

Steps with explanation to push files into github using cli commands:

Pushing files to GitHub using the command-line interface (CLI) involves several steps, especially if you're setting up Git for the first time or starting with a new repository. Here's a comprehensive step-by-step guide to help you set up your environment, commit your changes, and push them to GitHub.

Step 1: Install Git

Before you begin, make sure Git is installed on your system. You can download Git from git-scm.com and follow the installation instructions specific to your operating system.

Step 2: Set Up Your Git Configuration

When using Git for the first time, you need to set up your user name and email address. This information is important because every Git commit uses this data, and it appears in your commits on GitHub.

- **Set Your Username:**

```
git config --global user.name "Your Name"
```

- **Set Your Email Address:**

```
git config --global user.email your.email@example.com
```

These commands only need to be run once per machine (using the `--global` flag sets it for all repositories on your system). You can also set configuration options on a per-repository basis by omitting the `--global` flag.

Step 3: Create or Clone a Repository

- **Create a New Repository:** If you're starting a new project, create a new directory, navigate into it, and type:

```
git init
```

- **Clone an Existing Repository:** If the repository already exists on GitHub and you want to contribute to it, clone it:

```
git clone https://github.com/username/repository.git
```

Replace <https://github.com/username/repository.git> with the URL of the GitHub repository.

Step 4: Add a Remote Repository

If you initialized a new repository or need to link an existing local repository to GitHub:

- **Check Existing Remotes:**

```
git remote -v
```

- **Add Remote Repository:**

```
git remote add origin https://github.com/username/repository.git
```

Replace <https://github.com/username/repository.git> with the URL of your GitHub repository. The origin is a conventional name used for your main remote repository.

Step 5: Make Changes and Commit

After you've made changes to your project:

- **Add Changes to the Staging Area:**

`git add`

This command stages all changed files. You can also stage files individually using `git add filename`.

- **Commit the Changes:**

`git commit -m "A meaningful commit message describing what you have done"`

Step 6: Push Your Changes to GitHub

- **Push to GitHub:**

`git push origin main`

This command pushes your commits to the main branch of your remote repository named origin. Replace main with another branch name if you're working on a different branch.

Additional Commands and Considerations

- **Check Status:**

`git status`

Use this often to see the state of your working directory and staging area.

- **Pull Latest Changes:**

`git pull origin main`

Before pushing, especially if collaborating with others, pull the latest changes from the remote to avoid conflicts.

- **Branching:**

`git branch new-branch`

`git checkout new-branch`

Use branches to manage different features or experiments.

This guide covers the fundamental CLI commands needed to manage and push your projects to GitHub. Remember to secure your GitHub account with a strong password and, optionally, two-factor authentication (2FA) for better security.