Table of Contents

Introduction

CI/CD - Serverless Ebook using Gitbook CLI, Github Pages, Github Actions CI/CD, and Calibre

In this tutorial we are going to create an ebook instance using Github, then publish it to the Github pages in an automated manner (on every push to upstream) managed by Github Actions, and it will not deploy only the web version, but the ebook files as wall (in `.pdf` , `.epub` , and `.mobi` format).

For every incoming push to the upstream, Github Actions (CI/CD) will trigger certain processes (like compiling and generating the ebook), then the result will be pushed to the `gh-pages` branch, make it publicly accessible.

Prerequisites

Gitbook CLI

Install gitbook CLI (if you haven't). Do follow the guide on https://github.com/GitbookIO/gitbook-cli.

Github account

Ensure you have a Github account.

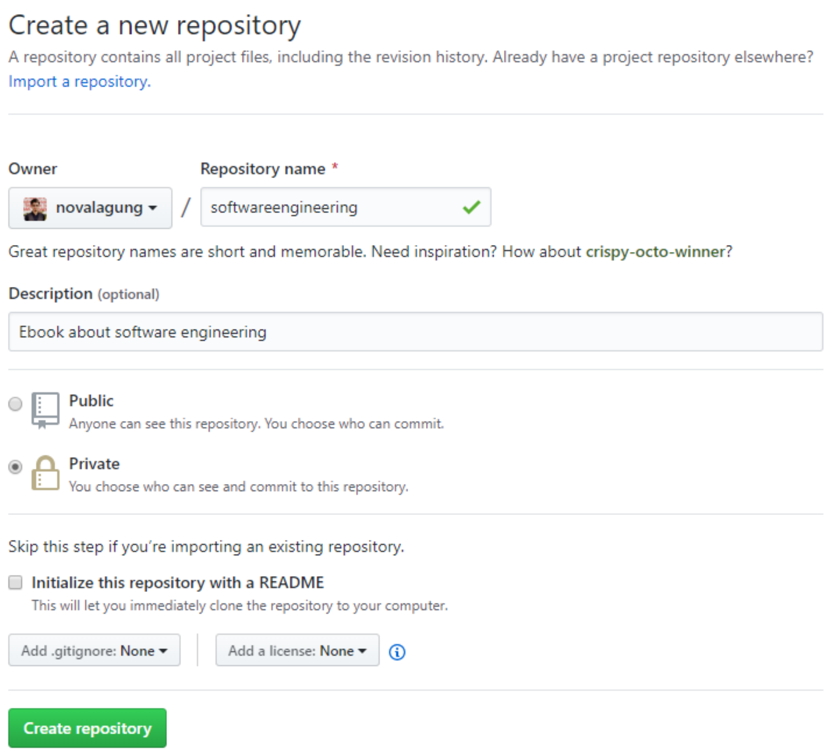
Git client

Ensure you have Git client installed in your local machine.

Guide

Create a Github repo

First, create a new repo in your Github account, it can be a private one or public, doesn't matter. Just for the sake of this tutorial, I am going to pick `softwareengineering` as the repo name.



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner: novalagung / Repository name *: softwareengineering ✓

Great repository names are short and memorable. Need inspiration? How about crispy-octo-winner?

Description (optional): Ebook about software engineering

○ Public
Anyone can see this repository. You choose who can commit.

● Private
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

□ Initialize this repository with a README
This will let you immediately clone the repository to your computer.

