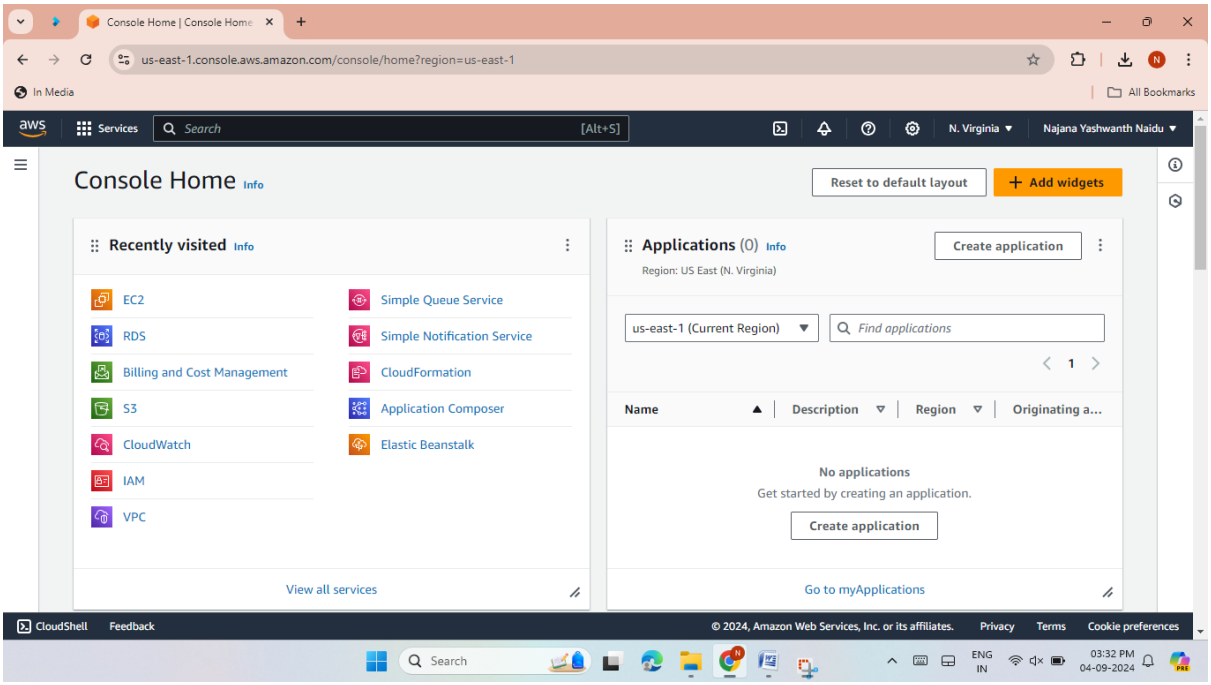
GIT MINI PROJECT

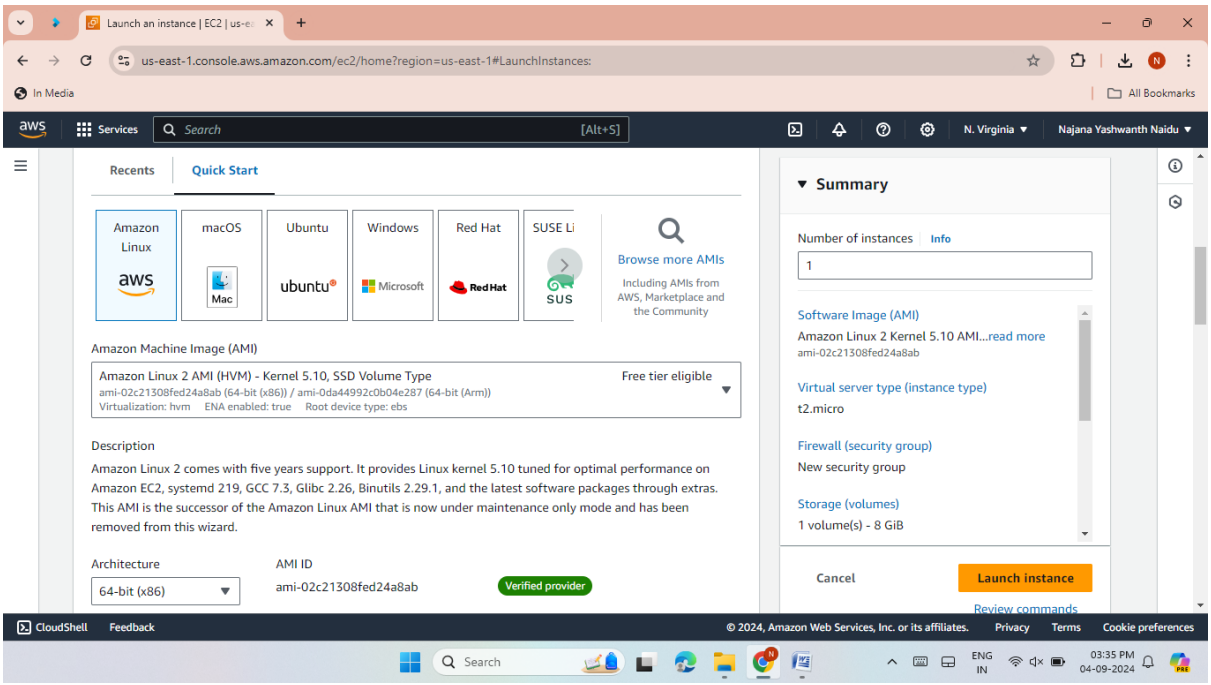
N Yashwanth Naidu

LAB 1: CREATE EC2 INSTANCE

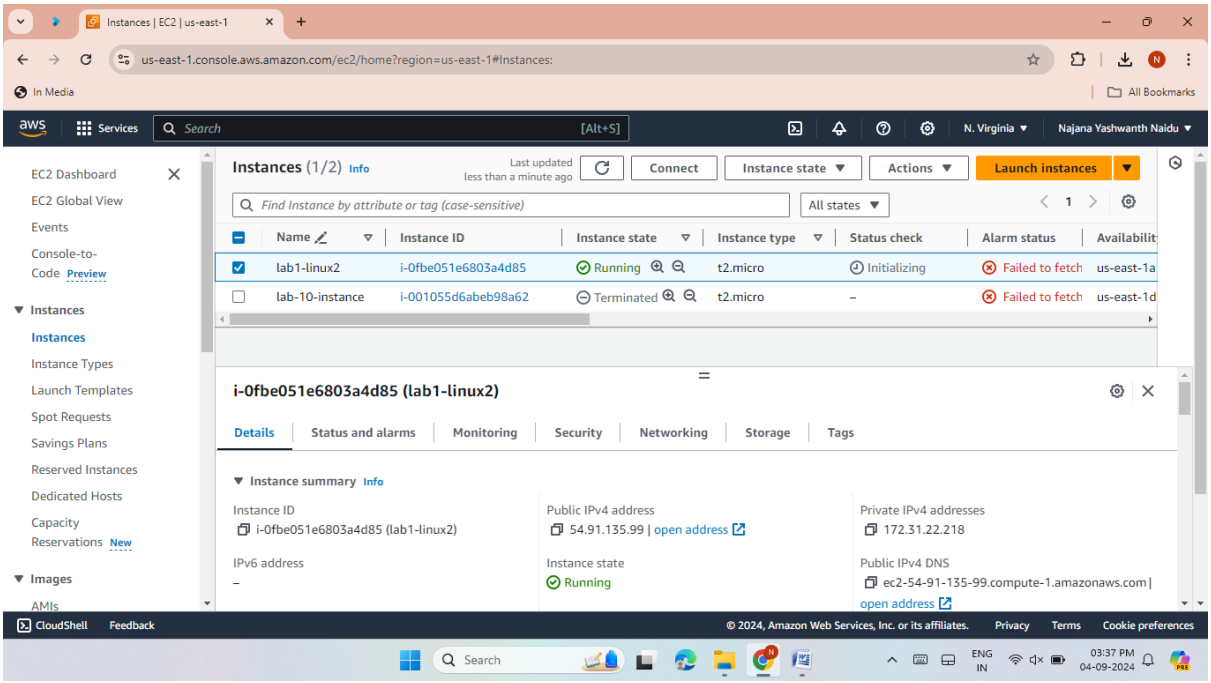
- login to AWS console.



