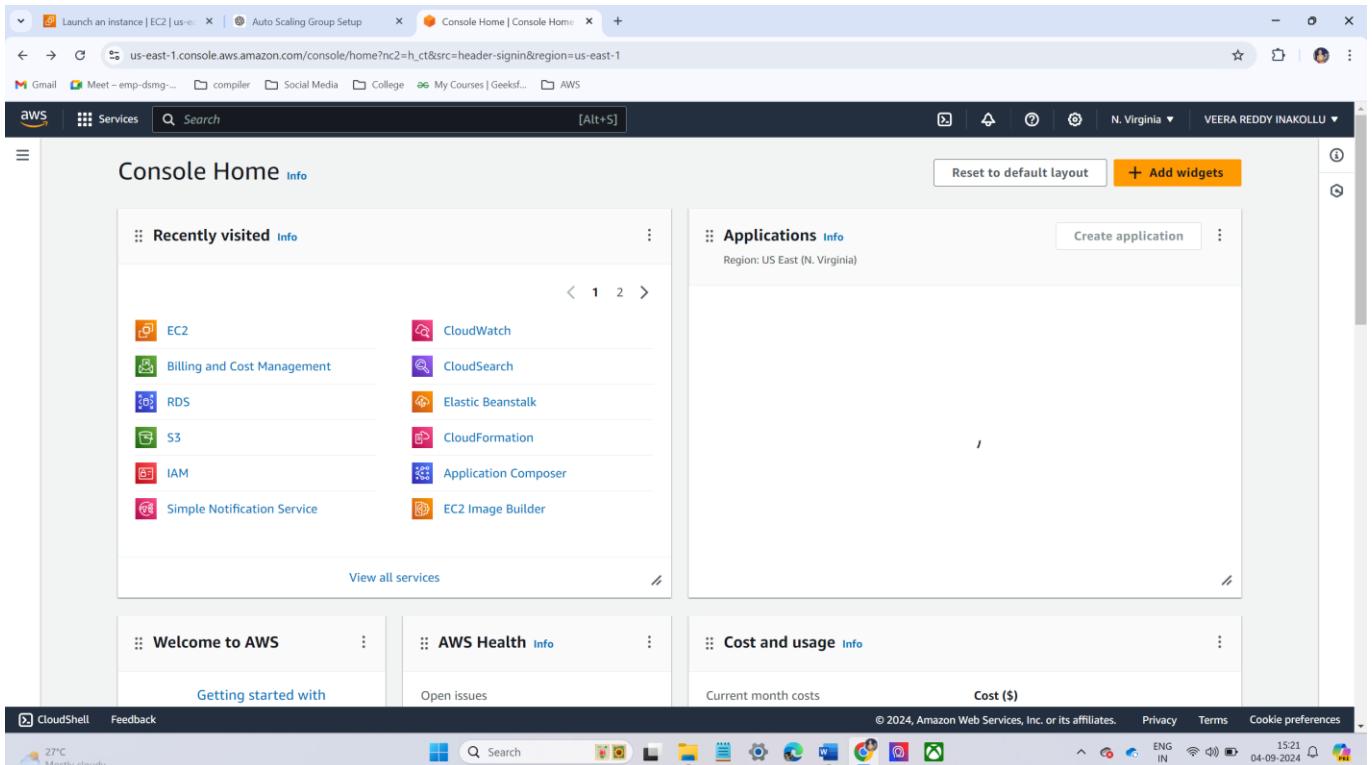


# GIT VERSION CONTROL SYSTEM BY USING AMAZON WEB SERVICES

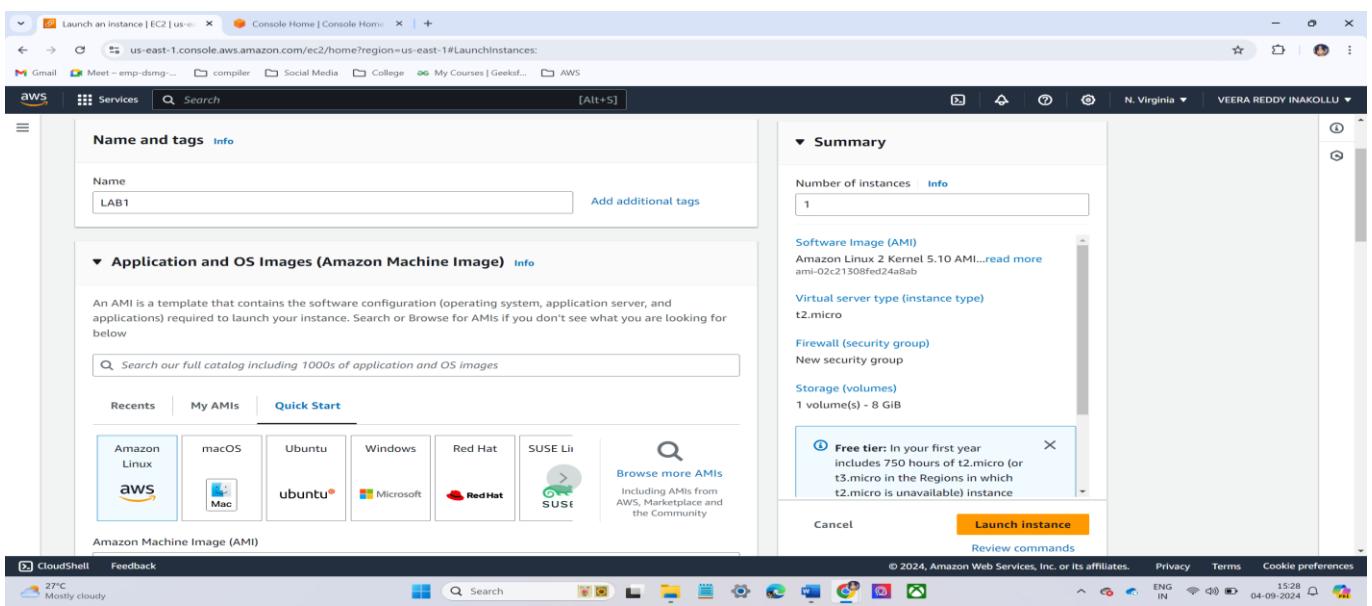
VEERA REDDY INAKOLLU

## LAB -1 CREATING EC2 INSTANCE

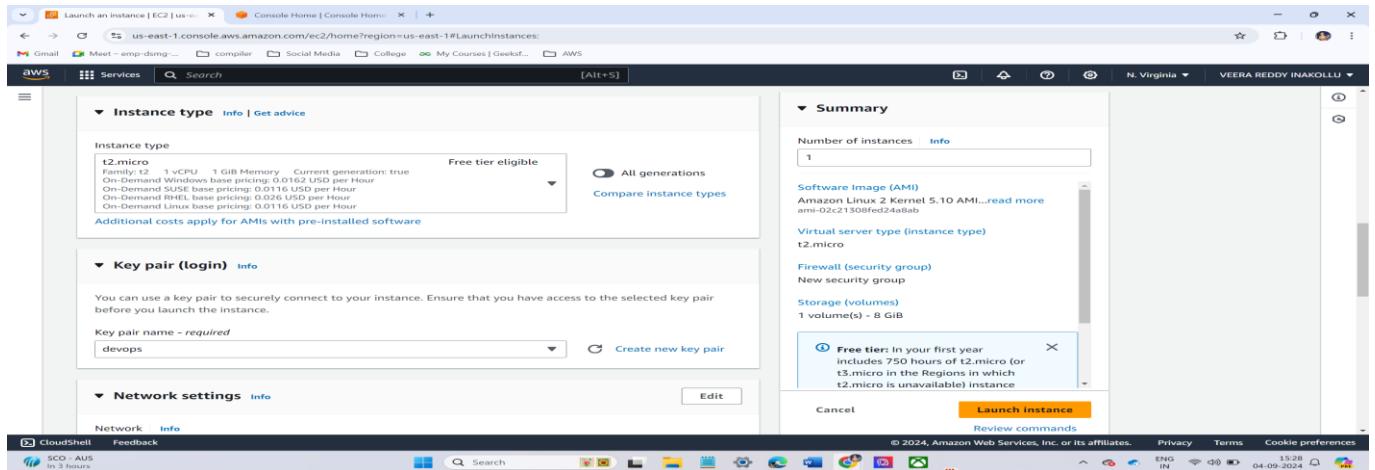
- Login into aws console.



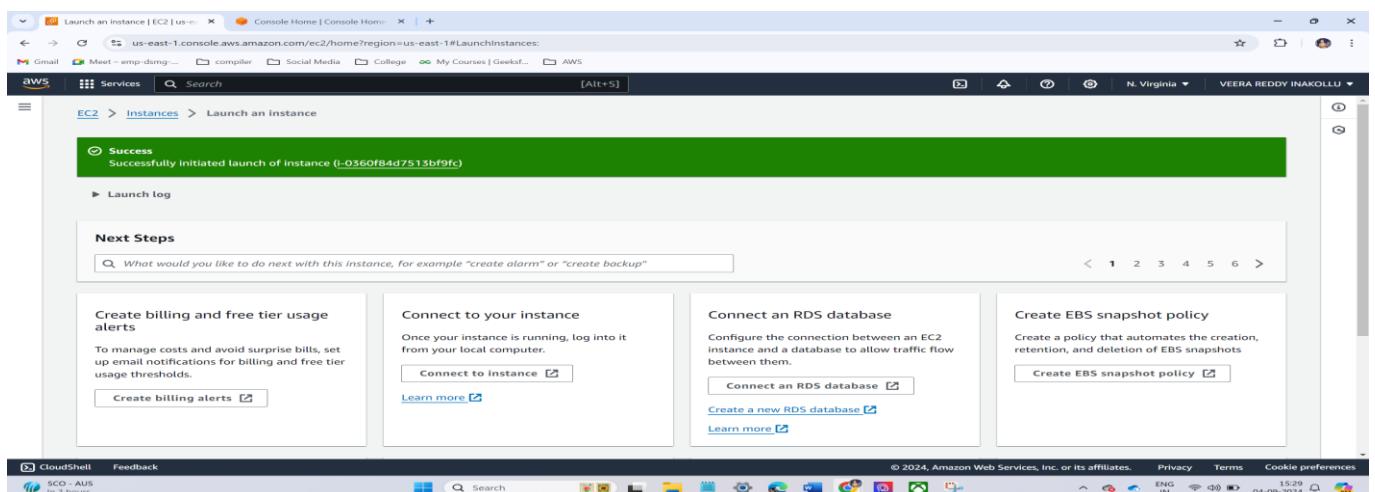
- Create EC2 server with amazon Linux.
- Go to **EC2** console →choose **Launch instance**.
- Configure instance details.
- Name: choose name
- Choose **Amazon Linux** operating system.



- Instance type = t2. Micro.
- Choose existing key pair.



- Instance launch successfully.



- Connect instance successfully

```

ec2-user@ip-172-31-86-241:~ + ~
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

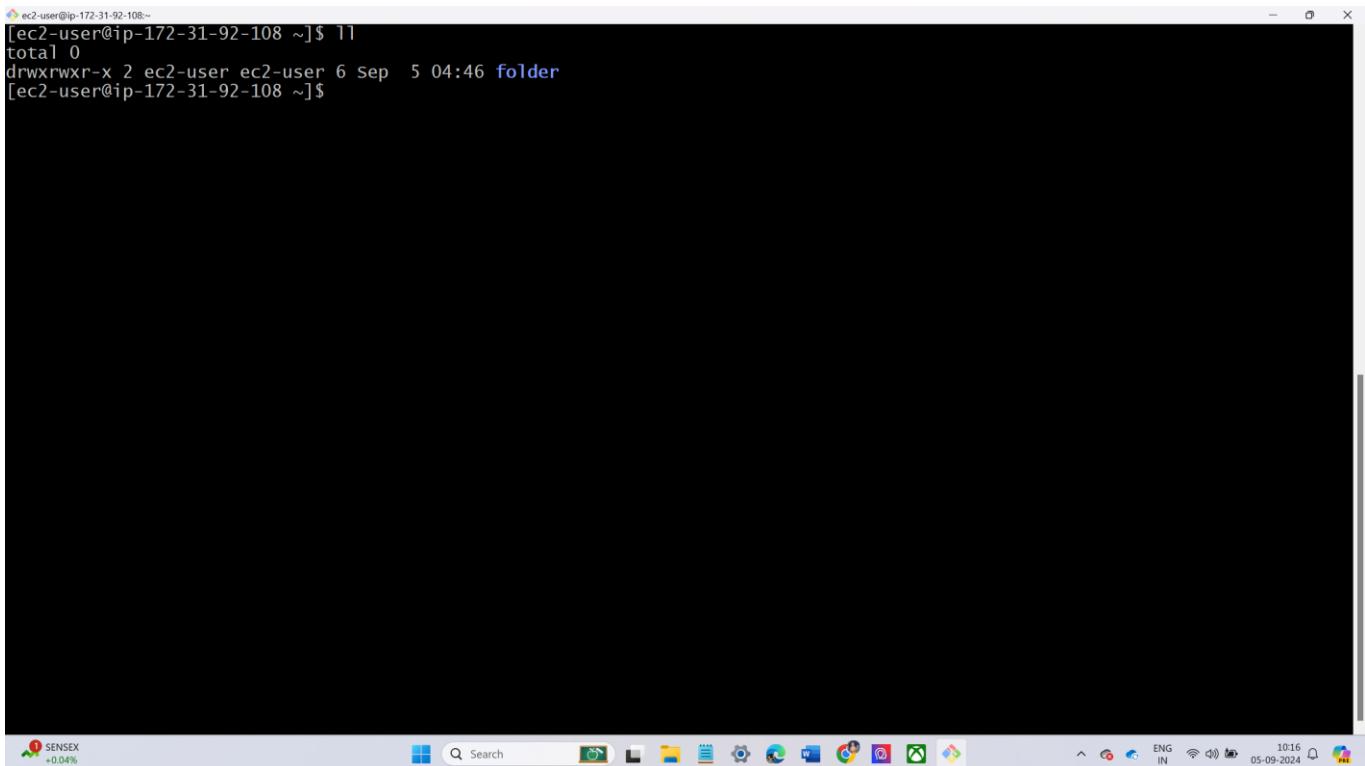
C:\Users\jawah>cd Socialprachar
C:\Users\jawah\Socialprachar>cd "veera reddy"
C:\Users\jawah\Socialprachar\veera reddy>cd keys

C:\Users\jawah\Socialprachar\veera reddy>ssh -i "devops.pem" ec2-user@ec2-54-89-183-194.compute-1.amazonaws.com
Last login: Wed Sep 4 10:15:21 2024 from 103.213.202.11
#
# ## Amazon Linux 2
## \### AL2 End of Life is 2025-06-30.
## /-->
## /--> A newer version of Amazon Linux is available!
## /--> / Amazon Linux 2023, GA and supported until 2028-03-15.
## /--> https://aws.amazon.com/linux/amazon-linux-2023/
[ec2-user@ip-172-31-86-241 ~]$ |

