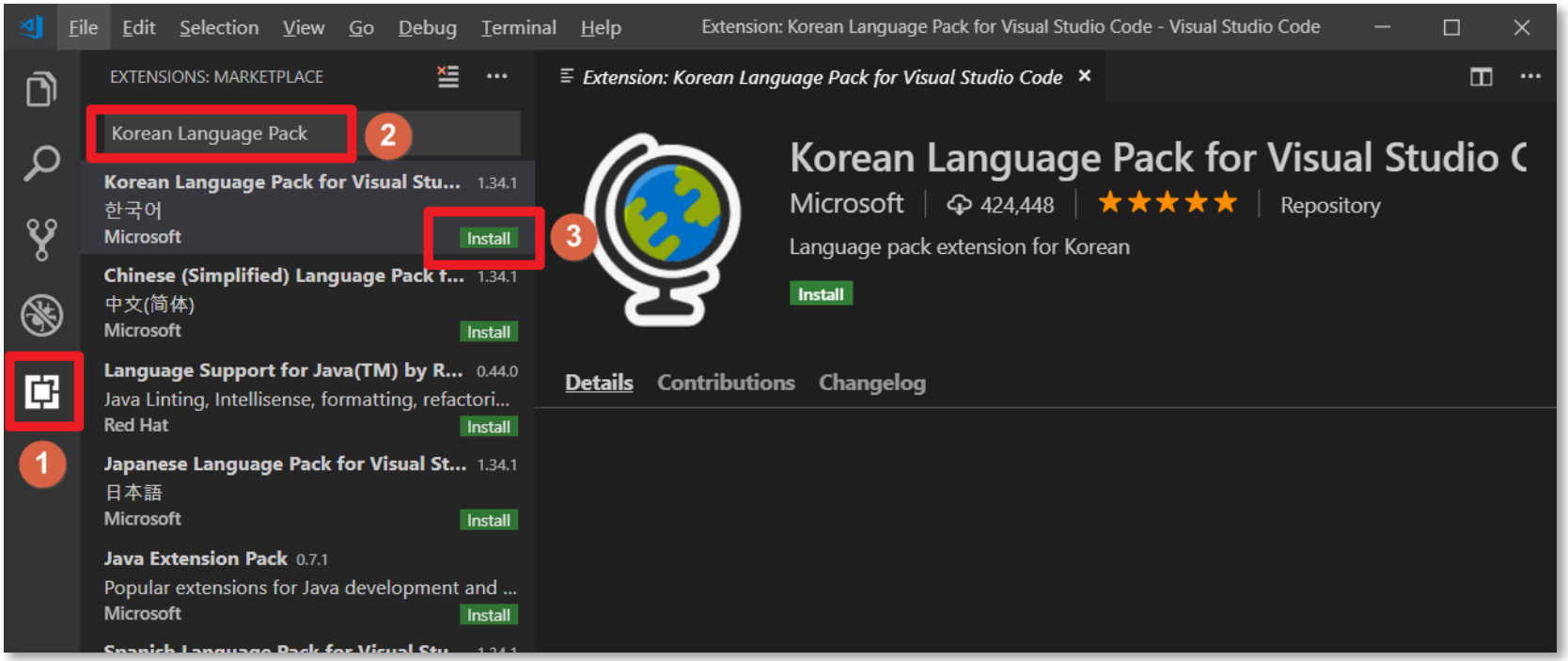
Visual studio code

- https://code.visualstudio.com/에서 다운로드 후 설치



Visual studio code

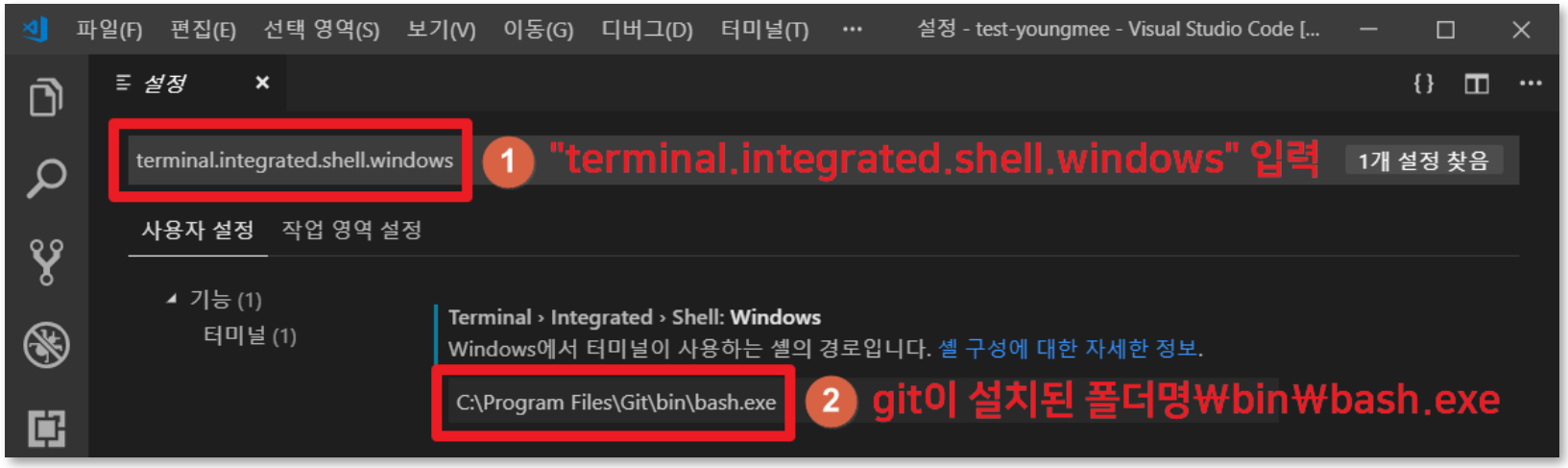
- 다음의 순서로 언어팩 설치



Visual studio code

- 터미널을 git bash로 변경

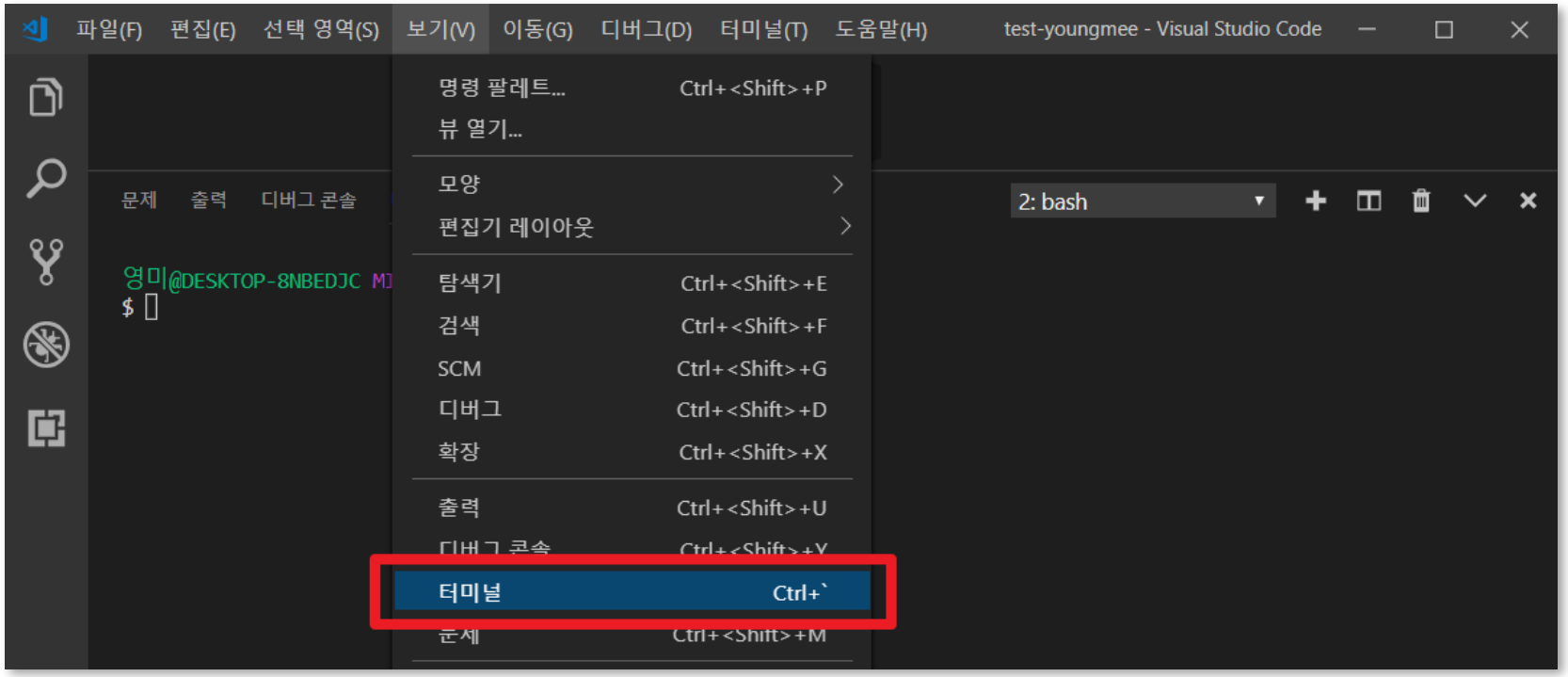
  1. <CTRL> + < , > 를 눌러 설정화면으로 이동
  2. 검색창에 "terminal.integrated.shell.windows" 입력
  3. 표시된 텍스트 상자에 "git이 설치된 폴더명\bin\bash.exe" 입력



  4. VSCode(Visual Studio Code) 재시작

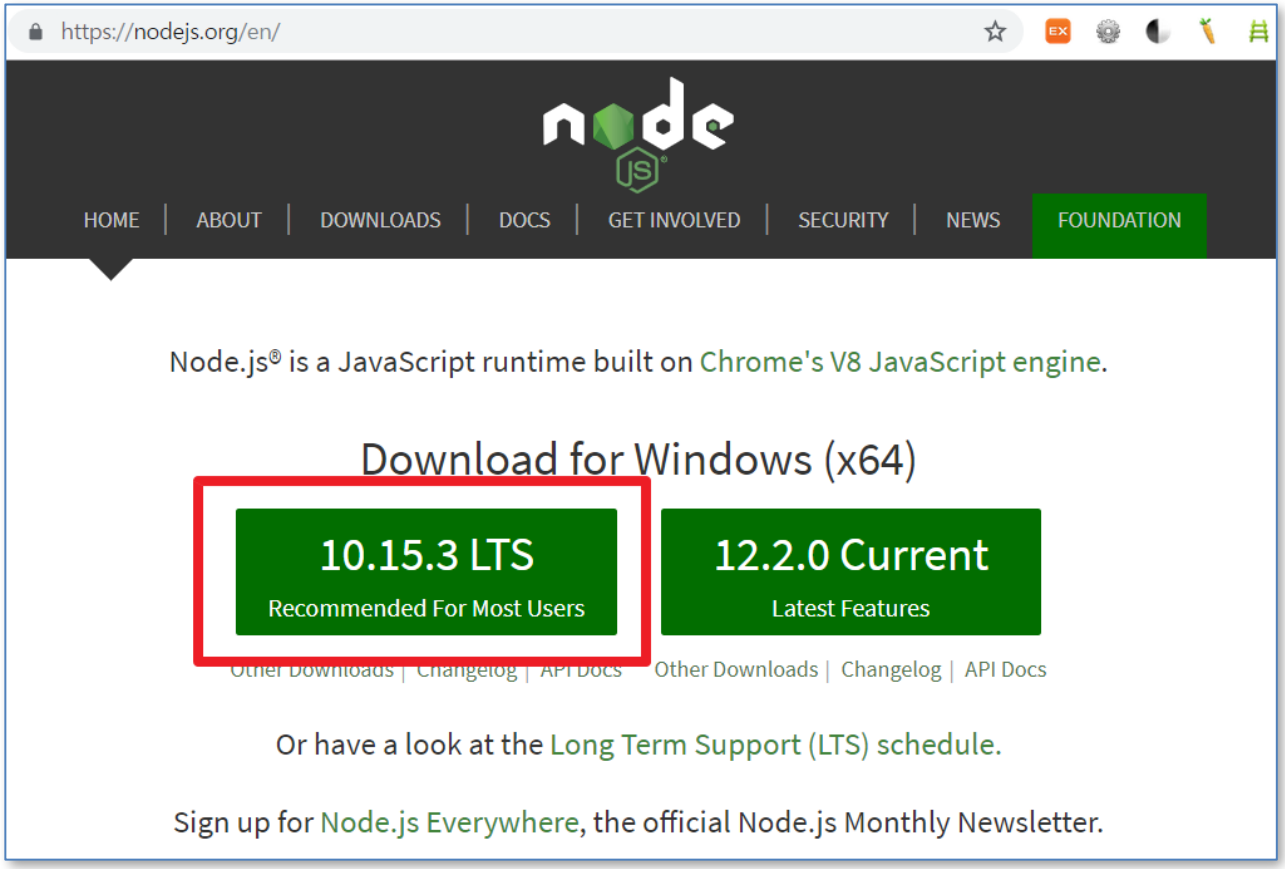
Visual studio code

- [보기]-[터미널] 클릭하여 터미널 사용



설치 전 요구사항

- git
- node.js
  - https://nodejs.org/en/에서 다운로드 받아 설치



hexo

- 쉽고 빠르고 강력한 블로그 프레임워크
- markdown 지원
- 쉽지만 **웹 프로그래밍**에 대한 **기본적인 지식**이 **필요**함.
- 작업 환경 **CLI**(command line interface) 방식임.
- github의 pages 서비스를 활용하여 정적 사이트를 만들 수 있음.

hexo 설치 및 환경설정

멀티캠퍼스

hexo 설치

- 터미널에서 **npm install -g hexo-cli** 실행하여 설치

```
문제   출력   디버그 콘솔   터미널                    1: bash

영미@DESKTOP-8NBEDJC MINGW64 ~
$ npm install -g hexo-cli
C:\Users\영미\AppData\Roaming\npm\hexo -> C:\Users\영미\AppData\Roaming\npm\node_modules\he
i\bin
\hexo
C:\Users\영미\AppData\Roaming\npm\hexo -> C:\Users\영미\AppData\Roaming\npm\node_modules\he
xo-cli\bin\hexo
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.9 (node_modules\hexo-cli\node_
modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.9: want
ed {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})

+ hexo-cli@1.1.0
updated 1 package in 50.183s

영미@DESKTOP-8NBEDJC MINGW64 ~
$
```

폴더 생성 및 이동

- 폴더 생성
  - **mkdir** 폴더명
- 폴더 이동
  - **cd** 폴더명

```
영미@DESKTOP-8NBEDJC MINGW64 ~
$ mkdir test-youngmee

영미@DESKTOP-8NBEDJC MINGW64 ~
$ cd test-youngmee

영미@DESKTOP-8NBEDJC MINGW64 ~/test-youngmee
$
```

hexo 초기화

- 웹사이트 초기화 : **hexo init**

```
영미@DESKTOP-8NBEDJC MINGW64 ~/test-youngmee
$ hexo init
INFO  Cloning hexo-starter to ~\test-youngmee
Cloning into 'C:\Users\영미\test-youngmee'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 71 (delta 0), reused 0 (delta 0), pack-reused 68
Unpacking objects: 100% (71/71), done.
Submodule 'themes/landscape' (https://github.com/hexojs/hexo-theme-landscape.git) registere
d for path 'themes/landscape'
Cloning into 'C:/Users/영미/test-youngmee/themes/landscape'...
remote: Enumerating objects: 893, done.
remote: Total 893 (delta 0), reused 0 (delta 0), pack-reused 893
```

~

```
found 3 vulnerabilities (2 low, 1 moderate)
  run `npm audit fix` to fix them, or `npm audit` for details
INFO  Start blogging with Hexo!

영미@DESKTOP-8NBEDJC MINGW64 ~/test-youngmee
$
```

hexo 기본 테마 적용

- **npm install** 를 이용하여 필요한 패키지 다운로드
  - node_modules 폴더가 없을 시 생성
  - 현재 프로젝트의 package.json에 있는 패키지를 node_modules 폴더에 설치

```
영미@DESKTOP-8NBEDJC MINGW64 ~/test-youngmee
$ npm install
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.9 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.9: want
ed {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})

audited 4698 packages in 13.512s
found 3 vulnerabilities (2 low, 1 moderate)
  run `npm audit fix` to fix them, or `npm audit` for details

영미@DESKTOP-8NBEDJC MINGW64 ~/test-youngmee
$
```

