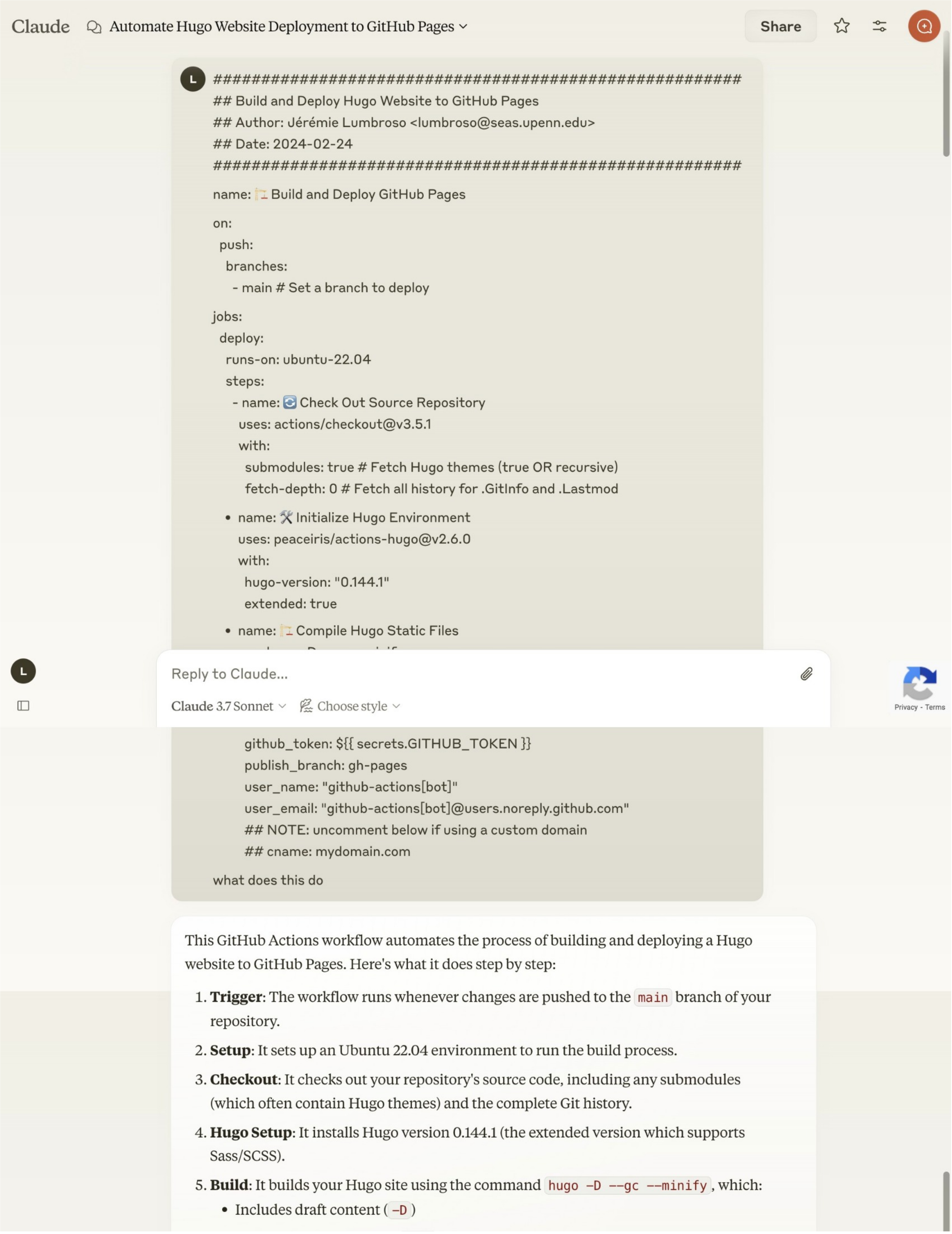


```
####################################################
## Build and Deploy Hugo Website to GitHub Pages
## Author: Jérémie Lumbroso <lumbroso@seas.upenn.edu>
## Date: 2024-02-24
####################################################

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

     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
```

what does this do

This GitHub Actions workflow automates the process of building and deploying a Hugo website to GitHub Pages. Here's what it does step by step:

1. **Trigger**: The workflow runs whenever changes are pushed to the `main` branch of your repository.
2. **Setup**: It sets up an Ubuntu 22.04 environment to run the build process.
3. **Checkout**: It checks out your repository's source code, including any submodules (which often contain Hugo themes) and the complete Git history.
4. **Hugo Setup**: It installs Hugo version 0.144.1 (the extended version which supports Sass/SCSS).
5. **Build**: It builds your Hugo site using the command `hugo -D --gc --minify`, which:
   - Includes draft content (`-D`)

    - Runs garbage collection (`--gc`) to clean up unused resources
    - Minifies the output files (`--minify`) to reduce their size
6. **Deploy**: It publishes the built site to GitHub Pages by:
    - Creating or updating a branch called `gh-pages`
    - Using a GitHub token for authentication
    - Setting the Git user information for the commit
    - (There's a commented out option to specify a custom domain)

Once completed, your Hugo website will be available at `https://[username].github.io/[repository-name]/` (or at your custom domain if you uncomment and configure the CNAME option).

so is this specifically for a hugo website?

Yes, this workflow is specifically designed for a Hugo website. Hugo is a popular open-source static site generator written in Go.

The workflow is tailored to Hugo's requirements in several ways:

1. It installs Hugo specifically (version 0.144.1) using the `peaceiris/actions-hugo` action
2. It uses the extended version of Hugo which supports Sass/SCSS processing
3. It runs the `hugo` command with specific flags to build the site
4. It checks out submodules, which is common for Hugo sites as themes are often included as Git submodules

If you were using a different static site generator (like Jekyll, Gatsby, Next.js, etc.), you would need a different workflow with different build commands and potentially different environment setup steps.

This workflow assumes you have a Hugo project already set up with the necessary configuration files and content in your repository.

how does it deploy github pages?

