**Uploading Terraform Project to GitHub**

This document provides a step-by-step guide that I developed for uploading a Terraform project to GitHub, resolving common Git issues along the way.

**Prerequisites**

* **Git installed** on the local machine
* **GitHub account** with a repository created
* **Terraform project** initialized and ready to be uploaded

**Step 1: Navigate to the Project Directory**

Open a terminal (PowerShell, Command Prompt, or Git Bash) and navigate to the Terraform project folder:

cd path\to\your\terraform-project

**Step 2: Initialize Git Repository**

If the project is not already a Git repository, initialize it:

git init

This creates a hidden .git folder, marking the directory as a Git repository.

**Step 3: Add Remote Repository**

Add the GitHub repository as the remote origin:

git remote add origin https://github.com/YOUR\_GITHUB\_USERNAME/YOUR\_REPOSITORY.git

To verify the remote repository:

git remote -v

**Step 4: Add Files and Commit**

Stage all the files in the repository:

git add .

Commit the files with a message:

git commit -m "Initial commit - Upload Terraform project"

**Step 5: Check Branch and Pull Latest Changes**

Ensure you're on the correct branch:

git branch

If the branch is not main, switch to it:

git checkout -b main

Fetch the latest changes from GitHub and rebase local changes:

git pull origin main --rebase

If there are merge conflicts, resolve them, then run:

git add .

git commit -m "Resolved merge conflicts"

**Step 6: Push to GitHub**

Finally, push local changes to GitHub:

git push -u origin main

If the push is successful, the Terraform project will be uploaded to GitHub.

**Step 7: Verify on GitHub**

1. Go to the GitHub repository.
2. Refresh the page and check if the Terraform files appear.

**Troubleshooting**

* **"fatal: not a git repository"** → Run git init in the project directory.
* **"error: failed to push some refs to..."** → Run git pull origin main --rebase before pushing.
* **"Updates were rejected because the remote contains work..."** → Pull the latest changes from GitHub first.

**Conclusion**

By following these steps, I have successfully uploaded my Terraform project to GitHub. Now, I can collaborate, version control, and share my project easily.