

## **TASK -1**

### **GIT – Version control system (VCS) by using Amazon Web Services**

#### **Introduction:**

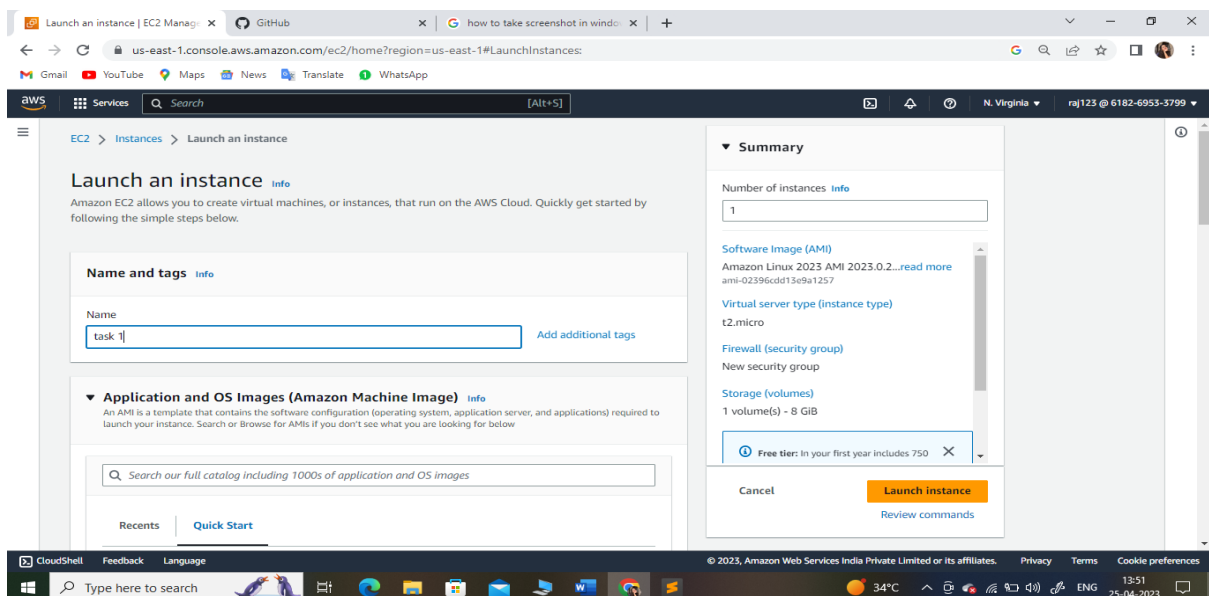
1. You can have code in local machine as well as in remote location
2. For remote location, we can use – GitHub/GitLab/bitbucket/AWS CodeCommit (so that we can interact with our team instead of working individually).

#### **What is GIT**

1. Git is the most common and widely used version control system in the world.
2. It is an open-source system.
3. Git was developed by Linus Torvalds in 2005.
4. Git is an example of Distributed Version Control System (DVCS).
5. DVCS means rather than having one place for full version history of the software.
6. Here, you can have a separate copy of the code in your local machines as well with the full history of changes.
7. Git works on the branching strategy, which means you can have many branches from your code just like a tree having branches connected to its trunk.
8. By using Git, we can interact with our team instead of working individually.

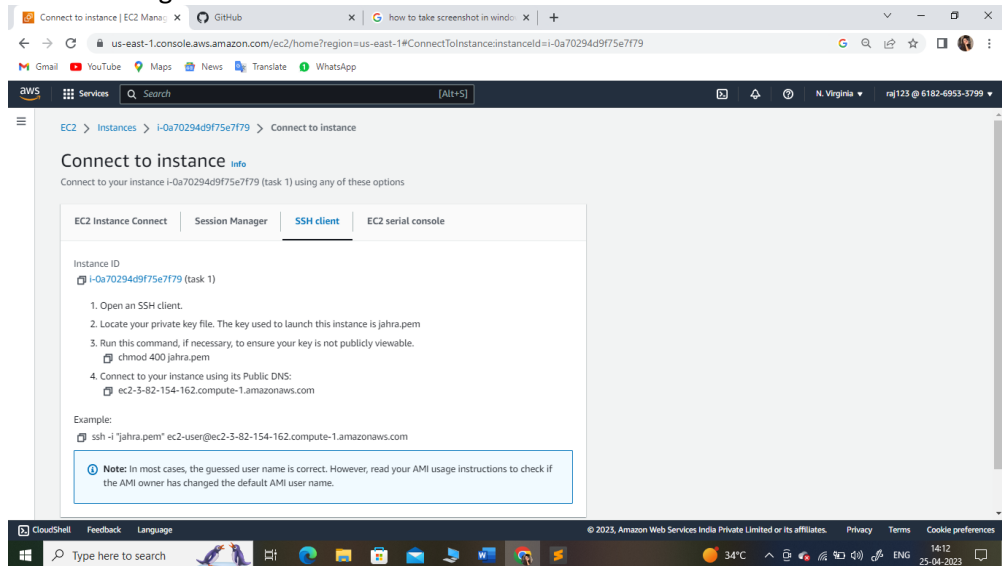
#### **Step - 1: (Creating EC2 instance)**