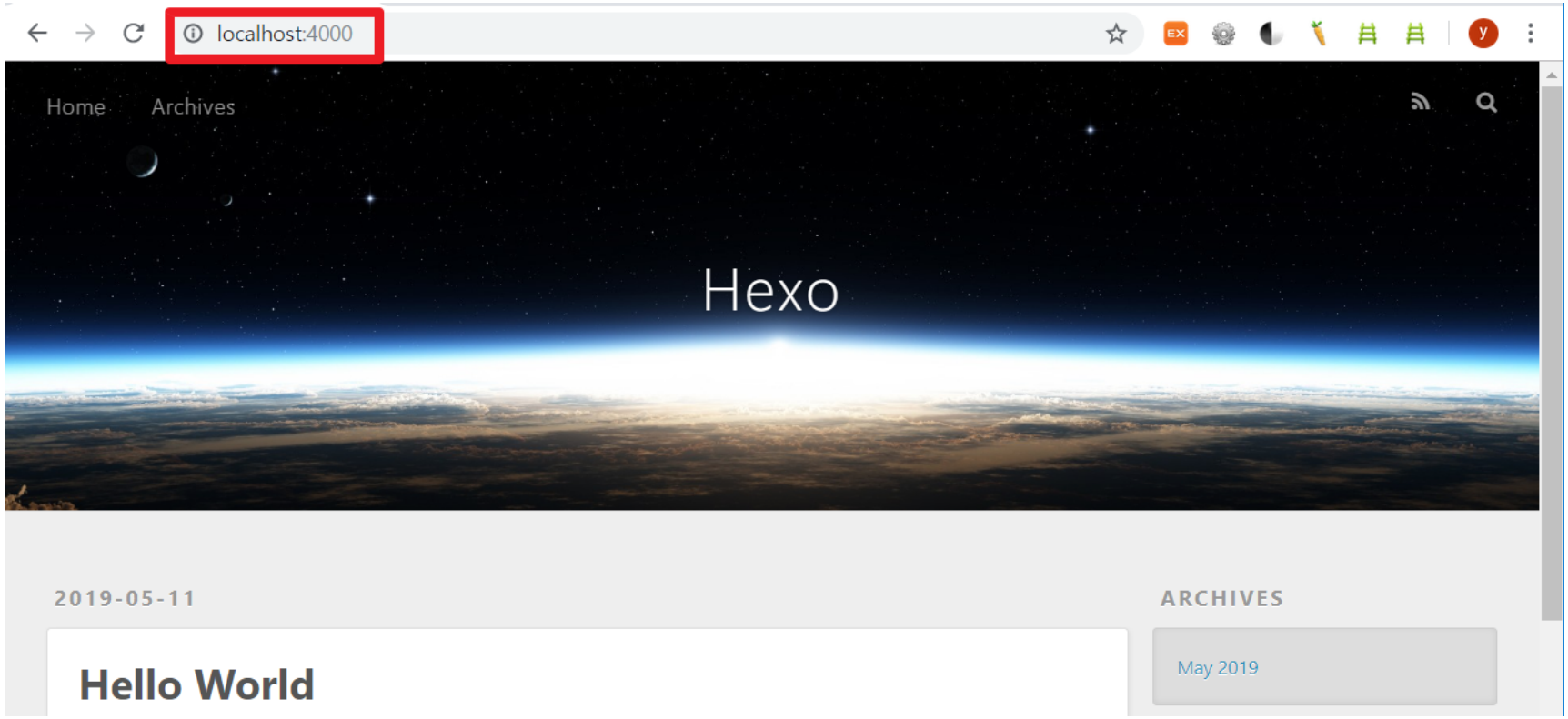
hexo 로컬 서버 구동

- **hexo server** 를 이용하여 서버 구동
  - 서버를 멈출때는 <CTRL> + <C>

```
영미@DESKTOP-8NBEDJC MINGW64 ~/test-youngmee
$ hexo server
INFO  Start processing
INFO  Hexo is running at http://localhost:4000 . Press Ctrl+C to stop.
```

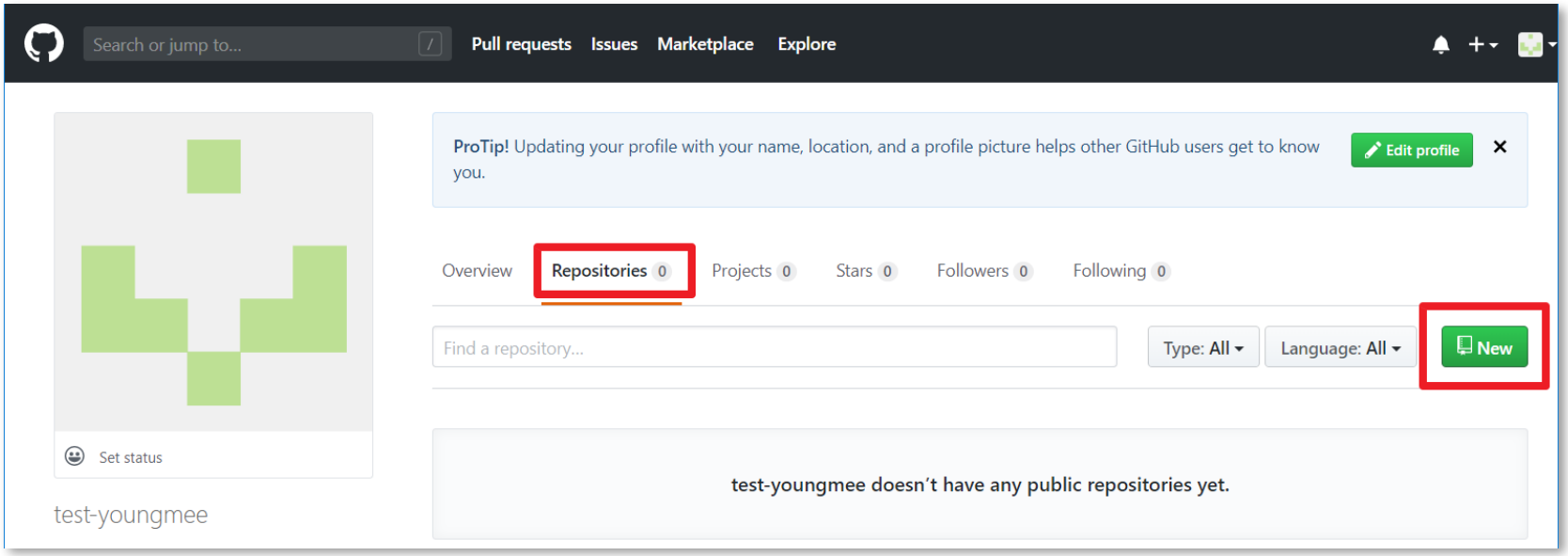
hexo 로컬 서버 구동

- 로컬 서버는 **http://localhost:4000**



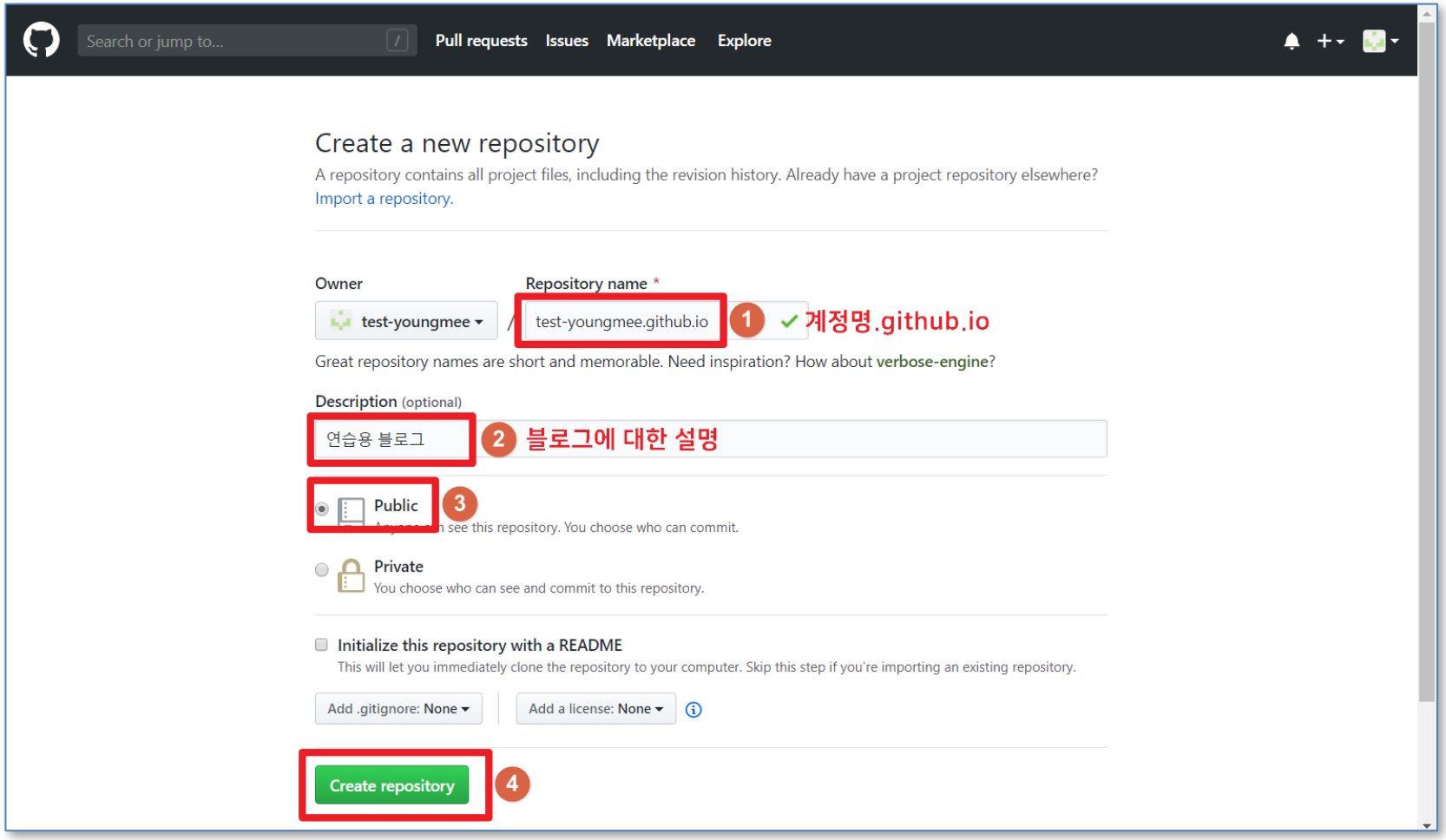
온라인 저장소 생성

- https://github.com 로그인
- Repositories 에서 [New] 버튼 클릭



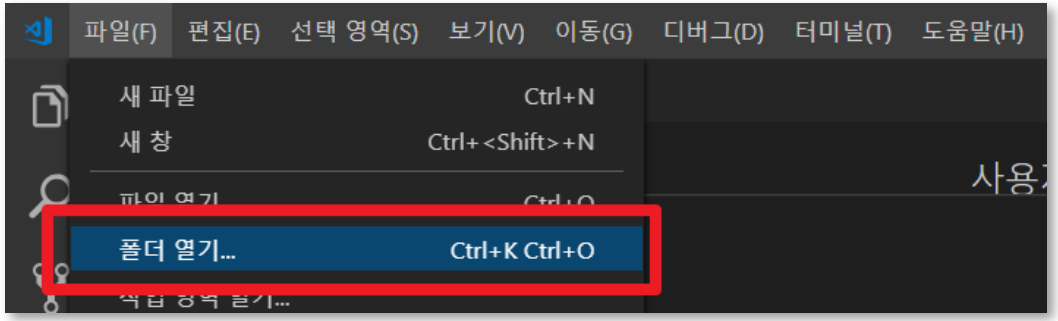
온라인 저장소 생성

- "계정명.github.io"를 이름으로 공개용 저장소 생성

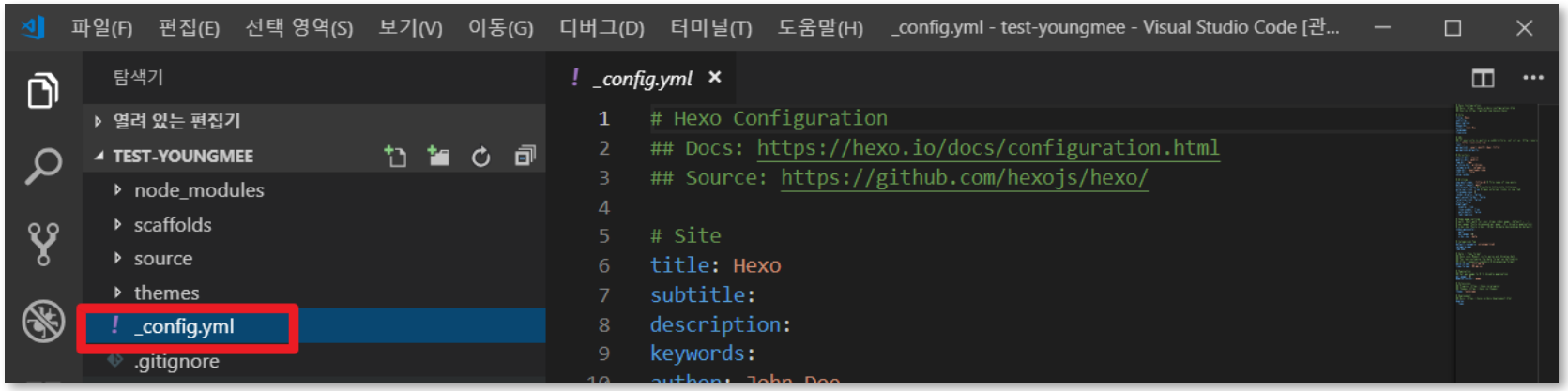


환경 설정(_config.yml)

- [파일]-[폴더 열기] 클릭하여 작업중인 폴더 선택



- [보기]-[탐색기] 클릭
- "**_config.yml**" 파일 클릭



환경 설정(_config.yml)

- title 및 author 정보 수정
- url은 **https://본인id.github.io**로 수정

_config.yml

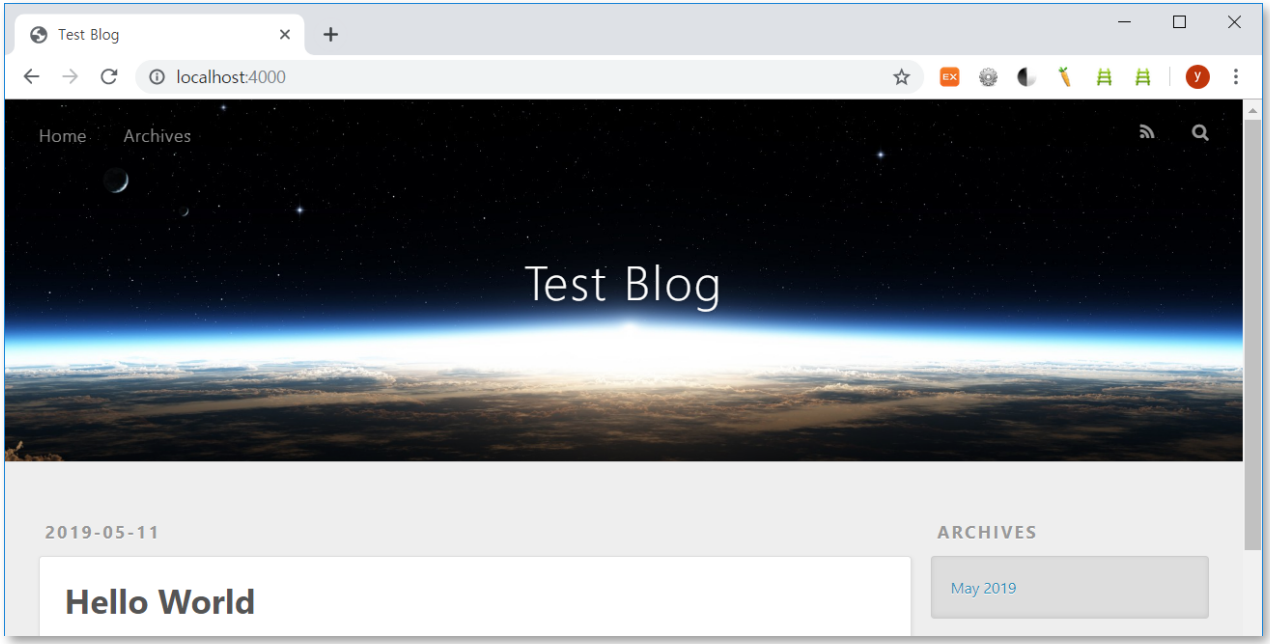
```yaml
# Site
title: Test Blog
subtitle:
description:
keywords:
author: anne
language:
timezone:

# URL
## If your site is put in a subdirectory, set url as 'http://yoursit
url: http://test-youngmee.github.io
root: /
permalink: :year/:month/:day/:title/
permalink_defaults:
```

환경 설정(_config.yml)

- 터미널에서 **hexo server** 실행

```
영미@DESKTOP-8NDEDJC MINGW64 ~/test-youngmee
$ hexo server
INFO  Start processing
INFO  Hexo is running at http://localhost:4000 . Press Ctrl+C to stop.
```

