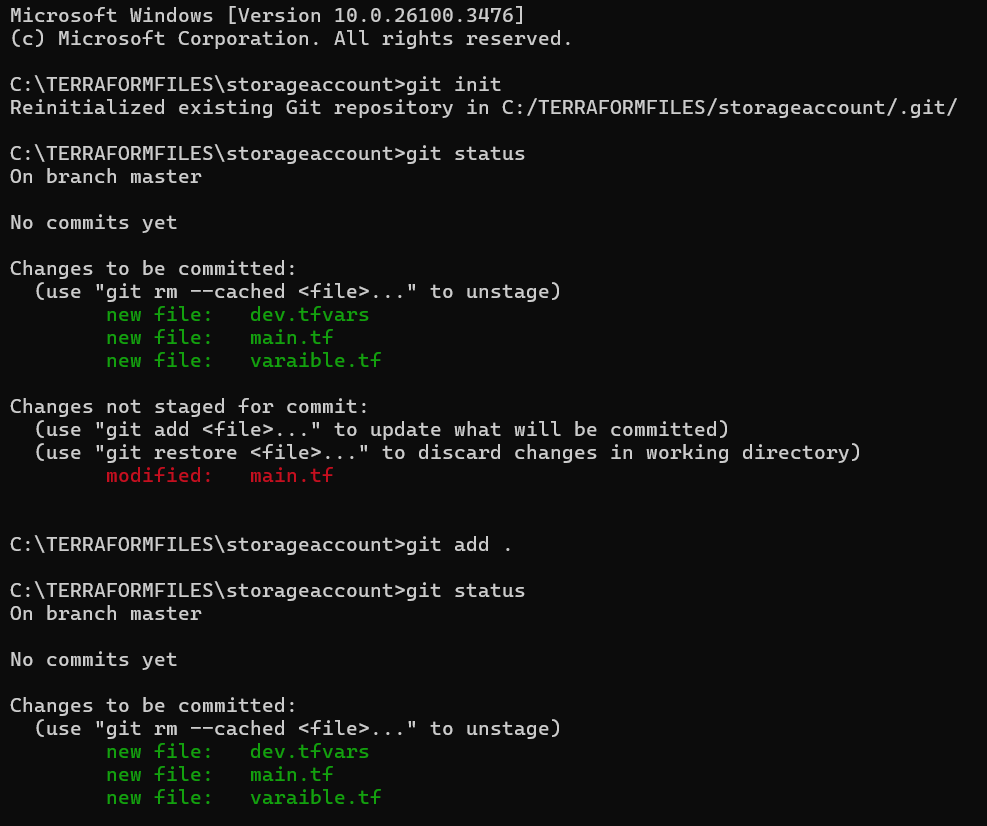
TASK-4

GIT COMMANDS

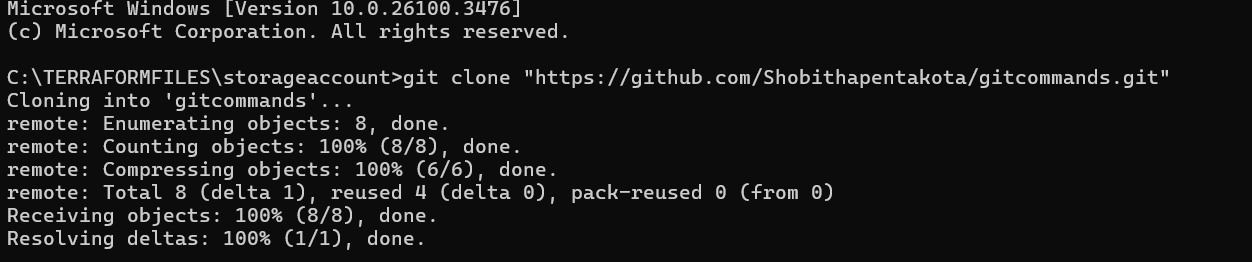
Git is a powerful version control system, and mastering its commands is essential for smooth collaboration and efficient development. Here's a breakdown of commonly used Git commands, categorized for clarity:

**1. Basic Setup and Initialization:**

* **git init:** Initializes a new Git repository in the current directory.