1. Login to AWS console (raj123).



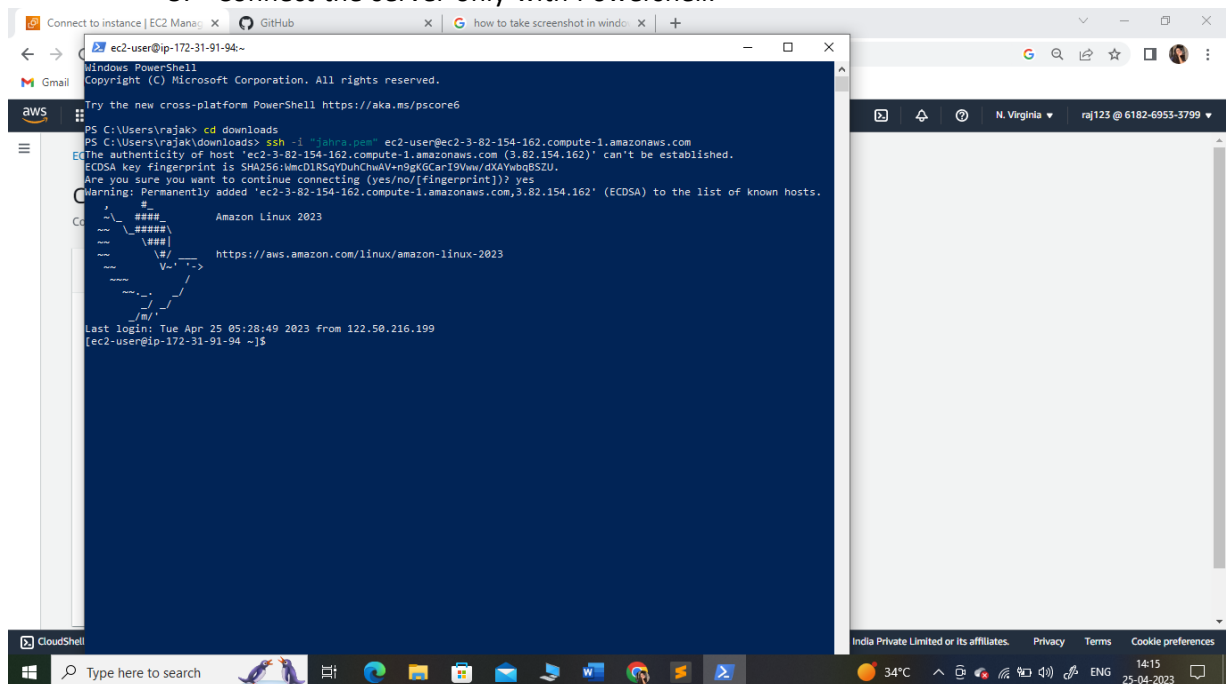
2. Create a server with Amazon Linux.