Add .gitignore: None

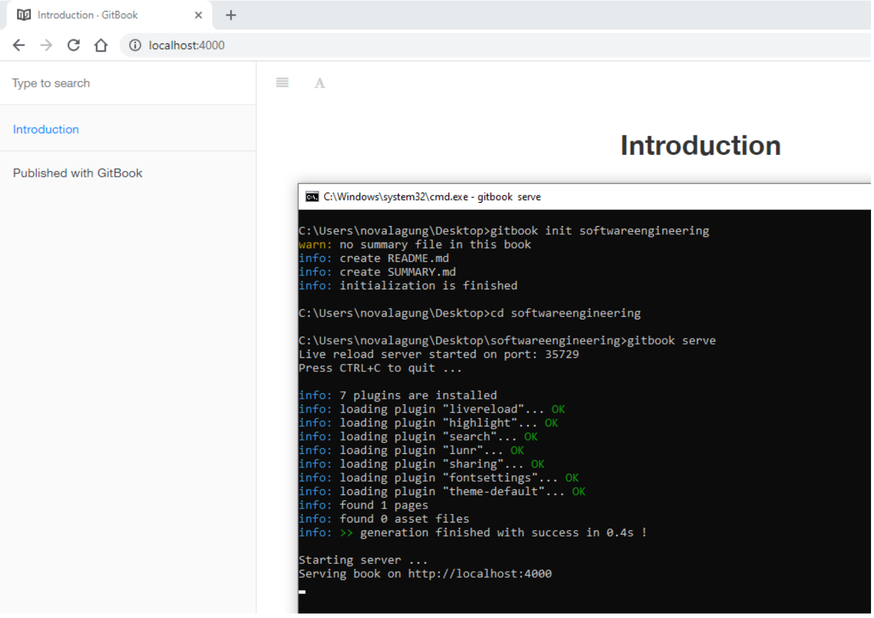
Add a license: None

Create a new Gitbook project

Next, use `gitbook` command line to initialize a new project, name it anyhing. Here I'll use `softwareengineering` , the same one as the git repo name.

After the project setup is finished, try to test it locally.

```
gitbook init softwareengineering
cd softwareengineering
gitbook serve
```



```
C:\Users\novalagung\Desktop>gitbook init softwareengineering
warn: no summary file in this book
info: create README.md
info: create SUMMARY.md
info: initialization is finished

C:\Users\novalagung\Desktop>cd softwareengineering

C:\Users\novalagung\Desktop\softwareengineering>gitbook serve
Live reload server started on port: 35729
Press CTRL+C to quit ...

info: 7 plugins are installed
info: loading plugin "livereload"... OK
info: loading plugin "highlight"... OK
info: loading plugin "search"... OK
info: loading plugin "lunr"... OK
info: loading plugin "sharing"... OK
info: loading plugin "fontsettings"... OK
info: loading plugin "theme-default"... OK
info: found 1 pages
info: found 0 asset files
info: >> generation finished with success in 0.4s !

Starting server ...
Serving book on http://localhost:4000
```

As we can see from image above, the web version of the book is running up.

Next, we are going to use Github Action plugin peaceiris/actions-gh-pages to automate pushing resources from git repo server to the `gh-pages` .

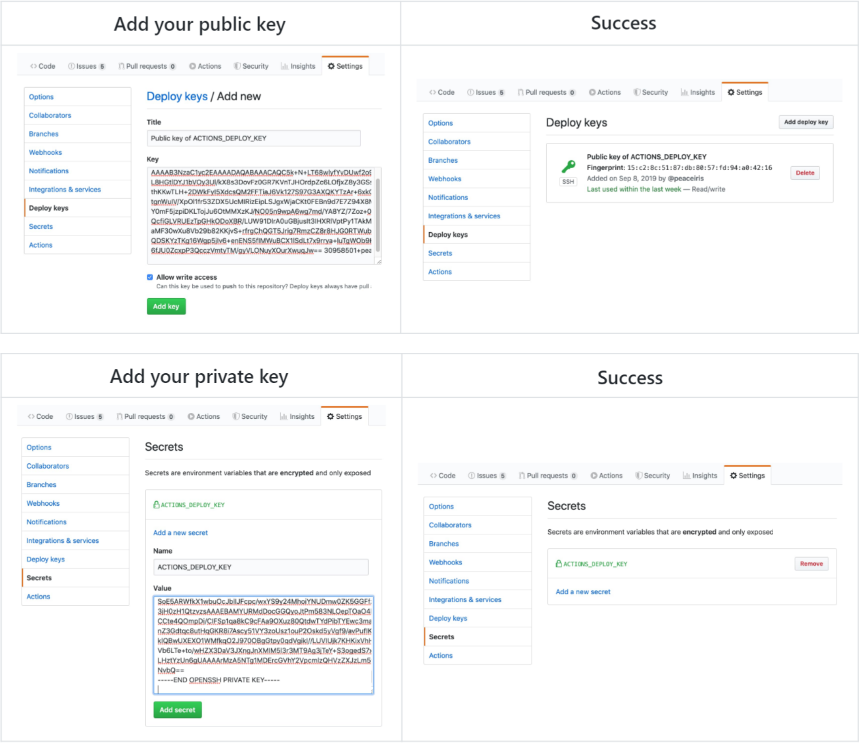
To make this scenario happen, first, generate new key pair using `ssh-keygen` command below. We will use the keys as Github deploy key.

```
ssh-keygen -t rsa -b 4096 -C "$(git config user.email)" -f gh-pages -N ""
# You will get 2 files:
#   gh-pages.pub (public key)
#   gh-pages     (private key)
```