- 새로고침하여 웹 사이트 변경 내용 확인



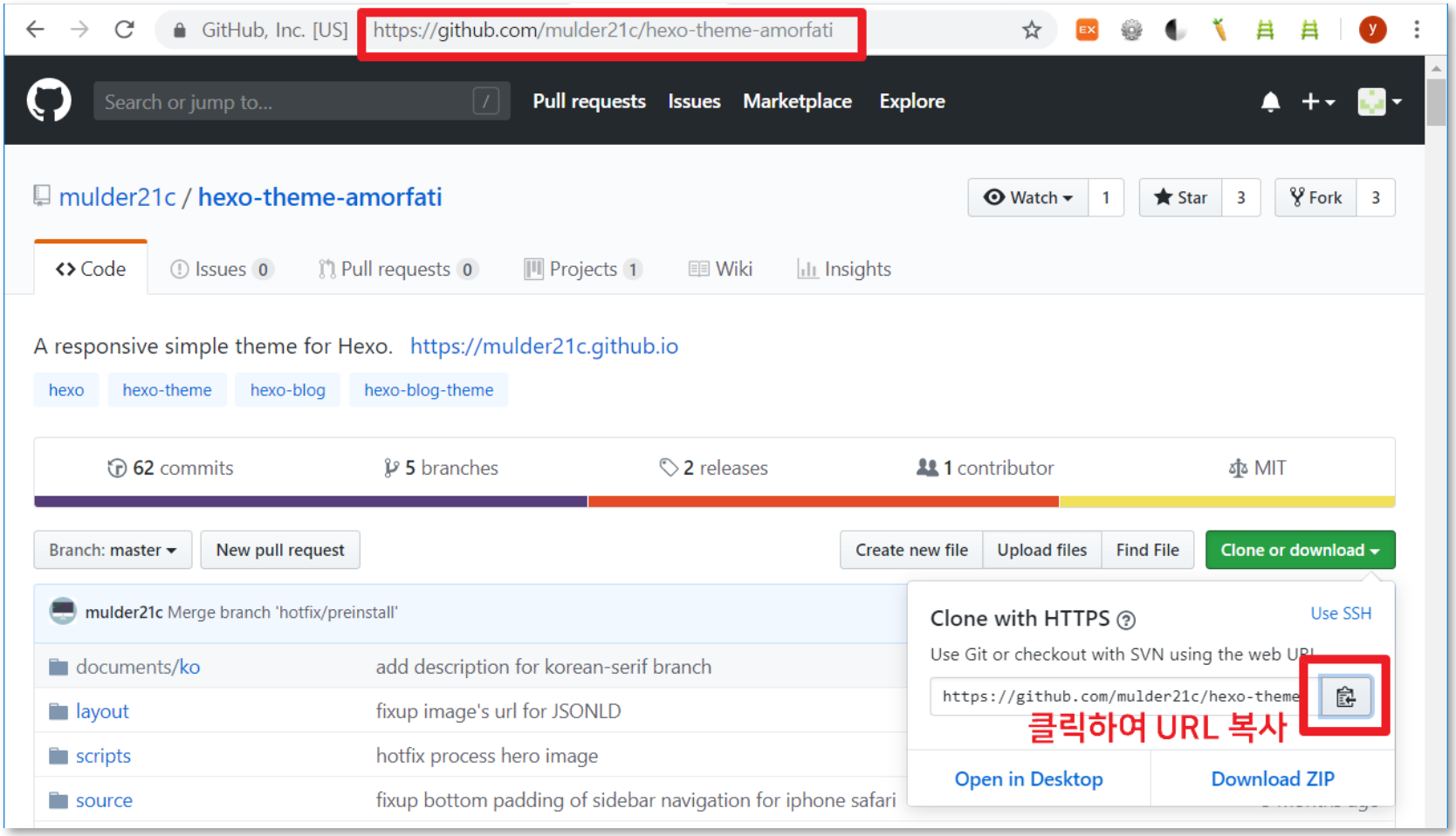
hexo 테마 변경

멀티캠퍼스

hexo 테마 선택

- https://github.com/mulder21c/hexo-theme-amorfati 에서

  무료 테마 사용을 위한 URL 복사



hexo 테마 위치

- 프로젝트 폴더안의 themes 폴더로 이동
  - **cd themes**

```
영미@DESKTOP-8NBEDJC MINGW64 ~/test-youngmee
$ ls
_config.yml  node_modules  package-lock.json  source
db.json      package.json  scaffolds          themes

영미@DESKTOP-8NBEDJC MINGW64 ~/test-youngmee
$ cd themes

영미@DESKTOP-8NBEDJC MINGW64 ~/test-youngmee/themes
$
```

hexo 테마 복사

- 온라인 테마 저장소를 themes 폴더로 복제
  - **git clone https://github.com/mulder21c/hexo-theme-amorfati**

```
영미@DESKTOP-8NBEDJC MINGW64 ~/test-youngmee/themes
$ git clone https://github.com/mulder21c/hexo-theme-amorfati
.git
Cloning into 'hexo-theme-amorfati'...
remote: Enumerating objects: 29, done.
remote: Counting objects: 100% (29/29), done.
remote: Compressing objects: 100% (23/23), done.
remote: Total 774 (delta 12), reused 17 (delta 6), pack-reused 745
Receiving objects: 100% (774/774), 98.47 MiB | 3.06 MiB/s, done.
Resolving deltas: 100% (331/331), done.

영미@DESKTOP-8NBEDJC MINGW64 ~/test-youngmee/themes
$ ls
hexo-theme-amorfati  landscape

영미@DESKTOP-8NBEDJC MINGW64 ~/test-youngmee/themes
$
```

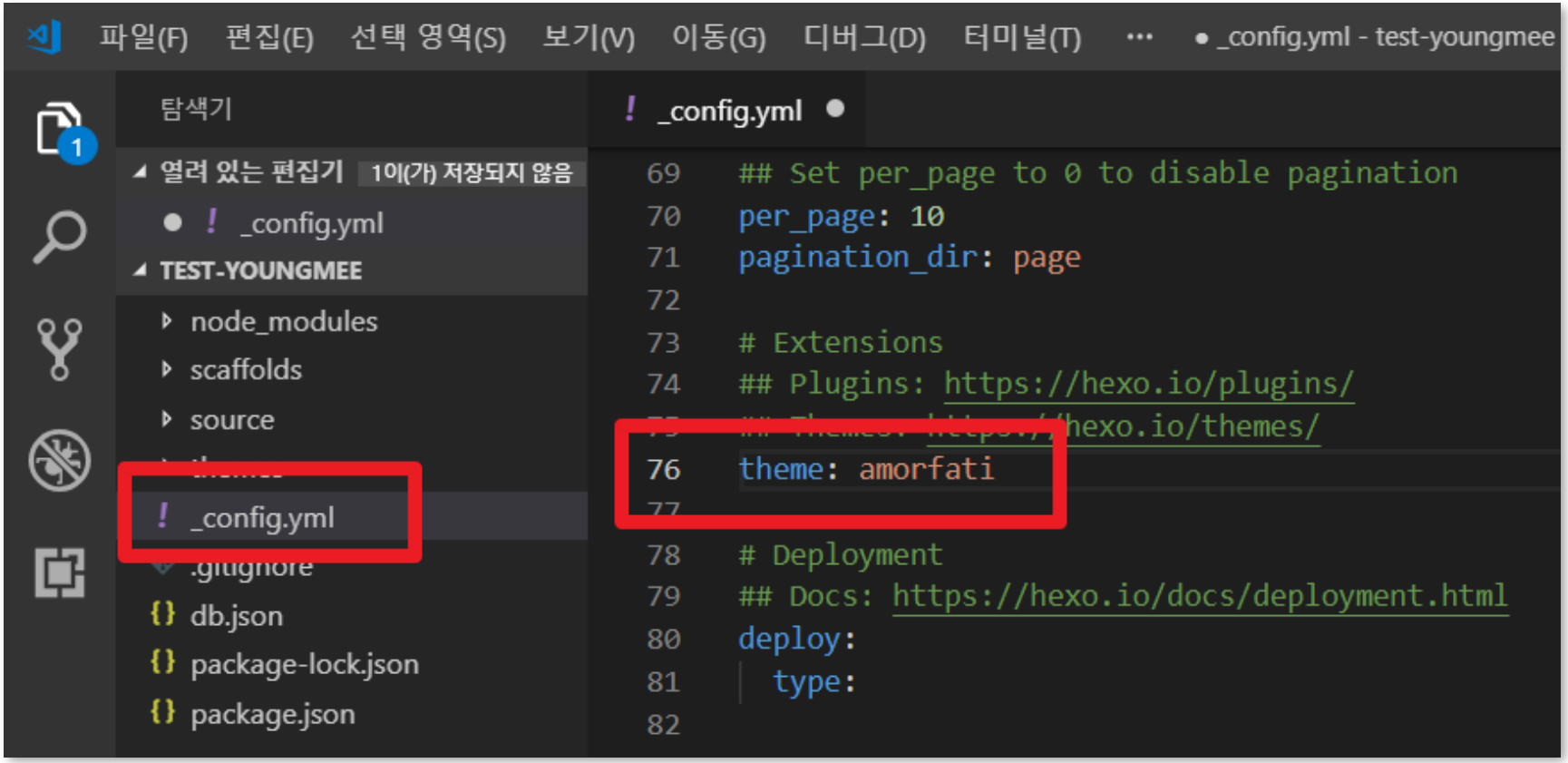
hexo 테마 패키지 다운로드

- 폴더 이동 후 **npm install** 명령어로 필요한 패키지 다운로드

```
영미@DESKTOP-8NBEDJC MINGW64 ~/test-youngmee/themes
$ ls
hexo-theme-amorfati  landscape

영미@DESKTOP-8NBEDJC MINGW64 ~/test-youngmee/themes
$ mv hexo-theme-amorfati amorfati

영미@DESKTOP-8NBEDJC MINGW64 ~/test-youngmee/themes
$ ls
amorfati  landscape

영미@DESKTOP-8NBEDJC MINGW64 ~/test-youngmee/themes
$ cd amorfati

영미@DESKTOP-8NBEDJC MINGW64 ~/test-youngmee/themes/amorfati (master)
$ npm install
```

hexo 테마 적용을 위한 환경설정

- "**_config.yml**"파일의 내용 중 **theme**의 내용을 **amorfati**로 변경 후 저장



hexo 테마 적용 확인

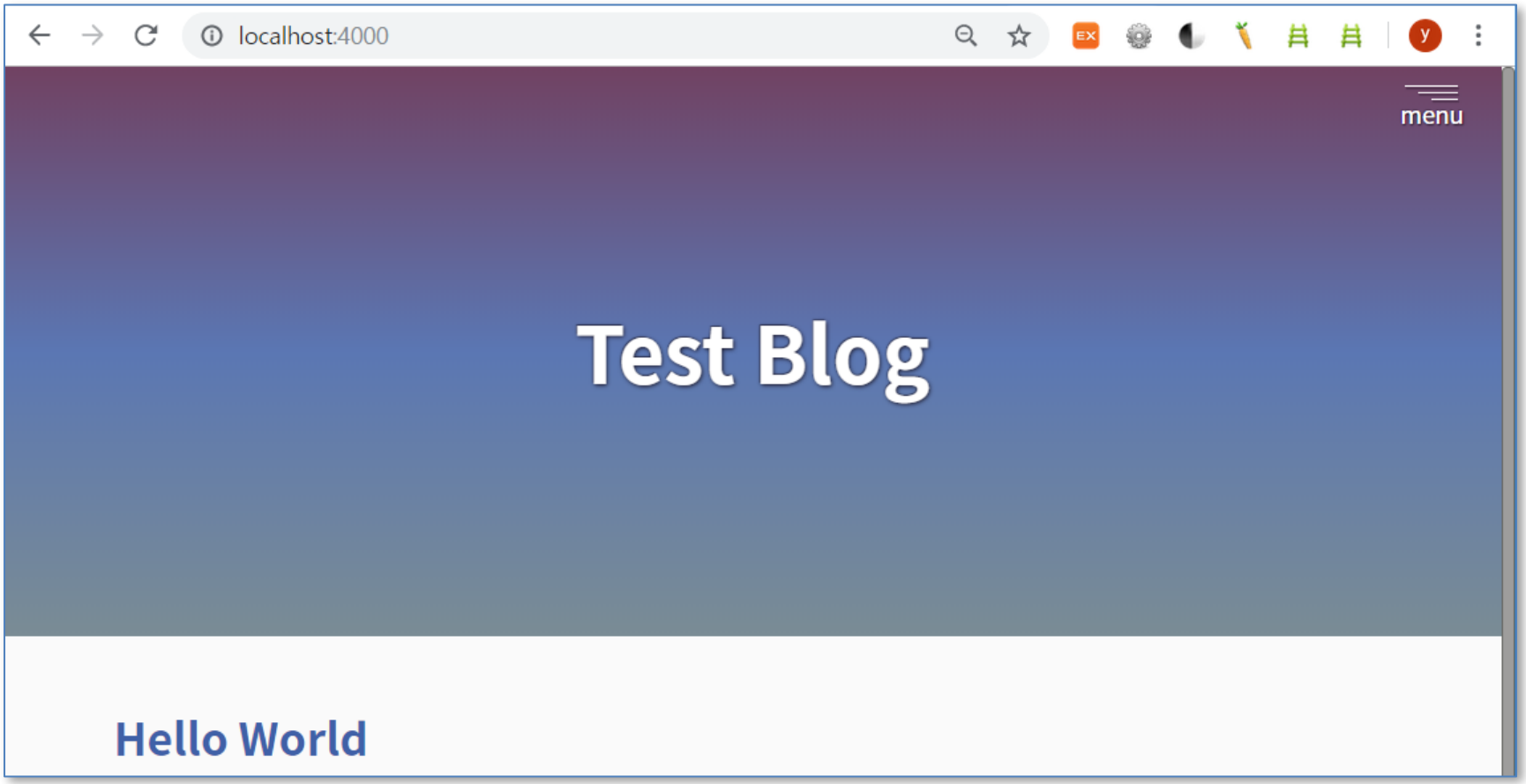
- 프로젝트 폴더로 이동 후 hexo 서버 구동
  - **hexo server**

```
영미@DESKTOP-8NBEDJC MINGW64 ~/test-youngmee/themes/amorfati (master)
$ cd ..

영미@DESKTOP-8NBEDJC MINGW64 ~/test-youngmee/themes
$ cd ..

영미@DESKTOP-8NBEDJC MINGW64 ~/test-youngmee
$ hexo server
INFO  Start processing
INFO  Hexo is running at http://localhost:4000 . Press Ctrl+C to stop.
```

