# TASK -1

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

### Introduction:

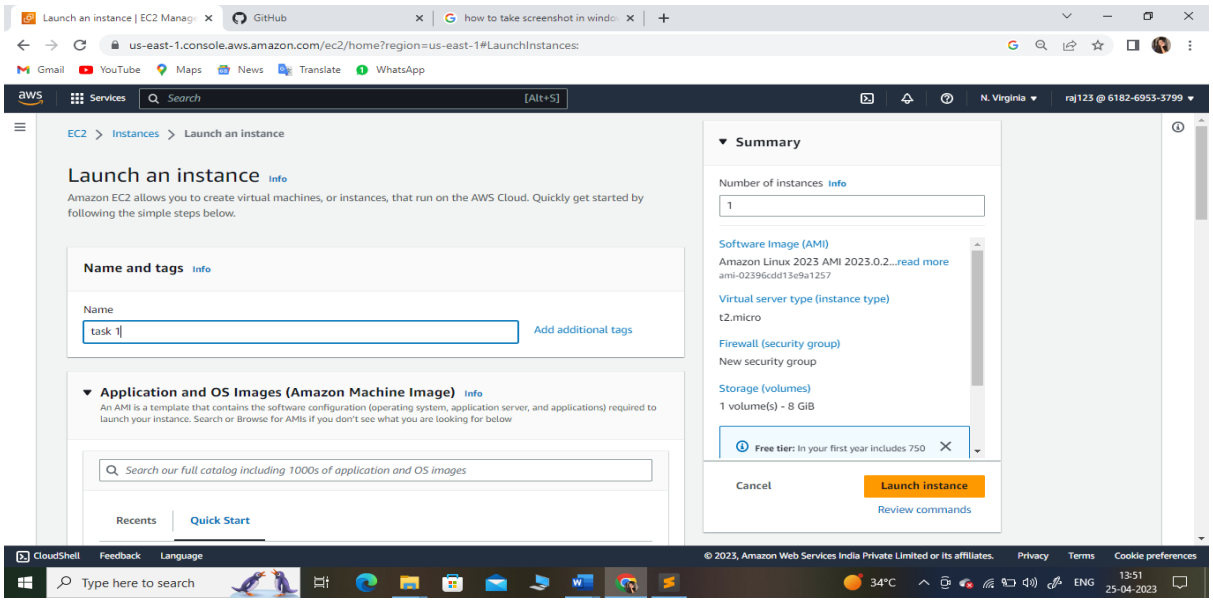
1. You can have code in local machine as well as in remote location
2. For remote location, we can use – GitHub/GitLab/bitbucket/AWS CodeCommit (so thatwe can interact with our team instead of working individually).

### What is GIT

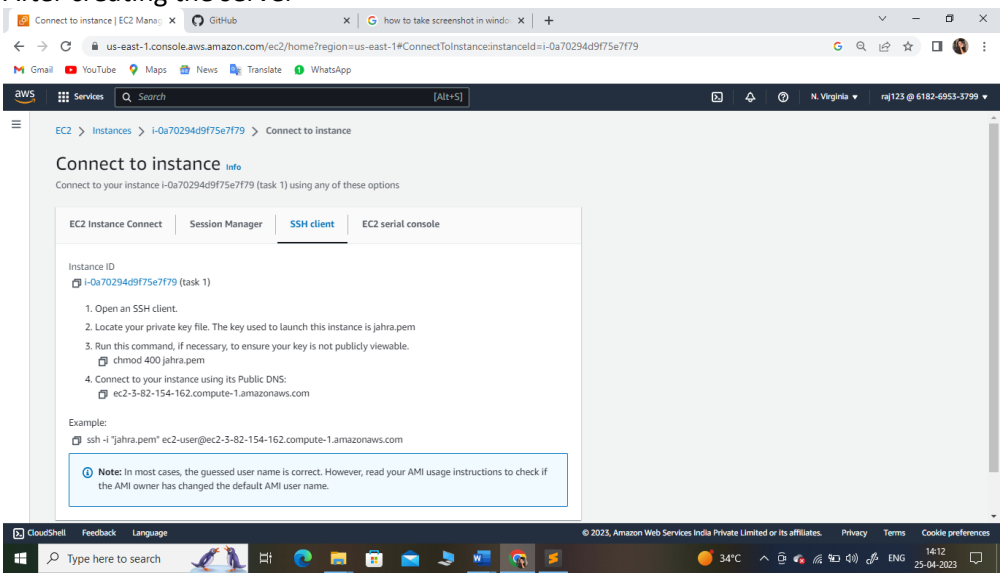
1. Git is the most common and widely used version control system in the world.
2. It is an open-source system.
3. Git was developed by Linus Torvalds in 2005.
4. Git is an example of Distributed Version Control System (DVCS).
5. DVCS means rather than having one place for full version history of the software.
6. Here, you can have a separate copy of the code in your local machines as well with the fullhistory of changes.
7. Git works on the branching strategy, which means you can have many branches from yourcode 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)