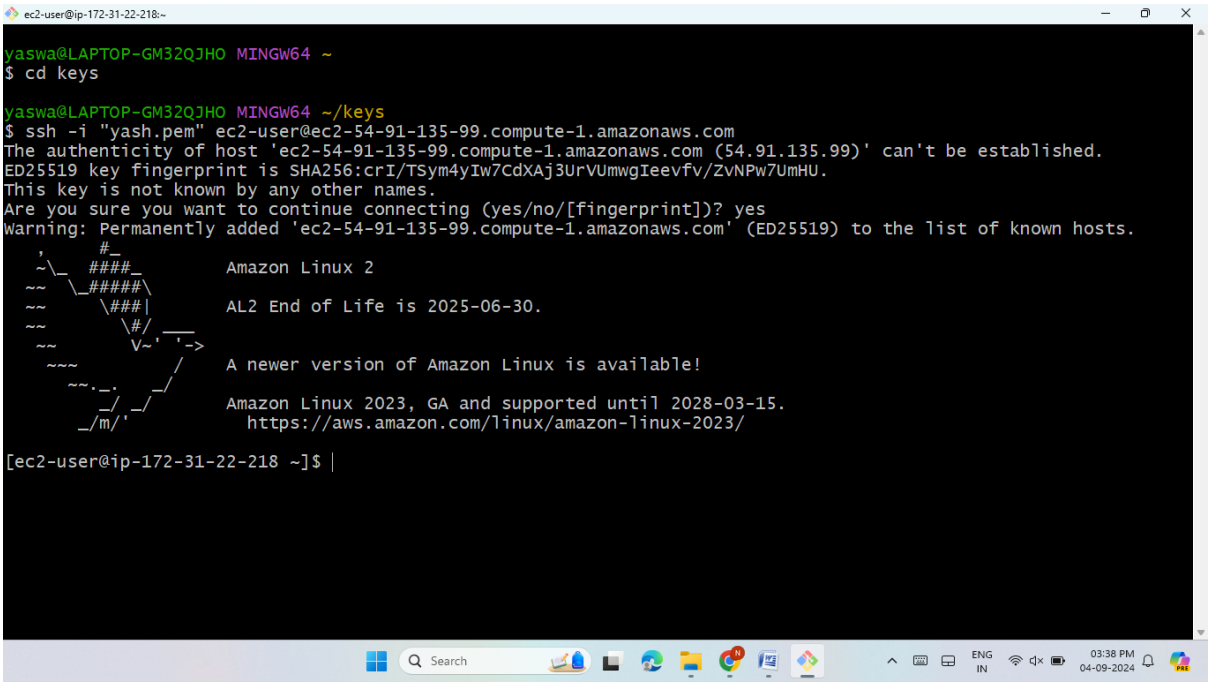
- Create a server with Amazon Linux 2 AMI/RHEL/UBUNTU.



➢ Here the instance was launched.

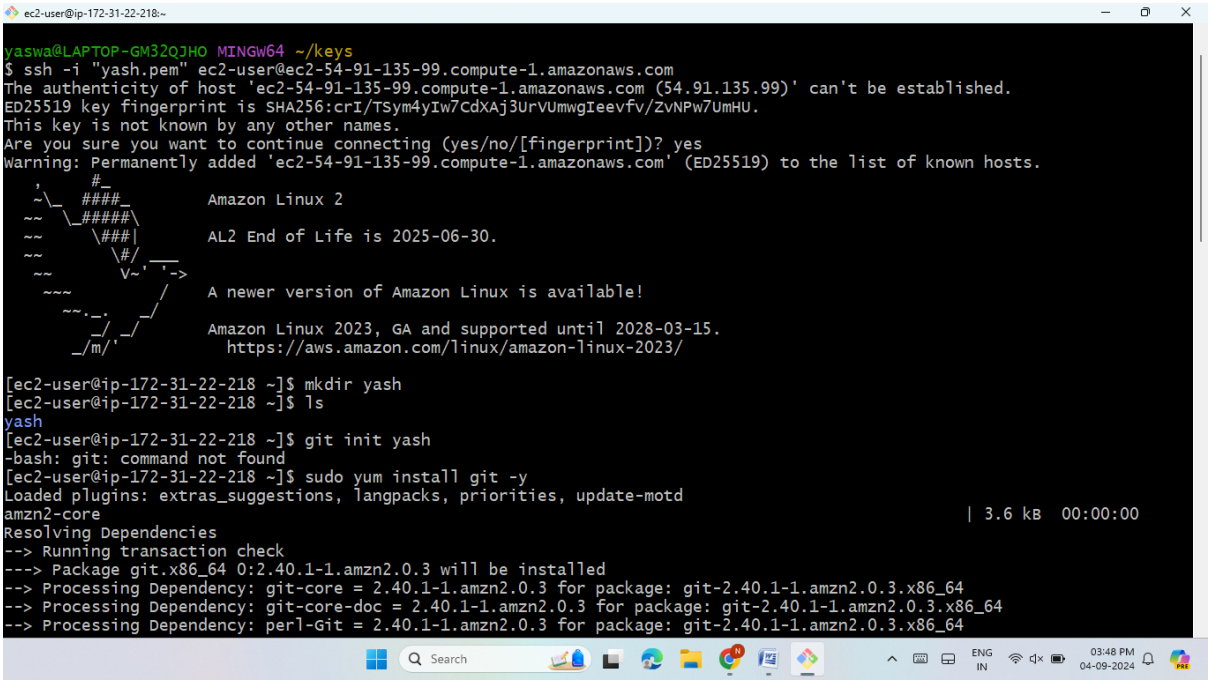


➢ Connect the server only with putty/Gitbash.

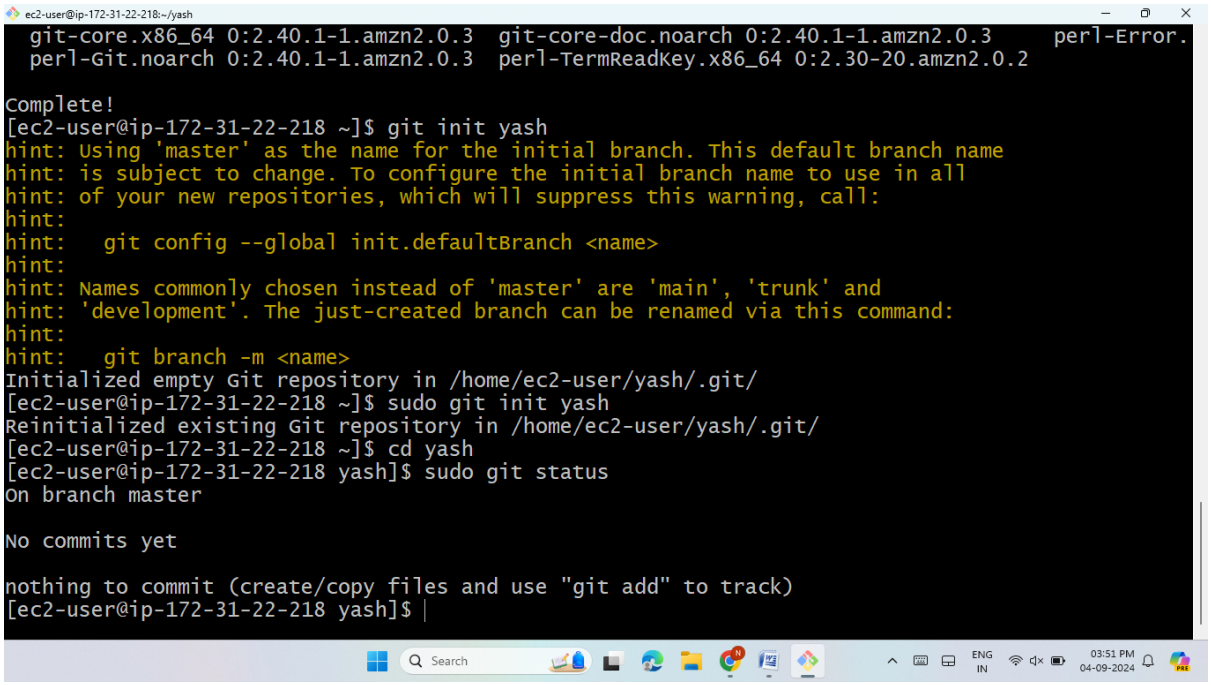


LAB 2 (CREATE REPO IN LOCAL MACHINE)