The above command generates two files:

- `gh-pages.pub` file as the public key
- `gh-pages` file as the private key

Upload these two files into repo's project keys and secret menu respectively. To do that, open the repo, click **Settings**, then do follow the steps below:



Add your public key

Deploy keys / Add new

Title: Public key of ACTIONS_DEPLOY_KEY

Key:

```
AAAAB3NzaC1yc2EAAAADAQABAAACAQC5k+N+LT68wlyfYvDUwf2o
L8HGtlDYJ1bVOy3Ul/kX8s3DovFz0GR7KVnTJHOrdpZc6LOfjxZ8y3GS
thKKwTLH+2DWkFyI5XdcsQM2FFTiaJ6Vk127S97G3AXQKYTzAr+6xk
tgnWuIV/XpOl1fr53ZDX5UcMIRizEipLSJgxWjaCKt0FEBn9d7E7Z94X8
Y0mF5jzpiDKLTojJu6OtMMXzKJ/NO05n9wpA6wg7md/YA8YZ/7Zoz+0
QcfiGLVRUEzTpGHkODoXBR/LUW91DIrA0uGBjuslt3IHXRlVptPy1TAkM
aMF30wXu8Vb29b82KKjvS+rfrgChQGT5Jrig7RmzCZ8r8HJG0RTWub
QDSKYzTKg16Wgp5jlv6+enENS5fIMWuBCX1lSdLt7x9rrya+luTgWOb9
6fJU0ZcxpP3QcczVmtyTM/gyVLONuyXOurXwuqJw== 30958501+pea
```

☑ Allow write access
Can this key be used to push to this repository? Deploy keys always have pull

Success

Deploy keys

Public key of ACTIONS_DEPLOY_KEY
Fingerprint: 15:c2:8c:51:87:db:80:57:fd:94:a0:42:16
Added on Sep 8, 2019 by @peaceiris
Last used within the last week — Read/write

Add your private key

Secrets

Secrets are environment variables that are **encrypted** and only exposed

ACTIONS_DEPLOY_KEY

Add a new secret

Name: ACTIONS_DEPLOY_KEY

Value:

```
SoE5ARWfkX1wbuOcJbllJFcpc/wxYS9y24MhoiYNUDmw0ZK5GGF
3jH0zH1QtzvzsAAAEBAMYURMdDocGGQyoJtPm583NLOepTOaO4
CCte4QOmpDi/ClFSp1qa8kC9cFAa9OXuz80QtdwTYdPibTYEwc3ma
nZ3Gdtqc8utHqGKR8i7Ascy51VY3zoUsz1ouP2Oskd5yVgf9/avPufIK
klQBwUXEXO1WMfkqO2J970OBgGtpy0qdVgikl//LUVlUjk7KHKixVhH
Vb6LTe+to/wHZX3DaV3JXngJnXMIM5I3r3MT9Ag3jTeY+S3ogedS7x
LHztYzUn6gUAAAArMzA5NTg1MDErcGVhY2VpcmlzQHVzZXJzLm5
NvbQ==
-----END OPENSSH PRIVATE KEY-----
```

Success

Secrets

Secrets are environment variables that are **encrypted** and only exposed

ACTIONS_DEPLOY_KEY

Create Github workflow CI/CD file for generating the web version of the ebook

****

Now we are going to make Github able to automatically deploy the web version of the ebook on every push. And we want that to be applied into the first push as well.