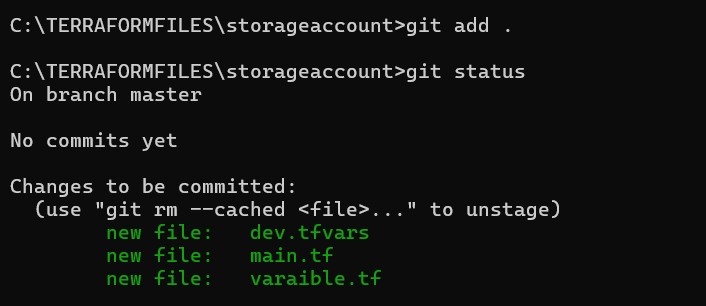


* **git clone <repository\_url>:** Creates a copy of a remote Git repository on your local machine.

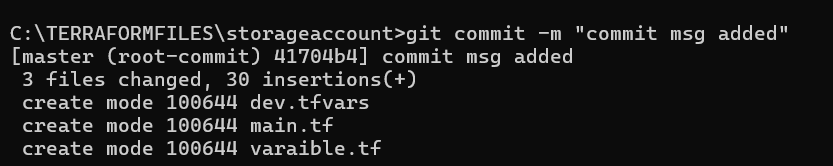


**2. Working with Changes:**

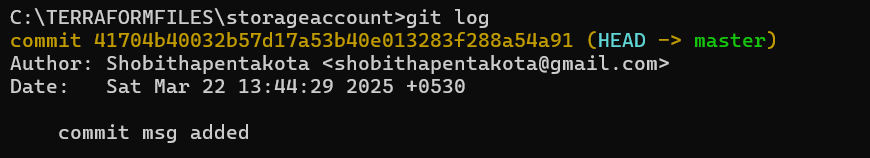
* **git status:** Shows the current status of your working directory and staging area.
* **git add <file>:** Adds a specific file to the staging area.
* **git add .:** Adds all changes in the current directory to the staging area.



* **git commit -m "Commit message":** Creates a new commit with a descriptive message.
* **git commit -a -m "Commit message":** Adds all modified files and creates a commit.

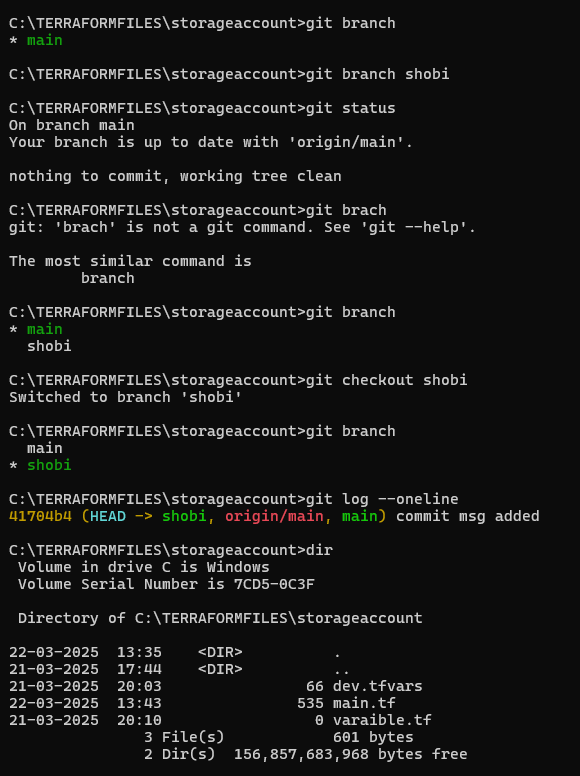


* **git rm <file>:** Removes a file from the working directory and staging area.
* **git mv <old\_file> <new\_file>:** Renames or moves a file.
* **git log** : Give the commit id