hexo 테마 적용 확인

Go

- 새로고침하여 테마가 변경된 웹사이트 확인

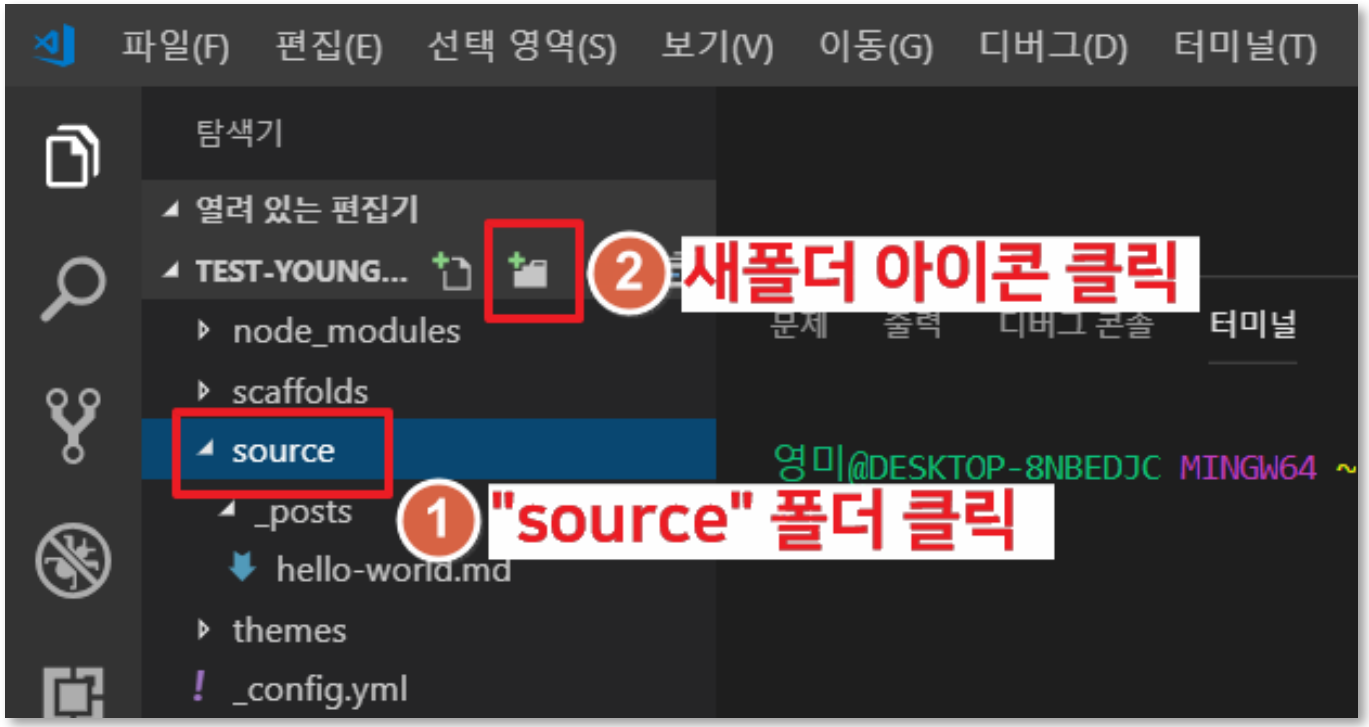


hexo 새글 작성

멀티캠퍼스

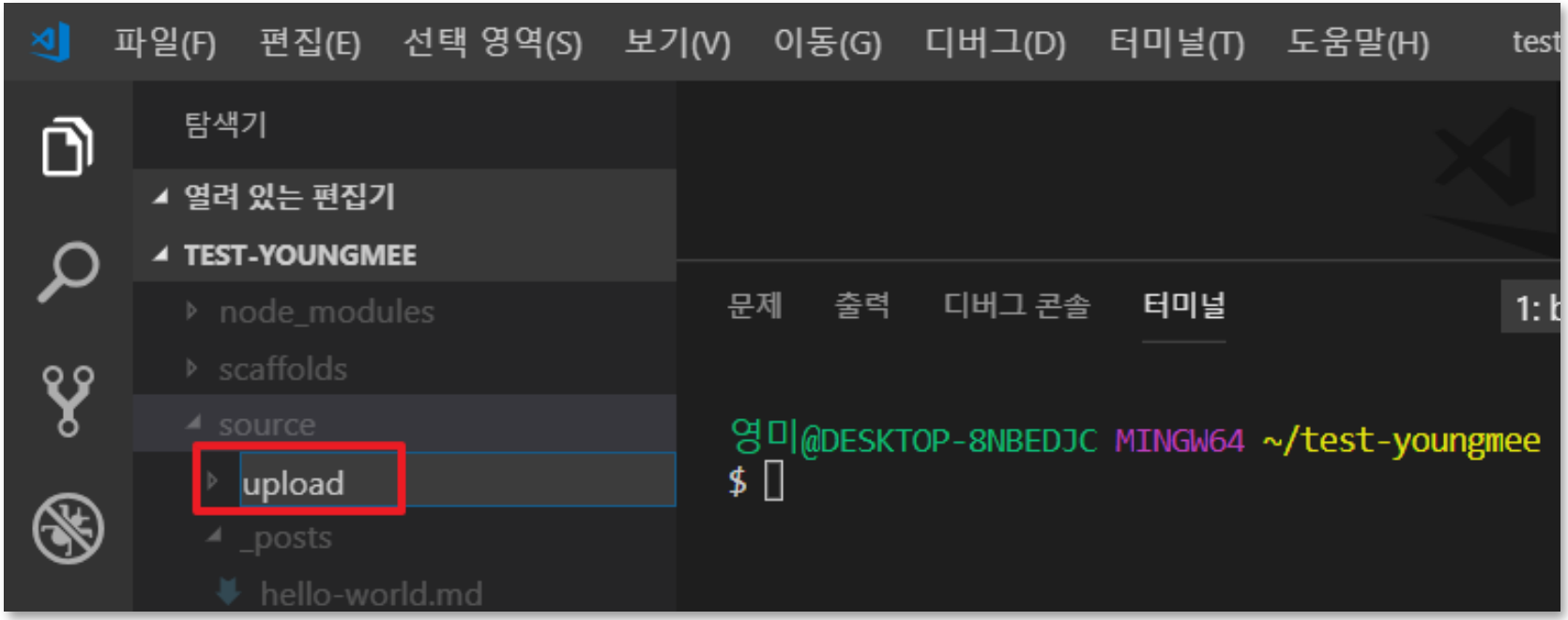
upload 폴더 생성

- 탐색기에서 source폴더에 클릭한 후 새폴더 아이콘 클릭하여 폴더 생성



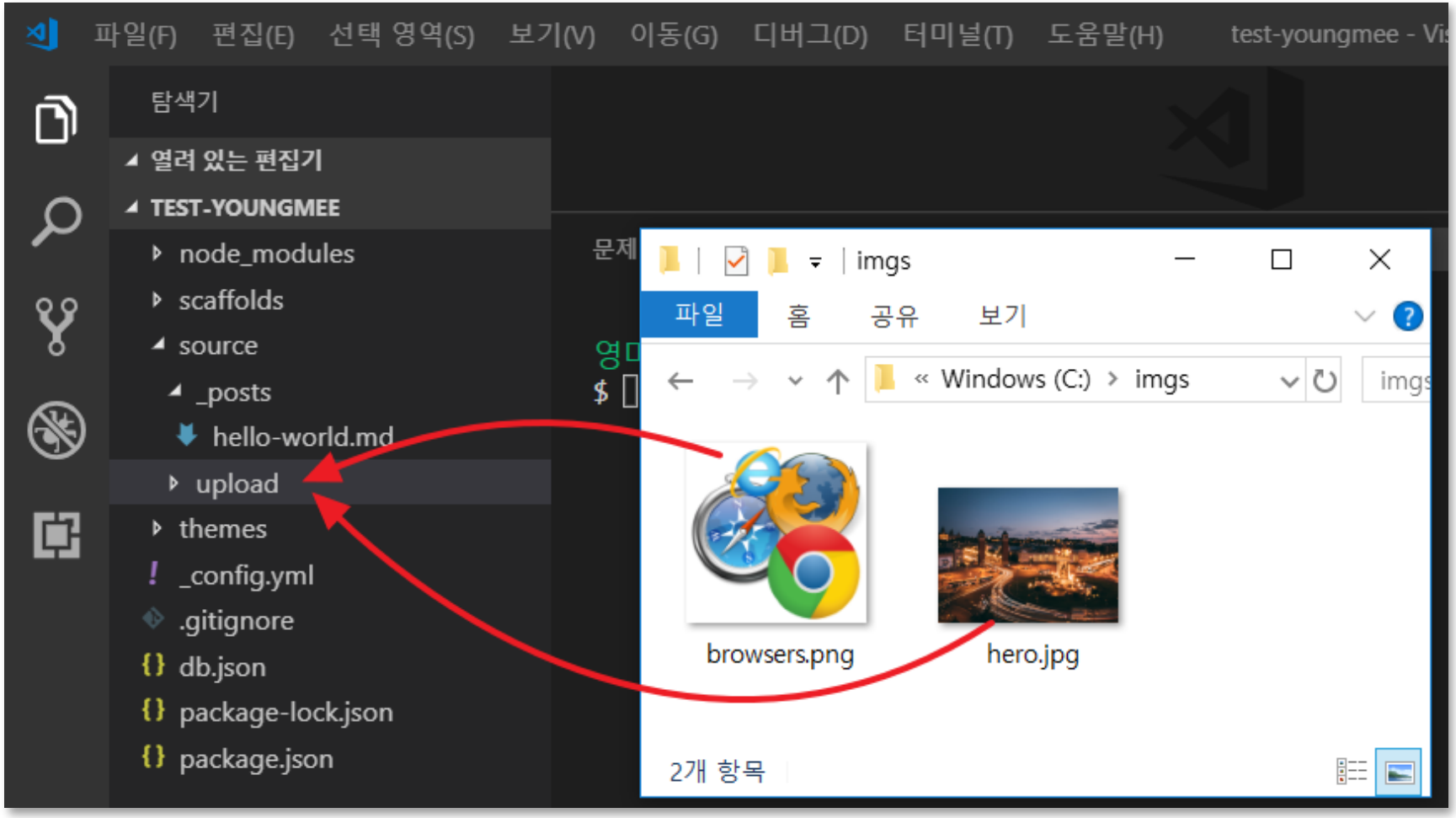
upload 폴더 생성

- 새로 만들어진 폴더의 이름으로 "upload"를 입력한 후 <Enter>



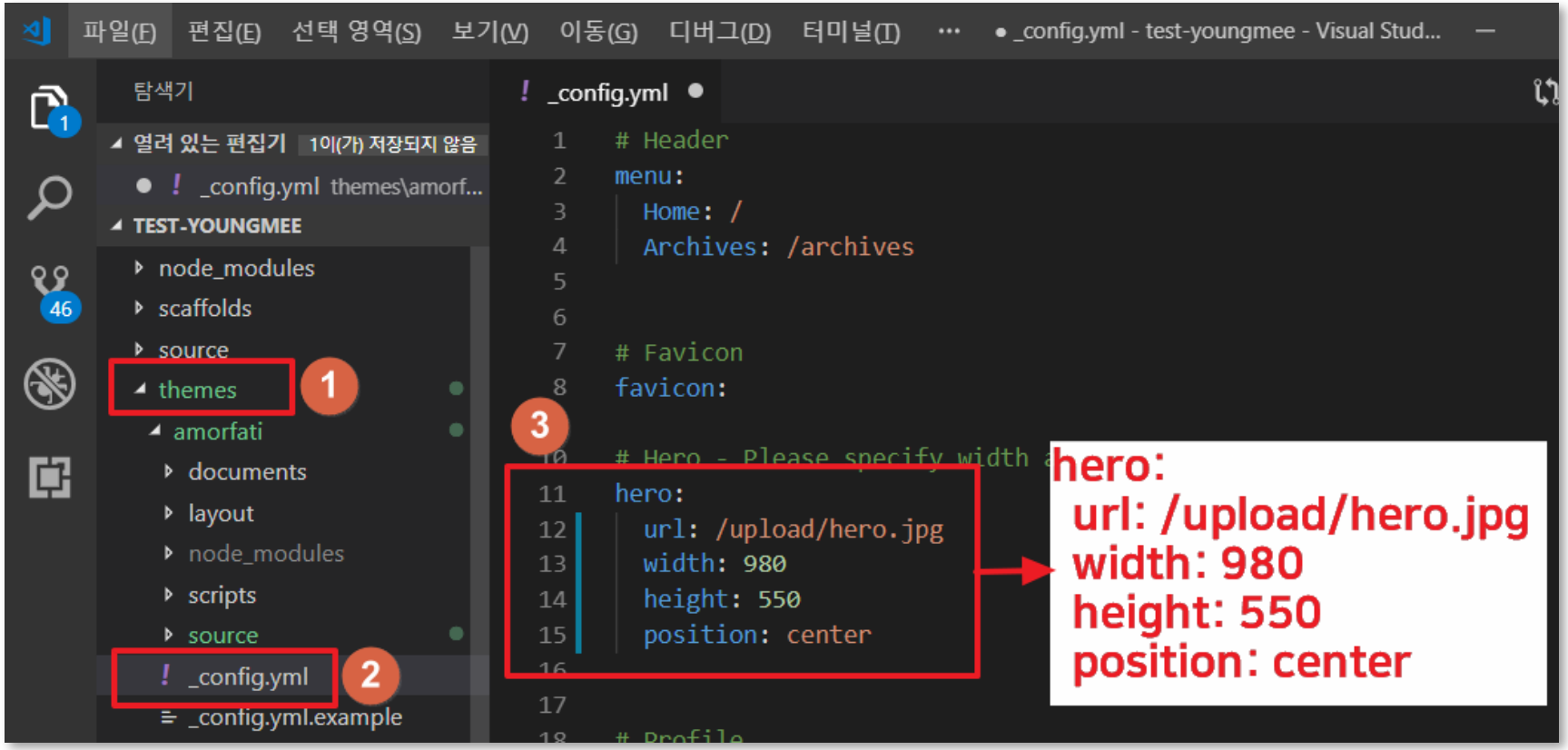
이미지 복사

- 사용할 이미지들을 upload 폴더로 drag & drop하여 복사



hero 이미지 수정(환경설정)

- 탐색기에서 themes 폴더안의 "_config.yml" 파일 선택
- hero 영역의 내용을 수정



hexo 서버 구동

- hexo server 명령으로 서버 시작

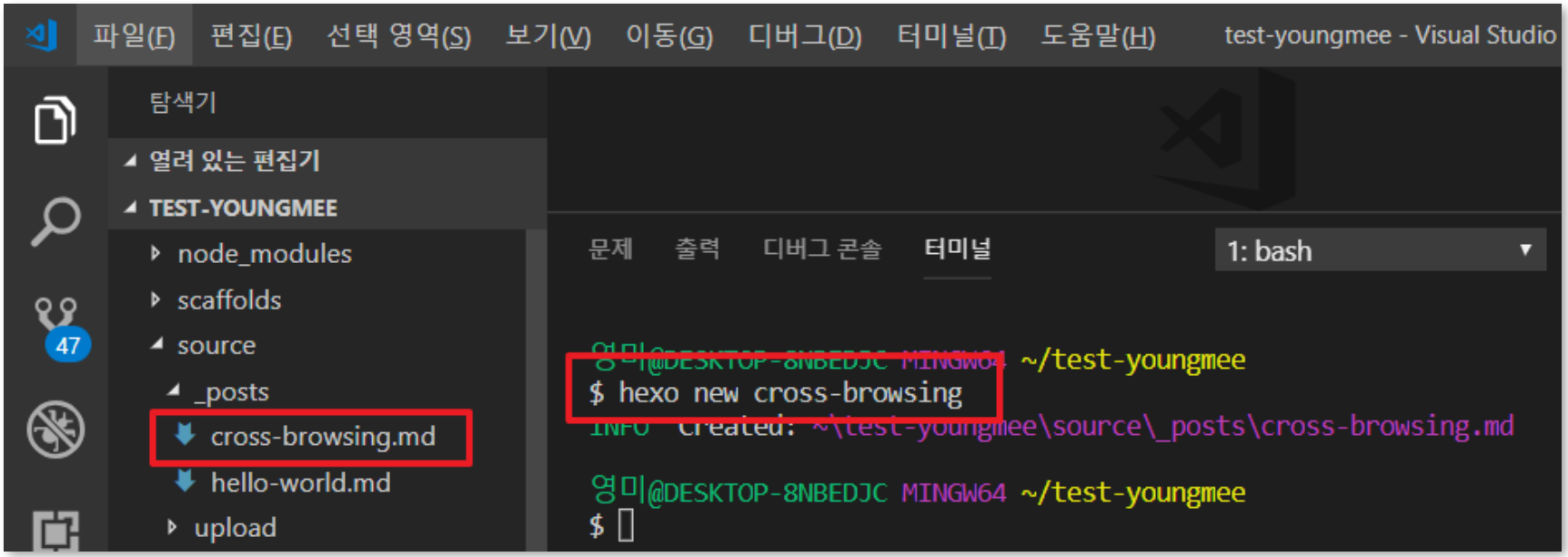
```
영미@DESKTOP-8NDEDJC MINGW64 ~/test-youngmee
$ hexo server
INFO  Start processing
INFO  Hexo is running at http://localhost:4000 . Press Ctrl+C to stop.
```

수정된 hero 이미지 확인

- 새로고침하여 변경된 내용 확인

새 글(new post) 생성

- hexo new 파일명으로 md 파일 생성
- 탐색기에서 source>_posts 안에 생성된 md 파일 확인



새 글(new post) 작성

- cross-browsing.md 파일의 front-matter 수정

```
---
title: cross-browsing
date: 2019-05-12
03:01:46
tags:
---
```

```
---
title: cross-browsing
date: 2019-05-09
tags:
- web browser
- web standards
- web accessibility
categories:
- 개발노트
- A11y
thumbnail: upload/browsers.png
---
```

새 글(new post) 등록 확인

- 새로고침하여 추가된 acticle 확인

새 글(new post) 수정

- markdown을 이용하여 cross-browsing.md 파일의 내용 추가

cross-browsing.md

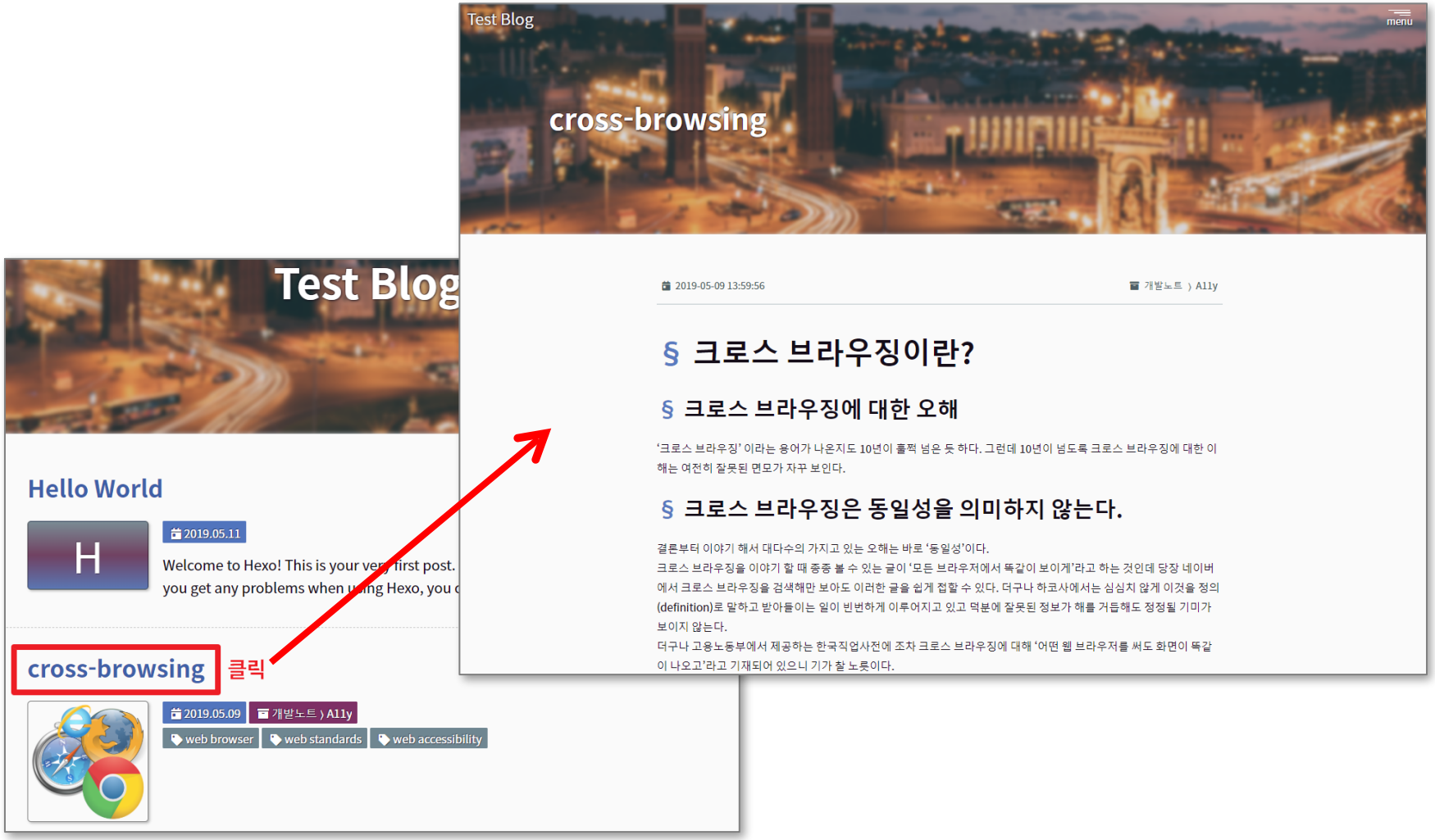
```
1   ---
2   title: cross-browsing
3   date: 2019-05-09 13:59:56
4   tags:
5   - web browser
6   - web standards
7   - web accessibility
8   categories:
9   - 개발노트
10  - A11y
11  thumbnail: upload/brows
12  ---
13
```

```
# 크로스 브라우징이란?

## 크로스 브라우징에 대한 오해
'크로스 브라우징' 이라는 용어가 나온지도 10년이 훌쩍 넘은 듯 하다. ...

## 크로스 브라우징은 동일성을 의미하지 않는다.
결론부터 이야기 해서 대다수의 가지고 있는 오해는 바로 '동일성'이다.
 ...

## 크로스 브라우징은 동등성(등가성)을 의미한다.
> 웹 페이지의 상호 호환성(Cross Browsing) 구축에 대해 이야기하면 기본적인 오해가 있다. ...
```