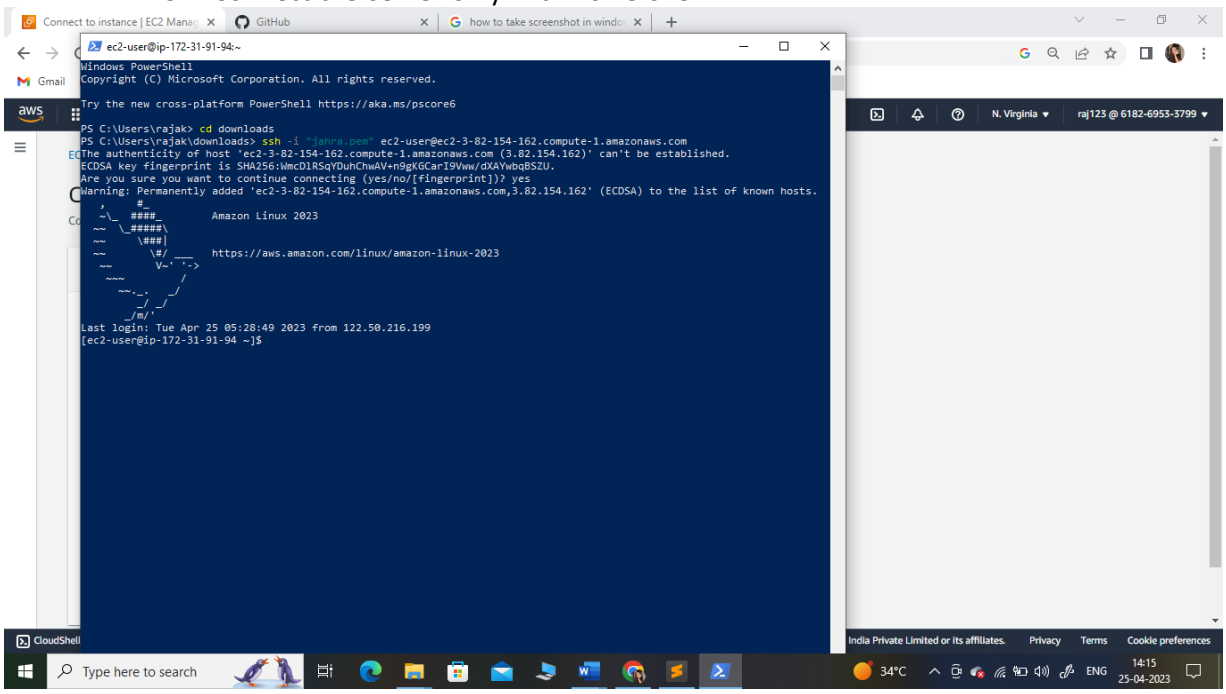
1. Login to AWS console (raj123).