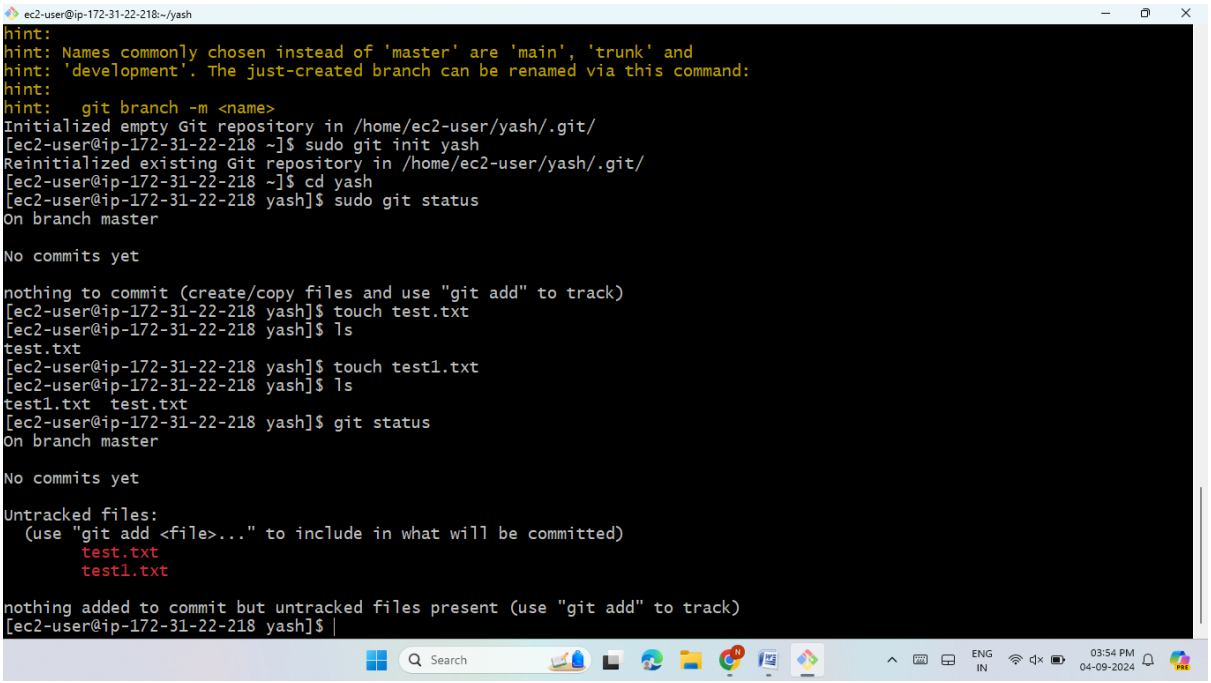
- Create a folder on my local machine using the command **mkdir**.
- Install Git in local machine using the command sudo **yum –y install.**



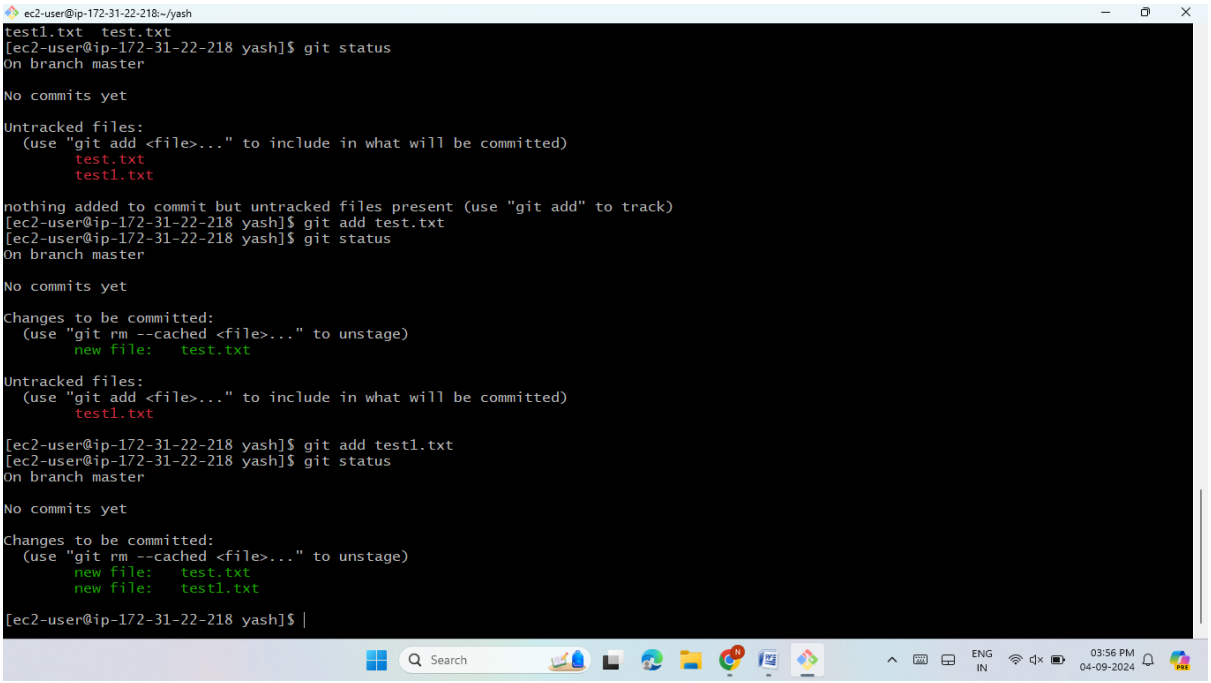
- Initialize this folder using **git init** command.
- Go inside this folder and run **git status** command to check the status.



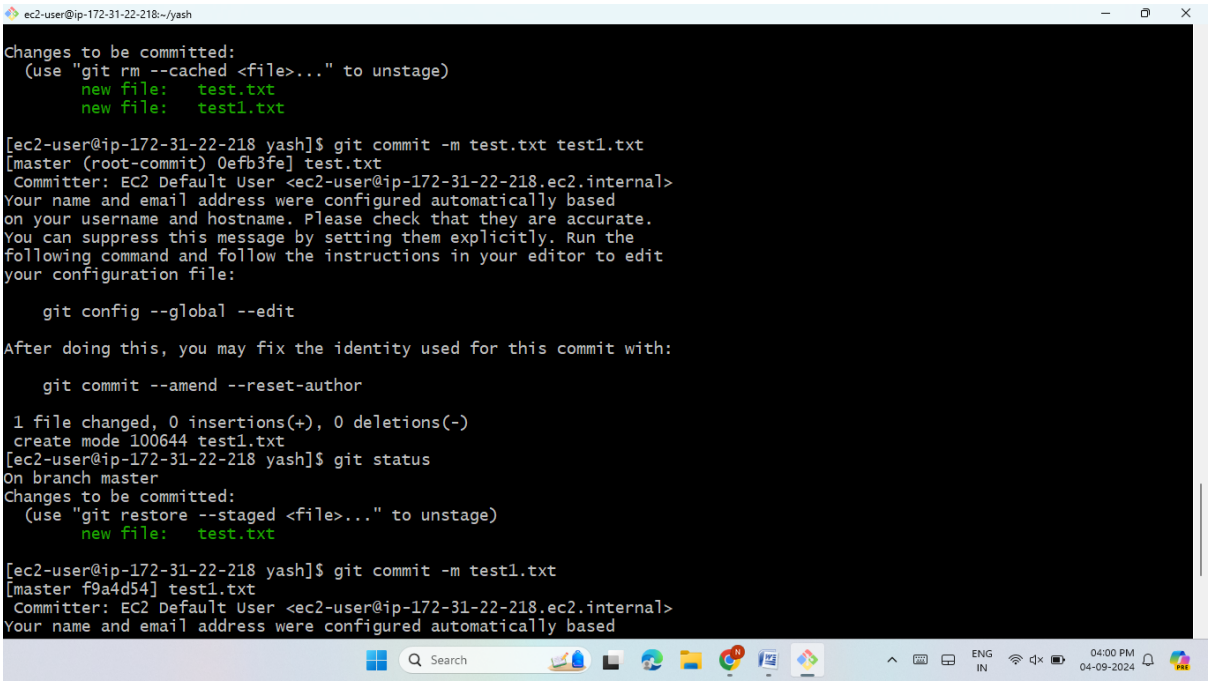
- Created some empty files using **touch** command.
- Again, run **git status** to see the changes, you can notice that the file is available but not tracked by Git.



