


# GitHub Actions Workflow for Hugo Website Deployment

 [claude.ai/chat/69620292-858b-4f57-8b8c-eef86b1ed792](https://claude.ai/chat/69620292-858b-4f57-8b8c-eef86b1ed792)



BY ANTHROPIC

AG

```
##### ## Build and Deploy  
Hugo Website to GitHub Pages ## Author: Jérémie Lumbroso <lumbroso@seas.upenn.edu>  
## Date: 2024-02-24 #####
```

name: 🏗️ Build and Deploy GitHub Pages

on: push: branches: - main # Set a branch to deploy

jobs: deploy: runs-on: ubuntu-22.04 steps: - name: 🔀 Check Out Source Repository uses: actions/checkout@v3.5.1 with: submodules: true # Fetch Hugo themes (true OR recursive)  
fetch-depth: 0 # Fetch all history for .GitInfo and .Lastmod

- name: ⚙️ Initialize Hugo Environment uses: peaceiris/actions-hugo@v2.6.0 with: hugo-version: "0.123.4" extended: true
- name: 🏗️ Compile Hugo Static Files run: hugo -D --gc --minify

- name: 🚀 Publish to GitHub Pages uses: peaceiris/actions-gh-pages@v3.9.3 with:  
github\_token: \${{ secrets.GITHUB\_TOKEN }} publish\_branch: gh-pages user\_name:  
"github-actions[bot]" user\_email: "github-actions[bot]@users.noreply.github.com" ##  
NOTE: uncomment below if using a custom domain ## cname: mydomain.com

This is .yaml file for github actions. Can you explain it to me line by line

I'll explain this GitHub Actions workflow file line by line:

```
yaml
```

```
name: 🏗️ Build and Deploy GitHub Pages
```

This is the name of your workflow that appears in the GitHub Actions tab. It includes an emoji for visual distinction.

```
yaml
```

```
on:  
  push:  
    branches:  
      - main
```


This section defines when the workflow triggers - in this case, it runs whenever code is pushed to the `main` branch.

```
yaml
```

```
jobs:  
  deploy:  
    runs-on: ubuntu-22.04
```

Defines a job named "deploy" that runs on Ubuntu 22.04. This is the virtual environment where your actions will execute.


yaml

```
steps:
  - name:  Check Out Source Repository
    uses: actions/checkout@v3.5.1
    with:
      submodules: true
      fetch-depth: 0
```

First step: checks out your repository code using `actions/checkout@v3.5.1`

- `submodules: true` fetches any Git submodules (often used for Hugo themes)
- `fetch-depth: 0` fetches the complete Git history, needed for Hugo's `.GitInfo` and `.Lastmod` features

yaml

```
- name:  Initialize Hugo Environment
  uses: peaceiris/actions-hugo@v2.6.0
  with:
    hugo-version: "0.123.4"
    extended: true
```

Sets up Hugo in the environment:

- Uses the `peaceiris/actions-hugo` action at version 2.6.0
- Installs Hugo version 0.123.4
- `extended: true` installs the extended version of Hugo which includes SCSS/SASS support

yaml

```
- name: 🏗️ Compile Hugo Static Files
  run: hugo -D --gc --minify
```

Builds your Hugo site:

- `-D` includes draft content
- `--gc` runs garbage collection during build
- `--minify` minimizes the output files for better performance

yaml

```
- name: 🚀 Publish to GitHub Pages
  uses: peaceiris/actions-gh-pages@v3.9.3
  with:
    github_token: ${ secrets.GITHUB_TOKEN }
    publish_branch: gh-pages
    user_name: "github-actions[bot]"
    user_email: "github-actions[bot]@users.noreply.github.com"
```

Final step: deploys to GitHub Pages:

- Uses `peaceiris/actions-gh-pages` action to handle deployment
- `github_token` uses the automatic GitHub token for authentication
- `publish_branch: gh-pages` specifies the branch where the site will be published
- Sets the commit author to the GitHub Actions bot

yaml

```
## NOTE: uncomment below if using a custom domain
## cname: mydomain.com
```

A commented section that can be uncommented if you want to use a custom domain for your GitHub Pages site.

Would you like me to elaborate on any particular part of this workflow?



Claude does not have the ability to run the code it generates yet.

AG

can you explain what the with: key word means?

The `with:` keyword in GitHub Actions is used to specify input parameters or configuration options for an action. It's similar to passing arguments to a function in programming.

Let's look at a few examples from your workflow:

yaml

```
- uses: actions/checkout@v3.5.1
  with:
    submodules: true
    fetch-depth: 0
```

Here, `with:` is providing configuration options to the `checkout` action. It's telling the action to include submodules and fetch all git history.

yaml

```
- uses: peaceiris/actions-hugo@v2.6.0
  with:
    hugo-version: "0.123.4"
    extended: true
```

In this case, `with:` is specifying which version of Hugo to install and that you want the extended version.

Each action (specified by `uses:`) has its own set of input parameters that you can configure using `with:`. These parameters are defined by the action's creator and can be found in the action's documentation. Think of it like this:

- `uses:` says "I want to use this action"
- `with:` says "and I want to configure it this way"

If you were to look at this in a more programming-like pseudocode, it would be something like:

python

```
checkout_action(
    submodules=True,
    fetch_depth=0
)
```

The available parameters under `with:` are specific to each action, so you'll need to check the action's documentation to know what options are available.

①

Claude does not have the ability to run the code it generates yet.

AG

Moreover you the job with deploy what does that do exactly. Does it deploy the website for you?

The `deploy` job in your workflow doesn't actually deploy the website by itself - it's just a name for a collection of steps that together handle the building and deployment process. Let me break it down:

yaml

```
jobs:
  deploy:
    runs-on: ubuntu-22.04
    steps:
      # ... all the steps
```

The actual deployment happens in the final step using the `peaceiris/actions-gh-pages` action:

yaml

```
- name: 🚀 Publish to GitHub Pages
  uses: peaceiris/actions-gh-pages@v3.9.3
  with:
    github_token: ${ secrets.GITHUB_TOKEN }
    publish_branch: gh-pages
```

This step does several things:

1. Takes your built Hugo site (the static files generated in the previous step)
2. Creates or updates a branch called `gh-pages`
3. Commits the built files to this branch
4. Pushes the changes

Then, GitHub Pages itself (a separate GitHub service) detects these changes in the `gh-pages` branch and serves them as a website. The actual hosting is handled by GitHub Pages, not by your workflow.

You could have named the job anything else like `build` or `publish` - the name doesn't affect its functionality. For example, this would work exactly the same:

yaml

```
jobs:
  make_website: # Different name, same functionality
    runs-on: ubuntu-22.04
    steps:
      # ... same steps
```

Think of the job name like a label on a container - changing the label doesn't change what's inside. The actual deployment work is done by the specific actions and commands within the steps, not by the job name itself.



Claude does not have the ability to run the code it generates yet.



Claude can make mistakes. Please double-check responses.