2. Create a server with Amazon Linux.
3. After creating the server



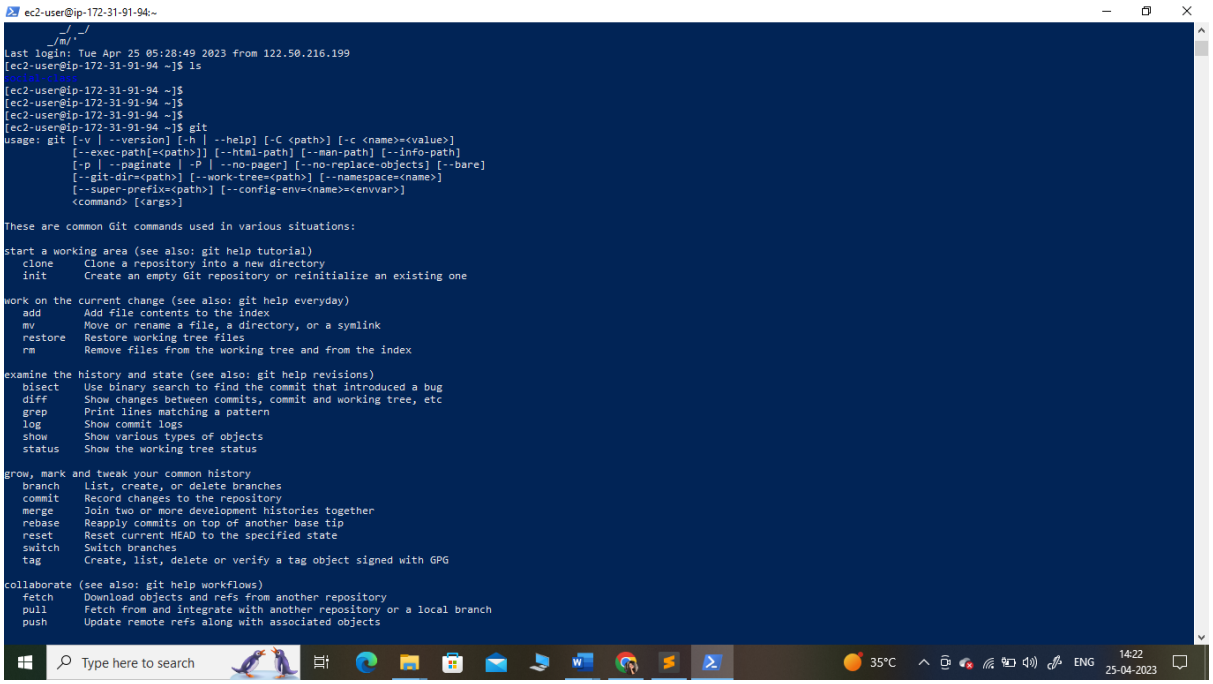

4. ssh -i "jahra.pem" ec2-user@ec2-44-202-168-209.compute-1.amazonaws.com
5. Connect the server only with Powershell.




## 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,

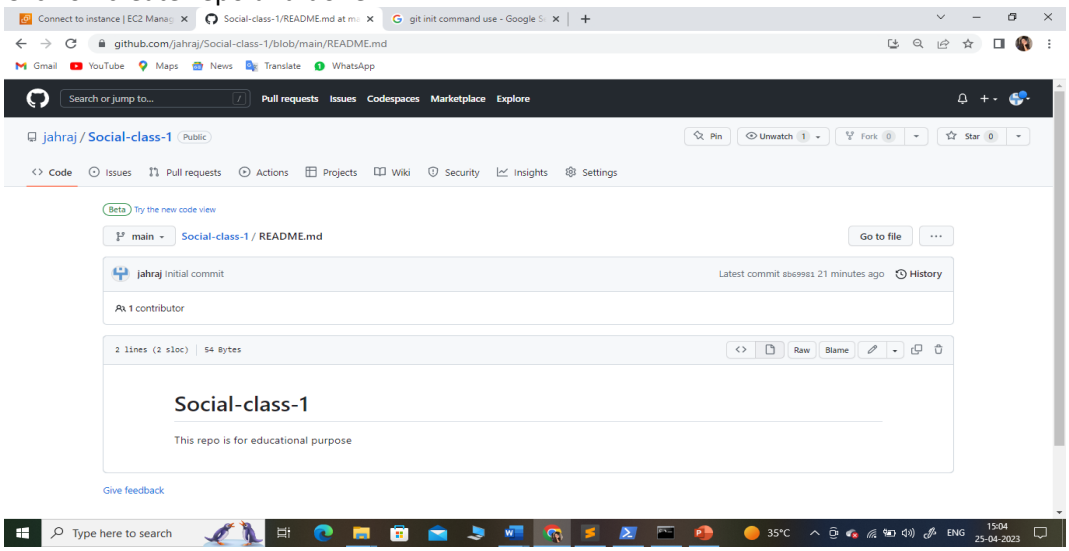



4. created some empty files using touch command
   - touch jahanavi.txt
5. Again, run git status to see the changes, you can notice that the file is available but nottracked by Git
6. Run git add <file.txt file1.txt> to stage this change (git will start tracking this file), you maycheck it using git 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.

