

```
#####  
## 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: "0.144.1"  
          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
```

Explain this code line by line

GitHub Actions Workflow Explanation

This is a GitHub Actions workflow file that automates the process of building and deploying a Hugo website to GitHub Pages. Let me explain each part in detail:

Header Section

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

This is just a comment block providing metadata about the workflow file, including its purpose, author, and creation date.

Workflow Name and Trigger

```
yaml  
  
name: 🏗️ Build and Deploy GitHub Pages  
on:  
  push:  
    branches:  
      - main # Set a branch to deploy
```

- `name`: Names the workflow " 🏗️ Build and Deploy GitHub Pages" (with an emoji)
- `on`: Defines when the workflow runs
- `push`: Triggers the workflow when code is pushed
- `branches`: `[main]`: Specifically triggers only when code is pushed to the "main" branch

Jobs Definition

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

- `jobs`: Contains all the jobs that this workflow will run
- `deploy`: Defines a job named "deploy"
- `runs-on: ubuntu-22.04`: Specifies that this job will run on an Ubuntu 22.04 virtual machine

Job Steps

```
yaml  
  
      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
```

- `steps`: Lists the sequence of tasks to perform in the job
- First step: Named " 📄 Check Out Source Repository"
- `uses: actions/checkout@v3.5.1`: Uses GitHub's checkout action (version 3.5.1) to clone the repository
- `submodules: true`: Also fetches any Git submodules (commonly used for Hugo themes)
- `fetch-depth: 0`: Fetches the entire 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: "0.144.1"  
            extended: true
```

- Second step: Named " 🐙 Initialize Hugo Environment"
- `uses: peaceiris/actions-hugo@v2.6.0`: Uses a predefined action to set up Hugo
- `hugo-version: "0.144.1"`: Specifies Hugo version 0.144.1
- `extended: true`: Uses the extended version of Hugo, which includes additional features like Sass/SCSS processing

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

- Third step: Named " 📦 Compile Hugo Static Files"
- `run: hugo -D --gc --minify`: Runs the Hugo command to build the site
 - `-D`: Includes draft posts in the build
 - `--gc`: Runs garbage collection during the build (cleans up unused cache files)
 - `--minify`: Minifies the output files (HTML, CSS, JS, etc.) to reduce size

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

- Fourth step: Named " 🚀 Publish to GitHub Pages"
- `uses: peaceiris/actions-gh-pages@v3.9.3`: Uses an action to deploy to GitHub Pages
- `github_token: ${{ secrets.GITHUB_TOKEN }}`: Uses an automatically provided token for GitHub API authentication
- `publish_branch: gh-pages`: Specifies that the built site should be pushed to the "gh-pages" branch
- `user_name` and `user_email`: Sets the Git commit author to the GitHub Actions bot
- The commented line `## cname: mydomain.com` can be uncommented if you want to use a custom domain for your GitHub Pages site

In summary, this workflow automates the process of:

- Checking out your repository code
- Setting up the Hugo environment
- Building your Hugo website
- Publishing the built website to GitHub Pages through the gh-pages branch

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

What causes the error "actions/checkout@v3.5.1" not allowed to be used in repository?

Error: "actions/checkout@v3.5.1 not allowed to be used in repository"

This error typically occurs due to GitHub's organization security settings that restrict which GitHub Actions can be used in repositories. There are several potential causes:

Main Causes

- Organization Restrictions:** Your organization has implemented a policy that restricts which actions can be used in workflows. Only specific actions that have been approved by your organization administrators are allowed.
- Enterprise Security Settings:** If you're part of a GitHub Enterprise account, administrators can configure allowed actions at the enterprise level.
- Repository Security Settings:** Individual repository settings might be configured to only allow specific actions.

Solutions

- Use an Allowed Version:**
 - Check which versions of `actions/checkout` are approved in your organization
 - Update your workflow to use an allowed version (could be older or newer)
- Request Permission:**
 - Contact your organization's GitHub administrators
 - Ask them to add `actions/checkout@v3.5.1` to the allowed actions list
- Use Organization-Defined Actions:**
 - Some organizations require using internal forks of popular actions
 - Check if there's an organization-approved version you should use instead
- Check Security Settings:**
 - If you have admin permissions, check:
 - Repository settings > Actions > General > Workflow permissions
 - Organization settings > Actions > General > Allowed actions
- Use the Default Version:**
 - Instead of pinning to a specific version (`v3.5.1`), try using `actions/checkout@v3` or even just `actions/checkout@main`
 - This might use an allowed version automatically

If you're the organization admin, you can modify these restrictions in the organization's GitHub settings under "Actions" > "General" > "Allowed actions".