수정된 새 글(new post) 확인

Go

- 새롭게 추가된 글의 제목을 클릭하여 추가된 글의 내용 확인



hexo 블로그를 github에 배포하기

멀티캠퍼스

배포(deployment)를 위한 패키지 설치

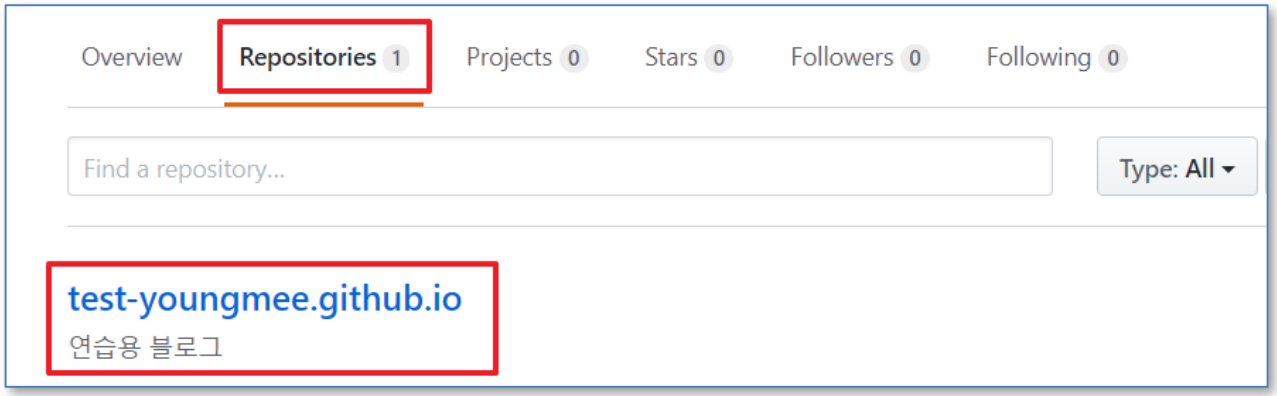
- hexo-deployer-git 설치
  - npm install hexo-deployer-git --save

```
영미@DESKTOP-8NBEDJC MINGW64 ~/test-youngmee
$ npm install hexo-deployer-git --save
npm WARN babel-eslint@10.0.1 requires a peer of eslint@>= 4.12.1 but none is installe
d. You must install peer dependencies yourself.
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.9 (node_modules\fsevents
):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.9
: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})

+ hexo-deployer-git@1.0.0
added 59 packages from 48 contributors and audited 7082 packages in 58.975s
found 6 vulnerabilities (5 low, 1 moderate)
  run `npm audit fix` to fix them, or `npm audit` for details
```

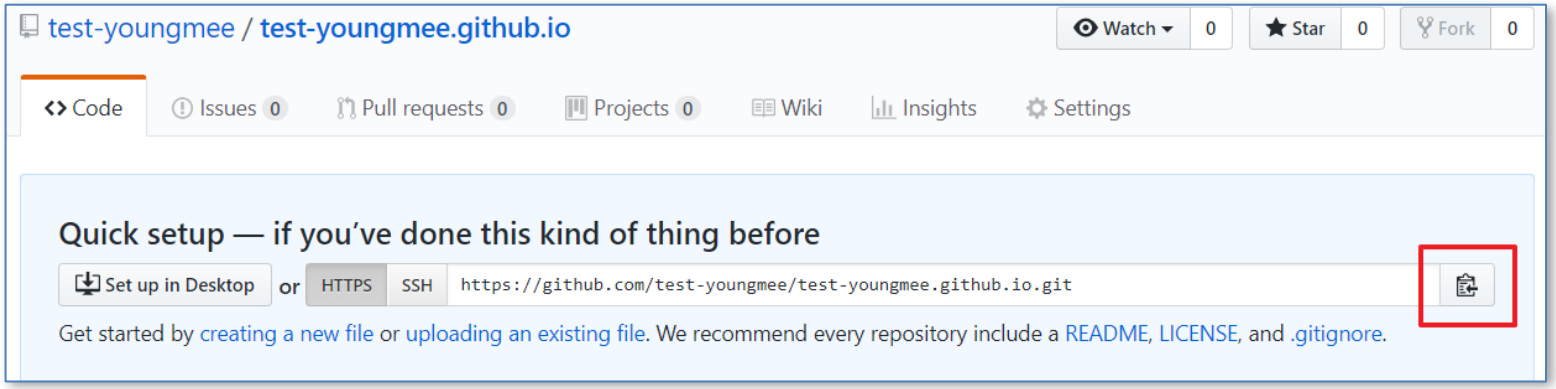
온라인 저장소 주소 복사

- Deployment를 위한 URL 복사

1. github.com의 Repositories 탭에서 사용할 저장소 클릭

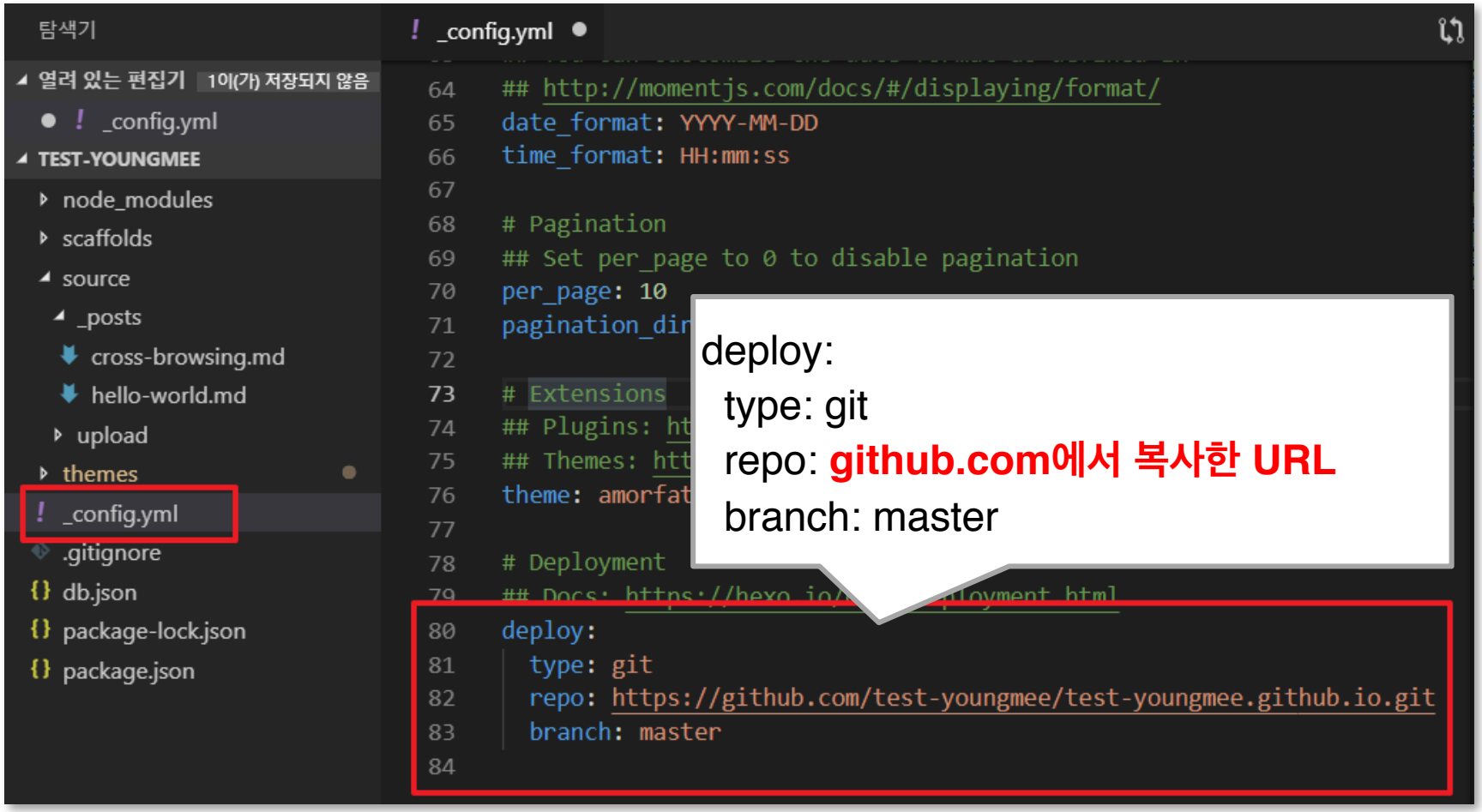


2. 표시된 곳을 클릭하여 URL 복사



배포(deployment) 주소 수정

- "_config.yml" 파일 수정

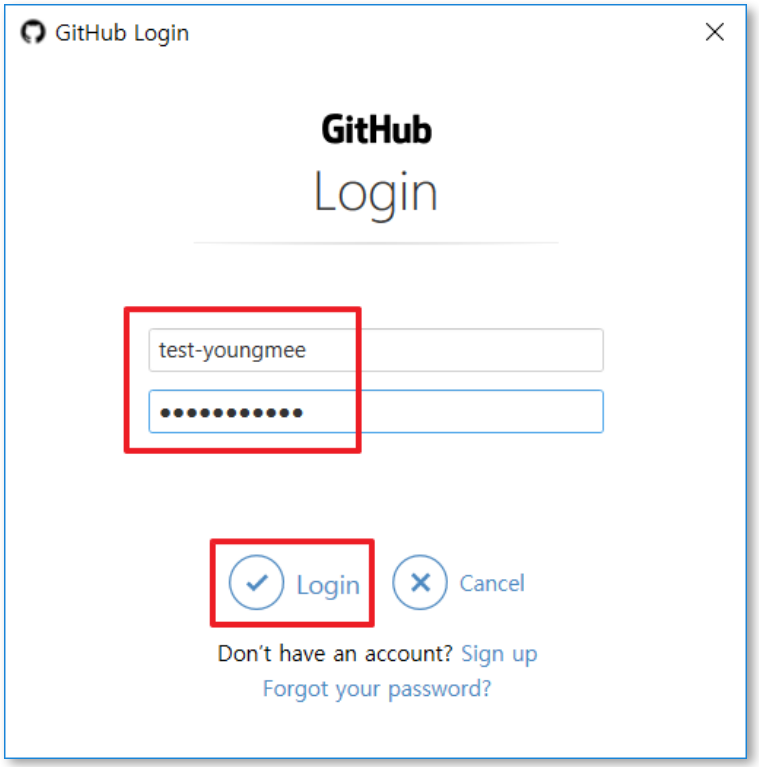


github에 배포(deployment)

- hexo deploy 명령어로 deployment 수행

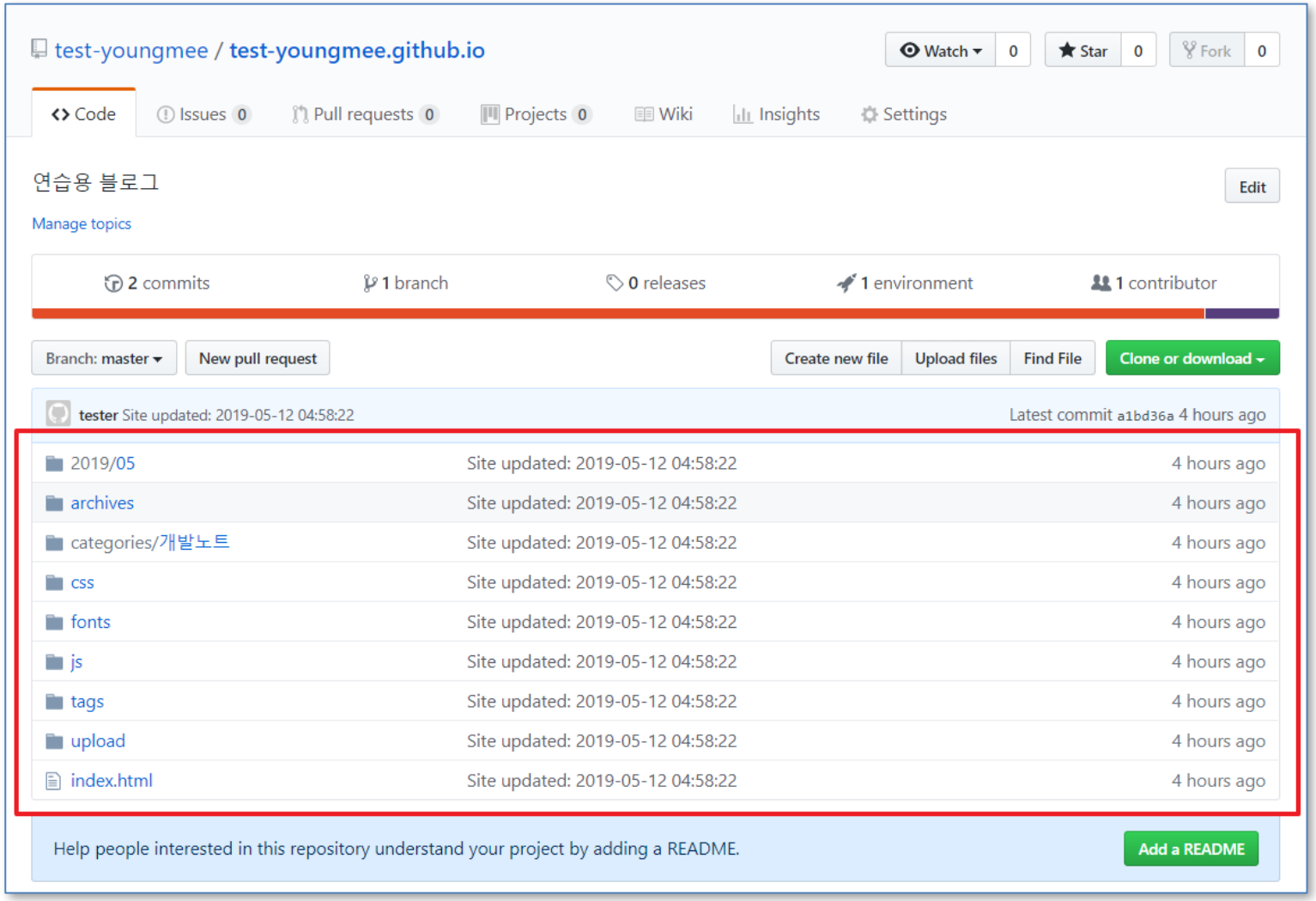
```
영미@DESKTOP-8NBEDJC MINGW64 ~/test-youngmee
$ hexo deploy
INFO  Start processing
INFO  Files loaded in 2.7 min
INFO  Generated: index.html
```

- 작업 진행 중 나타나는 창에 github의 아이디와 비밀번호 입력 후 [Login] 클릭



github에 배포(deployment)

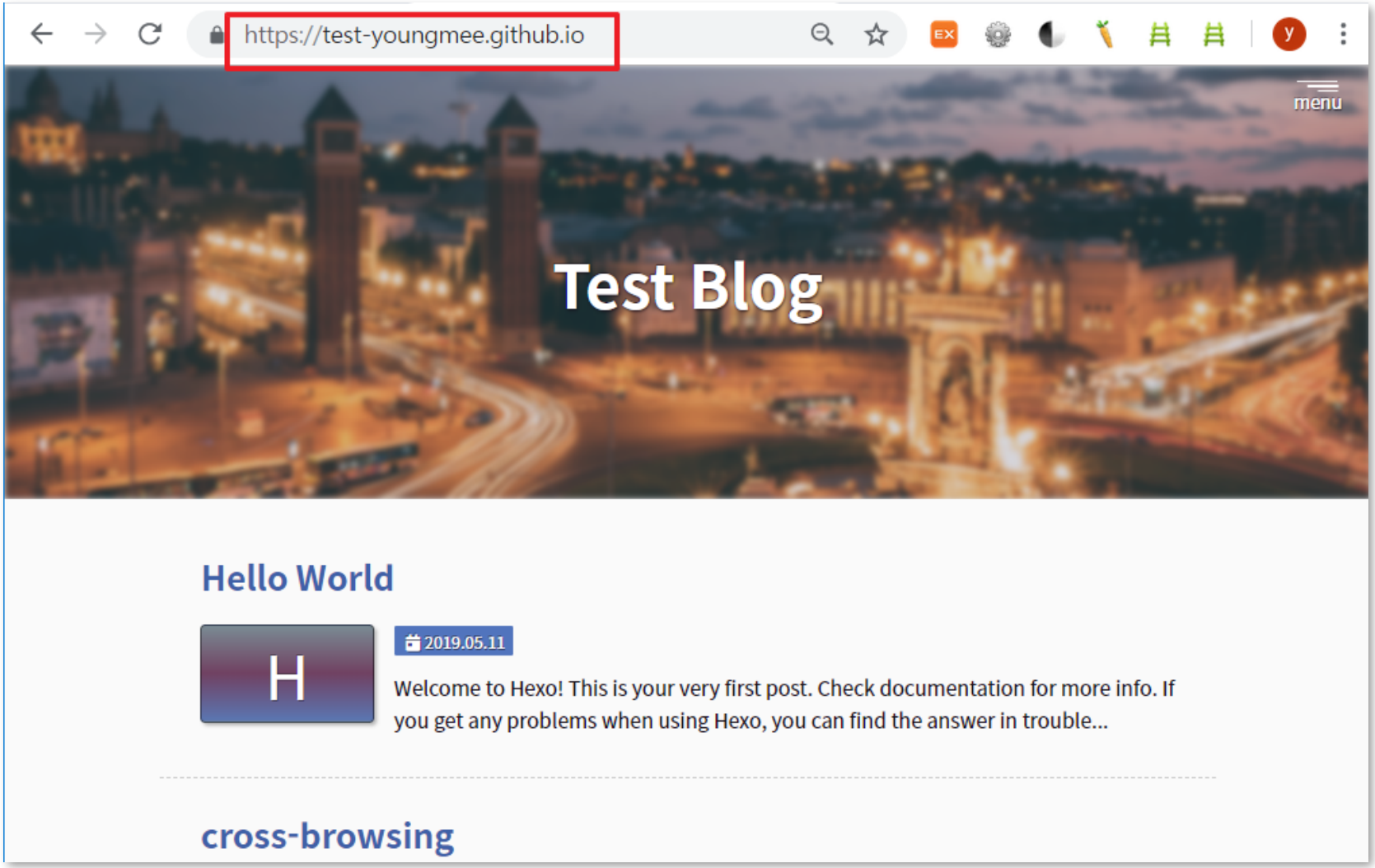
- hexo deploy 명령어 사용 후 github 온라인 저장소에 배포된 내용 확인