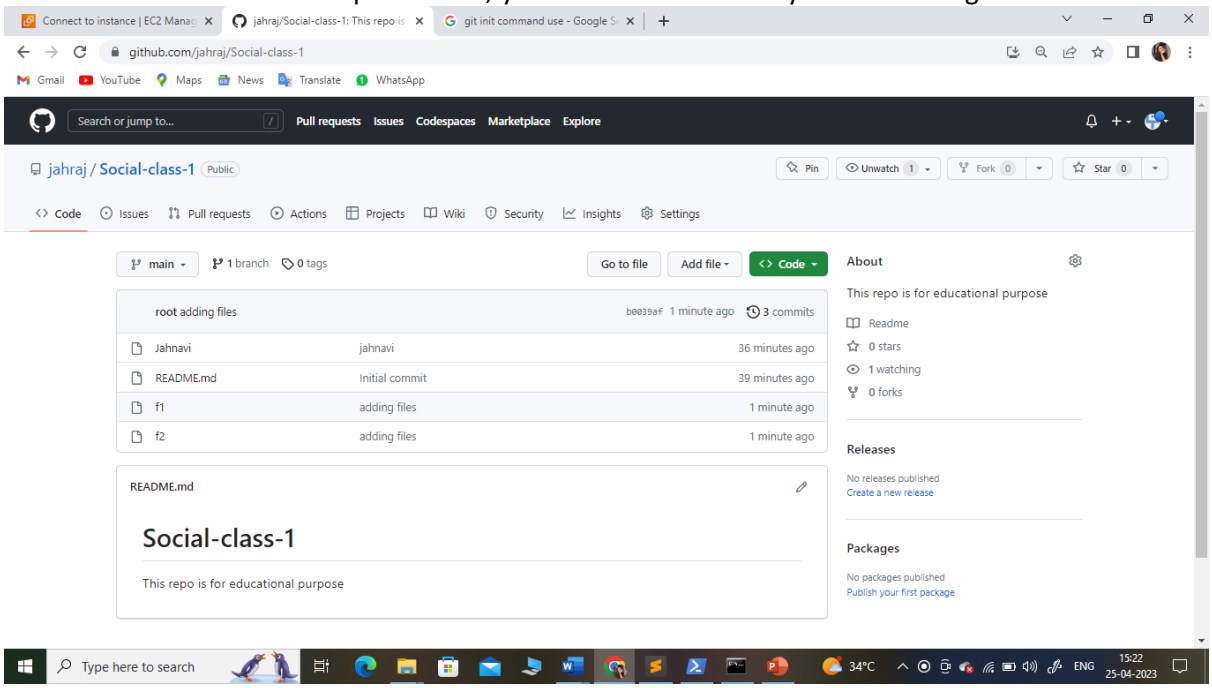


## STEP 4: (Working with Remote repo)

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 commandto 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.