```

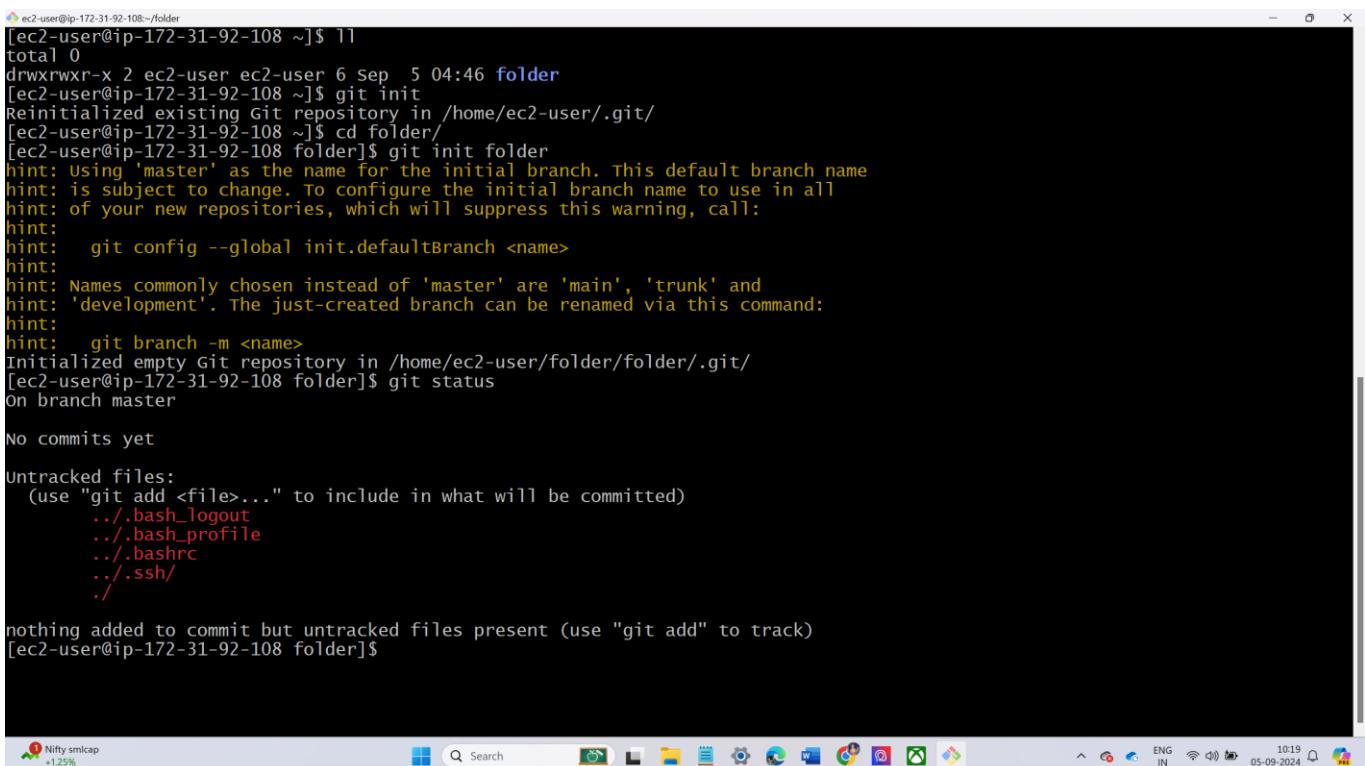
## LAB -2 – CREATE REPOSITORY IN LOCAL MACHINE.

- Create directory in local machine using touch command.



```
[ec2-user@ip-172-31-92-108 ~]$ ll
total 0
drwxrwxr-x 2 ec2-user ec2-user 6 Sep  5 04:46 folder
[ec2-user@ip-172-31-92-108 ~]$
```

- check the current status of Git repository:



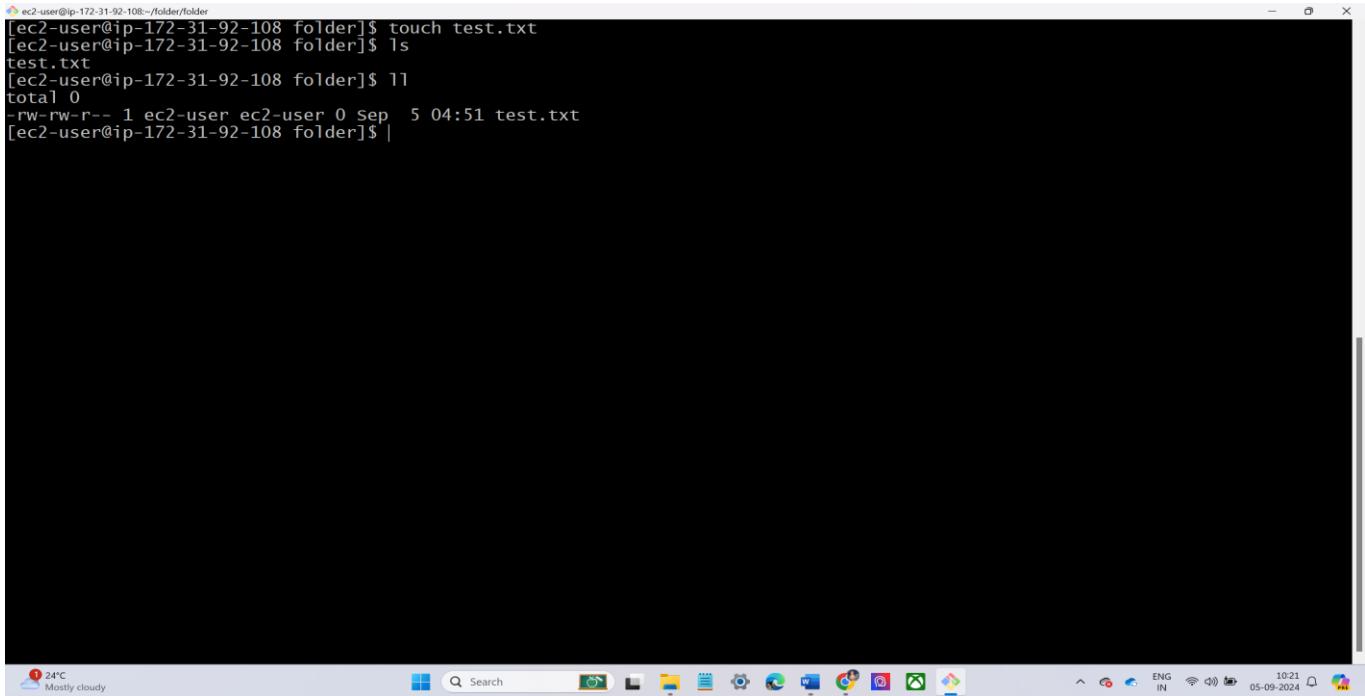
```
[ec2-user@ip-172-31-92-108 ~]$ ll
total 0
drwxrwxr-x 2 ec2-user ec2-user 6 Sep  5 04:46 folder
[ec2-user@ip-172-31-92-108 ~]$ git init
Reinitialized existing Git repository in /home/ec2-user/.git/
[ec2-user@ip-172-31-92-108 ~]$ cd folder/
[ec2-user@ip-172-31-92-108 folder]$ git init folder
hint: 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>
Initialized empty Git repository in /home/ec2-user/folder/.git/
[ec2-user@ip-172-31-92-108 folder]$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    ./._bash_logout
    ./._bash_profile
    ./._bashrc
    ./._ssh/
    ./

nothing added to commit but untracked files present (use "git add" to track)
[ec2-user@ip-172-31-92-108 folder]$
```

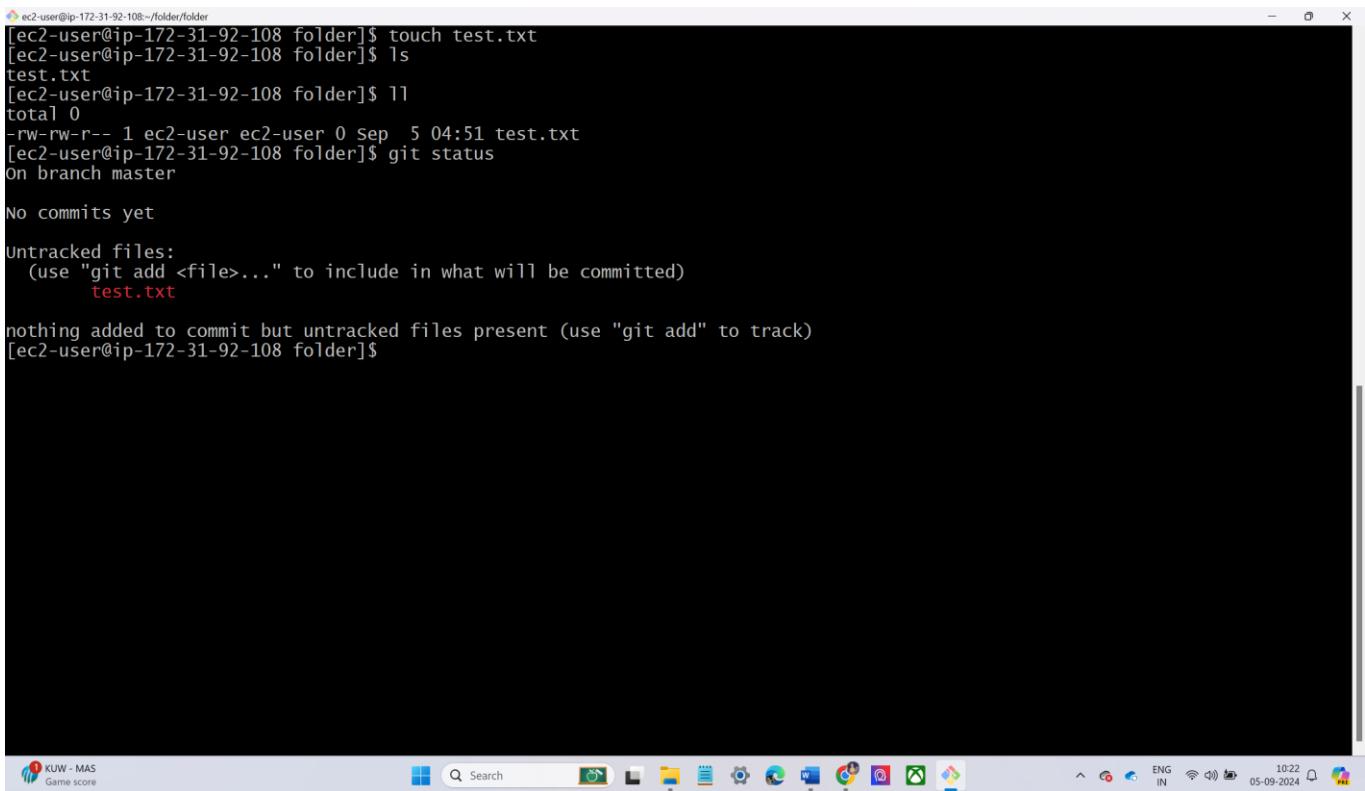
➤ Create some empty files using the touch command



```
ec2-user@ip-172-31-92-108:~/folder$ touch test.txt
[ec2-user@ip-172-31-92-108 folder]$ ls
test.txt
[ec2-user@ip-172-31-92-108 folder]$ ll
total 0
-rw-rw-r-- 1 ec2-user ec2-user 0 Sep  5 04:51 test.txt
[ec2-user@ip-172-31-92-108 folder]$ |
```

The screenshot shows a Windows desktop environment. A terminal window is open in the foreground, displaying the command `touch test.txt` followed by its output: a file named `test.txt` with permissions `rw-rw-r--` and size `0`. The desktop taskbar at the bottom shows various icons for applications like File Explorer, Edge, and Google Chrome. The system tray indicates the date as 05-09-2024 and the time as 10:21.