배포(deployment) 결과 확인

Go

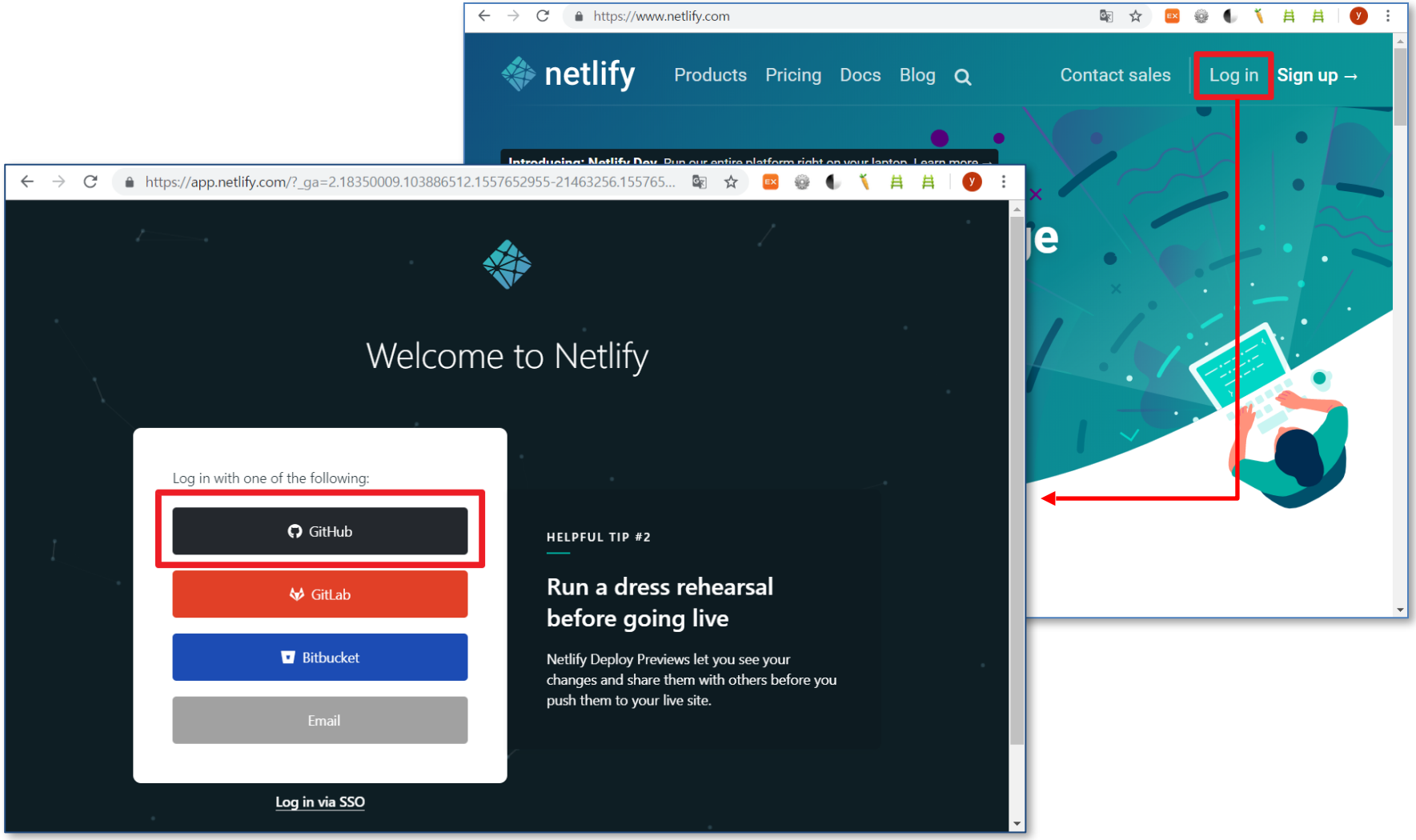
- 브라우저 주소 창에 저장소이름을 입력하여 온라인 저장소에 적용된 결과 확인



