

## GitHub CLI Workflow Documentation

Author: Guillermo Padilla Keymole

Course: 420-640-AB - Communication and Job Preparation for an IT Environment

Repository: Job Applications (JAC)

Link: <https://github.com/GuillermoPKeymole/Job-Applications-JAC->

### Objective:

This file documents the use of Git and GitHub CLI to manage, organize, and upload job application submissions to a private GitHub repository using PowerShell on Windows.

### Environment:

- Operating System: Windows 11
- Terminal: PowerShell
- Version Control: Git (CLI)
- Remote Host: GitHub (Private repository)

### Initial Git Setup:

Before performing any Git operations, global user information was configured to associate commits with my GitHub identity.

```
git config --global user.name "Guillermo Padilla Keymole"  
git config --global user.email "youremail@example.com"
```

```
PS C:\Users\Guill> git config --global user.name "Guillermo Padilla Keymole"  
PS C:\Users\Guill> git config --global user.email "guillermopkeymole@gmail.com"
```

These commands ensure that all future commits made from this system include the correct author name and email address linked to my GitHub account.

### Step-by-step GitHub CLI Workflow:

#### 1. Cloning the Repository from GitHub

This step pulls a remote GitHub repository into the local machine for editing and file management.

#### Command used:

```
git clone https://github.com/GuillermoPKeymole/Job-Applications-JAC-.git
```

```
PS C:\Users\Guill> git clone https://github.com/GuillermoPKeymole/Job-Applications-JAC-.git  
Cloning into 'Job-Applications-JAC-'...  
remote: Enumerating objects: 6, done.  
remote: Counting objects: 100% (6/6), done.  
remote: Compressing objects: 100% (4/4), done.  
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)  
Receiving objects: 100% (6/6), done.
```

Then change into the new local repo folder:

```
cd "Job-Applications-JAC-"
```

```
PS C:\Users\Guill> cd "Job-Applications-JAC-"
```

## 2. Creating a Submissions Directory

To keep submissions organized, a folder named 'submissions' was created inside the repo:

```
mkdir submissions
```

```
PS C:\Users\Guill\Job-Applications-JAC-> mkdir submissions

Directory: C:\Users\Guill\Job-Applications-JAC-

Mode                LastWriteTime         Length Name
----                -
d-----          2025-06-06 12:01 PM             submissions
```

## 3. Copying the Job Application PDF into the Repo

The job application PDF was located **in** OneDrive, inside the Documents folder. The filename contained parentheses, so backticks were used to escape them.

Command used:

```
cp "C:\Users\Guill\OneDrive\Documents\Job & CV JAC\`(FirstWeek`) Job Submission Applications GPK.pdf" submissions/Week1-Job-Applications.pdf
```

```
PS C:\Users\Guill\Job-Applications-JAC-> cp "C:\Users\Guill\OneDrive\Documents\Job & CV JAC\`(FirstWeek`) Job Submission Applications GPK.pdf" submissions/Week1-Job-Applications.pdf
```

This also renamed the file **for** simplicity **in** the repository.

## 4. Adding the New File to Git

The `git add .` command stages all changes (new files, updated files) to be committed.

```
git add .
```

```
PS C:\Users\Guill\Job-Applications-JAC-> git add .
```

## 5. Committing the Changes

This command saves the staged changes to the local repository with a message.

```
git commit -m "Add Week 1 job application PDF"
```

```
PS C:\Users\Guill\Job-Applications-JAC-> git commit -m "Add Week 1 job application PDF"
[main 4196199] Add Week 1 job application PDF
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 submissions/Week1-Job-Applications.pdf
```

## 6. Pushing to GitHub

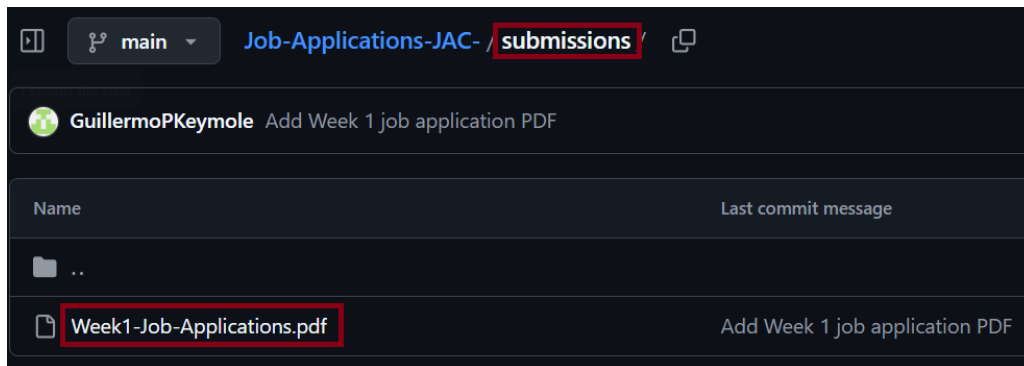
This command sends the local commits to the remote GitHub repository.

```
git push origin main
```

```
PS C:\Users\Guill\Job-Applications-JAC-> git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 20 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 578.05 KiB | 21.41 MiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/GuillermoPKeymole/Job-Applications-JAC-.git
d9b55f7..4196199  main -> main
```

**Result:**

The file Week1-Job-Applications.pdf was successfully added to the 'submissions' folder in the private GitHub repository.