➤ Check the status again



```
ec2-user@ip-172-31-92-108:~/folder$ touch test.txt
[ec2-user@ip-172-31-92-108 folder]$ ls
test.txt
[ec2-user@ip-172-31-92-108 folder]$ ll
total 0
-rw-rw-r-- 1 ec2-user ec2-user 0 Sep  5 04:51 test.txt
[ec2-user@ip-172-31-92-108 folder]$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    test.txt

nothing added to commit but untracked files present (use "git add" to track)
[ec2-user@ip-172-31-92-108 folder]$ |
```

The screenshot shows a Windows desktop environment. A terminal window is open in the foreground, displaying the command `git status` followed by its output: it shows a single untracked file `test.txt` and prompts to use `git add` to track it. The desktop taskbar at the bottom shows various icons for applications like File Explorer, Edge, and Google Chrome. The system tray indicates the date as 05-09-2024 and the time as 10:22.

- To stage the newly created file so Git starts tracking it
- Using command git add test.txt
- Check status.

```

[ec2-user@ip-172-31-92-108~/folder/folder]
[ec2-user@ip-172-31-92-108 folder]$ touch test.txt
[ec2-user@ip-172-31-92-108 folder]$ ls
test.txt
[ec2-user@ip-172-31-92-108 folder]$ ll
total 0
-rw-rw-r-- 1 ec2-user ec2-user 0 sep  5 04:51 test.txt
[ec2-user@ip-172-31-92-108 folder]$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    test.txt

nothing added to commit but untracked files present (use "git add" to track)
[ec2-user@ip-172-31-92-108 folder]$ git add test.txt
[ec2-user@ip-172-31-92-108 folder]$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   test.txt

[ec2-user@ip-172-31-92-108 folder]$

```

The screenshot shows a Windows terminal window with a black background and white text. It displays the creation of a file named 'test.txt' in a directory, followed by the output of 'git status' which shows that the file is untracked. Then, the command 'git add test.txt' is run, and the status is checked again, showing that the file is now staged for commit. The window has a standard Windows title bar at the top and a taskbar with various icons at the bottom.

- Commit the staged changes with a message.

```

[ec2-user@ip-172-31-92-108~/folder/folder]
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    test.txt

nothing added to commit but untracked files present (use "git add" to track)
[ec2-user@ip-172-31-92-108 folder]$ git add test.txt
[ec2-user@ip-172-31-92-108 folder]$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   test.txt

[ec2-user@ip-172-31-92-108 folder]$ git commit -m "added test files"
[master (root-commit) 0a9a347] added test files
  Committer: EC2 Default User <ec2-user@ip-172-31-92-108.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

1 file changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 test.txt
[ec2-user@ip-172-31-92-108 folder]$

```

This screenshot continues from the previous one, showing the execution of 'git commit -m "added test files"'. The terminal outputs the commit message, the commit hash, and the configuration message. It also provides instructions for fixing the identity if needed. The terminal window is identical to the one above, with a black background and white text, and a taskbar at the bottom.

- Check status again after committing.

```
ec2-user@ip-172-31-92-108:~/folder/folder
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    test.txt

nothing added to commit but untracked files present (use "git add" to track)
[ec2-user@ip-172-31-92-108 folder]$ git add test.txt
[ec2-user@ip-172-31-92-108 folder]$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   test.txt

[ec2-user@ip-172-31-92-108 folder]$ git commit -m "added test files".
[master (root-commit) 0a9a347] added test files
  Committer: EC2 Default User <ec2-user@ip-172-31-92-108.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

1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 test.txt
[ec2-user@ip-172-31-92-108 folder]$ git status
On branch master
nothing to commit, working tree clean
[ec2-user@ip-172-31-92-108 folder]$
```

24°C Mostly cloudy

Search

ENG IN 10:27 05-09-2024

## LAB -3 – CREATING REPOSITORY IN REMOTE LOCATION – GITHUB

- Open GitHub and log in with credentials.
- On GitHub dashboard, click on the green "New" button located on the upper-right side.

**Create a new repository**

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Required fields are marked with an asterisk (\*).

**Owner \*** veerareddy1422    **Repository name \*** MINIPROJECT  
 MINIPROJECT is available.

Great repository names are short and memorable. Need inspiration? How about [turbo-meme](#)?

**Description (optional)**

**Visibility**

- Public** Anyone on the internet can see this repository. You choose who can commit.
- Private** You choose who can see and commit to this repository.

**Initialize this repository with:**

- Add a README file** This is where you can write a long description for your project. [Learn more about READMEs](#).

**Add .gitignore**

.gitignore template: [None](#)

Choose which files not to track from a list of templates. [Learn more about ignoring files](#).

SENSEX +0.01%

Search

ENG IN 10:32 05-09-2024

- Fill in the Repository Details
- Provide a repository name: Enter a unique name
- Choose visibility: Private
- Initialize with a README.md file.
- Create the Repository.

**Create a new repository**

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Required fields are marked with an asterisk (\*).

Owner \* Repository name \*

veerareddy1422 / MINIPROJECT

MINIPROJECT is available.

Great repository names are short and memorable. Need inspiration? How about [turbo-meme](#) ?

Description (optional)

Public Anyone on the internet can see this repository. You choose who can commit.

Private You choose who can see and commit to this repository.

Initialize this repository with:

Add a README file This is where you can write a long description for your project. [Learn more about READMEs](#).

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. [Learn more about ignoring files](#).

veerareddy1422 / MINIPROJECT

**Code** Issues Pull requests Actions Projects Security Insights Settings

**MINIPROJECT** Private

main 1 Branch 0 Tags

Go to file Add file Code

veerareddy1422 Initial commit 607fe1b · now 1 Commit

README.md Initial commit now

README

**About**

No description, website, or topics provided.

Readme Activity 0 stars 0 watching 0 forks

**Releases**

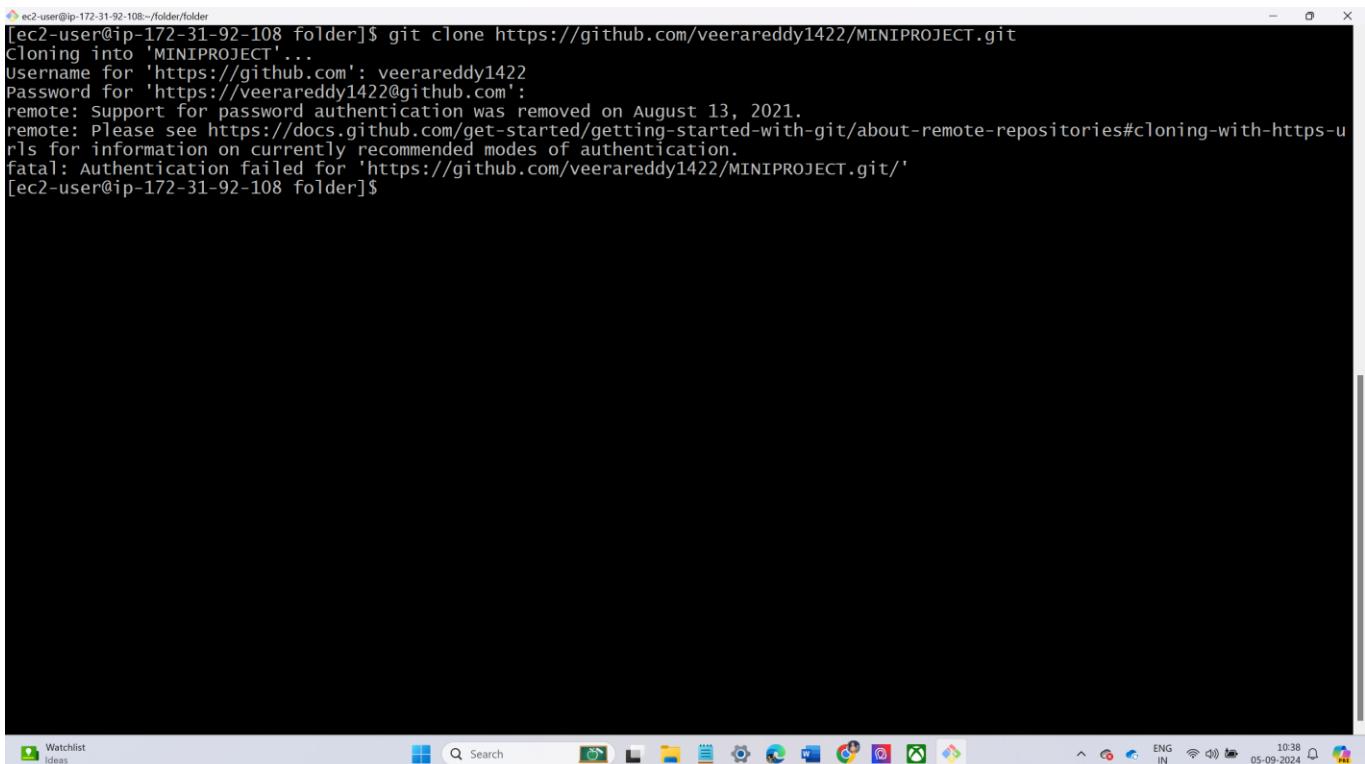
No releases published [Create a new release](#)

**Packages**

No packages published [Publish your first package](#)

## LAB -4 – WORKING WITH REMOTE REPOSITORY.

- Clone the Remote Repository using command **git clone**.
- After running the command, Git will ask you for your **GitHub username** and **password**.
- You'll receive an error message due to the deprecation of password-based authentication.



```
ec2-user@ip-172-31-92-108:~/folder/folder$ git clone https://github.com/veerareddy1422/MINIPROJECT.git
Cloning into 'MINIPROJECT'...
Username for 'https://github.com': veerareddy1422
Password for 'https://veerareddy1422.github.com':
remote: Support for password authentication was removed on August 13, 2021.
remote: Please see https://docs.github.com/get-started/getting-started-with-git/about-remote-repositories#cloning-with-https-urls for information on currently recommended modes of authentication.
fatal: Authentication failed for 'https://github.com/veerareddy1422/MINIPROJECT.git/'
[ec2-user@ip-172-31-92-108:~/folder/folder]$
```

- To overcome the above error, we can use your personal access token.
- Create a Personal Access Token.
  - Go to GitHub Settings -> Click on your profile picture at the top-right corner and select **Settings**.
  - Scroll down and select **Developer settings** from the left sidebar.
  - Create a New Token → Select **Personal access tokens** → **Tokens (classic)** → **Generate new token**.

New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Note

My Token for Git Operations

What's this token for?

Expiration \*

7 days The token will expire on Thu, Sep 12 2024

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

Scope	Description
<input checked="" type="checkbox"/> repo	Full control of private repositories
<input type="checkbox"/> repo:status	Access commit status
<input type="checkbox"/> repo_deployment	Access deployment status
<input type="checkbox"/> public_repo	Access public repositories
<input type="checkbox"/> repo:invite	Access repository invitations
<input type="checkbox"/> security_events	Read and write security events
<input type="checkbox"/> workflow	Update GitHub Action workflows
<input type="checkbox"/> write:packages	Upload packages to GitHub Package Registry
<input type="checkbox"/> read:packages	Download packages from GitHub Package Registry

- After generating token again clone the repository.
- Again, ask username and password. In password give token address.

```
[ec2-user@ip-172-31-92-108 ~]$ git clone https://github.com/veerareddy1422/MINIPROJECT.git
Cloning into 'MINIPROJECT'...
Username for 'https://github.com': veerareddy1422
Password for 'https://veerareddy1422@github.com':
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
[ec2-user@ip-172-31-92-108 ~]$
```

- Make changes to local repository.
- Go to cloned repository.
- Create some sample files using touch command

```

ec2-user@ip-172-31-92-108:~/folder/folder/MINIPROJECT$ git clone https://github.com/veerareddy1422/MINIPROJECT.git
Cloning into 'MINIPROJECT'...
Username for 'https://github.com': veerareddy1422
Password for 'https://veerareddy1422@github.com':
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
[ec2-user@ip-172-31-92-108 folder]$ cd MINIPROJECT/
[ec2-user@ip-172-31-92-108 MINIPROJECT]$ touch file1.txt file2.txt
[ec2-user@ip-172-31-92-108 MINIPROJECT]$ ll
total 4
-rw-rw-r-- 1 ec2-user ec2-user 0 Sep  5 05:22 file1.txt
-rw-rw-r-- 1 ec2-user ec2-user 0 sep  5 05:22 file2.txt
-rw-rw-r-- 1 ec2-user ec2-user 13 Sep  5 05:17 README.md
[ec2-user@ip-172-31-92-108 MINIPROJECT]$ |

```

The screenshot shows a Windows desktop environment with a terminal window open. The terminal window title is 'Terminal' and it contains the command-line session above. The desktop taskbar at the bottom shows various icons for applications like File Explorer, Edge, and Google Chrome. The system tray indicates the date as 05-09-2024 and the time as 10:52. The weather widget on the taskbar shows '24°C Mostly cloudy'.