Create a new workflow file named `deploy.yml` , place it in `<yourproject>/.github/workflows` , then fill it with the configuration below:

```
# file ./softwareengineering/.github/workflow/deploy.yml

name: 'deploy website and ebooks'

on:
  push:
    branches:
      - master

jobs:
  job_deploy_website:
    name: 'deploy website'
    runs-on: ubuntu-latest
    steps:
    - uses: actions/checkout@v1
    - uses: actions/setup-node@v1
      with:
        node-version: '10.x'
    - name: 'Installing gitbook cli'
      run: npm install -g gitbook-cli
    - name: 'Generating distributable files'
      run: |
        gitbook install
        gitbook build
    - uses: peaceiris/actions-gh-pages@v2.5.0
      env:
        ACTIONS_DEPLOY_KEY: ${{ secrets.ACTIONS_DEPLOY_KEY }}
        PUBLISH_BRANCH: gh-pages
        PUBLISH_DIR: ./_book
```

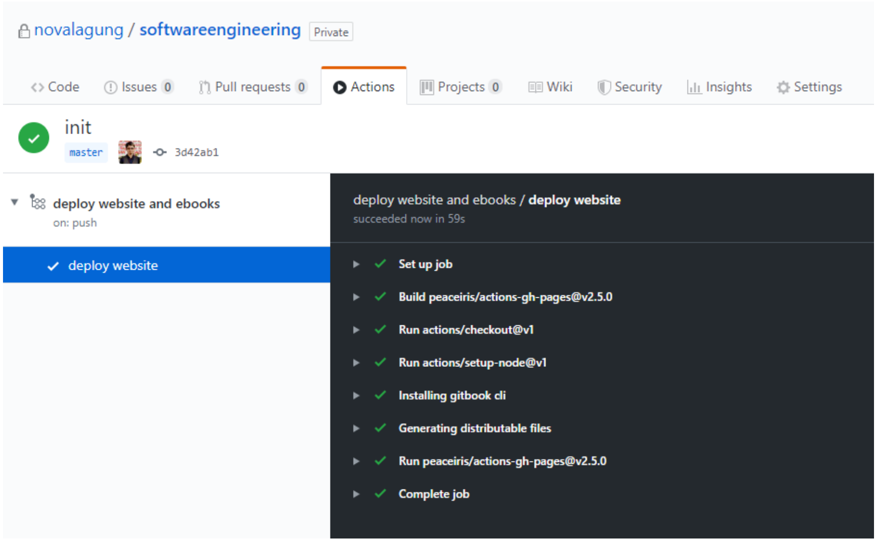
In summary, the workflow above will do these things sequentially:

- Trigger this workflow on every push happens on `master` branch.
- Install `nodejs`.
- Install `gitbook` CLI.
- Build the project.
- use `peaceiris/actions-gh-pages` plugin to deploy the built result to `gh-pages` branch. The Github deploy key that we just uploaded is used by this plugin.

Push project to Github repo

```
cd softwareengineering

# ignore certain directory
touch .gitignore
echo '_book' >> .gitignore

# init git repo
git init
git add .
git commit -m "init"
git remote add origin git@github.com:novalagung/softwareengineering.git

# push
git push origin master
```

Navigate to browser, open your Github repo, click `Actions`, watch a workflow process that currently is running.