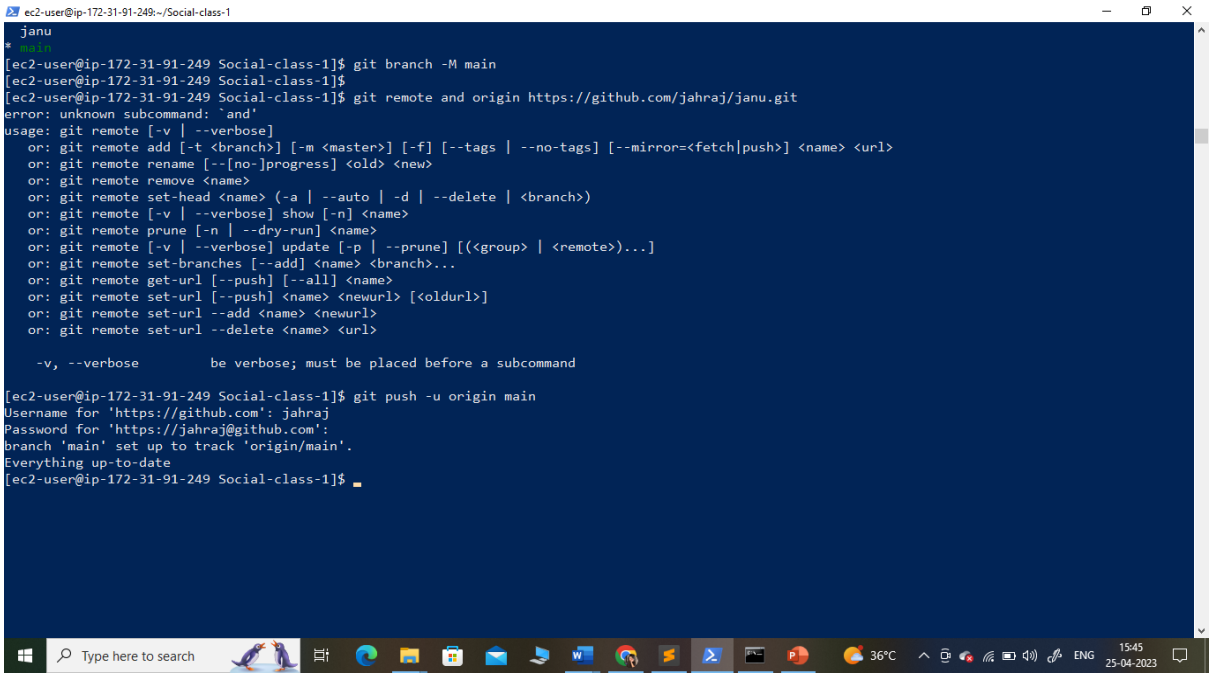



## STEP 5: (Pushing a locally created repo to GitHub)

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

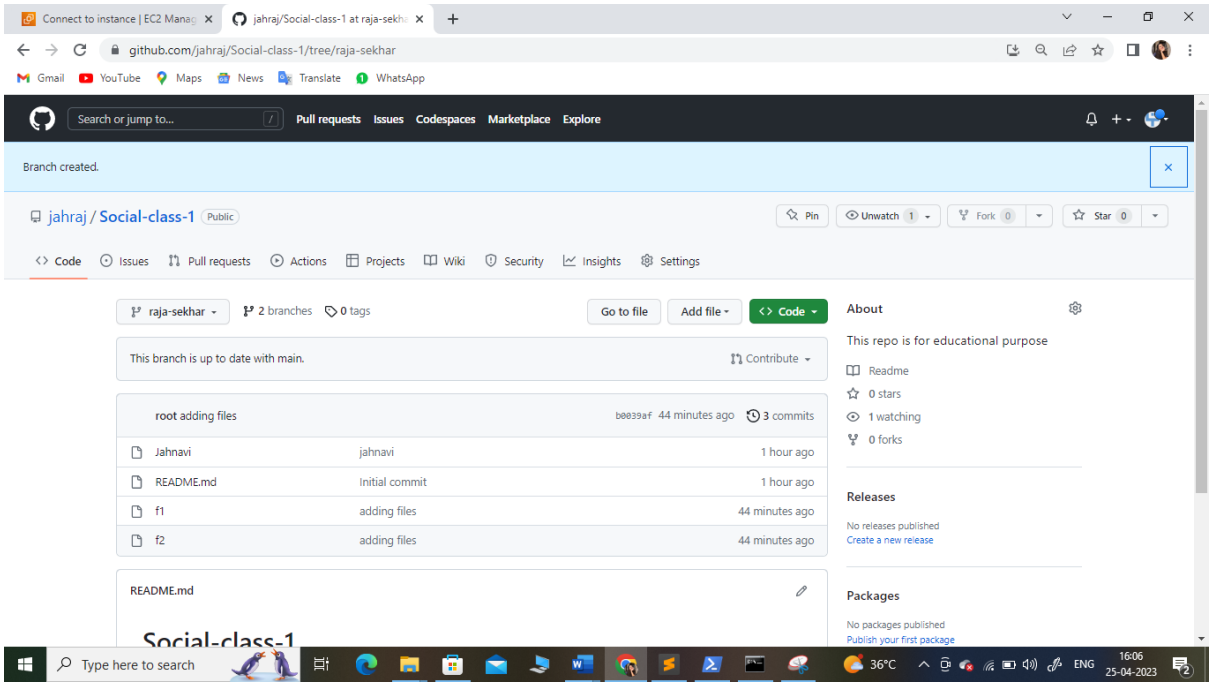
3. Come to your local machine and run the following commands from inside your local repo
    a. git branch -M main
    b. git remote add origin < ghp_sSn6Q1nekIdx9m2TcykOKyA7aFceZb2pyWGk >
    C. git push -u origin main




