How I used Claude Code to analyze a PR to find specific files

This blog post documents the steps I followed to use Claude Code to find specific files within a GitHub Pull Request (PR). The task was to review multiple PRs for our official OCP (OpenShift Container Platform) documentation across multiple enterprise distributions. Each PR had around 80 files, making a manual review for my product, OADP (application backup and restore), very time-consuming. These steps and prompts may be helpful to anyone trying to do something similar, though you may need to modify the prompts for your specific use case.

Step 1: Set Up Claude Code

First, set up Claude Code by following the instructions in its official documentation. To test your setup, open a terminal and type:

```
claude tell me a joke
```

If Claude responds with a cliché developer joke, your setup is correct, and you're ready to proceed.

Step 2: Connect to Your GitHub Account

Next, connect Claude to your GitHub account using the following prompt:

```
claude connect to GitHub
```

- Claude will run the `gh --version` command to check if the GitHub CLI is installed.
- If you don't have the CLI installed, Claude will provide a link to install it.
- Claude will then ask for permission to run `gh auth login`. Say **yes**.
- A one-time authentication code will be provided. Use the GitHub link and this code to authenticate Claude as a device on your account. You will also need your GitHub FreeOTP token to complete the process.

After authenticating, return to your terminal and confirm your connection with this prompt:

```
claude am i connected to GH
```

Claude should respond in the affirmative.

Step 3: Analyze the PR

Once connected, you can start analyzing a PR. I began by providing context to Claude:

```
claude I need you to analyze a PR. Should I give you the PR link?
```

Next, I provided the specific PR link and my request:

```
claude here is the PR link: https://github.com/openshift/openshift-docs/pull/****. Here is
what I want you to do: Give me a list of files of the PR contained in this directory
/backup_and_restore/application_backup_and_restore
```

- Claude will run the **gh pr view** command and ask for your permission to proceed.
- In my case, Claude didn't find any files in that specific directory and provided a one-line summary of the PR's contents.

To ensure Claude searched thoroughly, I used a more granular prompt:

```
claude analyze the PR again and give me a list of files containing any backup and restore
content.
```

- This time, Claude ran two **gh pr view** commands to use a **JQuery** and **grep** for "backup and restore" keywords.
- It successfully returned a file that contained relevant content. I verified that the file did not belong to my product, and I was satisfied with the result.

Step 4: Analyze Multiple PRs

To analyze multiple PRs, I used a concise prompt for the remaining 7 PRs:

```
claude do the same for https://github.com/openshift/openshift-docs/pull/***.
```

On some PRs, Claude failed to run the **gh pr view diff** command but seamlessly switched to using the **gh api** command to complete the task. This adaptability saved me a significant amount of manual effort.