3. After creating the server



4. `ssh -i "jahra.pem" ec2-user@ec2-44-202-168-209.compute-1.amazonaws.com`

5. Connect the server only with Powershell.



## Step - 2: (Create repo in local machine)

1. Create a folder on my local machine

2. Initialize this folder using git clone command

3. Go inside this folder and run git status command to check the status,

```
ec2-user@ip-172-31-91-94:~$ ls
.git
last login: Tue Apr 25 05:28:49 2023 from 122.50.216.199
[ec2-user@ip-172-31-91-94 ~]$ ls
.git
[ec2-user@ip-172-31-91-94 ~]$ cd .git
[ec2-user@ip-172-31-91-94 ~]$ git
usage: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]
           [--exec-path<path>] [--html-path] [--man-path] [--info-path]
           [-p | --paginate] [-P | --no-pager] [--no-replace-objects] [--bare]
           [--git-dir<path>] [--work-tree<path>] [--namespace<name>]
           [--super-prefix<path>] [--config-env<name>=<envvar>]
           <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
  clone Clone a repository into a new directory
  init   Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
  add    Add file contents to the index
  mv     Move or rename a file, a directory, or a symlink
  restore Restore working tree files
  rm     Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)
  bisect Use binary search to find the commit that introduced a bug
  diff   Show changes between commits, commit and working tree, etc
  grep   Print lines matching a pattern
  log    Show commit logs
  show   Show various types of objects
  status Show the working tree status

grow, mark and tweak your common history
  branch List, create, or delete branches
  commit Record changes to the repository
  merge  Join two or more development histories together
  rebase Reapply commits on top of another base tip
  reset  Reset current HEAD to the specified state
  switch Switch branches
  tag    Create, list, delete or verify a tag object signed with GPG

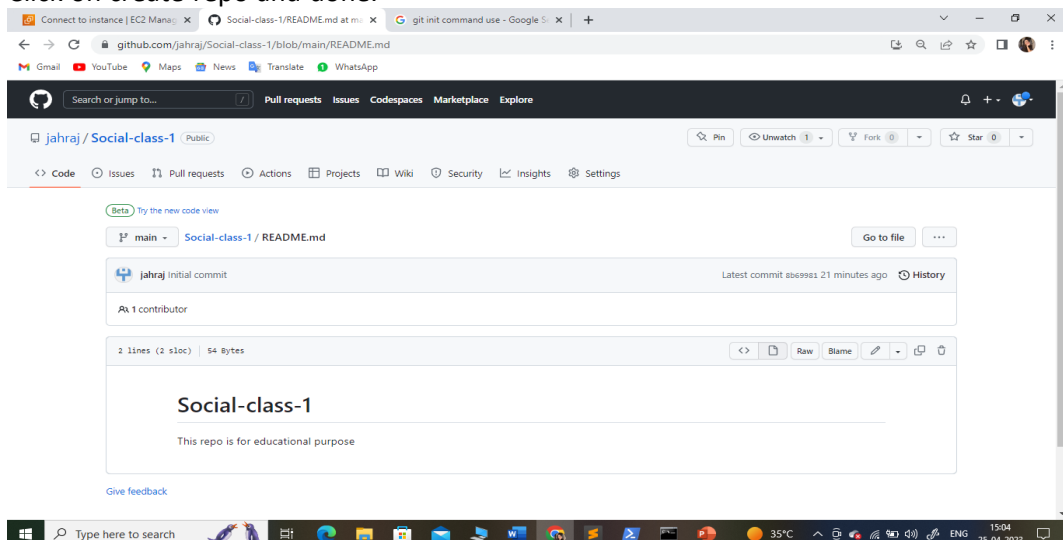
collaborate (see also: git help workflows)
  fetch Download objects and refs from another repository
  pull  Fetch from and integrate with another repository or a local branch
  push  Update remote refs along with associated objects
```

4. created some empty files using touch command  
- touch jahanavi.txt
5. Again, run git status to see the changes, you can notice that the file is available but not tracked by Git
6. Run git add <file.txt file1.txt> to stage this change (git will start tracking this file), you may check it using git status once again
7. Now commit our changes by running git commit -m "added files" file.txt file1.txt
8. Run git status once again and it will show you that the working tree is clean.

```
ec2-user@ip-172-31-91-249:~/jahanavi.  
[ec2-user@ip-172-31-91-249 jahanavi.]$ git status  
On branch master  
Changes to be committed:  
  (use "git restore --staged <file>..." to unstage)  
    new file:   file.txt  
  
[ec2-user@ip-172-31-91-249 jahanavi.]$ git commit -m "addind files" file.txt file1.txt  
[master 844f434] addind files  
  Committer: EC2 Default User <ec2-user@ip-172-31-91-249.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 file.txt  
[ec2-user@ip-172-31-91-249 jahanavi.]$ git status  
On branch master  
nothing to commit, working tree clean  
[ec2-user@ip-172-31-91-249 jahanavi.]$
```