- Save changes
- Stage the newly created files for the commit using **git add** command.
- Commit the changes with a message using **git commit** command.

```

Receiving objects: 100% (3/3), done.
[ec2-user@ip-172-31-92-108 folder]$ cd MINIPROJECT/
[ec2-user@ip-172-31-92-108 MINIPROJECT]$ touch file1.txt file2.txt
[ec2-user@ip-172-31-92-108 MINIPROJECT]$ ll
total 4
-rw-rw-r-- 1 ec2-user ec2-user 0 Sep  5 05:22 file1.txt
-rw-rw-r-- 1 ec2-user ec2-user 0 Sep  5 05:22 file2.txt
-rw-rw-r-- 1 ec2-user ec2-user 13 Sep  5 05:17 README.md
[ec2-user@ip-172-31-92-108 MINIPROJECT]$ git add .
[ec2-user@ip-172-31-92-108 MINIPROJECT]$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   file1.txt
    new file:   file2.txt

[ec2-user@ip-172-31-92-108 MINIPROJECT]$ git commit -m "Added sample files file1 and file2"
[main 85d5e4d] Added sample files file1 and file2
Committer: EC2 Default User <ec2-user@ip-172-31-92-108.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 file1.txt
create mode 100644 file2.txt
[ec2-user@ip-172-31-92-108 MINIPROJECT]$ |

```

This screenshot continues the terminal session from the previous one. It shows the user adding the staged files to the commit, committing them with a message, and then viewing the commit history. The terminal window title is 'Terminal'. The desktop taskbar and system tray are visible at the bottom, showing the same information as the first screenshot.

- Push Changes to the Remote Repository
- Run the **git push** command to push the committed changes to the remote repository.

```
[ec2-user@ip-172-31-92-108:~/folder/folder/MINIPROJECT]$ git push
Username for 'https://github.com': veerareddy1422
Password for 'https://veerareddy1422@github.com':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 315 bytes | 315.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/veerareddy1422/MINIPROJECT.git
  607fe1b..85d5e4d main -> main
[ec2-user@ip-172-31-92-108 MINIPROJECT]$
```

- Verify the Changes on GitHub.
- Go to GitHub repository in the browser and refresh the page. Now see the new files (file1 and file2) that were pushed from local machine.

The screenshot shows a GitHub repository page for 'veerareddy1422/MINIPROJECT'. The repository is private. It contains a single commit by 'EC2 Default User' adding 'file1' and 'file2'. The repository has 0 stars and 0 forks. There are sections for Releases, Packages, and Activity.

## LAB – 5 – PUSHING A LOCALLY CREATED REPOSITORY TO GITHUB

- Create a Local Repository and Initialize Git
- Create a Folder for Local Repo using command **mkdir**.

```
ec2-user@ip-172-31-92-108 ~]$ mkdir lab5
[ec2-user@ip-172-31-92-108 ~]$ ll
total 0
drwxrwxr-x 3 ec2-user ec2-user 36 Sep  5 04:50 folder
drwxrwxr-x 2 ec2-user ec2-user  6 Sep  5 05:44 lab5
[ec2-user@ip-172-31-92-108 ~]$
```

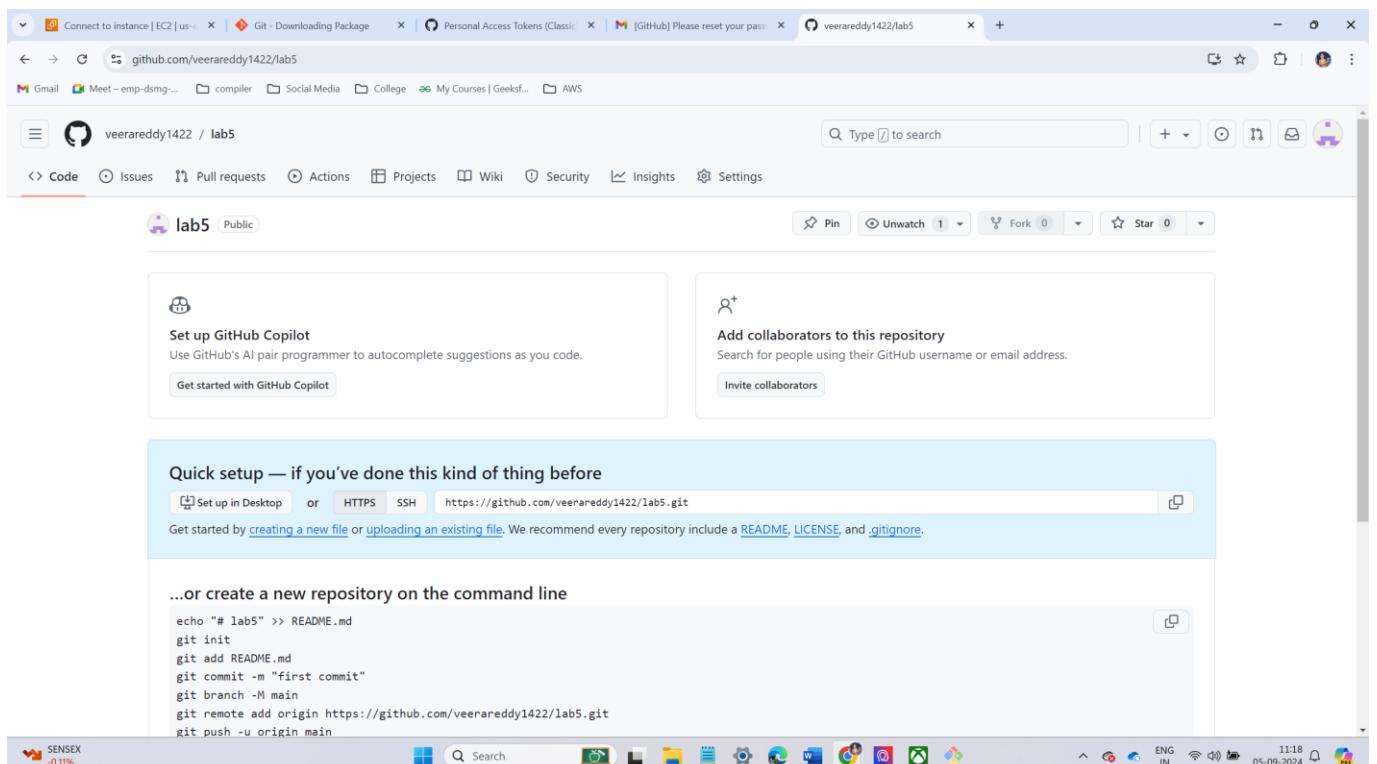
The screenshot shows a Windows terminal window with a black background and white text. The command `mkdir lab5` is entered, followed by `ll` to list the contents of the directory. The output shows a folder named `lab5` was created, containing a subfolder `folder`. The terminal window has a title bar and a taskbar at the bottom with various icons.

- Initialize the git on local repo using **git init** command

```
ec2-user@ip-172-31-92-108:~/lab5]$ mkdir lab5
[ec2-user@ip-172-31-92-108:~/lab5]$ ll
total 0
drwxrwxr-x 3 ec2-user ec2-user 36 Sep  5 04:50 folder
drwxrwxr-x 2 ec2-user ec2-user  6 Sep  5 05:44 lab5/
[ec2-user@ip-172-31-92-108:~/lab5]$ cd lab5/
[ec2-user@ip-172-31-92-108:~/lab5]$ 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>
Initialized empty Git repository in /home/ec2-user/lab5/.git/
[ec2-user@ip-172-31-92-108:~/lab5]$
```

The screenshot shows a Windows terminal window with a black background and white text. The command `git init` is entered in a directory named `lab5`. The terminal displays a series of hints about the `master` branch being used as the default. The terminal window has a title bar and a taskbar at the bottom with various icons.

- Create a New Repository on GitHub : **Provide a name** for the repository. Make sure the name is the same as local folder
- **Do not initialize** the GitHub repository with a README, gitignore, or license.
- After creating the repository, you will be redirected to a page showing the repository URL (HTTPS or SSH). Copy the **HTTPS URL**.



- Connect the Local Repo to the Remote GitHub Repo

```
ec2-user@ip-172-31-92-108:~/lab5
[ec2-user@ip-172-31-92-108 ~]$ mkdir lab5
[ec2-user@ip-172-31-92-108 ~]$ ll
total 0
drwxrwxr-x 3 ec2-user ec2-user 36 Sep  5 04:50 folder
drwxrwxr-x 2 ec2-user ec2-user  6 Sep  5 05:44 lab5
[ec2-user@ip-172-31-92-108 ~]$ cd lab5/
[ec2-user@ip-172-31-92-108 lab5]$ 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>
Initialized empty Git repository in /home/ec2-user/lab5/.git/
[ec2-user@ip-172-31-92-108 lab5]$ git branch -M main
[ec2-user@ip-172-31-92-108 lab5]$ git remote add origin https://github.com/veerareddy1422/lab5.git
[ec2-user@ip-172-31-92-108 lab5]$ git add .
[ec2-user@ip-172-31-92-108 lab5]$ git commit -m "Initial commit"
On branch main
Initial commit

nothing to commit (create/copy files and use "git add" to track)
[ec2-user@ip-172-31-92-108 lab5]$
```

- Whenever I am pushing local repo to GitHub it shows as an error like below.

```

ec2-user@ip-172-31-92-108:~/lab5
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>
Initialized empty Git repository in /home/ec2-user/lab5/.git/
[ec2-user@ip-172-31-92-108 lab5]$ git branch -M main
[ec2-user@ip-172-31-92-108 lab5]$ git remote add origin https://github.com/veerareddy1422/lab5.git
[ec2-user@ip-172-31-92-108 lab5]$ git add .
[ec2-user@ip-172-31-92-108 lab5]$ git commit -m "Initial commit"
On branch main
Initial commit

nothing to commit (create/copy files and use "git add" to track)
[ec2-user@ip-172-31-92-108 lab5]$ git push -u origin main
error: src refspec main does not match any
error: failed to push some refs to 'https://github.com/veerareddy1422/lab5.git'
[ec2-user@ip-172-31-92-108 lab5]$ git push -u origin main
error: src refspec main does not match any
error: failed to push some refs to 'https://github.com/veerareddy1422/lab5.git'
[ec2-user@ip-172-31-92-108 lab5]$ ^C
[ec2-user@ip-172-31-92-108 lab5]$ git pull origin main --rebase
fatal: Updating an unborn branch with changes added to the index.
[ec2-user@ip-172-31-92-108 lab5]$ git push origin main
error: src refspec main does not match any
error: failed to push some refs to 'https://github.com/veerareddy1422/lab5.git'
[ec2-user@ip-172-31-92-108 lab5]$ git push -f origin main
error: src refspec main does not match any
error: failed to push some refs to 'https://github.com/veerareddy1422/lab5.git'
[ec2-user@ip-172-31-92-108 lab5]$ git push -u origin master
error: src refspec master does not match any
error: failed to push some refs to 'https://github.com/veerareddy1422/lab5.git'
[ec2-user@ip-172-31-92-108 lab5]$ git branch -M main
[ec2-user@ip-172-31-92-108 lab5]$ git push -u origin main
error: src refspec main does not match any
error: failed to push some refs to 'https://github.com/veerareddy1422/lab5.git'
[ec2-user@ip-172-31-92-108 lab5]$ |

```

- To overcome the above error, I have checked the branch name.
- I have created different branch name.
- Again, I am pushing it works

```

ec2-user@ip-172-31-92-108:~/lab5
git config --global --edit
After doing this, you may fix the identity used for this commit with:
  git commit --amend --reset-author

1 file changed, 1 insertion(+)
create mode 100644 README.md
[ec2-user@ip-172-31-92-108 lab5]$ git branch -M main
[ec2-user@ip-172-31-92-108 lab5]$ git remote add origin https://github.com/veerareddy1422/lab5.git
error: remote origin already exists.
[ec2-user@ip-172-31-92-108 lab5]$ git push -u origin main
Username for 'https://github.com': veerareddy1422
Password for 'https://veerareddy1422@github.com':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 237 bytes | 237.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/veerareddy1422/lab5.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
[ec2-user@ip-172-31-92-108 lab5]$ git branch -M master
[ec2-user@ip-172-31-92-108 lab5]$ git remote add origin https://github.com/veerareddy1422/lab5.git
error: remote origin already exists.
[ec2-user@ip-172-31-92-108 lab5]$ git push -u origin master
Username for 'https://github.com': veerareddy1422
Password for 'https://veerareddy1422@github.com':
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:     https://github.com/veerareddy1422/lab5/pull/new/master
remote:
To https://github.com/veerareddy1422/lab5.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
[ec2-user@ip-172-31-92-108 lab5]$ |

```

- After pushing the GitHub will show like below.

The screenshot shows a GitHub repository page for 'veerareddy1422/lab5'. The main branch is 'main'. A dropdown menu shows 'main' is the default branch. The repository has 1 commit and 2 branches. The right sidebar shows basic repository statistics: Readme, Activity, 0 stars, 1 watching, 0 forks, and no releases published.

## LAB – 6 CREATING NEW BRANCH FROM MAIN BRANCH

- Go to Repo at the place of main and click view all branches.

The screenshot shows the 'Branches' page for the 'veerareddy1422/lab5' repository. It lists three branches: 'main', 'master', and a new branch 'nifty-smilecap'. The 'New branch' button is visible in the top right corner.