netlify 서비스 연동

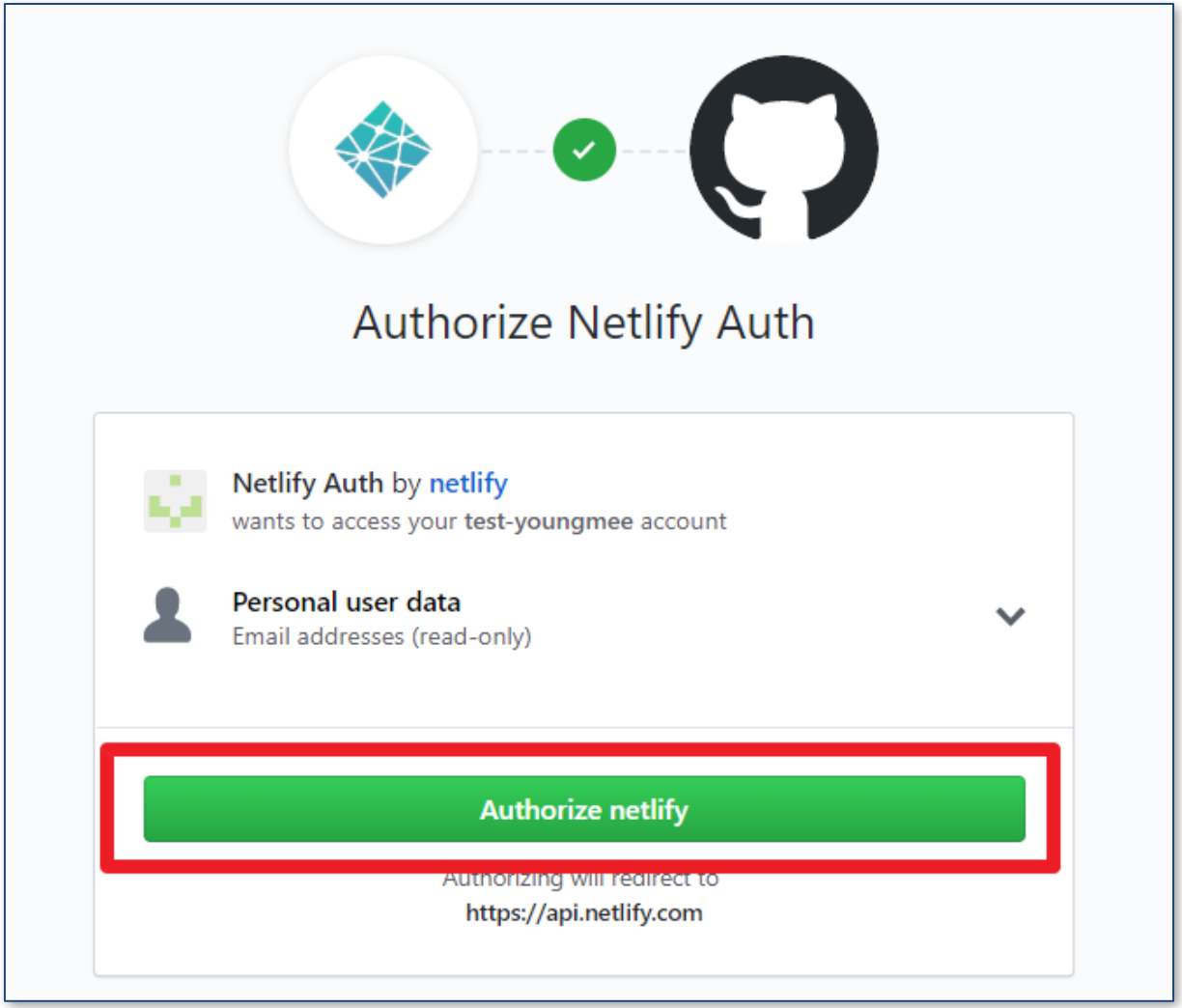
netlify와 github 연동

- https://www.netlify.com/ 사이트에서 [Login] 클릭 후 github로 로그인



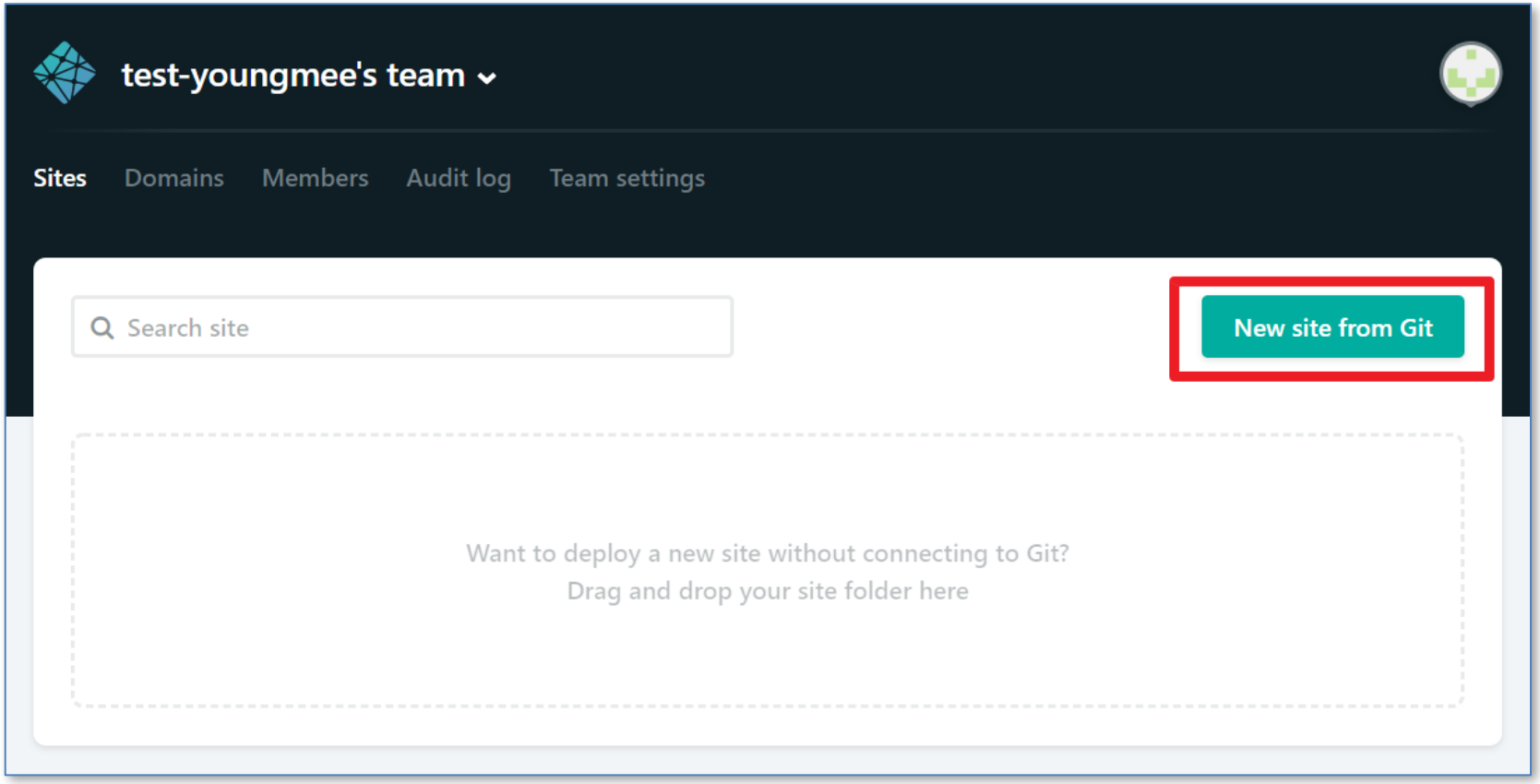
netlify와 github 연동

- [Authorize netlify] 클릭



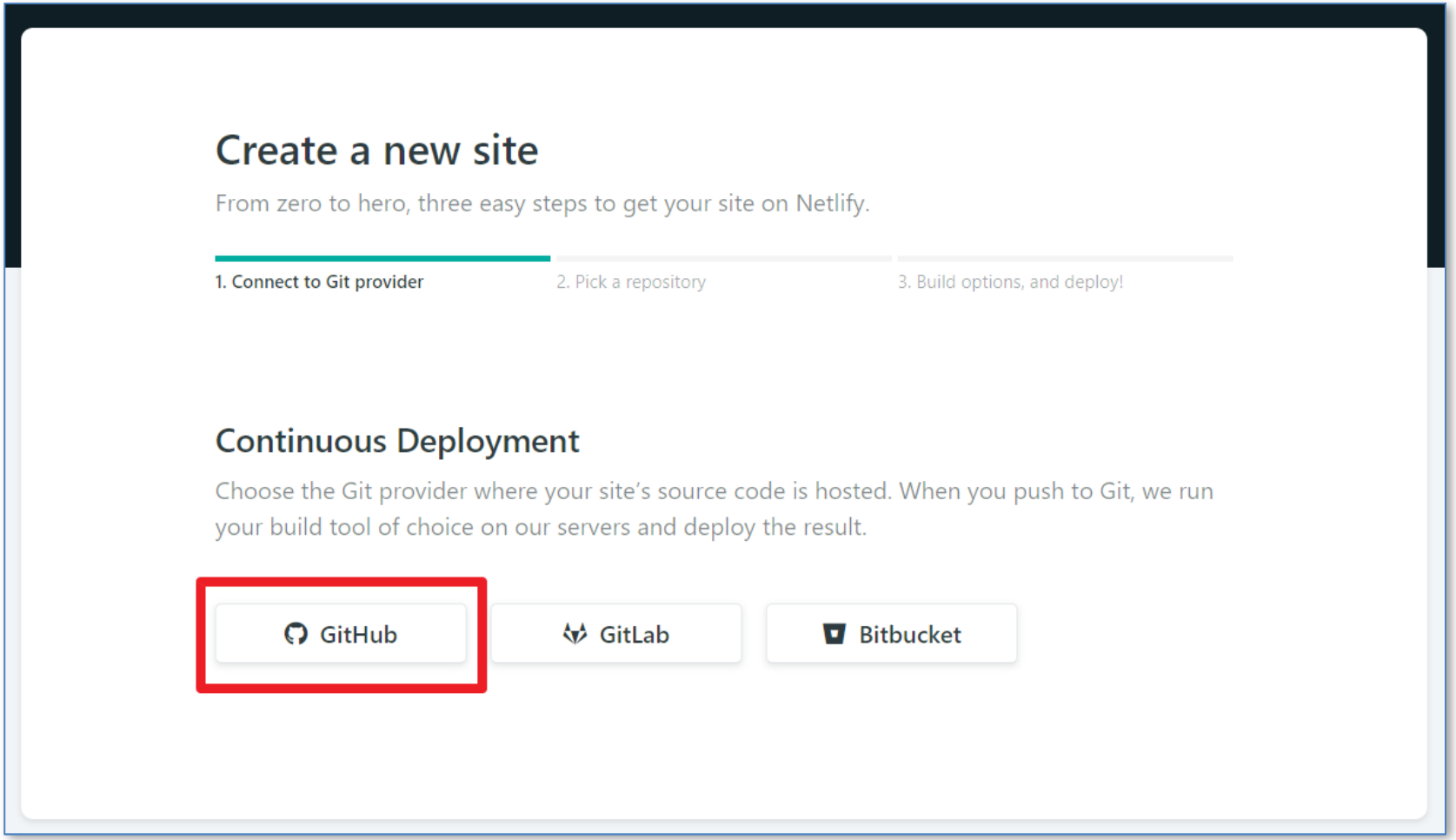
netlify와 github 연동

- [New site from Git] 을 클릭



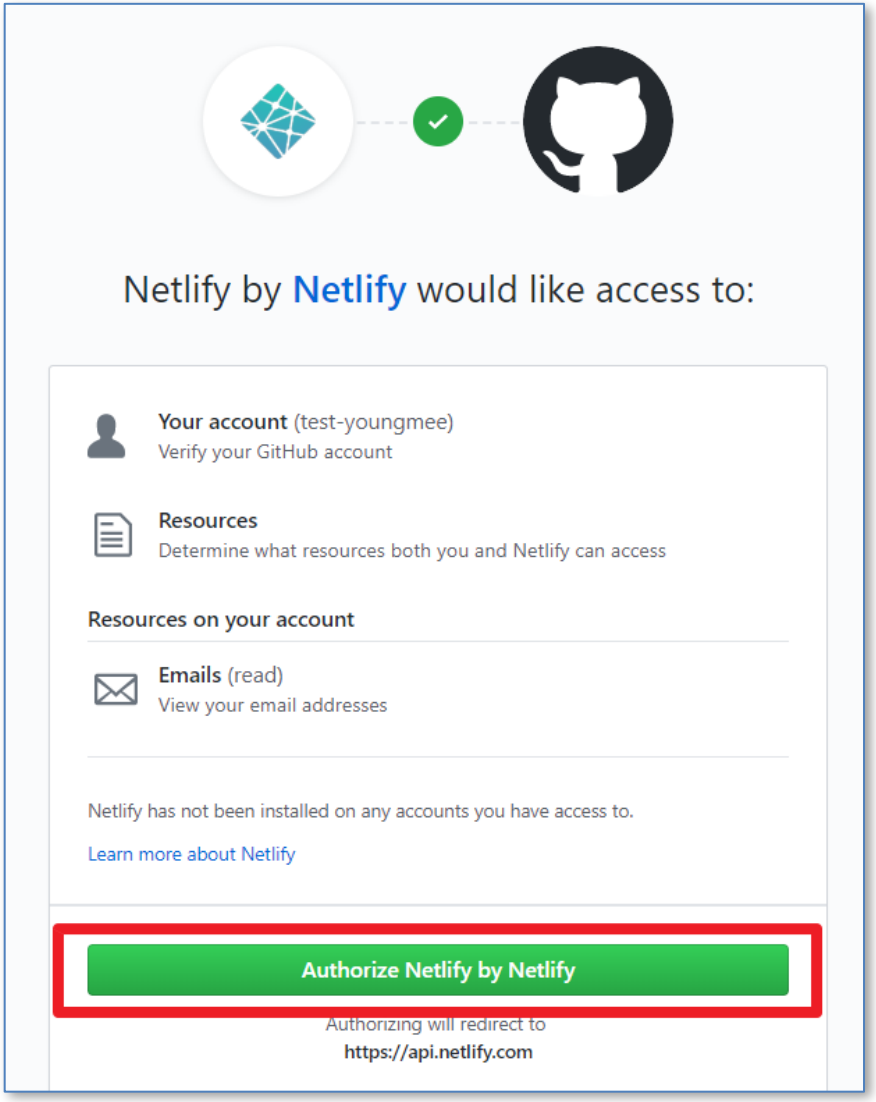
netlify와 github 연동

- [GitHub] 를 클릭하여 Deployment 진행



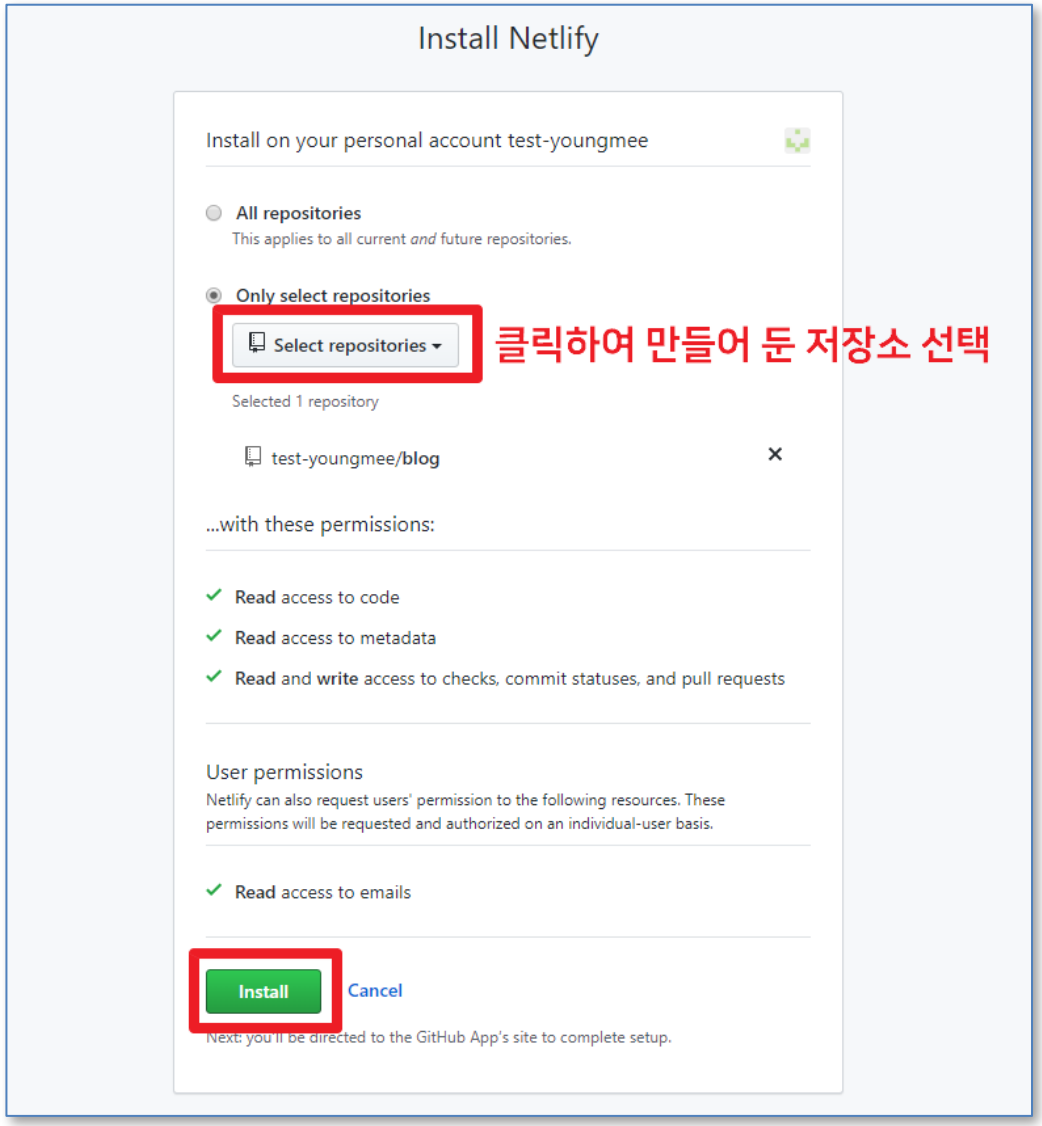
netlify와 github 연동

- [Authorize Netlify by Netlify] 클릭



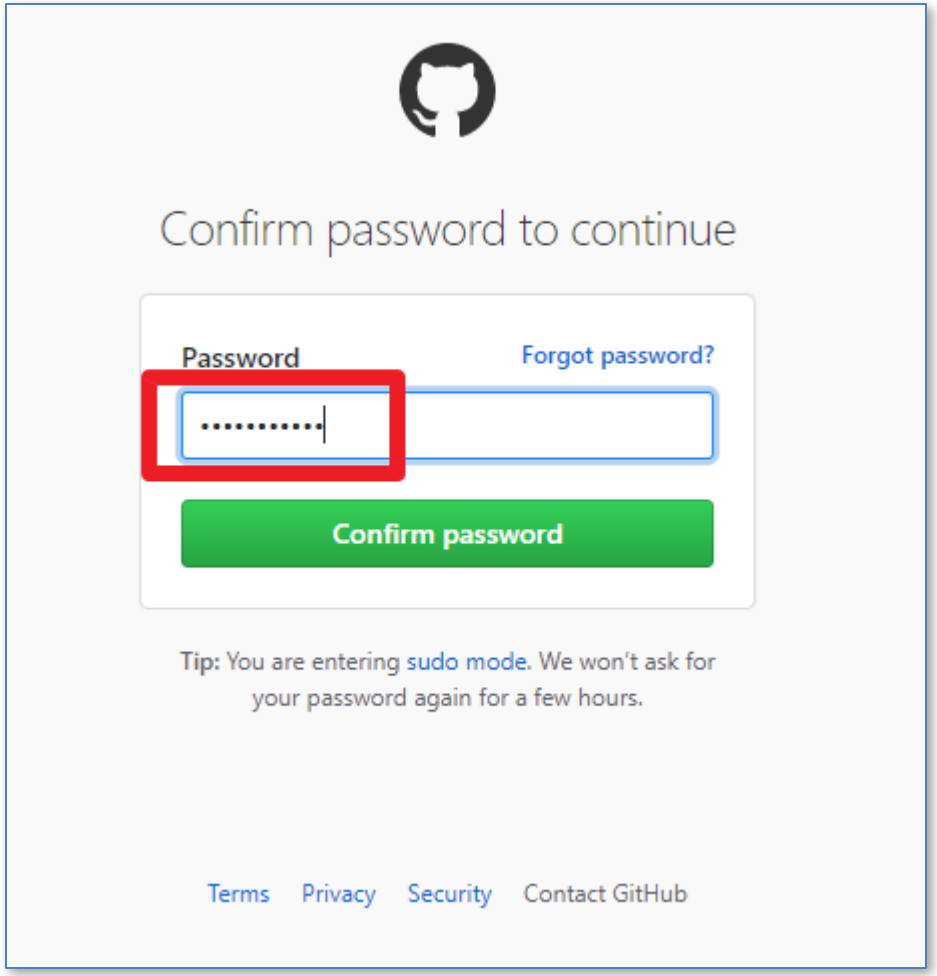
netlify와 github 연동

- [Select repositories] 클릭 후 사용할 저장소 선택



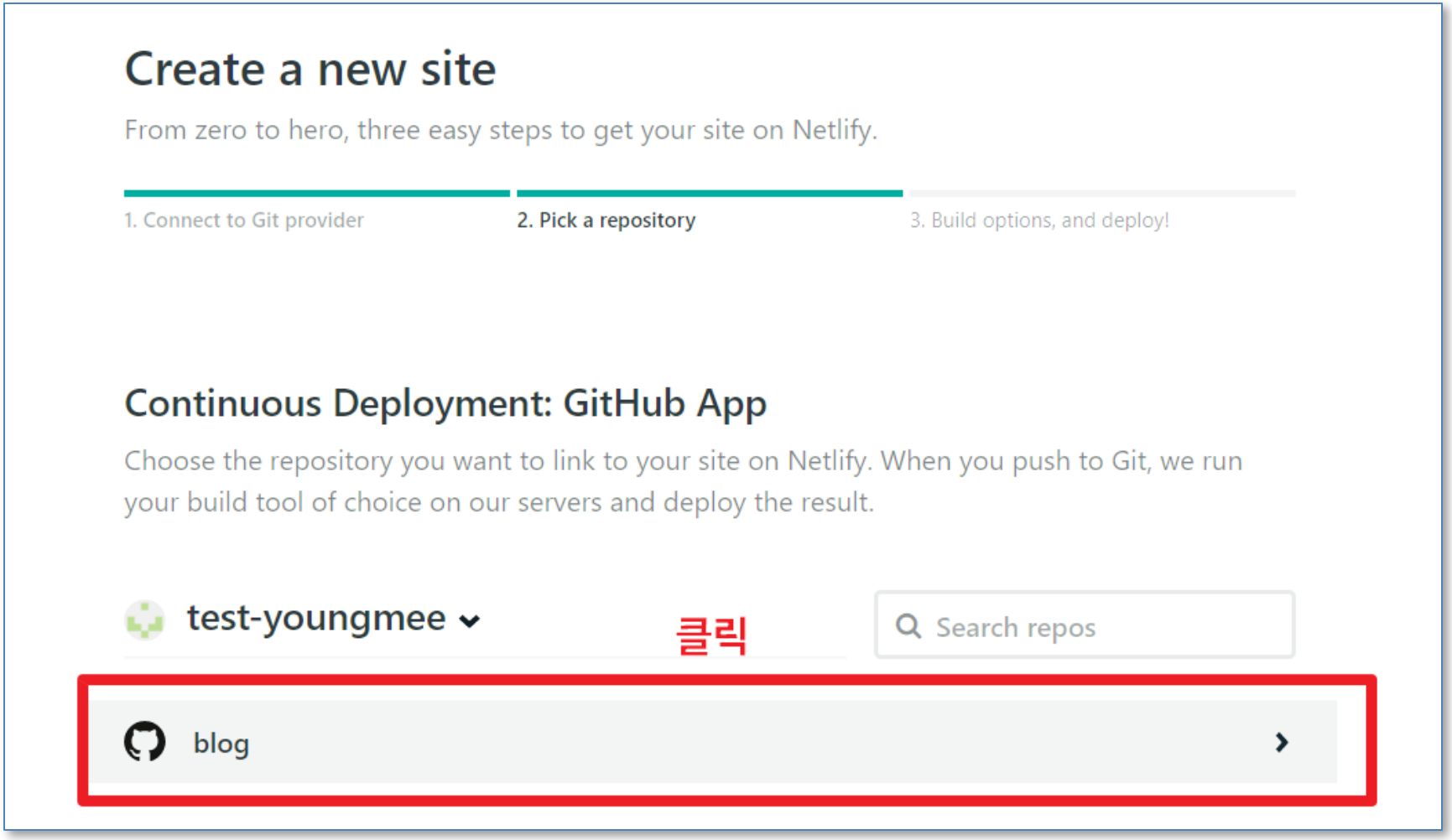