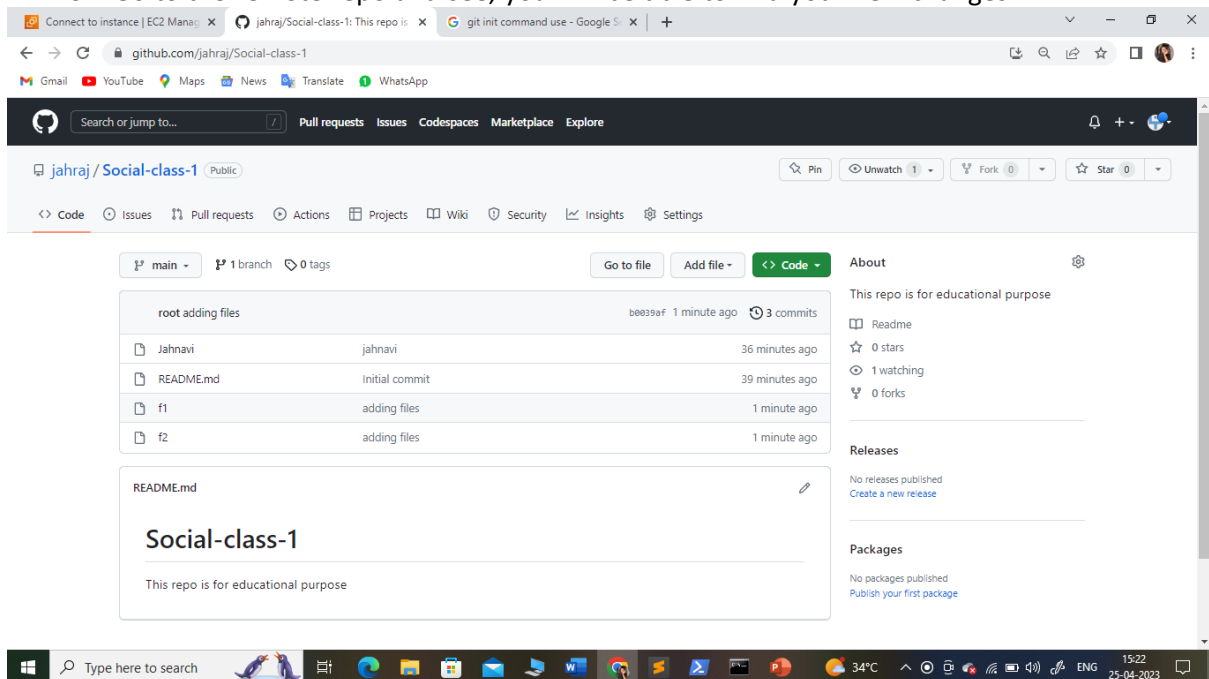
### STEP 3: (Creating repo in remote location – GitHub)

1. In remote location – GitHub.
2. Create a new repository by clicking on **new** button
  - a. Provide repo name ( Social Class 1 )
  - b. Select whether it is a private or public repo (recommended is private)
  - c. Initialize the repo by adding a **README.md** file.
3. Click on create repo and done.



## STEP 4: (Working with Remote repo)

1. Pick the clone URL of the repository from the GitHub repo.
2. Go to your local machine and clone this repo using git clone command  
Git clone <https://github.com/jahraj/Social-class-1.git>
3. Once cloned, go to the repo folder and add some sample files. we can use touch command to create empty files.
  - a. touch f1 f2
4. Stage these changes by running git add f1 f2
5. Commit these changes by running git commit -m "adding files" f1 f2
6. Go to the remote repo and see, you will be able to find your new changes.