## ➤ Now click on New Branch

The screenshot shows a GitHub repository page for 'veerareddy1422/lab5'. A modal window titled 'Create a branch' is open, prompting for a 'New branch name' (set to 'fetch') and a 'Source' (set to 'main'). The 'Create new branch' button is highlighted in green. The main interface shows three sections: 'Default' (with 'main' branch), 'Your branches' (with 'master' branch), and 'Active branches' (empty). The bottom status bar indicates it's 29°C Haze, 14:56, and the date is 05-09-2024.

➤ New branch was created.

The screenshot shows the same GitHub repository page after the 'fetch' branch has been created. The 'Your branches' section now includes the 'fetch' branch, which was last updated '1 minute ago'. The 'Active branches' section also lists the 'fetch' branch. The bottom status bar indicates it's 29°C Haze, 14:57, and the date is 05-09-2024.

- Now make some changes on newly created branch
- I have created some text files in new branch

```
ec2-user@ip-172-31-92-108:~/lab5$ git switch fetch
Switched to branch 'fetch'
[ec2-user@ip-172-31-92-108 lab5]$ git branch
* fetch
  master
[ec2-user@ip-172-31-92-108 lab5]$ touch file1.txt file2.txt
[ec2-user@ip-172-31-92-108 lab5]$ ls
file1.txt  file2.txt  README.md
[ec2-user@ip-172-31-92-108 lab5]$ git add .
[ec2-user@ip-172-31-92-108 lab5]$ git status
On branch fetch
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   file1.txt
    new file:   file2.txt

[ec2-user@ip-172-31-92-108 lab5]$ git commit -m "add files to new branch"
[fetch 6312f2c] add files to new branch
Committer: EC2 Default User <ec2-user@ip-172-31-92-108.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 file1.txt
create mode 100644 file2.txt
[ec2-user@ip-172-31-92-108 lab5]$ git status
On branch fetch
nothing to commit, working tree clean
```

29°C Haze 15:22 05-09-2024

```
ec2-user@ip-172-31-92-108:~/lab5$ 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 file1.txt
create mode 100644 file2.txt
[ec2-user@ip-172-31-92-108 lab5]$ git status
On branch fetch
nothing to commit, working tree clean
[ec2-user@ip-172-31-92-108 lab5]$ git push
fatal: The current branch fetch has no upstream branch.
To push the current branch and set the remote as upstream, use
  git push --set-upstream origin fetch

To have this happen automatically for branches without a tracking
upstream, see 'push.autoSetupRemote' in 'git help config'.

[ec2-user@ip-172-31-92-108 lab5]$ git push -all
error: did you mean `--all` (with two dashes)?
[ec2-user@ip-172-31-92-108 lab5]$ git push -u origin fetch
Username for 'https://github.com': veerareddy1422
Password for 'https://veerareddy1422@github.com':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 311 bytes | 311.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/veerareddy1422/lab5.git
  ad315db..6312f2c  fetch -> fetch
branch 'fetch' set up to track 'origin/fetch'.
[ec2-user@ip-172-31-92-108 lab5]$
```

29°C Haze 15:22 05-09-2024

- The changes are occurring only in **new branch** not in **main branch**
- The changes are shown in the figure below.

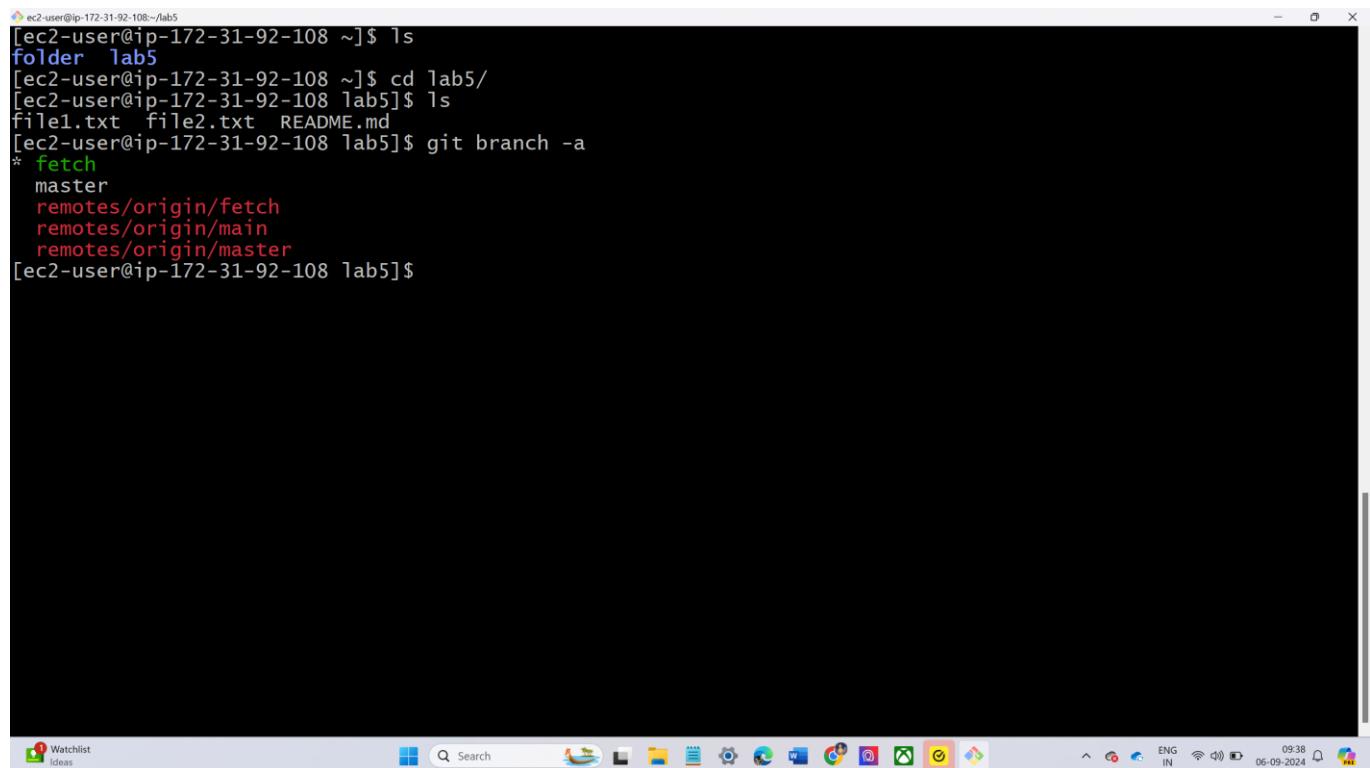
The screenshot shows the GitHub interface for the repository `veerareddy1422/lab5`. The `fetch` branch is selected, showing three commits from the EC2 Default User. The `main` branch is also listed, showing one commit. The `README` file is present in both branches. The GitHub UI includes a search bar, navigation links, and repository statistics like forks and stars.

- The main branch didn't have any effect.

The screenshot shows the GitHub interface for the repository `veerareddy1422/lab5`. The `main` branch is selected, showing three commits from the EC2 Default User. The `fetch` branch is also listed, showing one commit. The `README` file is present in both branches. The GitHub UI includes a search bar, navigation links, and repository statistics like forks and stars.

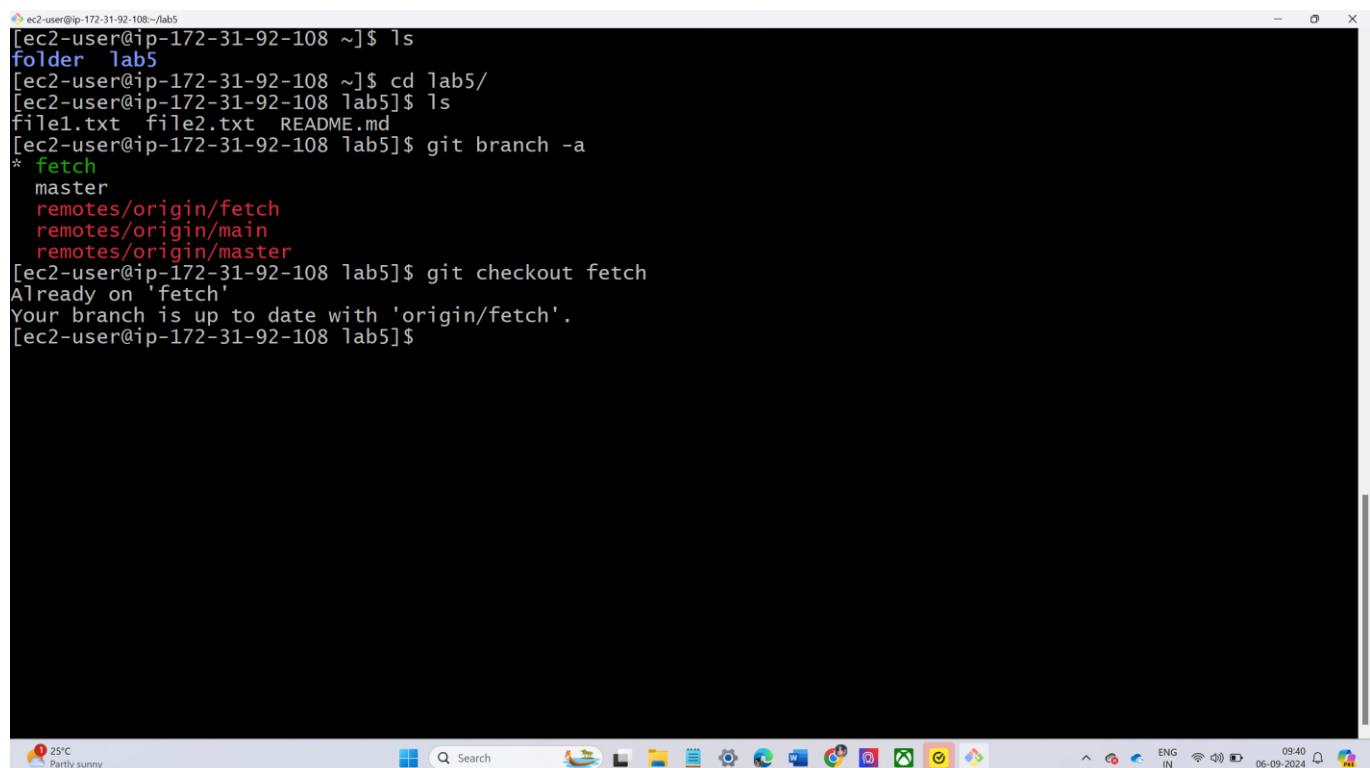
## LAB-7—PULL ALL BRANCHES IN YOUR LOCAL MACHINE

- List out all branches.
- Go to local machine with git repository and run command **git branch -a**.



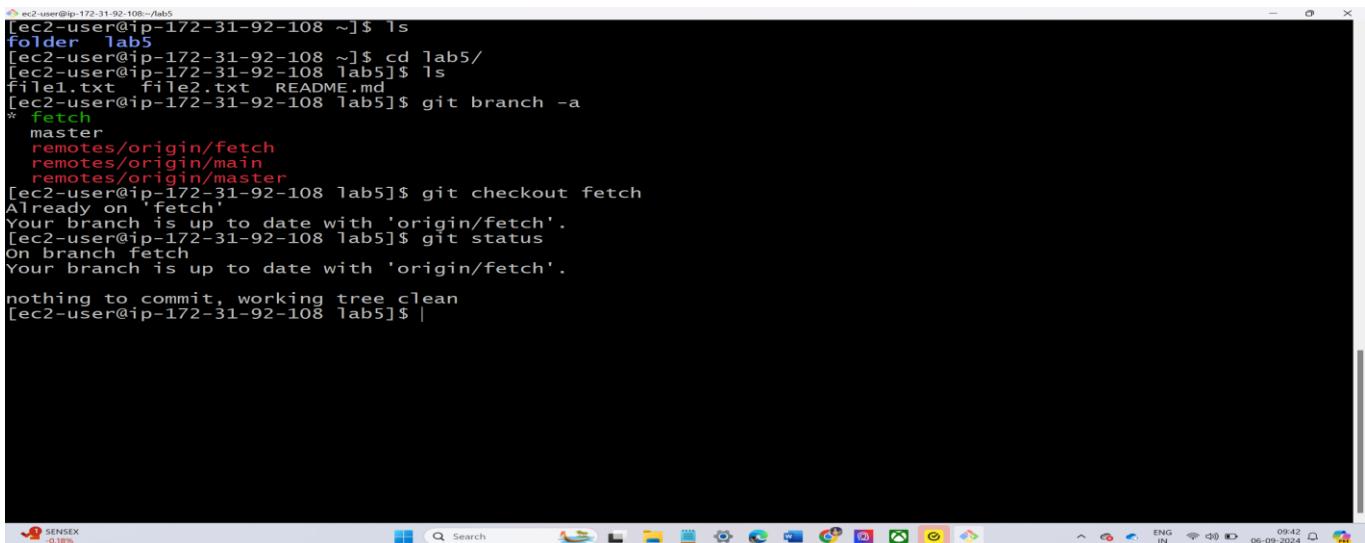
```
[ec2-user@ip-172-31-92-108 ~]$ ls
folder lab5
[ec2-user@ip-172-31-92-108 ~]$ cd lab5/
[ec2-user@ip-172-31-92-108 lab5]$ ls
file1.txt file2.txt README.md
[ec2-user@ip-172-31-92-108 lab5]$ git branch -a
* fetch
  master
    remotes/origin/fetch
    remotes/origin/main
    remotes/origin/master
[ec2-user@ip-172-31-92-108 lab5]$
```

- Checkout to the Feature Branch.
- switch to the feature branch you want to work on by running using command  
**git checkout <branch name>**



```
[ec2-user@ip-172-31-92-108 ~]$ ls
folder lab5
[ec2-user@ip-172-31-92-108 ~]$ cd lab5/
[ec2-user@ip-172-31-92-108 lab5]$ ls
file1.txt file2.txt README.md
[ec2-user@ip-172-31-92-108 lab5]$ git branch -a
* fetch
  master
    remotes/origin/fetch
    remotes/origin/main
    remotes/origin/master
[ec2-user@ip-172-31-92-108 lab5]$ git checkout fetch
Already on 'fetch'
Your branch is up to date with 'origin/fetch'.
[ec2-user@ip-172-31-92-108 lab5]$
```

- verify the status of the new branch using command **git status**.