**Reflections:**

Through this process, I gained experience in:

- Using Git and PowerShell for version control and automation
- Cloning, staging, committing, and pushing code and documents to GitHub
- Handling file paths and special characters in PowerShell
- Structuring a professional repository with private access control
- Maintaining clean naming and folder organization for academic or professional evaluation

This documentation reflects my ability to manage version-controlled projects using command-line tools and demonstrates a foundational understanding of GitHub workflows in real-world scenarios.

## GitHub CLI Collaborator Additional Documentation

Author: Guillermo Padilla Keymole  
Repository: Job Applications (JAC)  
Collaborator Added: pargol  
Date: June 6, 2025

### Purpose:

This document outlines the **process** used to install GitHub CLI, authenticate with **it**, and successfully add a collaborator to a private GitHub repository **using** GitHub's command-line interface and REST API.

### Tools Used:

- GitHub CLI (gh) version 2.74.0
- PowerShell on Windows 11
- GitHub.com account with admin access to the target repository

### Step 0: Install GitHub CLI (if not already installed)

Go to: <https://cli.github.com>

Download the installer **for** Windows and follow the **setup** instructions.  
After installation, close and reopen PowerShell.



### Step 1: Confirm GitHub CLI is Installed

Check **if** `gh` is available:

`gh --version`

```
PS C:\Users\Guill> gh --version
gh version 2.74.0 (2025-05-29)
https://github.com/cli/cli/releases/tag/v2.74.0
```

Expected output:

gh version 2.74.0 (2025-05-29)

### Step 2: Authenticate with GitHub

**Start** the authentication **process**:

`gh auth login`

```
PS C:\Users\Guill> gh auth login
? Where do you use GitHub? GitHub.com
? What is your preferred protocol for Git operations on this host? HTTPS
? Authenticate Git with your GitHub credentials? Yes
? How would you like to authenticate GitHub CLI? Login with a web browser

! First copy your one-time code: 0C8A-87AB
Press Enter to open https://github.com/login/device in your browser...
✓ Authentication complete.
- gh config set -h github.com git_protocol https
✓ Configured git protocol
✓ Logged in as GuillermoPKeymole
```

**Follow the interactive prompts:**

- Choose: GitHub.com
- Protocol: HTTPS
- Authenticate Git with GitHub credentials: Yes
- Authentication method: Login with a web browser
- Visit the link shown and enter the one-time code provided
- Once successful, you'll see confirmation and your GitHub username

**Result:**

Logged in as: GuillermoPKeymole

**Step 3: Add a Collaborator using GitHub API**

Run the following command to add a collaborator:

```
gh api -X PUT -H "Accept: application/vnd.github+json" /repos/GuillermoPKeymole/Job-Applications-JAC-/collaborators/pargol -f permission=push
```

```
PS C:\Users\Guill> gh api -X PUT -H "Accept: application/vnd.github+json" /repos/GuillermoPKeymole/Job-Applications-JAC-/collaborators/pargol -f permission=push
```

**Command breakdown:**

- gh api: executes a REST API call
- -X PUT: HTTP method used to update or create resource
- -H: sets the header to accept GitHub's recommended API format
- /repos/.../collaborators/pargol: specifies the user and repo
- -f permission=push: gives write access (can view, comment, and commit)

**Step 4: Confirmation**

The command returns a JSON response showing repository details, confirming that:

- The repository is private
- The request was processed
- The collaborator was added

```
"full_name": "GuillermoPKeymole/Job-Applications-JAC-",
"private": true,
"invitee": {
  "login": "Pargol",
"html_url": "https://github.com/Pargol",
"permissions": "write",
"created_at": "2025-06-06T18:44:41Z",
"url": "https://api.github.com/user/repository_invitations/283866916",
"html_url": "https://github.com/GuillermoPKeymole/Job-Applications-JAC-/invitations"
```

**Note:**

The collaborator (pargol) will receive an invitation and must accept it to gain access to the repository.

**Outcome:**

This successfully demonstrates GitHub CLI usage for authentication and collaborator management in a private repository context, using PowerShell and GitHub's API.