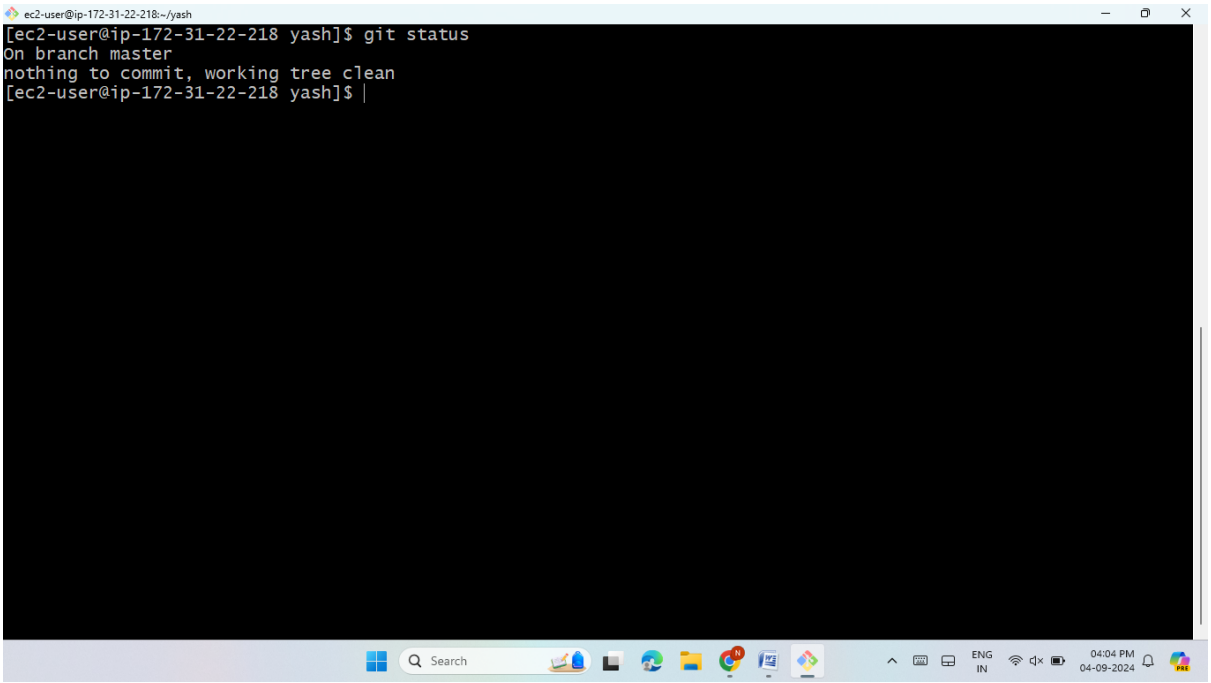
- Run **git add <filename>** to stage this change (git will start tracking this file), you may check is using **git status** once again.



- Now commit our changes by running **git commit –m “added test files”.**

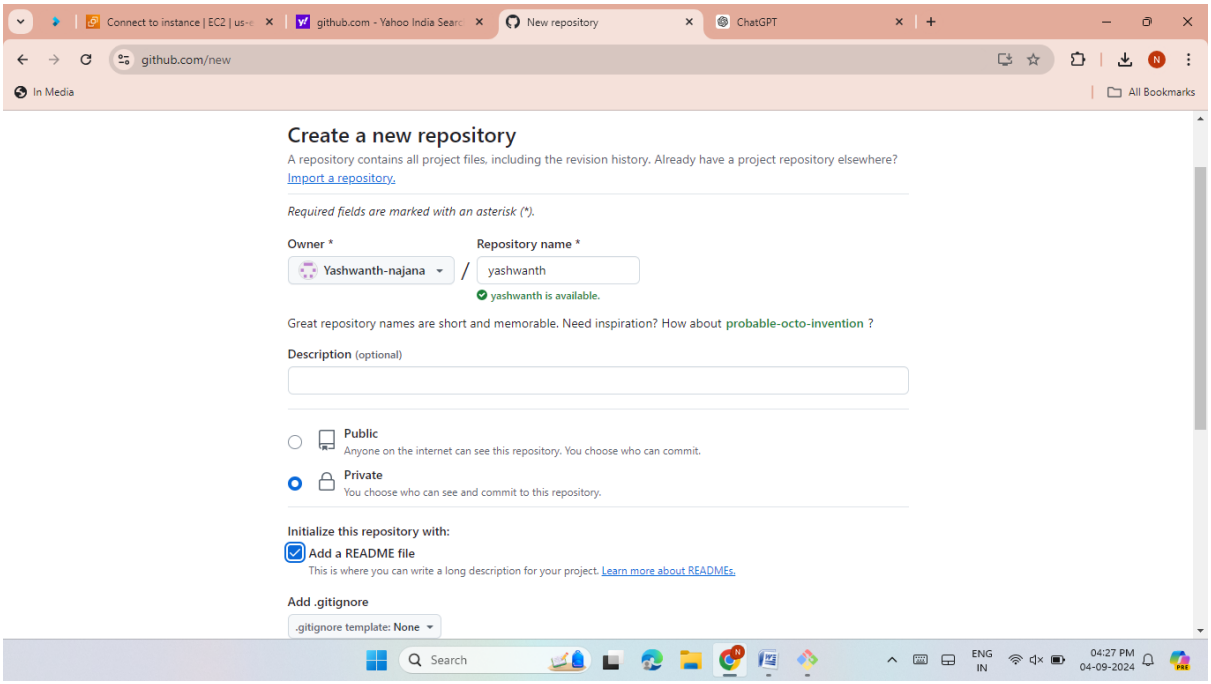


- Run **git status** once again and it will show you that the working tree is clean.