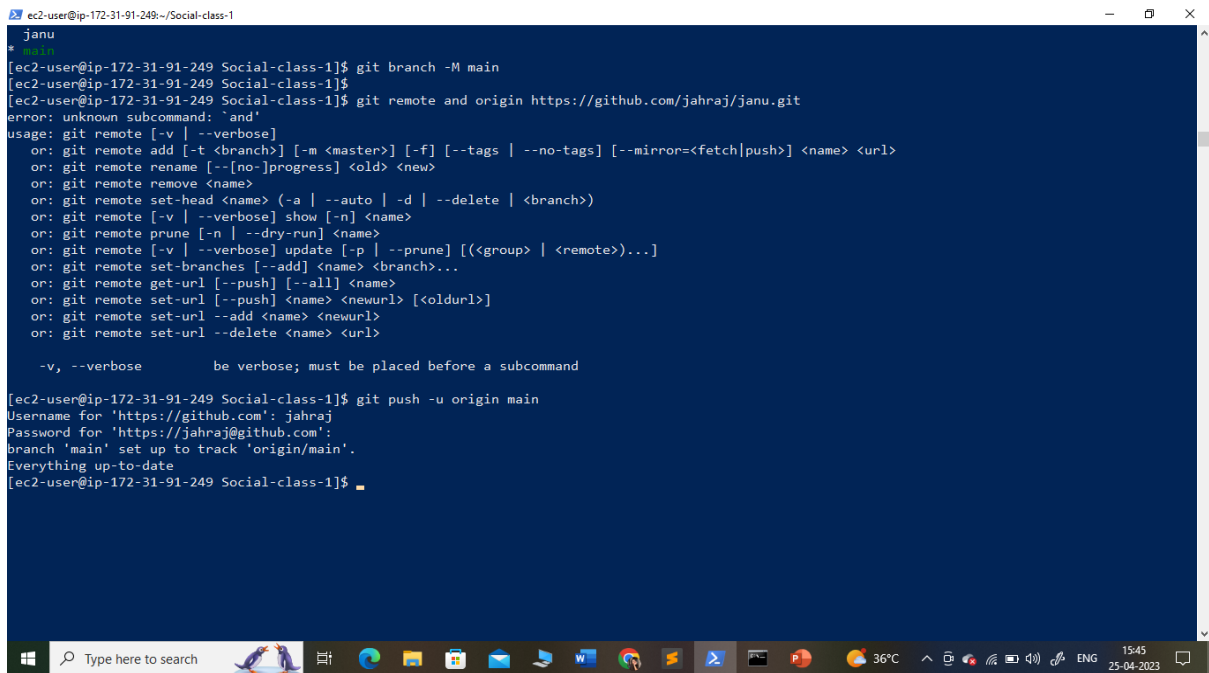


## STEP 5: (Pushing a locally created repo to GitHub)

1. Create one repo in our local machine and initialize it locally
2. Create one remote repo with the same name as local repo in GitHub and do not initialize

it

3. Come to your local machine and run the following commands from inside your local repo
  - a. `git branch -M main`
  - b. `git remote add origin < ghp\_sSn6Q1neklDx9m2TcykOKyA7aFceZb2pyWGk >`
  - c. `git push -u origin main`