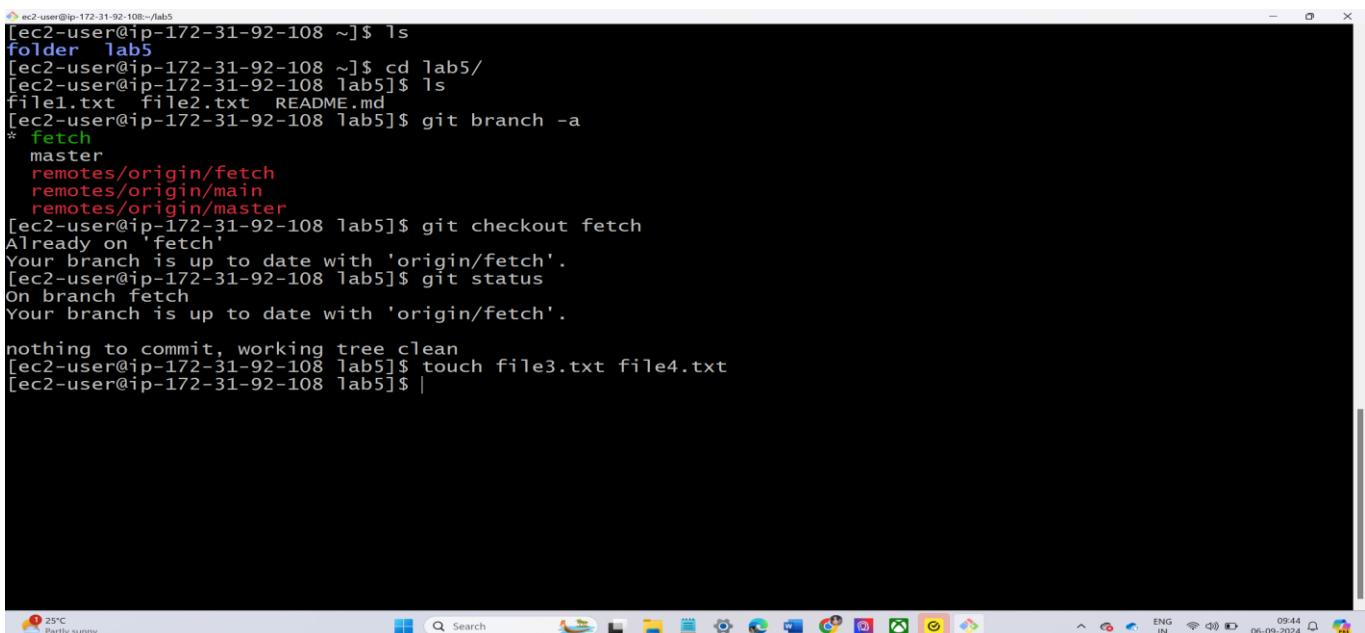


```
[ec2-user@ip-172-31-92-108 ~]$ ls
[ec2-user@ip-172-31-92-108 ~]$ cd lab5/
[ec2-user@ip-172-31-92-108 lab5]$ ls
file1.txt  file2.txt  README.md
[ec2-user@ip-172-31-92-108 lab5]$ git branch -a
* fetch
  master
  remotes/origin/fetch
  remotes/origin/main
  remotes/origin/master
[ec2-user@ip-172-31-92-108 lab5]$ git checkout fetch
Already on 'fetch'
Your branch is up to date with 'origin/fetch'.
[ec2-user@ip-172-31-92-108 lab5]$ git status
On branch fetch
Your branch is up to date with 'origin/fetch'.

nothing to commit, working tree clean
[ec2-user@ip-172-31-92-108 lab5]$ |
```

The screenshot shows a Windows desktop environment with a terminal window open. The terminal window displays the output of running 'git status' on a branch named 'fetch'. It shows that the branch is up-to-date with 'origin/fetch' and there is nothing to commit.

- Add new files into the branch
- Before adding we can create the file and then add it into the branch.
- For creating the files using the command **touch <files name>**.

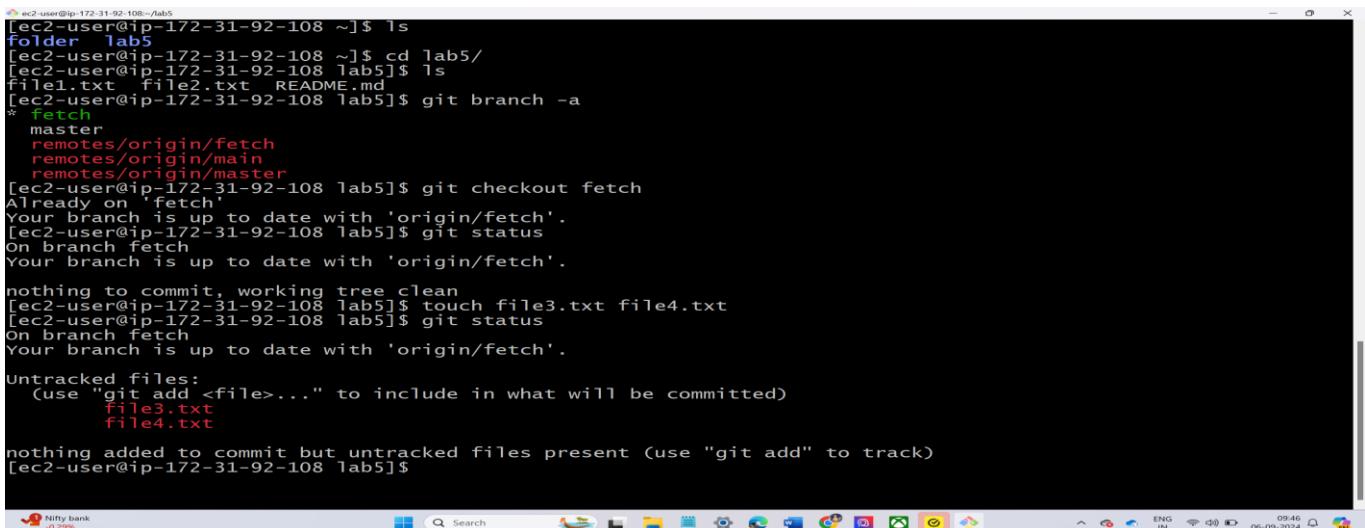


```
[ec2-user@ip-172-31-92-108 ~]$ ls
[ec2-user@ip-172-31-92-108 ~]$ cd lab5/
[ec2-user@ip-172-31-92-108 lab5]$ ls
file1.txt  file2.txt  README.md
[ec2-user@ip-172-31-92-108 lab5]$ git branch -a
* fetch
  master
  remotes/origin/fetch
  remotes/origin/main
  remotes/origin/master
[ec2-user@ip-172-31-92-108 lab5]$ git checkout fetch
Already on 'fetch'
Your branch is up to date with 'origin/fetch'.
[ec2-user@ip-172-31-92-108 lab5]$ git status
On branch fetch
Your branch is up to date with 'origin/fetch'.

nothing to commit, working tree clean
[ec2-user@ip-172-31-92-108 lab5]$ touch file3.txt file4.txt
[ec2-user@ip-172-31-92-108 lab5]$ |
```

The screenshot shows a Windows desktop environment with a terminal window open. The terminal window displays the output of running 'git status' on the 'fetch' branch. It shows that the branch is up-to-date with 'origin/fetch' and there is nothing to commit. At the bottom of the terminal window, the command 'touch file3.txt file4.txt' is shown, indicating that new files have been created.

➤ Now again check the status



```
[ec2-user@ip-172-31-92-108 ~]$ ls
[ec2-user@ip-172-31-92-108 ~]$ cd lab5/
[ec2-user@ip-172-31-92-108 lab5]$ ls
file1.txt file2.txt README.md
[ec2-user@ip-172-31-92-108 lab5]$ git branch -a
* fetch
  master
  remotes/origin/fetch
  remotes/origin/main
  remotes/origin/master
[ec2-user@ip-172-31-92-108 lab5]$ git checkout fetch
Already on 'fetch'
Your branch is up to date with 'origin/fetch'.
[ec2-user@ip-172-31-92-108 lab5]$ git status
On branch fetch
Your branch is up to date with 'origin/fetch'.

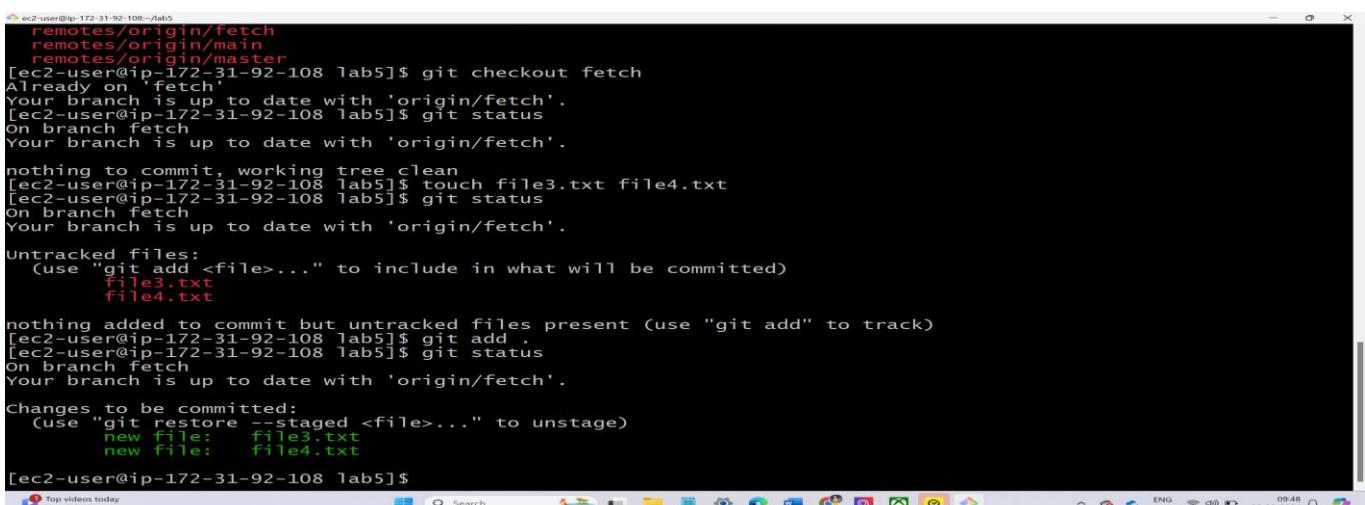
nothing to commit, working tree clean
[ec2-user@ip-172-31-92-108 lab5]$ touch file3.txt file4.txt
[ec2-user@ip-172-31-92-108 lab5]$ git status
On branch fetch
Your branch is up to date with 'origin/fetch'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    file3.txt
    file4.txt

nothing added to commit but untracked files present (use "git add" to track)
[ec2-user@ip-172-31-92-108 lab5]$
```