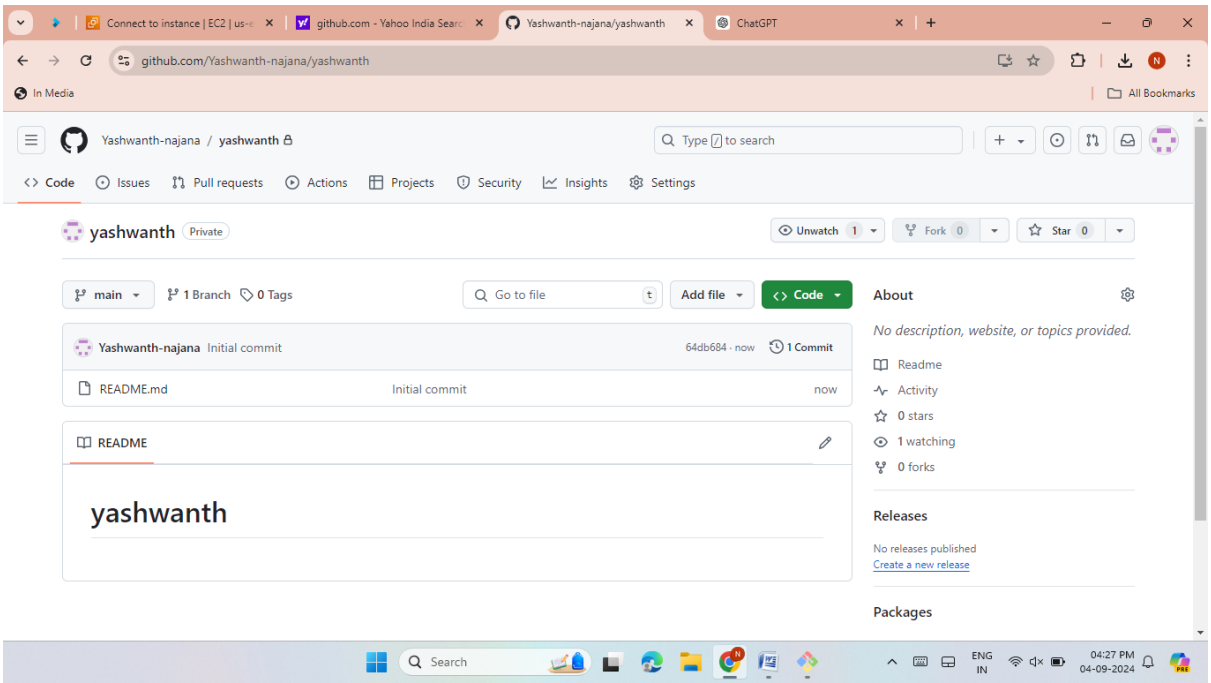


LAB 3 (CREATING REPO IN REMOTE LOCATION – GITHUB)

- In remote location – GitHub.
- Create a new repository by clicking on new button.
- Provide repo name.
- Select whether it is a private or public repo. (I selected private)
- Initialize the repo by adding a README.md file.
- Click on create repo and done.

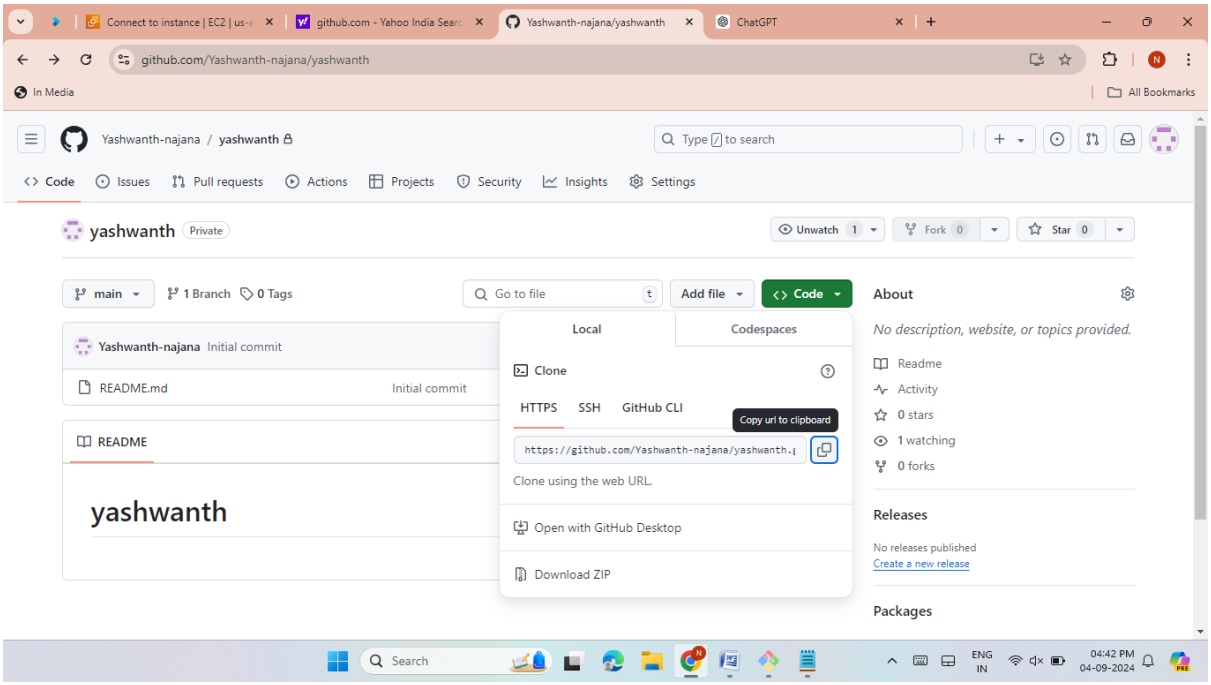


- Here the new repository is created.

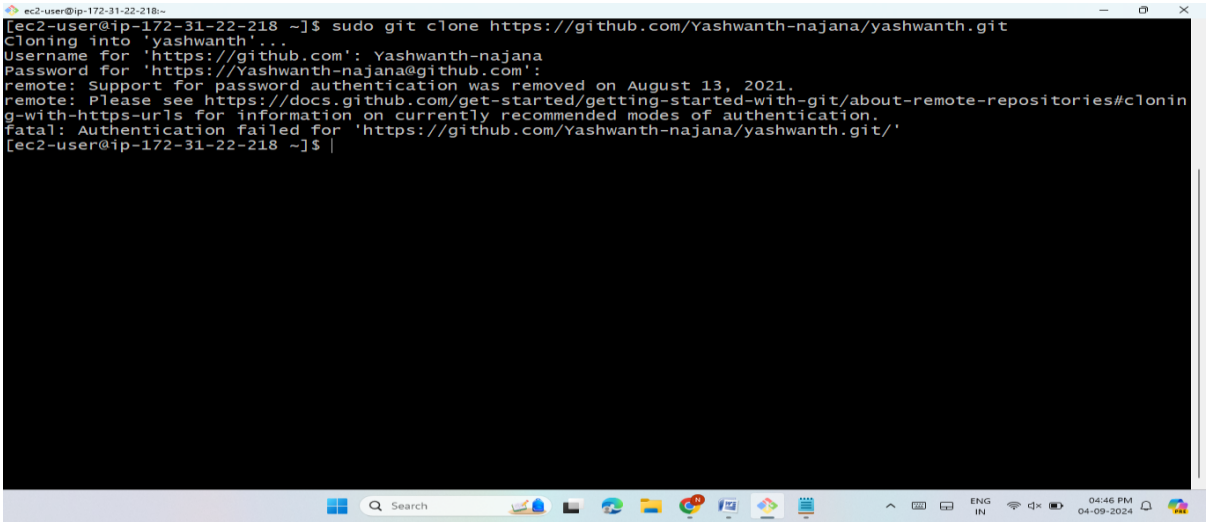


LAB-4 (WORKING WITH REMOTE REPO)

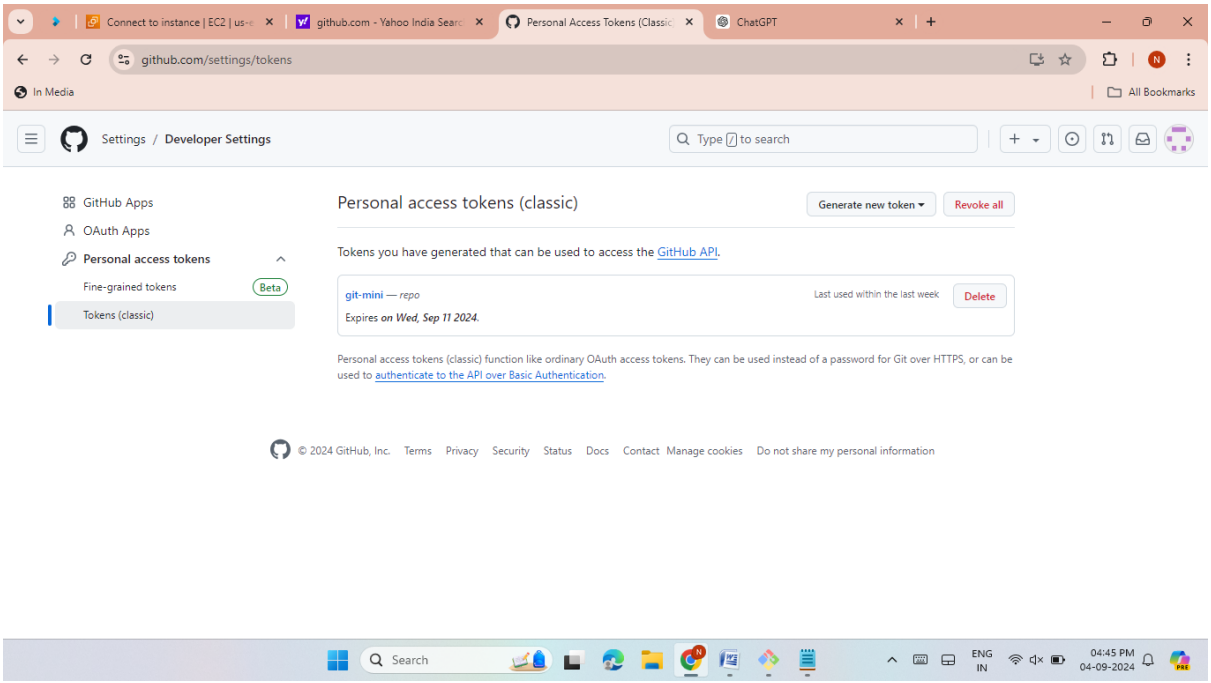
- Take the remote repo to your local.
- Machine make the changes.
- Commit the changes.
- Push the changes to remote.
- Pick the clone URL of the repository from the Github repo.