netlify와 github 연동

- github의 비밀번호 입력 후 [Confirm password] 클릭



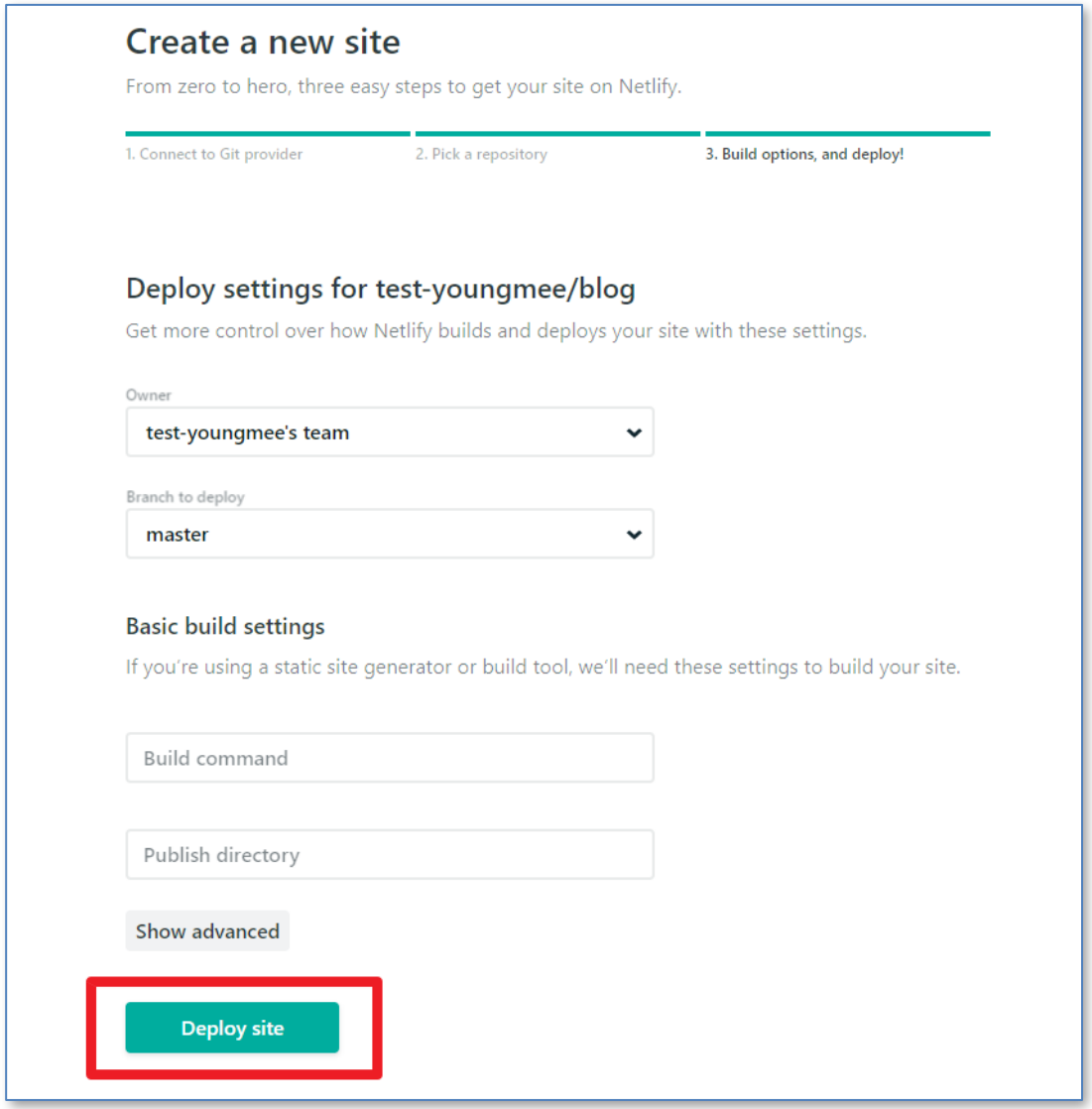
netlify와 github 연동

- github에서 만들어 둔 blog 저장소 선택



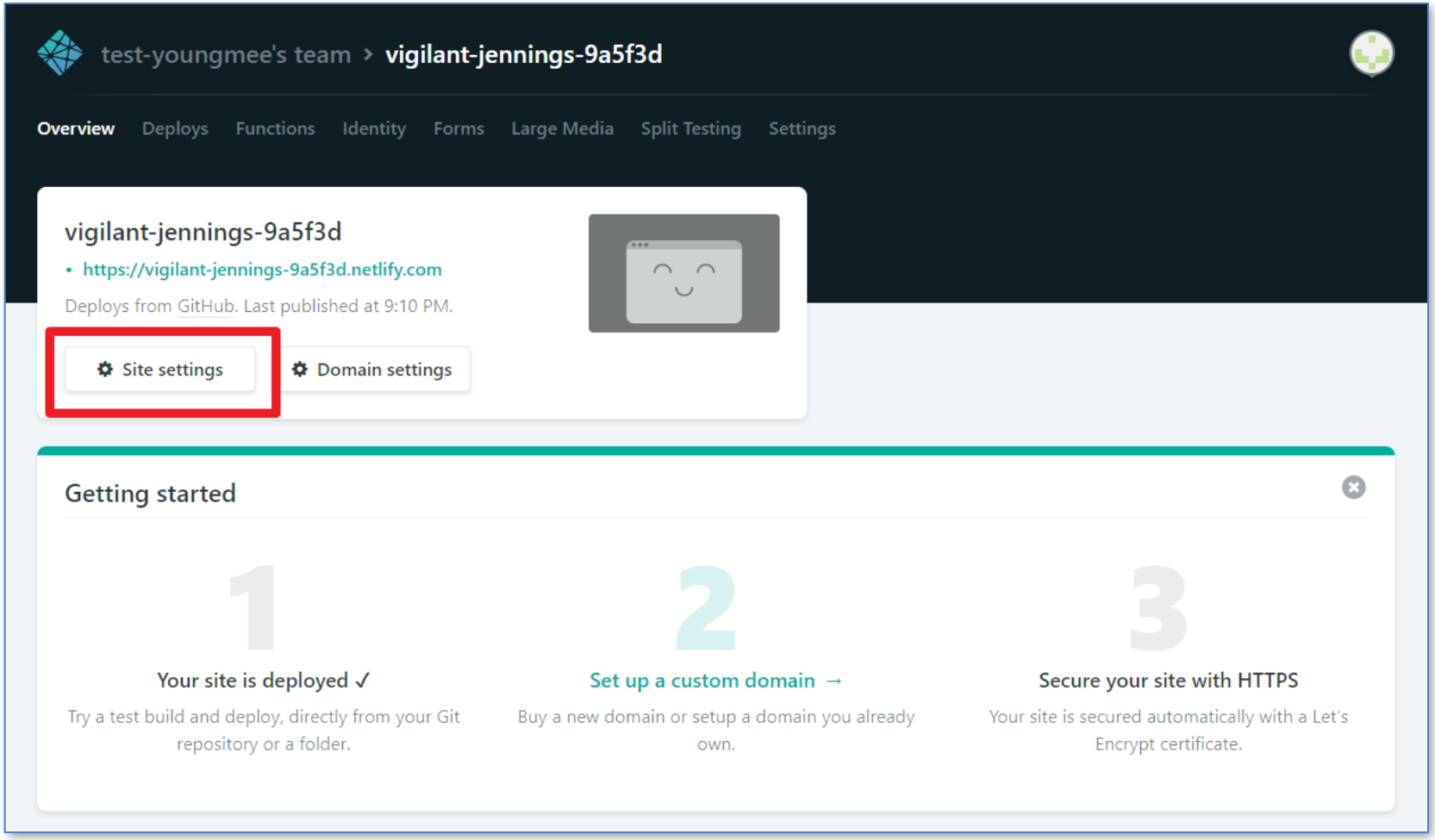
netlify와 github 연동

- [Deploy site] 클릭



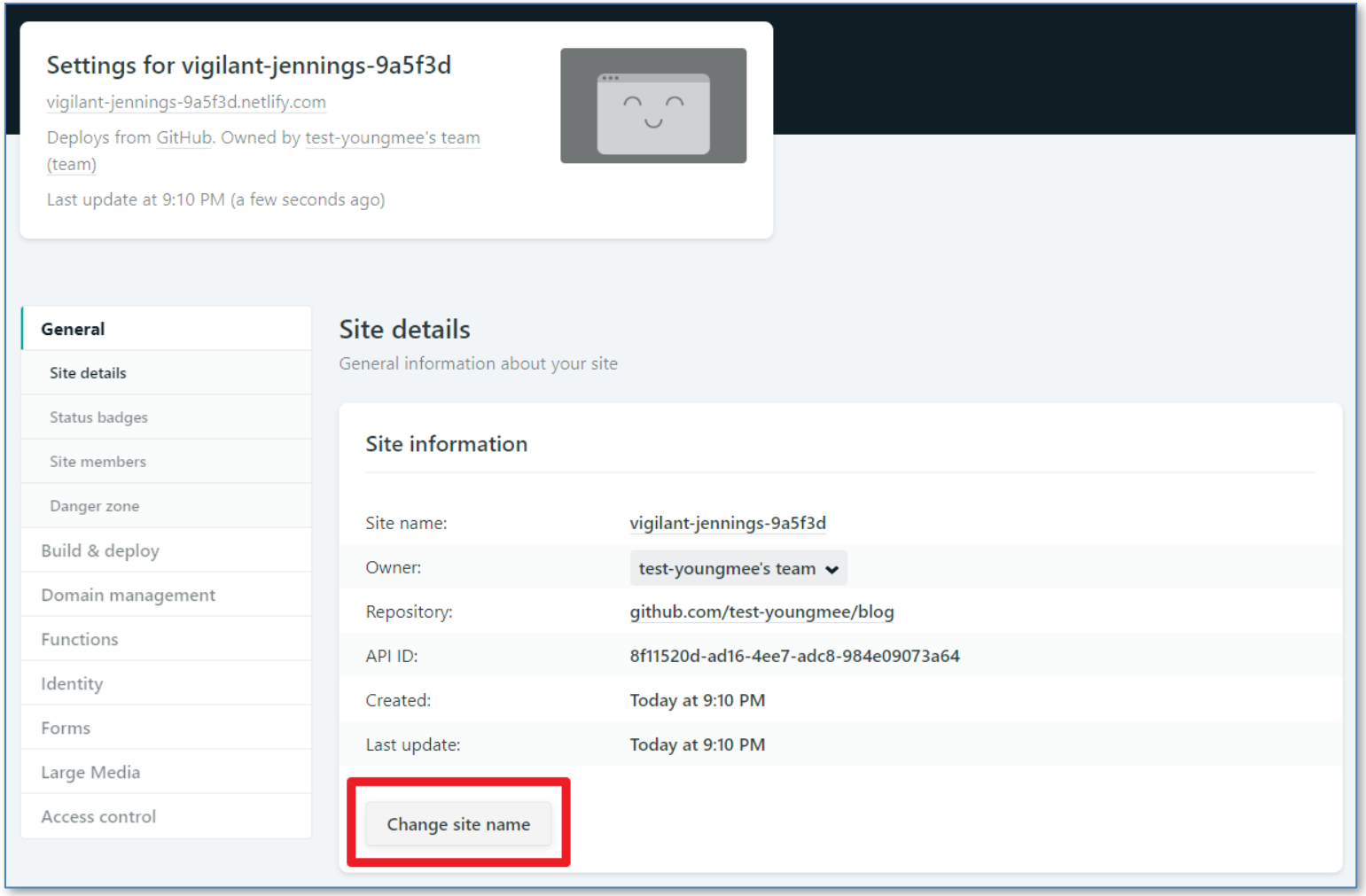
netlify 설정 변경

- 설정 변경을 위해 [Site settings] 클릭



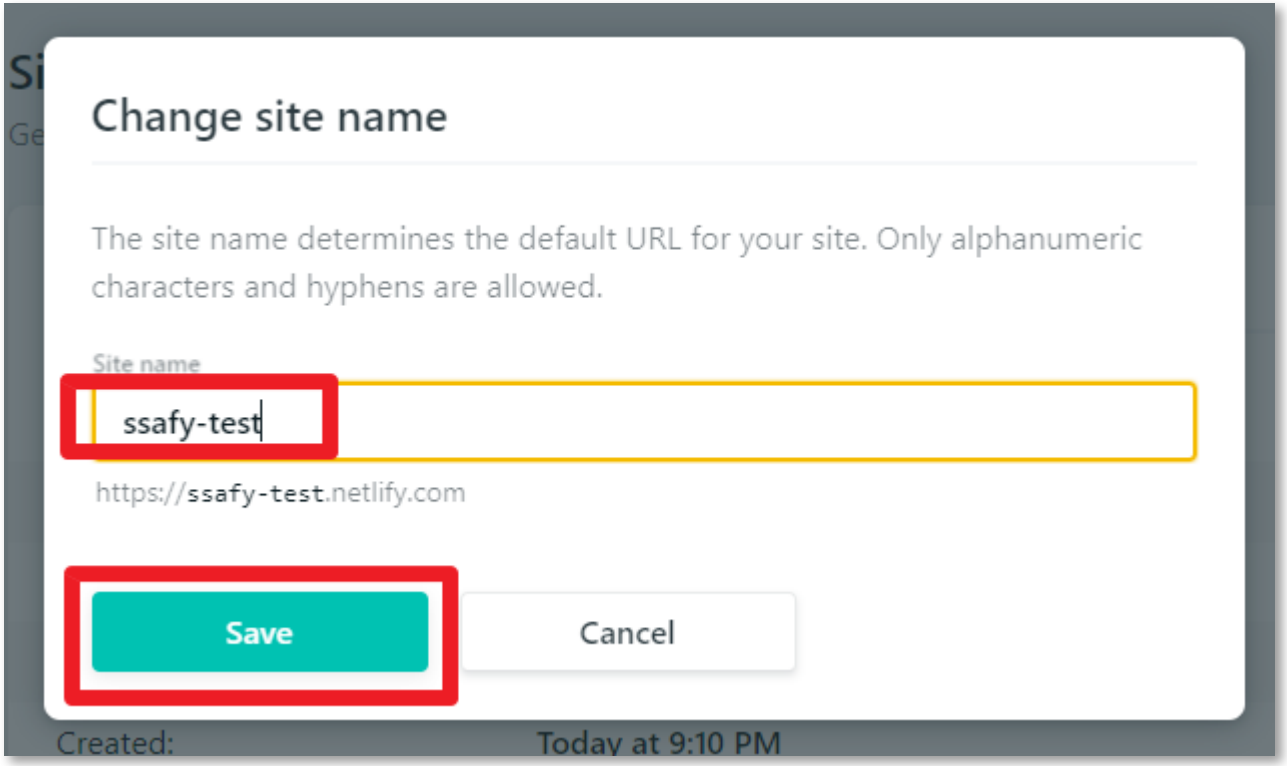
netlify 설정 변경

- [Change site name] 클릭



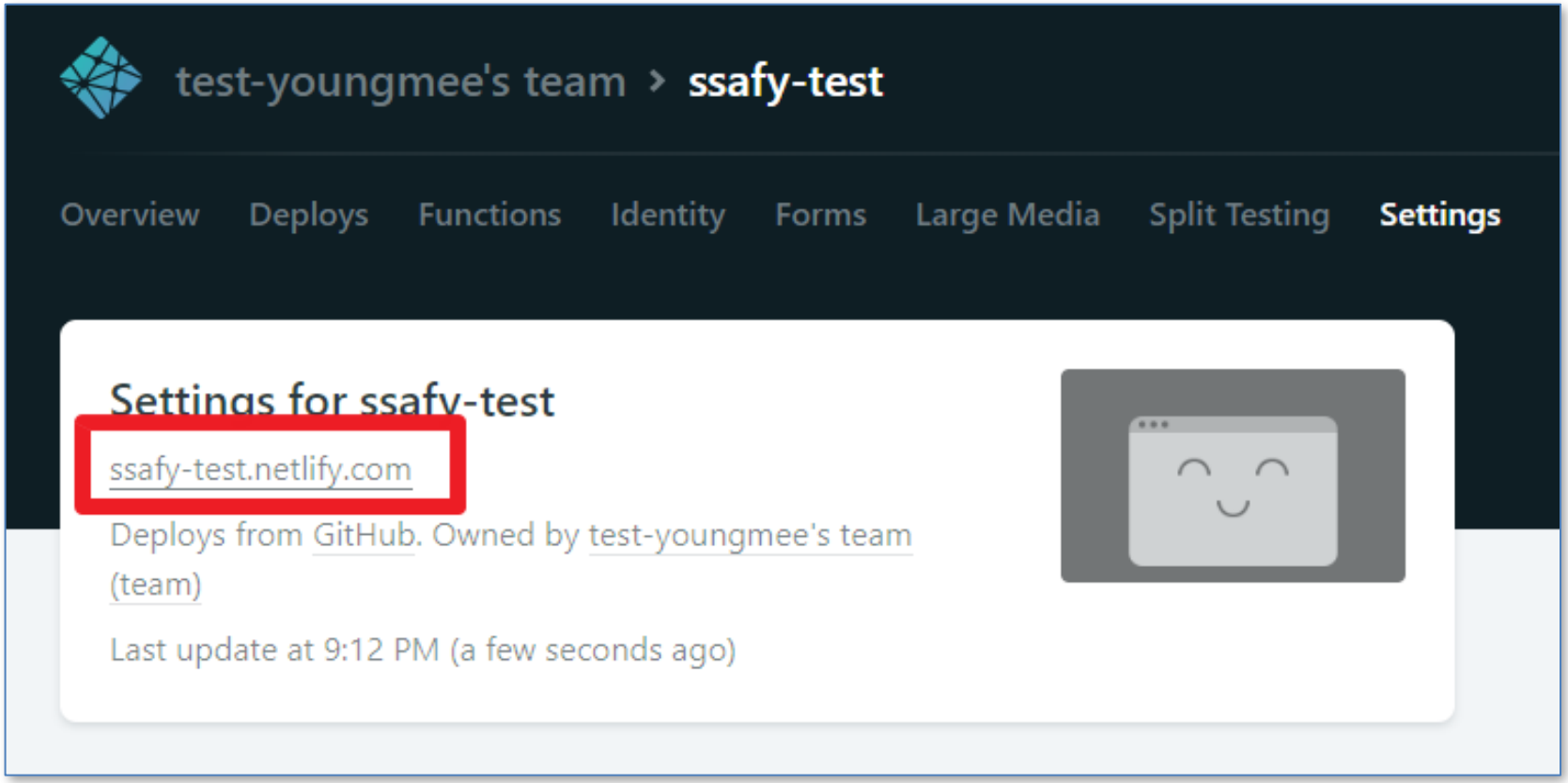
서비스 도메인 변경

- netlify에서 사용할 사이트명 입력 후 [Save] 클릭

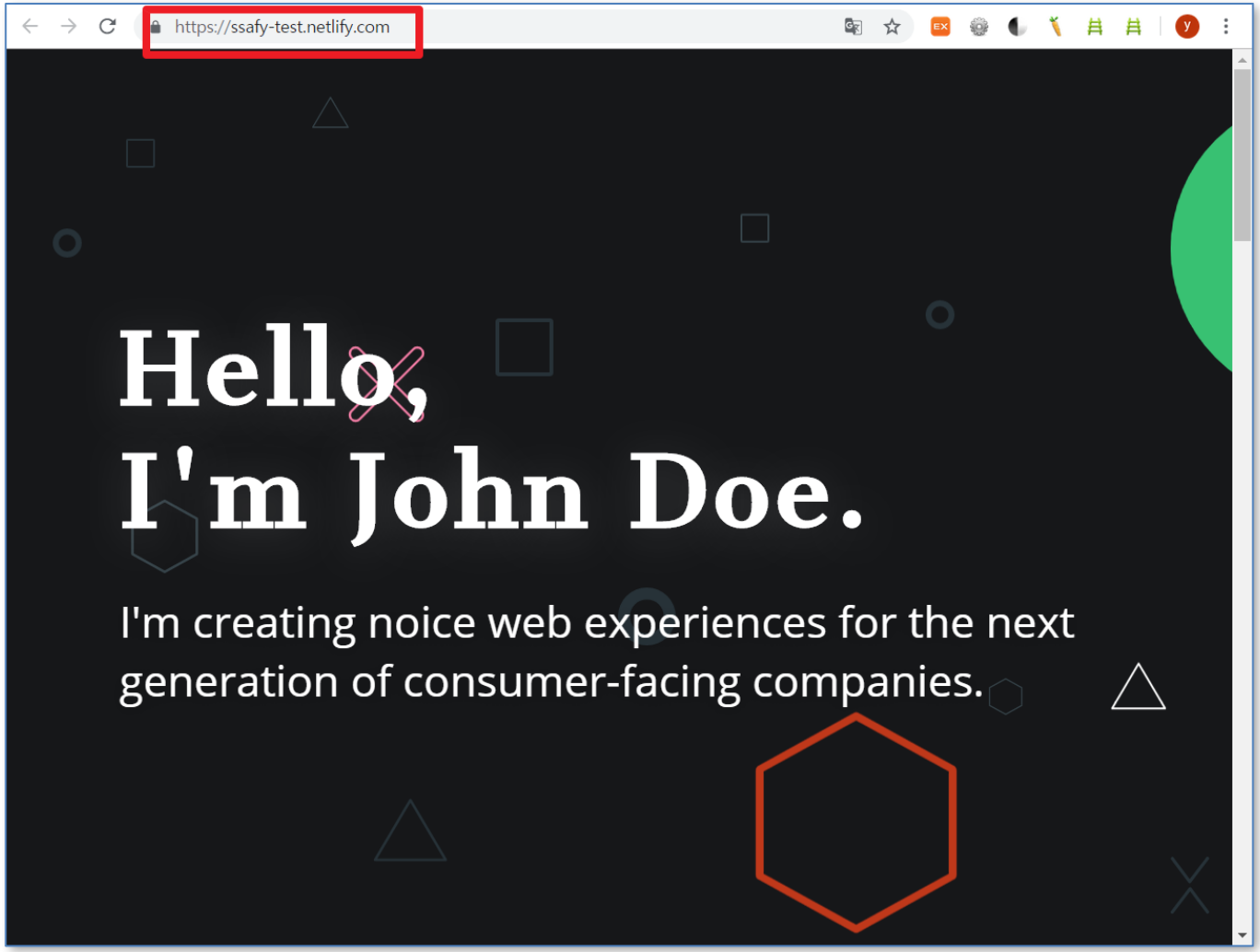


netlify 연동 결과 확인

- 표시된 형태의 도메인 네임 사용 가능



netlify 연동 결과 확인

Go

- 웹 브라우저에 https://ssafy-test.netlify.com/ 를 입력하여 적용된 결과 확인