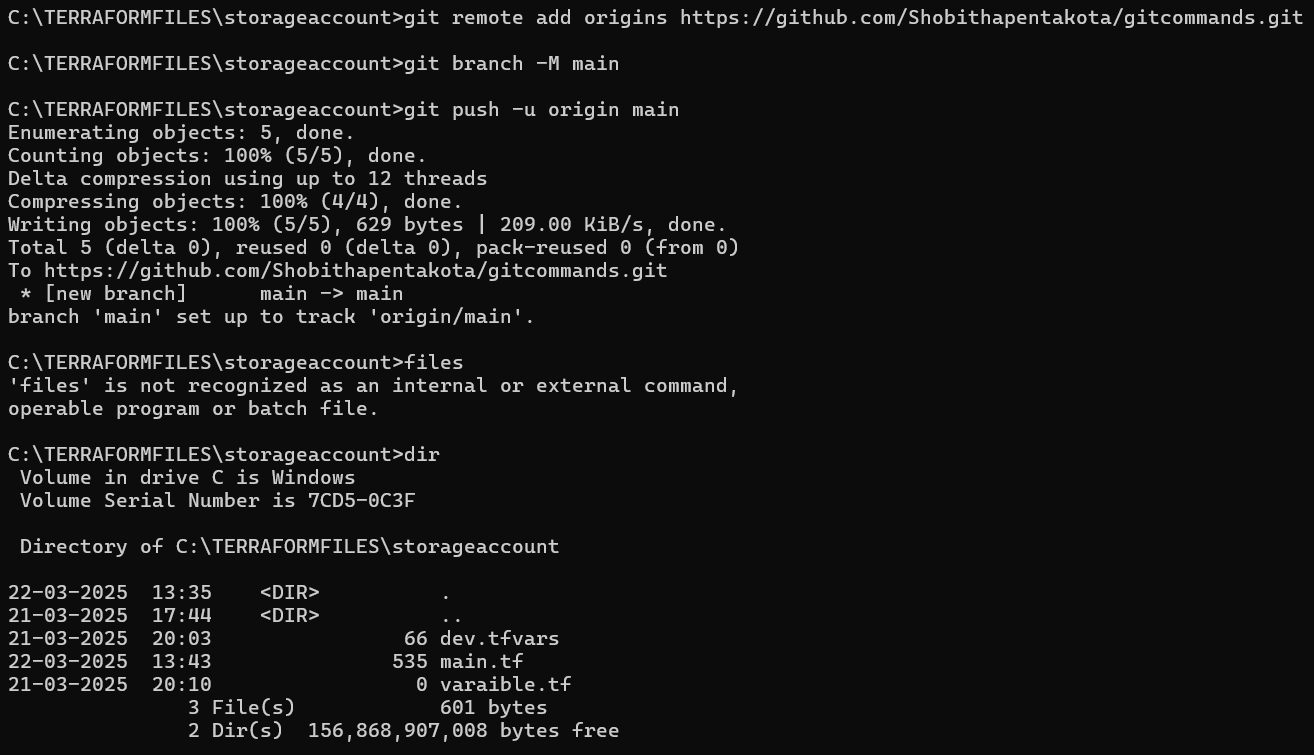
**3. Branching and Merging:**

* **git branch:** Lists all local branches.
* **git branch <branch\_name>:** Creates a new branch.
* **git checkout <branch\_name>:** Switches to the specified branch.
* **git checkout -b <branch\_name>:** Creates and switches to a new branch.
* **git merge <branch\_name>:** Merges the specified branch into the current branch.
* **git branch -d <branch\_name>:** Deletes a local branch (if merged).
* **git branch -D <branch\_name>:** Force deletes a local branch.
* **git stash:** Temporarily saves changes that you don't want to commit immediately.
* **git stash pop:** Applies the stashed changes and removes them from the stash.
* **git stash list:** Lists all stashed changes.



**4. Remote Repositories:**

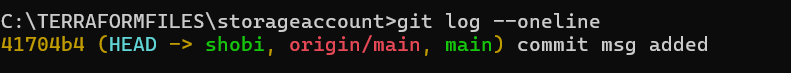
* **git remote add <remote\_name> <repository\_url>:** Adds a remote repository.
* **git remote -v:** Lists all remote repositories and their URLs.
* **git fetch <remote\_name>:** Downloads changes from the remote repository without merging them.
* **git pull <remote\_name> <branch\_name>:** Downloads changes and merges them into the current branch.



* **git push <remote\_name> <branch\_name>:** Uploads local commits to the remote repository.
* **git push -u <remote\_name> <branch\_name>:** sets the upstream branch, so future pulls and pushes can be done by just using git pull, and git push.
* **git remote remove <remote\_name>:** Removes a remote repository.

**5. History and Inspection:**

* **git log:** Shows the commit history.
* **git log --oneline:** Shows a concise commit history.
* **git show <commit\_hash>:** Shows the details of a specific commit.
* **git checkout <commit\_hash> <file>:** Checks out a specific version of a file.
* **git reset --hard <commit\_hash>:** Resets the current branch to a specific commit (use with caution).
* **git revert <commit\_hash>:** Creates a new commit that undoes the changes of a previous commit.
* **git log –**oneline:7 num of our commit id.



* **git tag <tag\_name>:** Creates a tag for a specific commit.
* **git tag:** lists the existing tags.
* **git push --tags:** Pushes tags to the remote.
* **Git pull :** To get code from the Git repo to local repo.