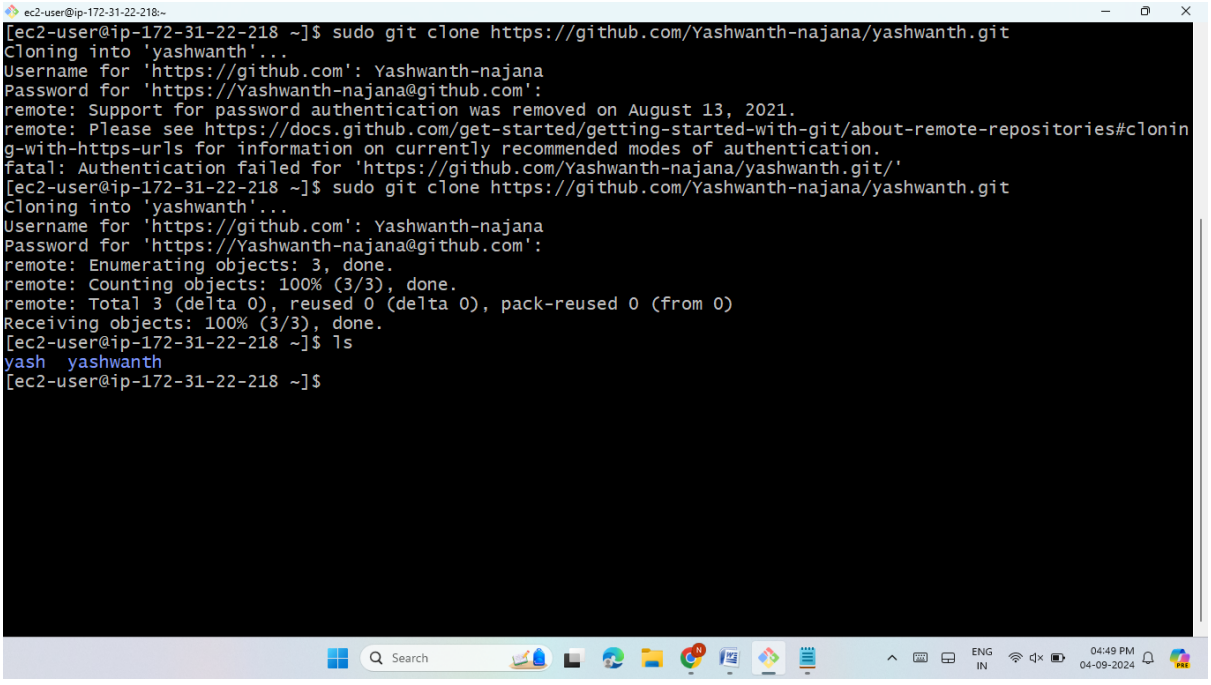
- Go to your local machine and clone this repo using git clone command.
- you will be asked to provide username and password.
- Here you will get one error for password based authentication depreciation.



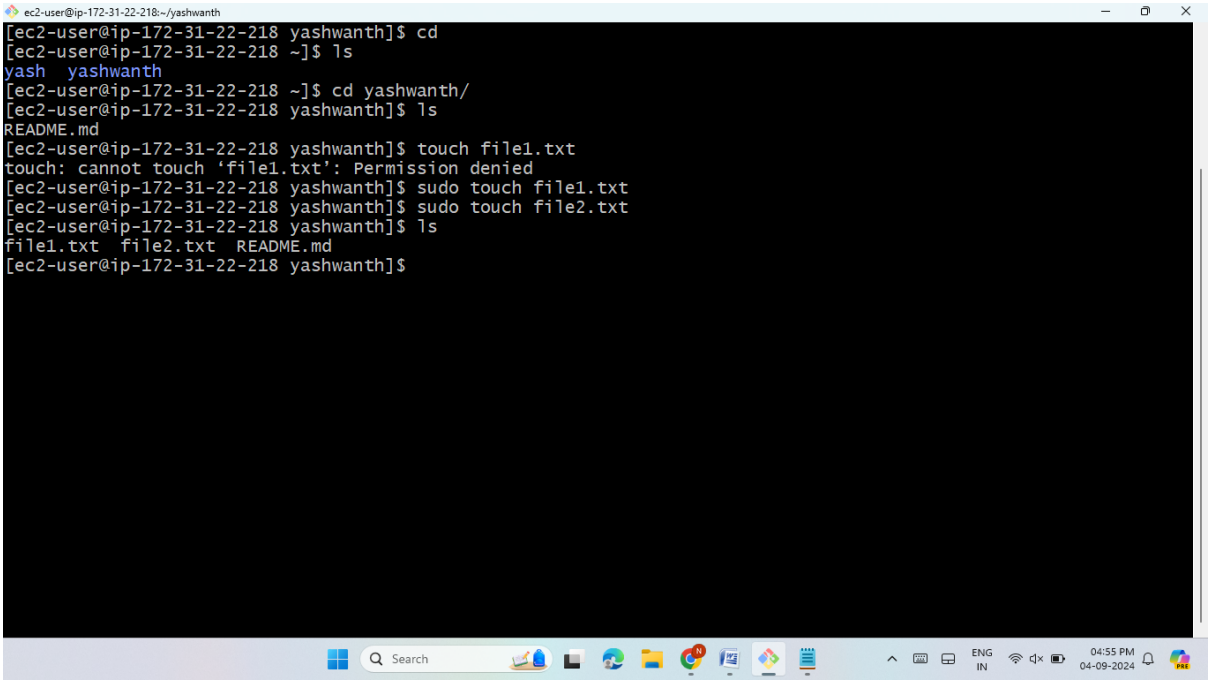
- Now , we need to create the personal access token to work with this repo.
- Go to settings →go to developer settings → go to personal access token→ select tokens (classic) → select generate token.