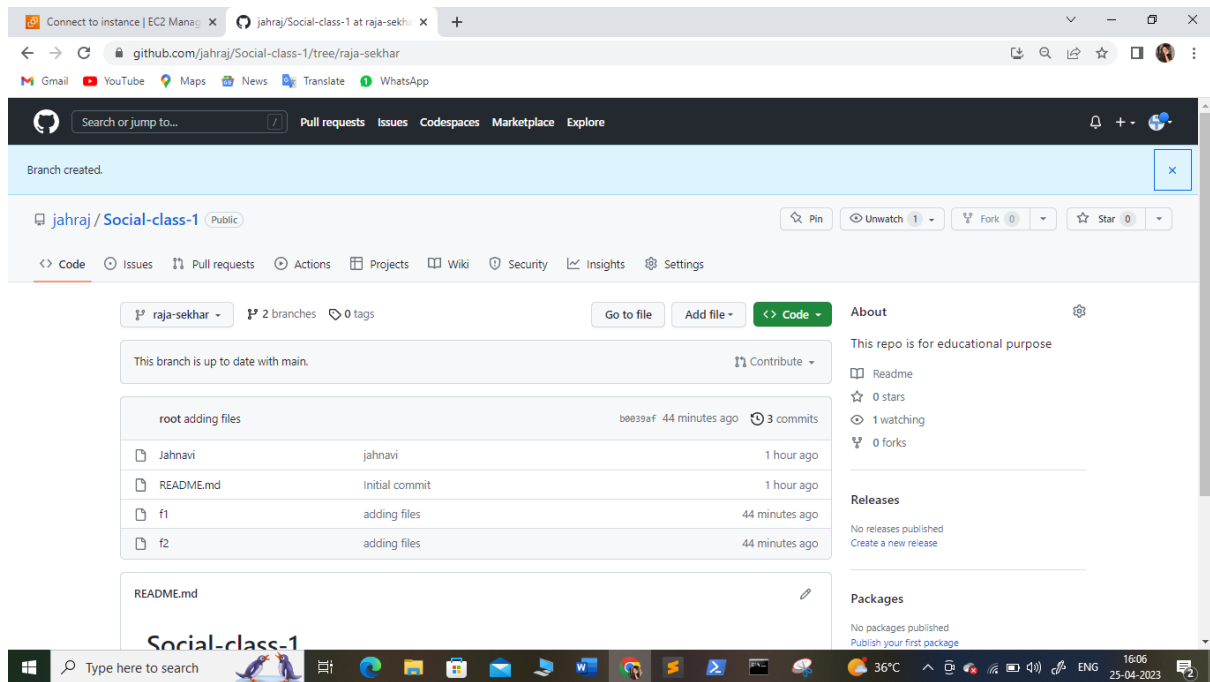
```
ec2-user@ip-172-31-91-249:~/Social-class-1
janu
* main
[ec2-user@ip-172-31-91-249 Social-class-1]$ git branch -M main
[ec2-user@ip-172-31-91-249 Social-class-1]$
[ec2-user@ip-172-31-91-249 Social-class-1]$ git remote add origin https://github.com/jahraj/janu.git
error: unknown subcommand: 'and'
usage: git remote [-v | --verbose]
or: git remote add [-t <branch>] [-m <master>] [-f] [--tags | --no-tags] [--mirror=<fetch|push>] <name> <url>
or: git remote rename [--[no-]progress] <old> <new>
or: git remote remove <name>
or: git remote set-head <name> (-a | --auto | -d | --delete | <branch>)
or: git remote [-v | --verbose] show [-n] <name>
or: git remote prune [-n | --dry-run] <name>
or: git remote [-v | --verbose] update [-p | --prune] [(<group> | <remote>)...]
or: git remote set-branches [--add] <name> <branch>...
or: git remote get-url [--push] [--all] <name>
or: git remote set-url [--push] <name> <newurl> [<oldurl>]
or: git remote set-url --add <name> <newurl>
or: git remote set-url --delete <name> <url>

-v, --verbose          be verbose; must be placed before a subcommand

[ec2-user@ip-172-31-91-249 Social-class-1]$ git push -u origin main
Username for 'https://github.com': jahraj
Password for 'https://jahraj@github.com':
branch 'main' set up to track 'origin/main'.
Everything up-to-date
[ec2-user@ip-172-31-91-249 Social-class-1]$
```

## STEP 6: (Creating a new branch from your main branch)

1. Go to the repository at the place of main and click on the branch dropdown.
2. Type the name of your new branch that you want to create and click on create button.
3. A new branch will be created.
  - Raja sekhar