➤ Now move the files into the branch

➤ For moving the files into **staging area** using the command **git add**.



```
[ec2-user@ip-172-31-92-108 ~]$ ls
[ec2-user@ip-172-31-92-108 ~]$ cd lab5/
[ec2-user@ip-172-31-92-108 lab5]$ git checkout fetch
Already on 'fetch'
Your branch is up to date with 'origin/fetch'.
[ec2-user@ip-172-31-92-108 lab5]$ git status
On branch fetch
Your branch is up to date with 'origin/fetch'.

nothing to commit, working tree clean
[ec2-user@ip-172-31-92-108 lab5]$ touch file3.txt file4.txt
[ec2-user@ip-172-31-92-108 lab5]$ git status
On branch fetch
Your branch is up to date with 'origin/fetch'.

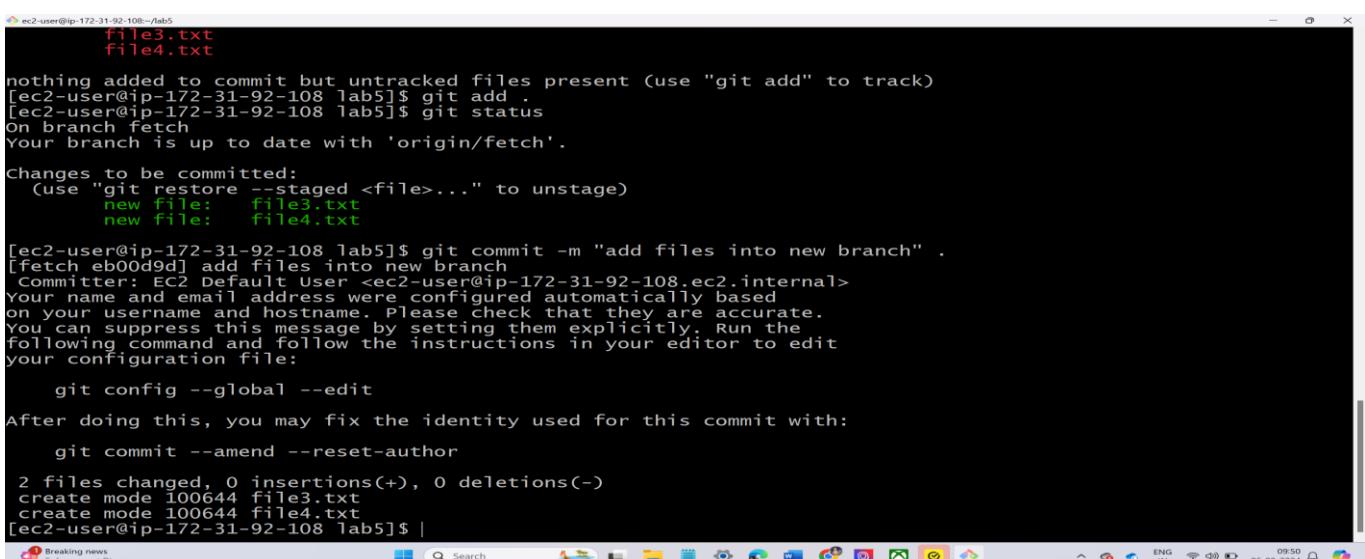
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    file3.txt
    file4.txt

nothing added to commit but untracked files present (use "git add" to track)
[ec2-user@ip-172-31-92-108 lab5]$ git add .
[ec2-user@ip-172-31-92-108 lab5]$ git status
On branch fetch
Your branch is up to date with 'origin/fetch'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   file3.txt
    new file:   file4.txt

[ec2-user@ip-172-31-92-108 lab5]$
```

➤ Now commit the files using command **git commit -m <message> filename**.



```
[ec2-user@ip-172-31-92-108 ~]$ ls
file3.txt
file4.txt

nothing added to commit but untracked files present (use "git add" to track)
[ec2-user@ip-172-31-92-108 lab5]$ git add .
[ec2-user@ip-172-31-92-108 lab5]$ git status
On branch fetch
Your branch is up to date with 'origin/fetch'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   file3.txt
    new file:   file4.txt

[ec2-user@ip-172-31-92-108 lab5]$ git commit -m "add files into new branch"
[fetch eb00d9d] add files into new branch
  Committer: EC2 Default User <ec2-user@ip-172-31-92-108.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

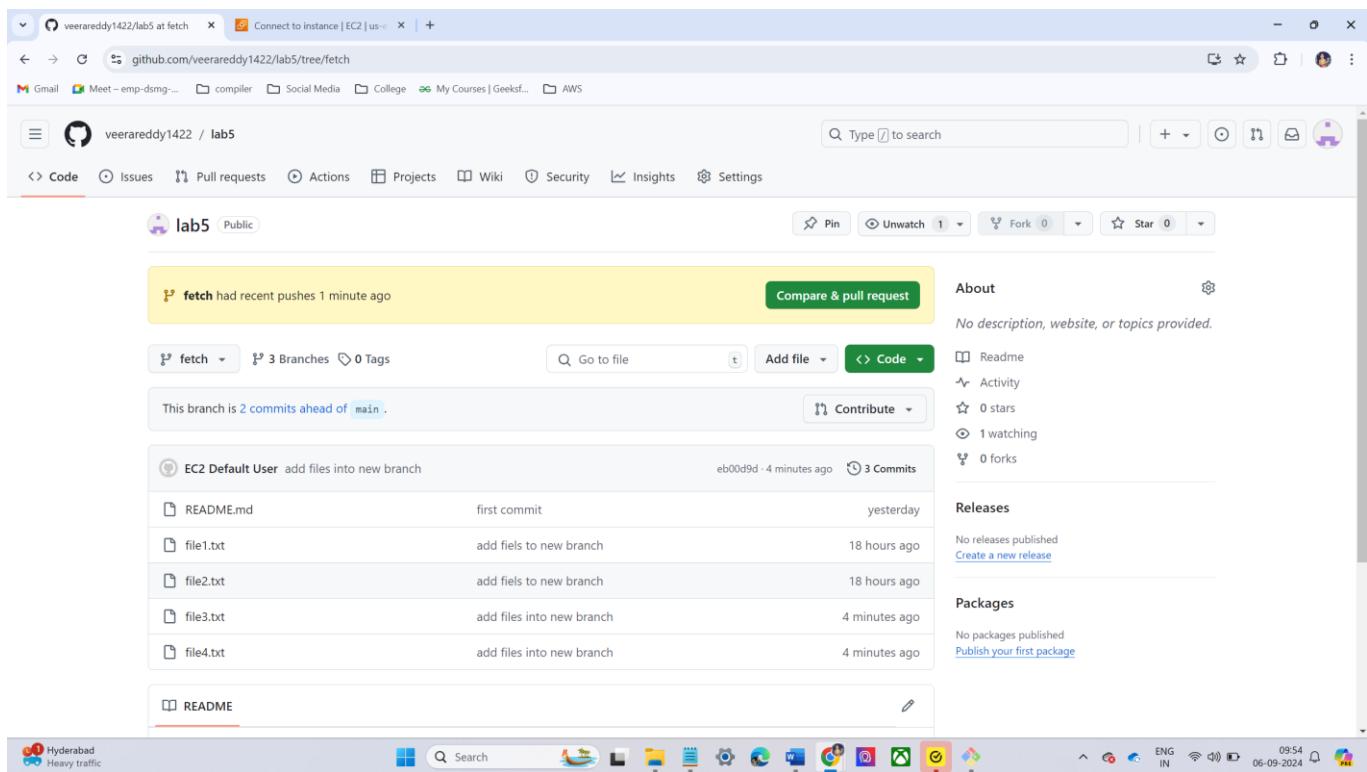
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file3.txt
create mode 100644 file4.txt
[ec2-user@ip-172-31-92-108 lab5]$
```

- Push the changes into remote branch using **git push** command

```
ec2-user@ip-172-31-92-108:~/lab5$ git restore --staged <file>..." to unstage)
new file:   file3.txt
new file:   file4.txt

[ec2-user@ip-172-31-92-108 lab5]$ git commit -m "add files into new branch"
[fetch eb00d9d] add files into new branch
Committer: EC2 Default User <ec2-user@ip-172-31-92-108.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 file3.txt
create mode 100644 file4.txt
[ec2-user@ip-172-31-92-108 lab5]$ git push origin fetch
Username for 'https://github.com': veerareddy1422
Password for 'https://veerareddy1422@github.com':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 310 bytes | 310.00 Kib/s, done.
Total 2 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/veerareddy1422/lab5.git
  6312f2c..eb00d9d  fetch -> fetch
[ec2-user@ip-172-31-92-108 lab5]$ |
```

- Now check the files in remote branch in git hub
- The new changes are present in your **feature branch**



- These changes are **not** yet available in the **main branch**.

The screenshot shows a GitHub repository named 'lab5'. The repository has 3 branches and 0 tags. It contains a README file with the text 'lab5'. The repository has 1 commit from 'EC2 Default User' made yesterday. The repository page includes sections for About, Releases, and Packages.

## LAB – 8 MERGE OUR FUTURE IN TO MAIN BRANCH

- Go to GitHub account and go repository
- **Click** branch down into new branch
- **Now** check the changes of the branch.
- Now go to pull requests tab
  - Go to left top → click settings → now click pull request

The screenshot shows the 'Pull requests' tab for the 'lab5' repository. A modal window titled 'Label issues and pull requests for new contributors' is open, prompting the user to discover issues labeled with 'good first issue'. Below the modal, there is a 'Welcome to pull requests!' message.

➤ Click on new pull request.

Comparing changes

Compare changes across branches, commits, tags, and more below. If you need to, you can also compare across forks.

base: main ▾ compare: main ▾

Choose different branches or forks above to discuss and review changes. [Learn about pull requests](#)

Create pull request

➤ In base branch drop down select main branch.

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also compare across forks or learn more about diff comparisons.

base: main ▾ compare: main ▾

There isn't anything to compare.

You'll need to use two different branch names to get a valid comparison.

Check out some of these sample comparisons.

base: main ▾ compare: main ▾

Showing 0 changed files with 0 additions and 0 deletions.

Split Unified

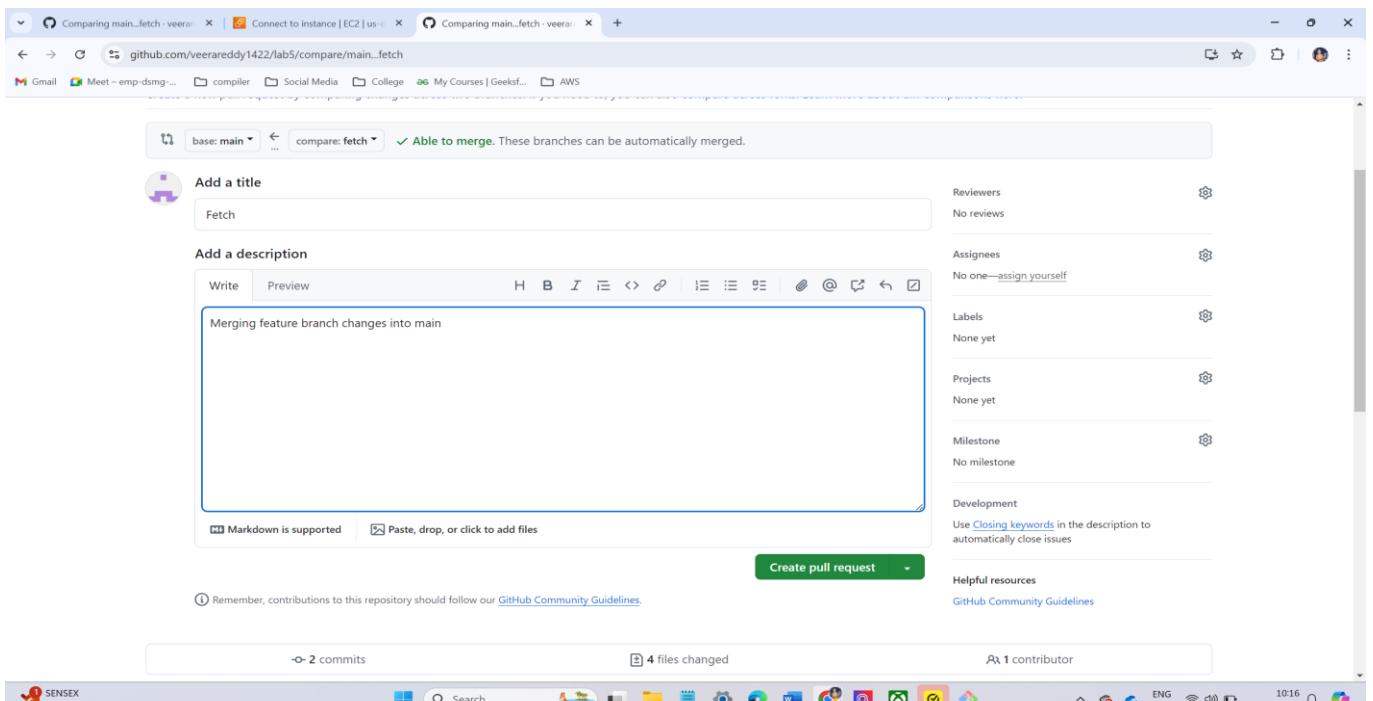
➤ In compare branch drop down select **fetch branch**.

The screenshot shows the GitHub 'Comparing changes' interface. At the top, it says 'base: main' and 'compare: fetch'. A green checkmark indicates 'Able to merge'. Below this, there's a summary: '2 commits', '4 files changed', and '1 contributor'. The commit history shows two commits from 'EC2 Default User': one on Sep 5, 2024, titled 'add file to new branch', and another on Sep 6, 2024, titled 'add files into new branch'. The bottom of the page has a 'Create pull request' button.