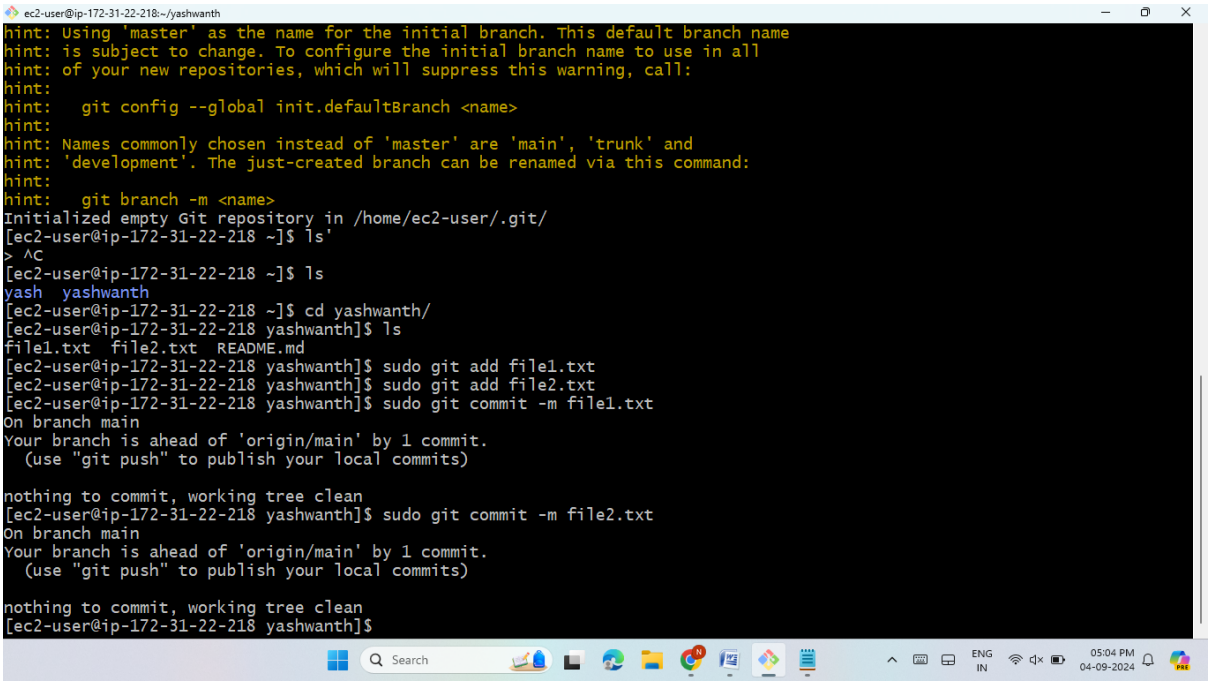
- Clone the repo again in your local machine, and this time provide the token in place of password while cloning.
- It successfully cloned the repository.



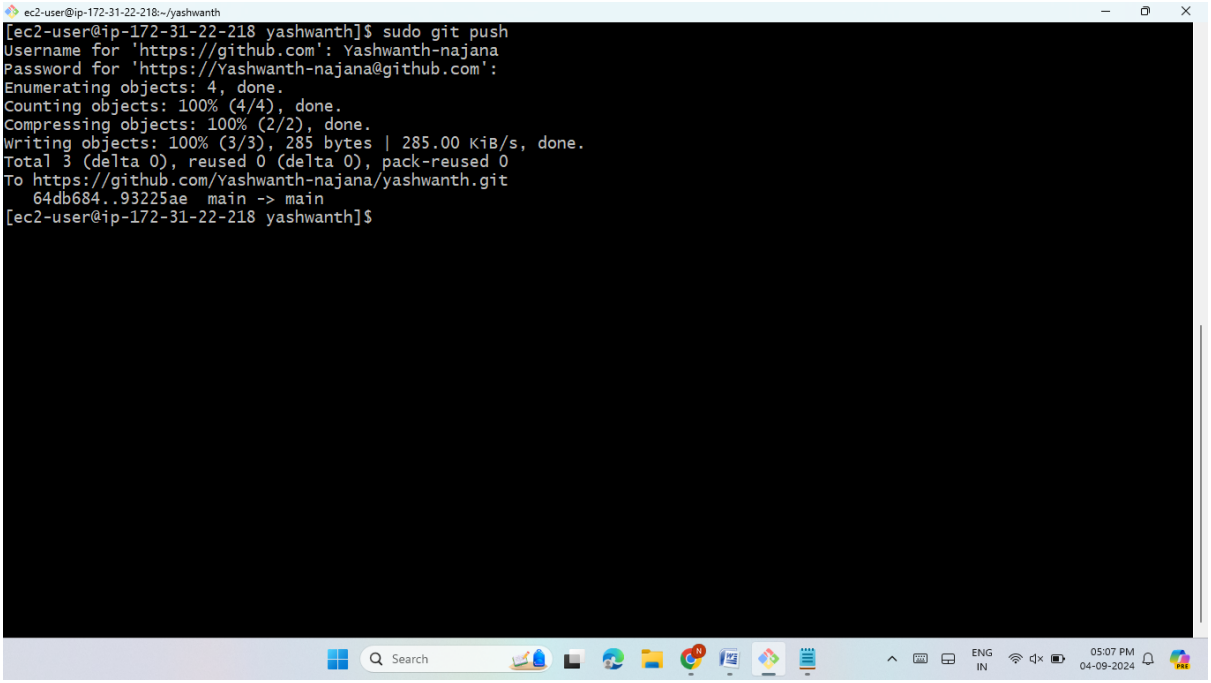
- Once cloned, go to the repo folder and add some sample files .we can use **touch** command to create empty files.
- First I gave **touch <u>filename</u>** but it shows permission denied.
- So we must give **sudo** before touch.



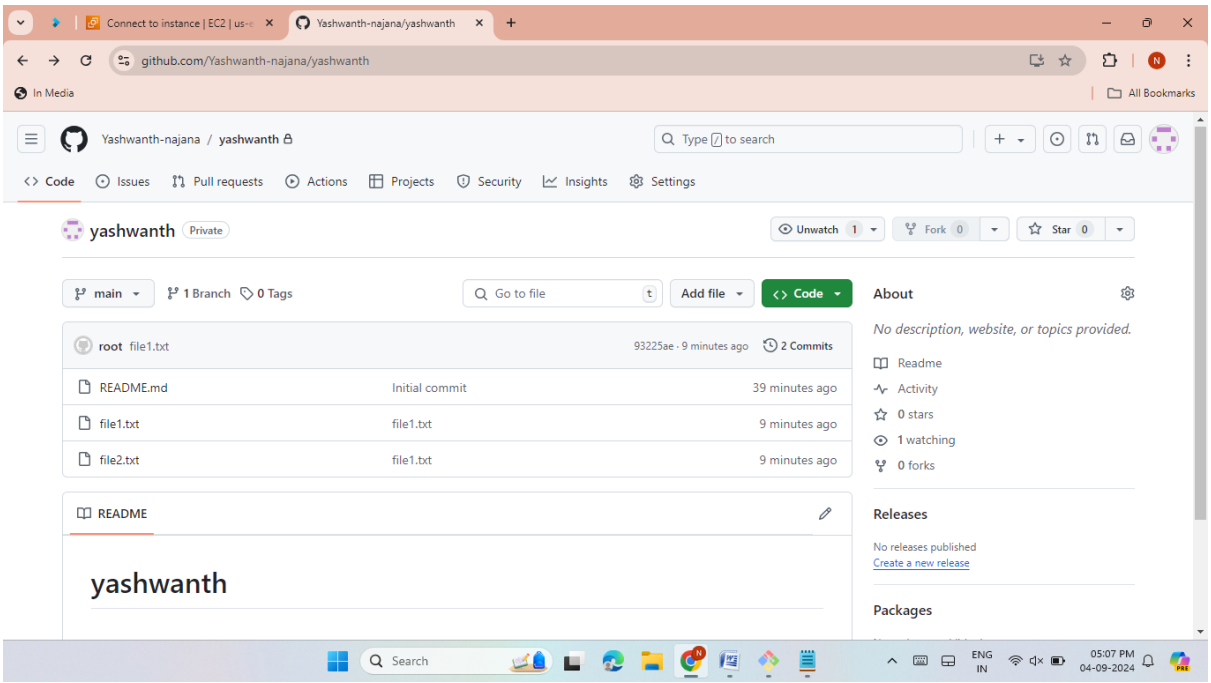
- Stage these changes by running **git add file1 file2.**
- Commit these changes by running git **commit – m "<any message>".**
- Check the status using **git status** command. It shows working tree is clean.



- **git push** → it will ask you for username and password , provide the username and personal access token in place of password which we created before.