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

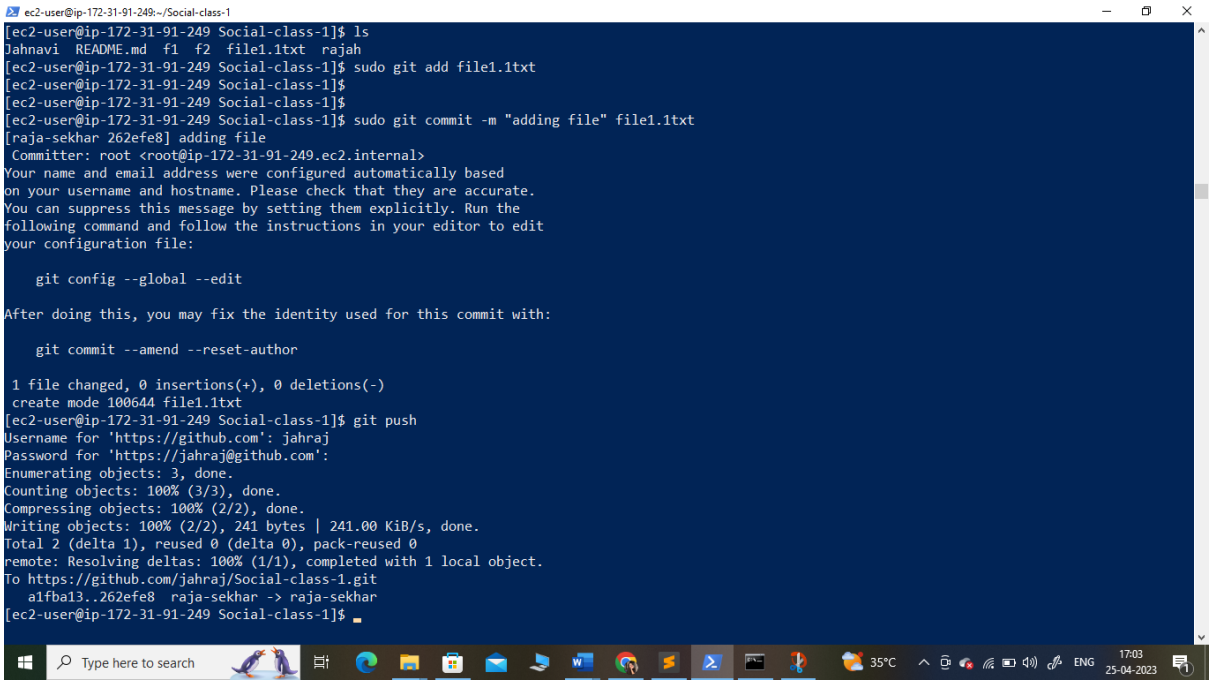
1. Go to the repository at the place of main and click on the branch dropdown.
2. Type the name of your new branch that you want to create and click on create button.
3. A new branch will be created.