## STEP 7: (Pull all the branches in your local machine)

1. Go to your local machine where you have the copy of your remote branch
2. Run the command "git pull" to pull all the new changes such as branches from the remote location
3. Checkout to the feature branch or the branch that you created in **(Lab – 6)**.
  - a. git checkout <Raja-sekhar>
4. Make some changes in this branch such as adding the files "touch file1.1txt"
5. git add <file1.1txt>
6. git commit -m "adding file" file1.1txt
7. git push

```
ec2-user@ip-172-31-91-249:~/Social-class-1
[ec2-user@ip-172-31-91-249 Social-class-1]$ ls
Jahnnavi README.md f1 f2 file1.1.txt rajah
[ec2-user@ip-172-31-91-249 Social-class-1]$ sudo git add file1.1.txt
[ec2-user@ip-172-31-91-249 Social-class-1]$
[ec2-user@ip-172-31-91-249 Social-class-1]$
[ec2-user@ip-172-31-91-249 Social-class-1]$ sudo git commit -m "adding file" file1.1.txt
[raja-sekhar 262efe8] adding file
Committer: root <root@ip-172-31-91-249.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

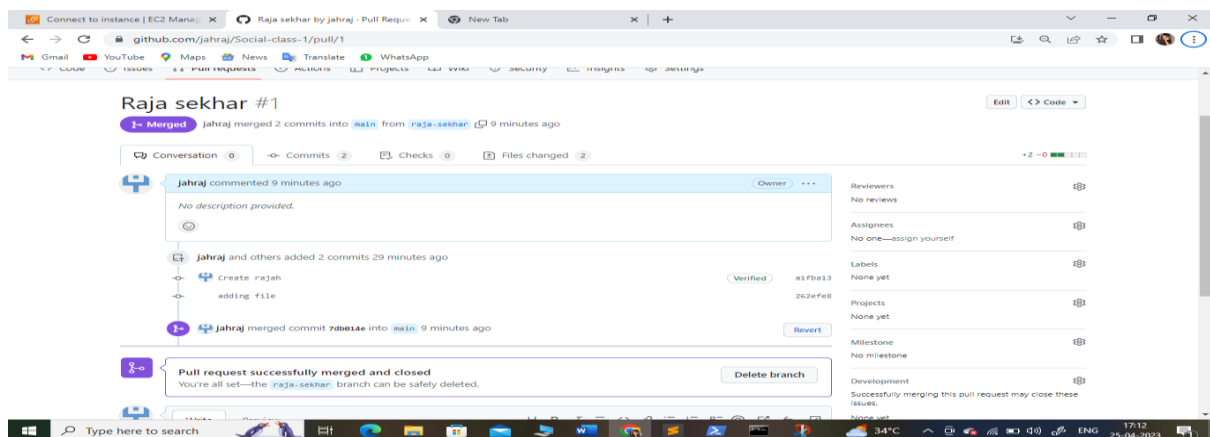
After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file1.1.txt
[ec2-user@ip-172-31-91-249 Social-class-1]$ git push
Username for 'https://github.com': jahraj
Password for 'https://jahraj@github.com':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 241 bytes | 241.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/jahraj/Social-class-1.git
   a1fba13..262efe8  raja-sekhar -> raja-sekhar
[ec2-user@ip-172-31-91-249 Social-class-1]$
```

## STEP - 8: (Merge our feature branch with main branch)

1. Go to your GitHub repository
2. Go to the pull request tab and click on create **Pull Request**
3. Click on create pull request and it will ask for a comment, just click again on create pull request
4. Go to pull request tab once again and click on the pull request available there
5. Click on review changes and then merge





## Lab 9: (Go to local machine)

1. Go to your local machine where you have the copy of your remote repo
2. Checkout to the main branch
  - Git checkout raja-sekhar
3. Now run the command "git pull" to pull all the new changes such as branches from the remote location
  - Sudo git pull
4. Here see that the new changes are only available in your main branch

```
ec2-user@ip-172-31-91-249:~/Social-class-1
fatal: not a git repository (or any of the parent directories): .git
[ec2-user@ip-172-31-91-249 ~]$ ls
Social-class-1  jahnavi
[ec2-user@ip-172-31-91-249 ~]$ cd Social-class-1
[ec2-user@ip-172-31-91-249 Social-class-1]$
[ec2-user@ip-172-31-91-249 Social-class-1]$ ls
Jahnavi README.md f1 f2 file1.txtt rajah
[ec2-user@ip-172-31-91-249 Social-class-1]$
[ec2-user@ip-172-31-91-249 Social-class-1]$ sudo git branch
  janu
  main
* raja-sekhar
[ec2-user@ip-172-31-91-249 Social-class-1]$ 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
Unpacking objects: 100% (1/1), 634 bytes | 634.00 KiB/s, done.
error: cannot update the ref 'refs/remotes/origin/main': unable to append to '.git/logs/refs/remotes/origin/main': Permission denied
From https://github.com/jahraj/Social-class-1
 ! b0039af..7db014e  main    -> origin/main (unable to update local ref)
[ec2-user@ip-172-31-91-249 Social-class-1]$ sudo git pull
From https://github.com/jahraj/Social-class-1
 ! b0039af..7db014e  main    -> origin/main
Already up to date.
[ec2-user@ip-172-31-91-249 Social-class-1]$
```

Thanking you

Raja Sekhar kanuri