init

master 3d42ab1

deploy website and ebooks
on: push

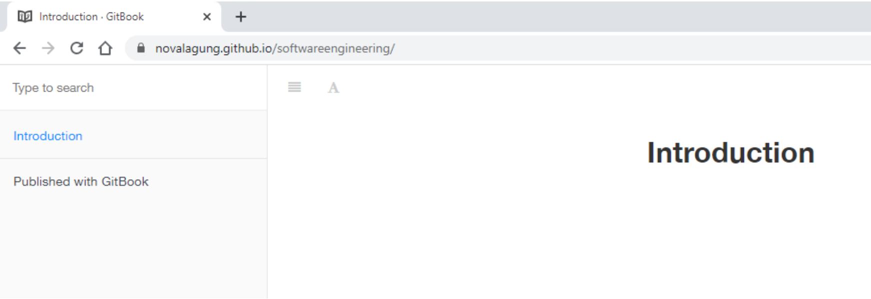
✓ deploy website

deploy website and ebooks / **deploy website**
succeeded now in 59s

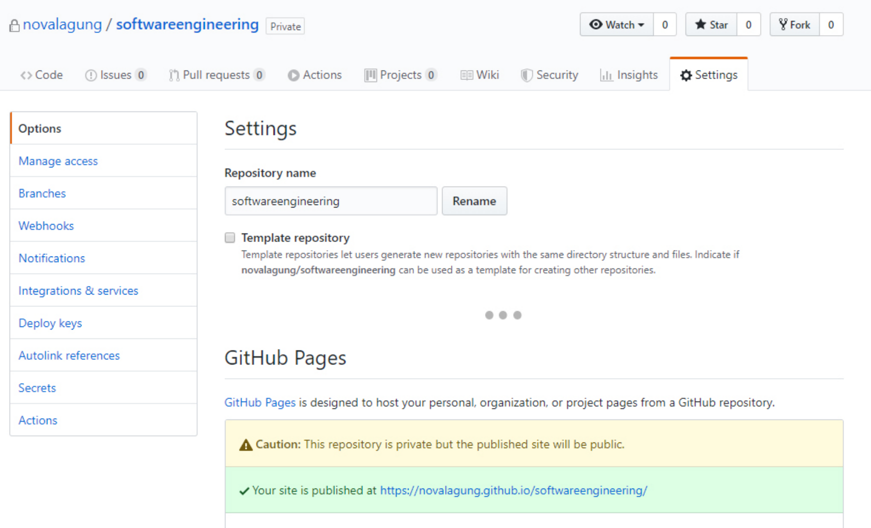
- ✓ Set up job
- ✓ Build peaceiris/actions-gh-pages@v2.5.0
- ✓ Run actions/checkout@v1
- ✓ Run actions/setup-node@v1
- ✓ Installing gitbook cli
- ✓ Generating distributable files
- ✓ Run peaceiris/actions-gh-pages@v2.5.0
- ✓ Complete job

After the workflow is complete, then try to open in the browser the following URL.

```
# https://<github-username>.github.io/<repo-name>
https://novalagung.github.io/softwareengineering/
```



If you are still not sure about what is the valid URL, open **Settings** menu of your Github repo then scrolls down a little bit until **Github Pages** section appears. The Github Pages URL will appear there.



Settings

Repository name

softwareengineering Rename

□ Template repository
Template repositories let users generate new repositories with the same directory structure and files. Indicate if **novalagung/softwareengineering** can be used as a template for creating other repositories.

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

⚠ **Caution:** This repository is private but the published site will be public.

✓ Your site is published at https://novalagung.github.io/softwareengineering/

Modify the workflow file to be able to generate the ebook files

Ok, now we will modify the workflow so it will be able to generate the ebook files (`.pdf` , `.epub` , and `.mobi`), not just the web version.

