GIT ASSIGNMENT SUBMISSION

Name: Vikram

Assignment: Git – Create Repository, Stage Files, Commit & Push

---

Problem Statement

You work for XYZ Corporation. As part of the DevOps workflow, you must demonstrate the ability to work with Git. Your task is to create files on a Linux server, initialise a Git repository, stage selected files, commit them, and push them to GitHub.

Tasks Performed

Task 1: Create a project folder and required files on Amazon Linux EC2

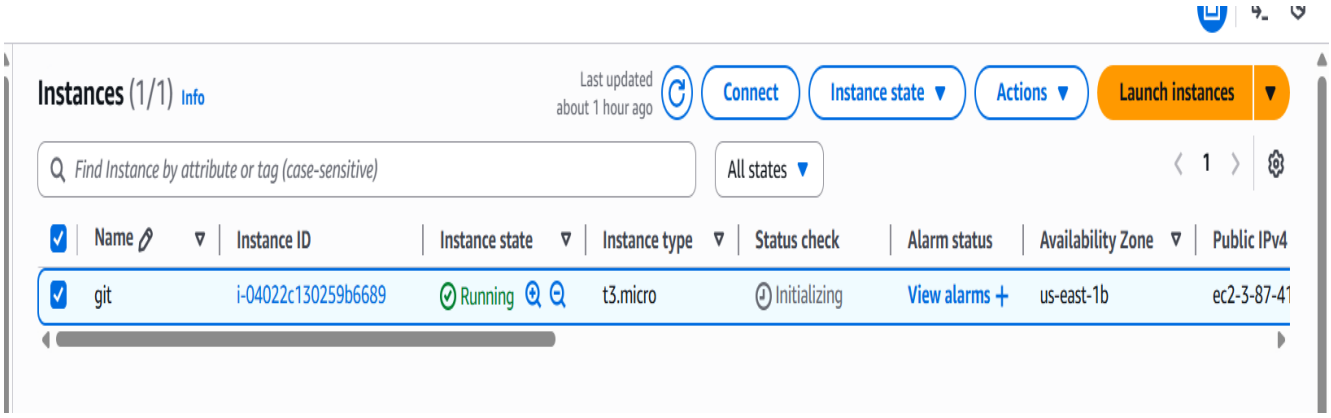
Steps taken:

1. Launched an Amazon Linux EC2 instance using EC2 Instance Connect.
2. Created a new project directory:

   mkdir myproject

   cd myproject
3. Created the required files:

   touch Code.txt Log.txt Output.txt



```
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-10-0-3-185 ~]$
```

```
[ec2-user@ip-10-0-3-185 ~]$ mkdir myproject
[ec2-user@ip-10-0-3-185 ~]$ cd myproject
[ec2-user@ip-10-0-3-185 myproject]$ touch Code.txt Log.txt Output.txt
[ec2-user@ip-10-0-3-185 myproject]$ ls
Code.txt  Log.txt  Output.txt
```

Task 2: Initialize Git repository

Steps taken:

1. Installed Git on the EC2 instance:

   sudo yum install git -y

2. Initialized a new Git repository inside the project folder:

   git init

```
[ec2-user@ip-10-0-3-185 ~]$ sudo yum install git -y
Last metadata expiration check: 0:00:35 ago on Tue Dec  2 04:50:53 2025.
Dependencies resolved.
================================================================================
 Package                 Architecture        Version                 Repository      Size
================================================================================
Installing:
 git                     x86_64              2.50.1-1.amzn2023.0.1   amazonlinux     53 k
Installing dependencies:
 git-core                x86_64              2.50.1-1.amzn2023.0.1   amazonlinux    4.9 M
 git-core-doc            noarch              2.50.1-1.amzn2023.0.1   amazonlinux    2.8 M
```

```
[ec2-user@ip-10-0-3-185 myproject]$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
hint:
hint: Disable this message with "git config set advice.defaultBranchName false"
Initialized empty Git repository in /home/ec2-user/myproject/.git/
```

Task 3: Stage only specific files

Steps taken:

1. Staged only Code.txt and Output.txt:

   git add Code.txt Output.txt

```
[ec2-user@ip-10-0-3-185 myproject]$ ls
Code.txt  Log.txt  Output.txt
[ec2-user@ip-10-0-3-185 myproject]$ git add Code.txt Output.txt
[ec2-user@ip-10-0-3-185 myproject]$ ls
Code.txt  Log.txt  Output.txt
```

Task 4: Commit the staged files

Steps taken:

1. Committed the staged files:

   git commit -m "Added Code and Output files"

```
[ec2-user@ip-10-0-3-185 myproject]$ git commit -m "Added Code and Output files"
[master (root-commit) fcfa2f8] Added Code and Output files
 Committer: EC2 Default User <ec2-user@ip-10-0-3-185.ec2.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 Code.txt
 create mode 100644 Output.txt
[ec2-user@ip-10-0-3-185 myproject]$
```

Task 5: Connect the repository to GitHub

Steps taken:

1. Created a GitHub repository.
2. Added the GitHub remote origin:

   git remote add origin https://github.com/<your-username>/<your-repo>.git
3. Renamed the local branch to main:

   git branch -M main

```
[ec2-user@ip-10-0-3-185 myproject]$ git remote add origin https://github.com/Vikky9387/git.git
git branch -M main
git push -u origin main
error: remote origin already exists.
Username for 'https://github.com': Vikky9387
Password for 'https://Vikky9387@github.com':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 250 bytes | 250.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'main' on GitHub by visiting:
remote:      https://github.com/Vikky9387/vikram/pull/new/main
remote:
To https://github.com/Vikky9387/vikram
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
[ec2-user@ip-10-0-3-185 myproject]$
```

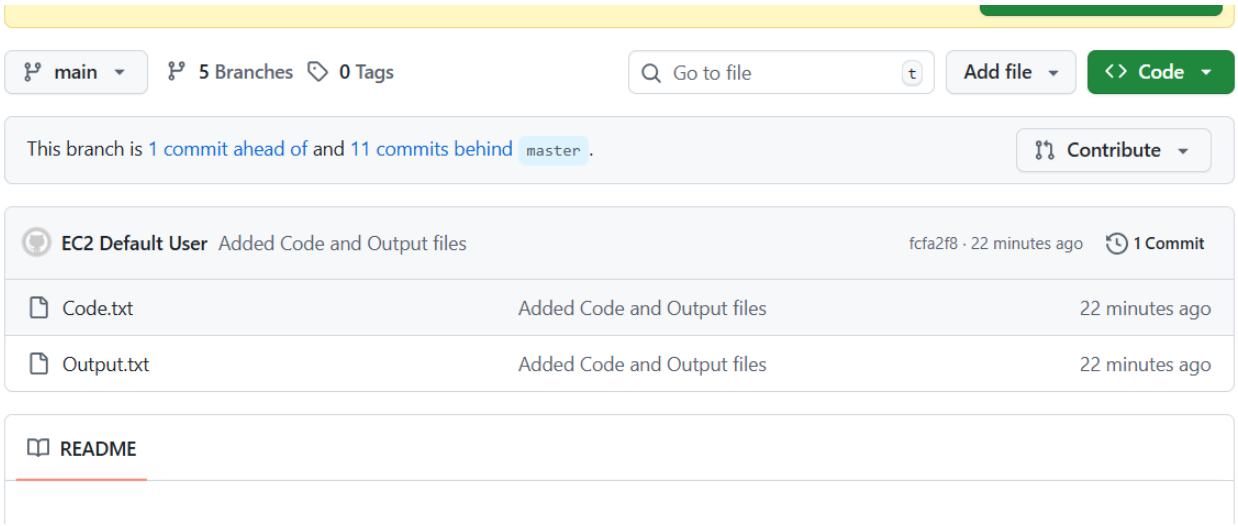
Task 6: Push the local commit to GitHub

Steps taken:

1. Generated a GitHub Personal Access Token (PAT) for authentication.
2. Pushed the commit to GitHub:

   git push -u origin main
3. Entered GitHub username and used PAT as password.
4. Verified that the files and commit were successfully uploaded to GitHub.

```
[ec2-user@ip-10-0-3-185 myproject]$ git remote add origin https://github.com/Vikky9387/git.git
git branch -M main
git push -u origin main
error: remote origin already exists.
Username for 'https://github.com': Vikky9387
Password for 'https://Vikky9387@github.com':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 250 bytes | 250.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'main' on GitHub by visiting:
remote:      https://github.com/Vikky9387/vikram/pull/new/main
remote:
To https://github.com/Vikky9387/vikram
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
[ec2-user@ip-10-0-3-185 myproject]$
```



Conclusion

Successfully created the required files on Amazon Linux EC2, initialized a Git repository, staged selected files, committed them, and pushed the commit to GitHub using secure authentication. This completes the Git workflow assignment as per the given requirements.