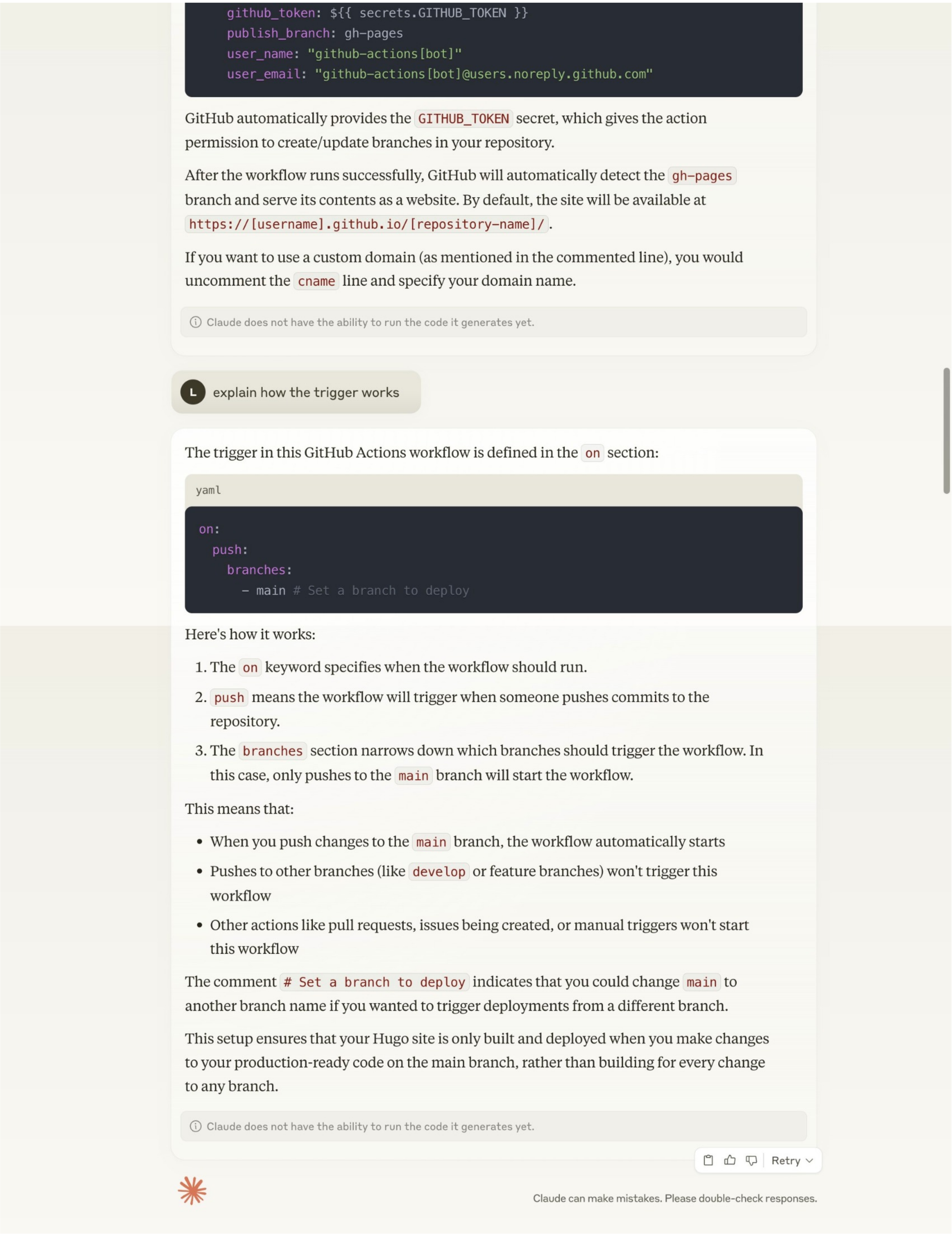
The deployment to GitHub Pages happens in the final step of the workflow, where it uses the `peaceiris/actions-gh-pages@v3.9.3` action. This action handles the entire GitHub Pages deployment process by:

1. Taking the built static files (which Hugo generates into the `public` directory by default)
2. Creating a new commit containing these files
3. Pushing this commit to a special branch called `gh-pages`
4. Setting up this branch to be served via GitHub Pages

Specifically, these lines handle the deployment:

yaml

```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"
```

GitHub automatically provides the `GITHUB_TOKEN` secret, which gives the action permission to create/update branches in your repository.

After the workflow runs successfully, GitHub will automatically detect the `gh-pages` branch and serve its contents as a website. By default, the site will be available at `https://[username].github.io/[repository-name]/`.

If you want to use a custom domain (as mentioned in the commented line), you would uncomment the `cname` line and specify your domain name.

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

L explain how the trigger works

The trigger in this GitHub Actions workflow is defined in the `on` section:

yaml

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

Here's how it works:

1. The `on` keyword specifies when the workflow should run.
2. `push` means the workflow will trigger when someone pushes commits to the repository.
3. The `branches` section narrows down which branches should trigger the workflow. In this case, only pushes to the `main` branch will start the workflow.

This means that:

- When you push changes to the `main` branch, the workflow automatically starts
- Pushes to other branches (like `develop` or feature branches) won't trigger this workflow
- Other actions like pull requests, issues being created, or manual triggers won't start this workflow

The comment `# Set a branch to deploy` indicates that you could change `main` to another branch name if you wanted to trigger deployments from a different branch.

This setup ensures that your Hugo site is only built and deployed when you make changes to your production-ready code on the main branch, rather than building for every change to any branch.

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

Claude can make mistakes. Please double-check responses.