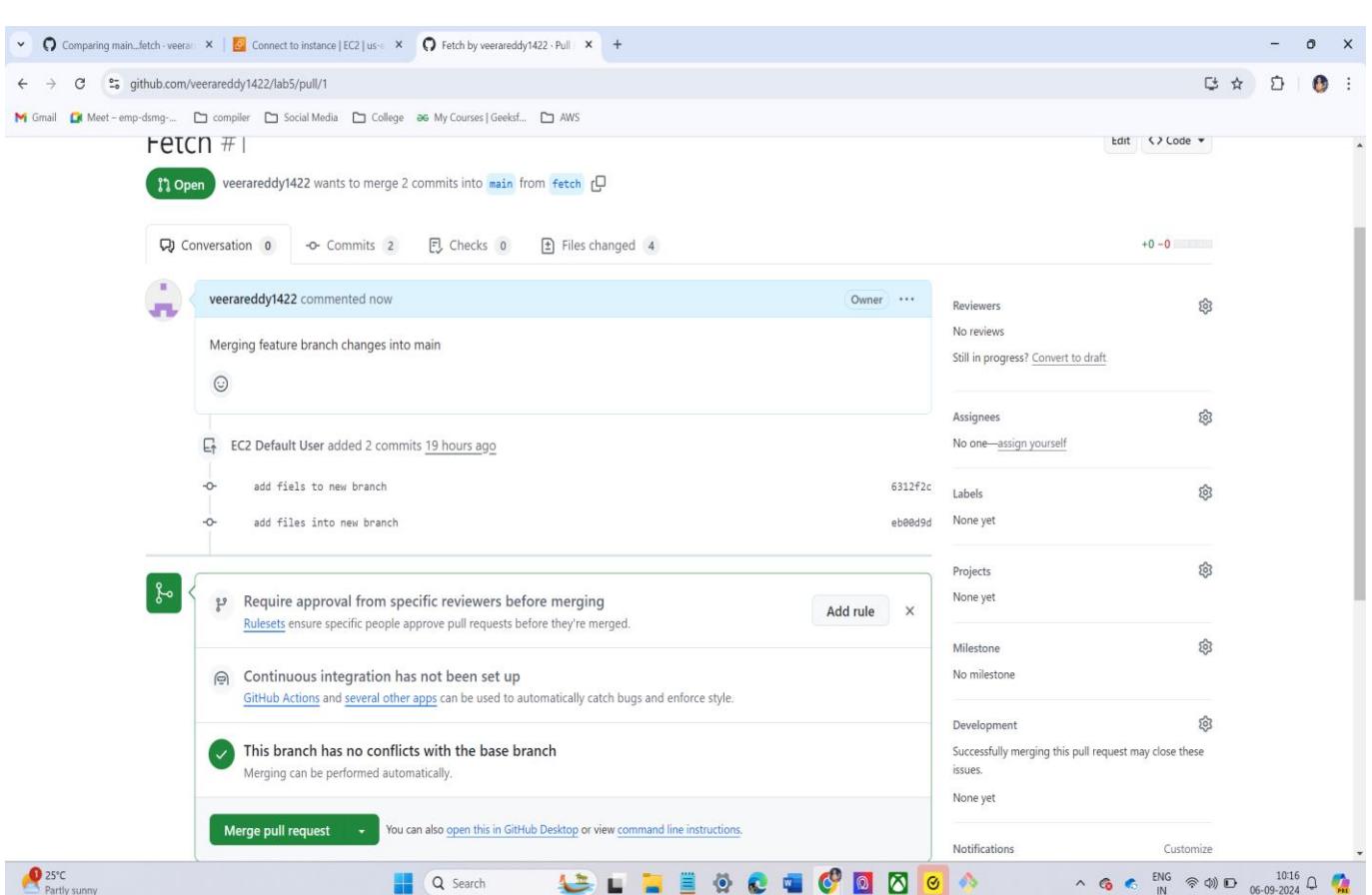
➤ Now click **Create pull request**.

The screenshot shows the GitHub 'Open a pull request' interface. It features a title field with 'Fetch' entered and a description editor with a placeholder 'Add your description here...'. To the right, there are sections for 'Reviewers' (No reviews), 'Assignees' (No one—assign yourself), 'Labels' (None yet), 'Projects' (None yet), 'Milestone' (No milestone), and 'Development' (Use Closing keywords in the description to). The bottom of the page includes a note about Markdown support and a file upload area.

- A new window will open where can add a comment or a description for the pull request
- After adding comments, click **Create Pull Request** again.



The screenshot shows the GitHub pull request creation interface. At the top, it says "base: main" and "compare: fetch". It indicates "Able to merge. These branches can be automatically merged." Below this, there's a "Add a title" field with "Fetch" entered, and an "Add a description" field containing "Merging feature branch changes into main". To the right, there are sections for "Reviewers" (No reviews), "Assignees" (No one—assign yourself), "Labels" (None yet), "Projects" (None yet), and "Milestone" (No milestone). Below the description, there's a note about using "Closing keywords" and a link to "GitHub Community Guidelines". At the bottom, there's a "Create pull request" button.

The screenshot shows the GitHub pull request details page for pull request #1. It displays "2 commits", "4 files changed", and "1 contributor". The pull request is titled "Fetch" and is described as merging 2 commits from the "fetch" branch into the "main" branch. A comment from "veerareddy1422" is shown: "Merging feature branch changes into main". Below the commit list, there are several status indicators: "EC2 Default User added 2 commits 19 hours ago" (with commits "add fields to new branch" and "add files into new branch"), a "Require approval from specific reviewers before merging" section (with a "Rulesets" link and an "Add rule" button), a "Continuous integration has not been set up" section (with a link to "GitHub Actions" and "several other apps"), and a "This branch has no conflicts with the base branch" section (with a green checkmark and a note that merging can be performed automatically). At the bottom, there's a "Merge pull request" button and a note about opening it in GitHub Desktop or viewing command line instructions. The interface includes standard Windows taskbar icons at the bottom.

## ➤ Review and Merge the Pull Request

Fetch #1

Open veerareddy1422 wants to merge 2 commits into `main` from `fetch`

Conversation 0 Commits 2 Checks 0 Files changed 4

veerareddy1422 commented 1 minute ago  
Merging feature branch changes into main

EC2 Default User added 2 commits 19 hours ago  
add files to new branch  
add files into new branch

Merge pull request #1 from veerareddy1422/fetch  
Fetch

This commit will be authored by ivreddy1422@gmail.com

Reviewers: No reviews  
Assignees: No one—assign yourself  
Labels: None yet  
Projects: None yet  
Milestone: No milestone  
Development: Successfully merging this pull request may close these issues.

Confirm merge Cancel

## ➤ Confirm by clicking Confirm Merge

Merged veerareddy1422 merged 2 commits into `main` from `fetch` now

veerareddy1422 commented 3 minutes ago  
Merging feature branch changes into main

EC2 Default User added 2 commits 19 hours ago  
add files to new branch  
add files into new branch

veerareddy1422 merged commit `40125fc` into `main` now

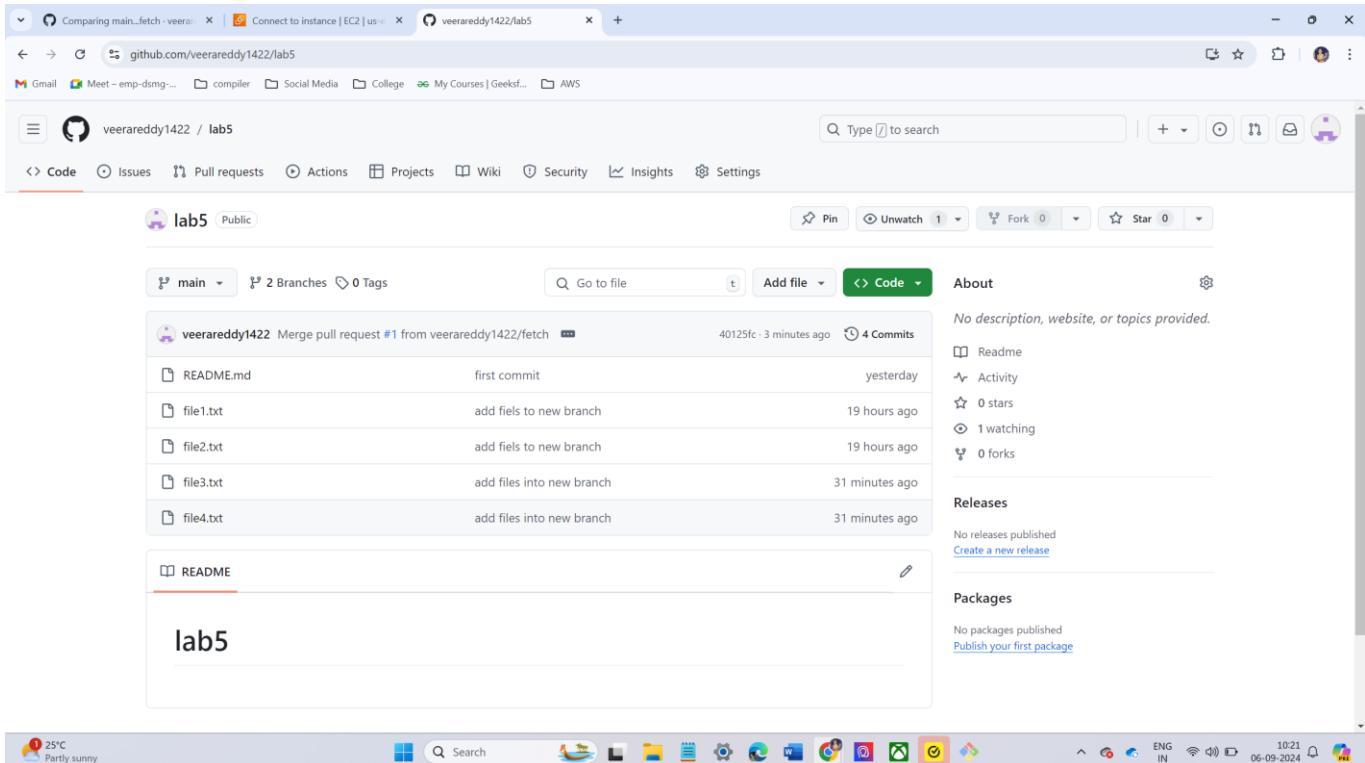
Pull request successfully merged and closed  
You're all set—the `fetch` branch can be safely deleted.

Add a comment

Write Preview

Reviewers: No reviews  
Assignees: No one—assign yourself  
Labels: None yet  
Projects: None yet  
Milestone: No milestone  
Development: Successfully merging this pull request may close these issues.  
None yet  
Notifications: Customize

- After merging, GitHub will offer the option to **Delete Branch**. click this if you no longer need the feature branch, or simply leave it as is.
- Verify the Changes in the **Main Branch**
- Go to the Code tab and verify that the changes from the feature branch (e.g., file3 and file4) are now present in the **main branch**.



## **LAB -9 – GOTO LOCAL MACHINE**

- Goto to local machine
  - Then go to directory
  - Now switch to **main branch**

- Pull the latest changes (including branches) from the remote repository using **git pull** command.

```
ec2-user@ip-172-31-92-108:~/lab5
The authenticity of host 'ec2-34-230-46-245.compute-1.amazonaws.com (34.230.46.245)' can't be established.
ED25519 key fingerprint is SHA256:42de07vUC1PSCHXlNE1B0LCr2ArMef50ztA24Kxt07c.
This host key is known by the following other names/addresses:
 ~/ssh/known_hosts:263: ec2-34-226-147-25.compute-1.amazonaws.com
 ~/ssh/known_hosts:265: ec2-44-201-122-110.compute-1.amazonaws.com
 ~/ssh/known_hosts:266: ec2-54-161-211-108.compute-1.amazonaws.com
Are you sure you want to continue connecting (yes/no/[fingerprint]? yes
Warning: Permanently added 'ec2-34-230-46-245.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Last Login: Fri Sep 6 04:05:45 2024 from 103.160.27.84

      #
      ####      Amazon Linux 2
~~~ \###      AL2 End of Life is 2025-06-30.
~~~ \#      A newer version of Amazon Linux is available!
~~~ .- /      Amazon Linux 2023, GA and supported until 2028-03-15.
~~~ /m/      https://aws.amazon.com/linux/amazon-linux-2023/
[ec2-user@ip-172-31-92-108 ~]$ ls
folder lab5
[ec2-user@ip-172-31-92-108 ~]$ cd lab5/
[ec2-user@ip-172-31-92-108 lab5]$ ls
file1.txt file2.txt file3.txt file4.txt README.md
[ec2-user@ip-172-31-92-108 lab5]$ git checkout main
branch 'main' set up to track 'origin/main'.
Switched to a new branch 'main'.
[ec2-user@ip-172-31-92-108 lab5]$ git pull
remote: Enumerating objects: 1, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (1/1), 890 bytes | 890.00 KiB/s, done.
From https://github.com/veerareddy1422/lab5
   ad315db..40125fc main           -> origin/main
Updating ad315db..40125fc
Fast-forward
 file1.txt | 0
 file2.txt | 0
 file3.txt | 0
 file4.txt | 0
 4 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file1.txt
 create mode 100644 file2.txt
 create mode 100644 file3.txt
 create mode 100644 file4.txt
[ec2-user@ip-172-31-92-108 lab5]$ |
```

Connect to instance | EC2 | us-east-1 | veerareddy1422/lab5

github.com/veerareddy1422/lab5

Gmail Meet – emp-dsmg... compiler Social Media College My Courses | GeeksforGeeks AWS

veerareddy1422 / lab5

Type / to search

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

lab5 Public

main 2 Branches 0 Tags Go to file Add file Code

veerareddy1422 Merge pull request #1 from veerareddy1422/fetch 40125fc · 2 days ago 4 Commits

README.md first commit 3 days ago

file1.txt add files to new branch 3 days ago

file2.txt add files to new branch 3 days ago

file3.txt add files into new branch 2 days ago

file4.txt add files into new branch 2 days ago

About No description, website, or topics provided.

Readme Activity 0 stars 1 watching 0 forks

Releases No releases published Create a new release

Packages No packages published Publish your first package

27°C Cloudy Search

ENG IN 08-09-2024 17:22