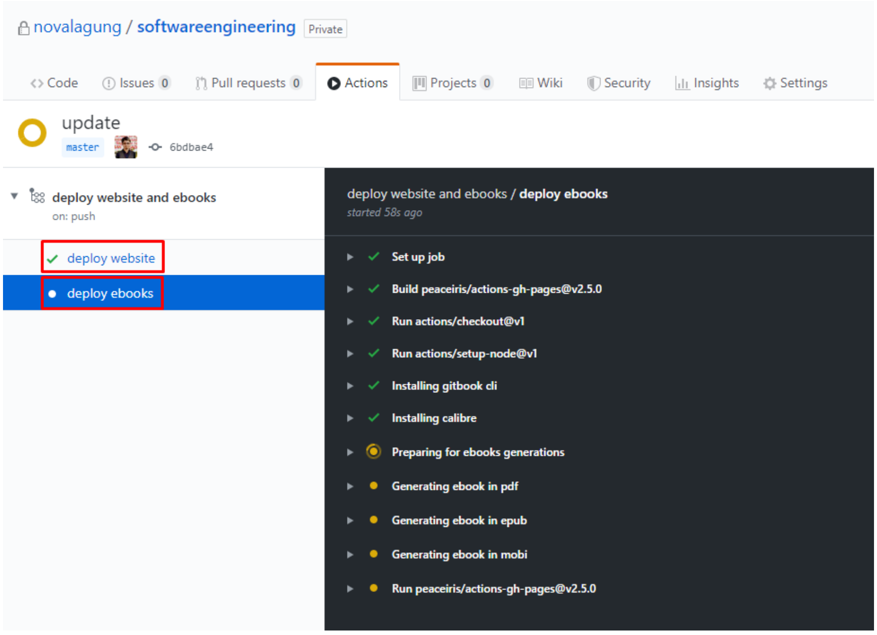
Do open the previous `deploy.yml` file, add a new job called `job_deploy_ebooks` .

```
# file ./softwareengineering/.github/workflow/deploy.yml

name: 'deploy website and ebooks'

on:
  push:
    branches:
      - master

env:
  ebook_name: 'softwareengineeringtutorial'

jobs:
  job_deploy_website:
    # ...
  job_deploy_ebooks:
    name: 'deploy ebooks'
    runs-on: ubuntu-latest
    steps:
    - uses: actions/checkout@v1
    - uses: actions/setup-node@v1
      with:
        node-version: '10.x'
    - name: 'Installing gitbook cli'
      run: npm install -g gitbook-cli
    - name: 'Installing calibre'
      run: |
        sudo -v
        wget -nv -O- https://download.calibre-ebook.com/linux-installer.sh | su
    - name: 'Preparing for ebooks generations'
      run: |
        gitbook install
        mkdir _book
    - name: 'Generating ebook in pdf'
      run: gitbook pdf ./ ./_book/${{ env.ebook_name }}.pdf
    - name: 'Generating ebook in epub'
      run: gitbook epub ./ ./_book/${{ env.ebook_name }}.epub
    - name: 'Generating ebook in mobi'
      run: gitbook mobi ./ ./_book/${{ env.ebook_name }}.mobi
    - uses: peaceiris/actions-gh-pages@v2.5.0
      env:
        ACTIONS_DEPLOY_KEY: ${{ secrets.ACTIONS_DEPLOY_KEY }}
        PUBLISH_BRANCH: ebooks
        PUBLISH_DIR: ./_book
```

The previous `job_deploy_website` is responsible for generating the web-based version of the ebook. This newly created `job_deploy_ebooks` has different purpose, which is to generate the files version of the ebook (`.pdf` , `.epub` , `.mobi`). The generated files later will be pushed to a branch named `ebooks` . The processes will be done by **Calibre**.

Ok, now let's push recent changes into upstream.

```
git add .
git commit -m "update"
git push origin master
```



update

master 6bdbae4

deploy website and ebooks
on: push

✓ deploy website

● deploy ebooks

deploy website and ebooks / **deploy ebooks**
started 58s ago

```
✓ Set up job
✓ Build peaceiris/actions-gh-pages@v2.5.0
✓ Run actions/checkout@v1
✓ Run actions/setup-node@v1
✓ Installing gitbook cli
✓ Installing calibre
● Preparing for ebooks generations
● Generating ebook in pdf
● Generating ebook in epub
● Generating ebook in mobi
● Run peaceiris/actions-gh-pages@v2.5.0
```

After the process complete, the ebooks will be available for download in these following URLs. Please adjust it to follow your Github profile and repo name.

```
https://github.com/novalagung/softwareengineering/raw/ebooks/softwareengineerin
https://github.com/novalagung/softwareengineering/raw/ebooks/softwareengineerin
https://github.com/novalagung/softwareengineering/raw/ebooks/softwareengineerin
```

FYI! Since the ebook files are accessible through Github direct link, this means the visibility of the repo needs to be public (not private). If you want the repo to be in private but keep the files accessible, then do push the files into `gh-pages` branch.