- Raja sekhar



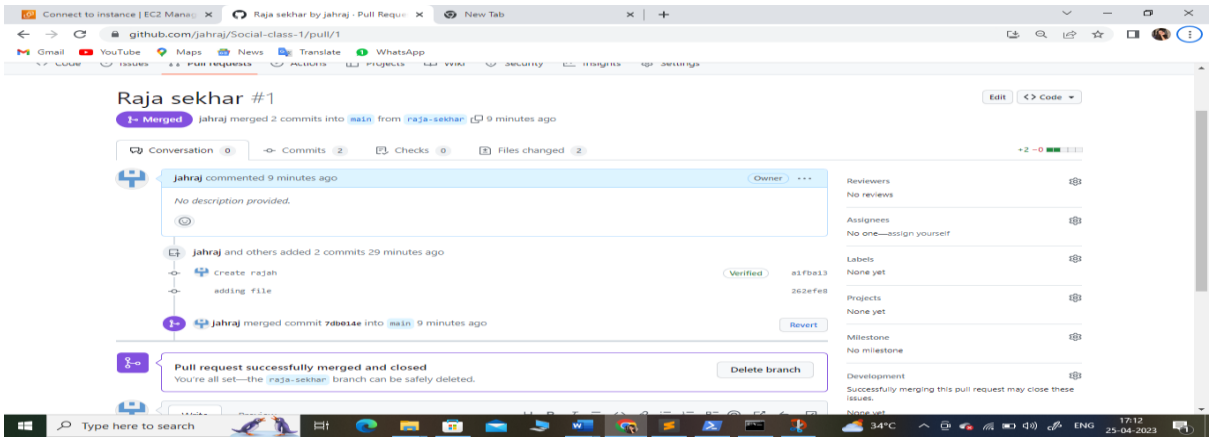

## STEP 7: (Pull all the branches in your local machine)

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



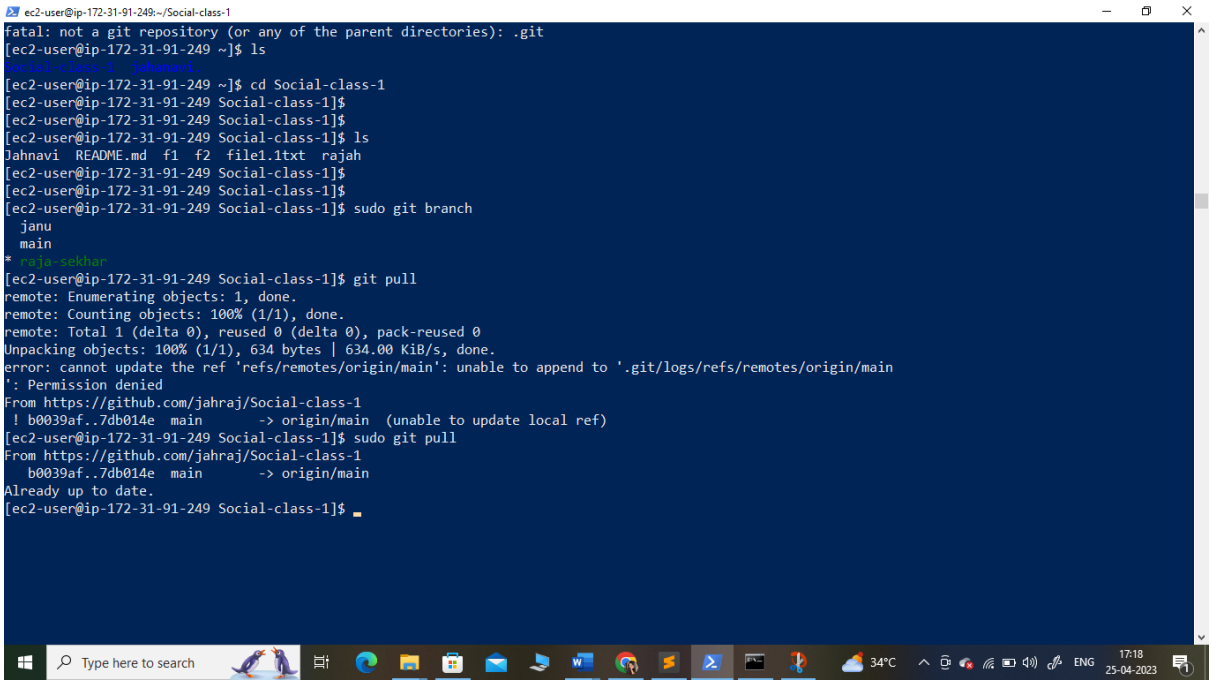

## STEP - 8: (Merge our feature branch with main branch)

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




Thanking you

Raja Sekhar kanuri