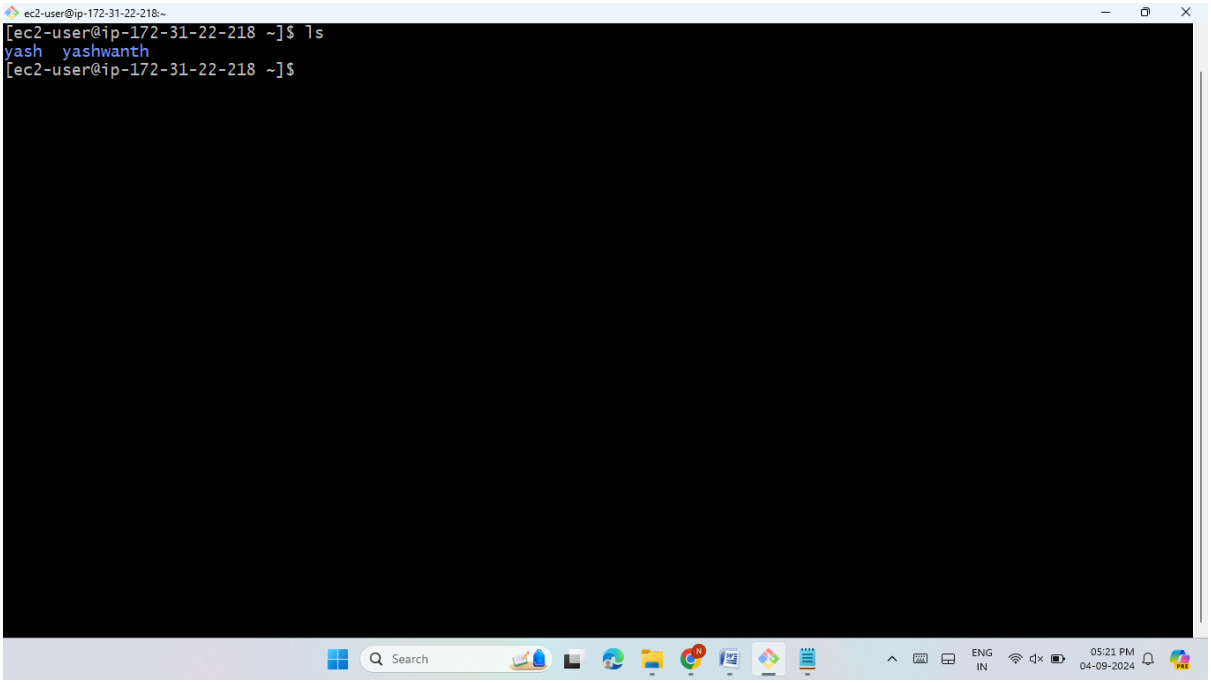


- Go to the remote repo and see , you will be able to find your new changes.
- The file1 and file2 is added in remote repository.

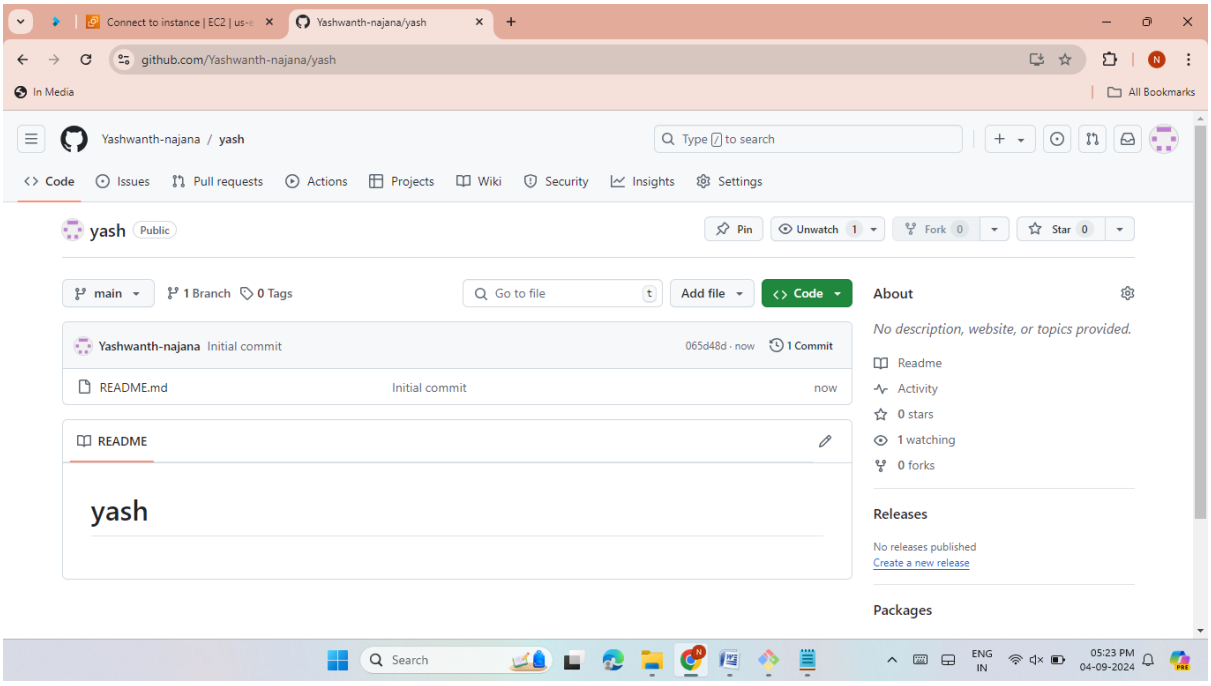


LAB 5: (PUSHING A LOCALLY CRATED REPO TO GITHUB)

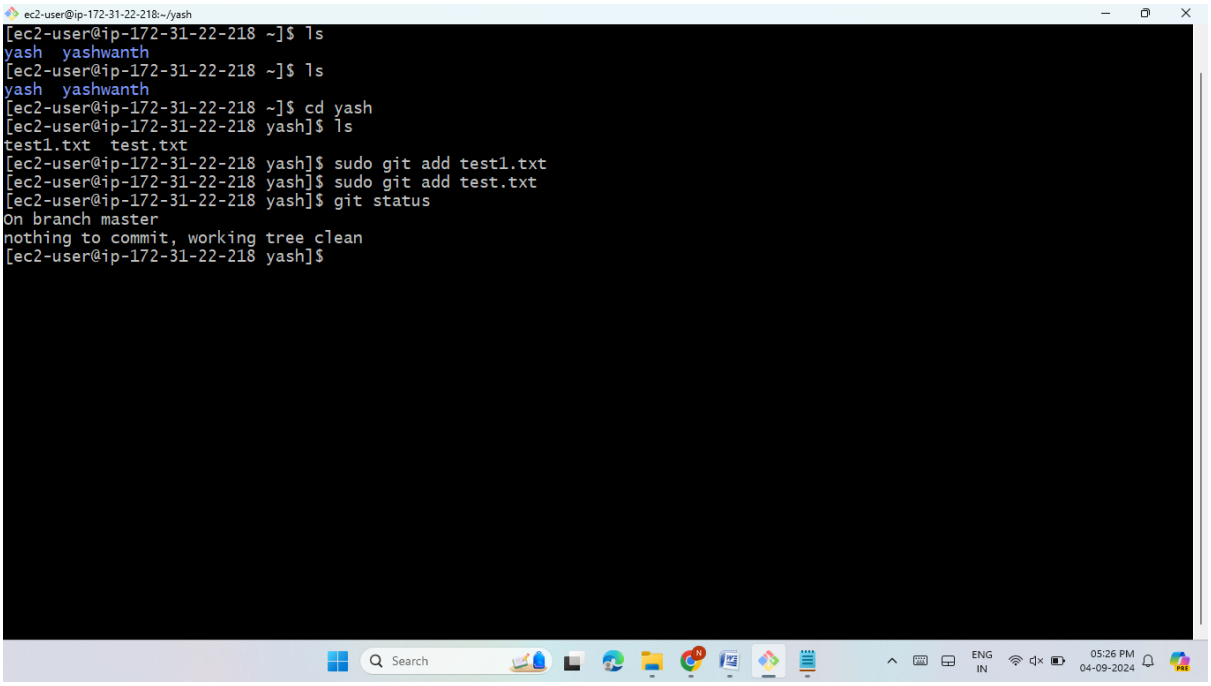
- Created one repo in our local machine and initialize is locally. (we have done it in lab -2)
- The new repository is yash.



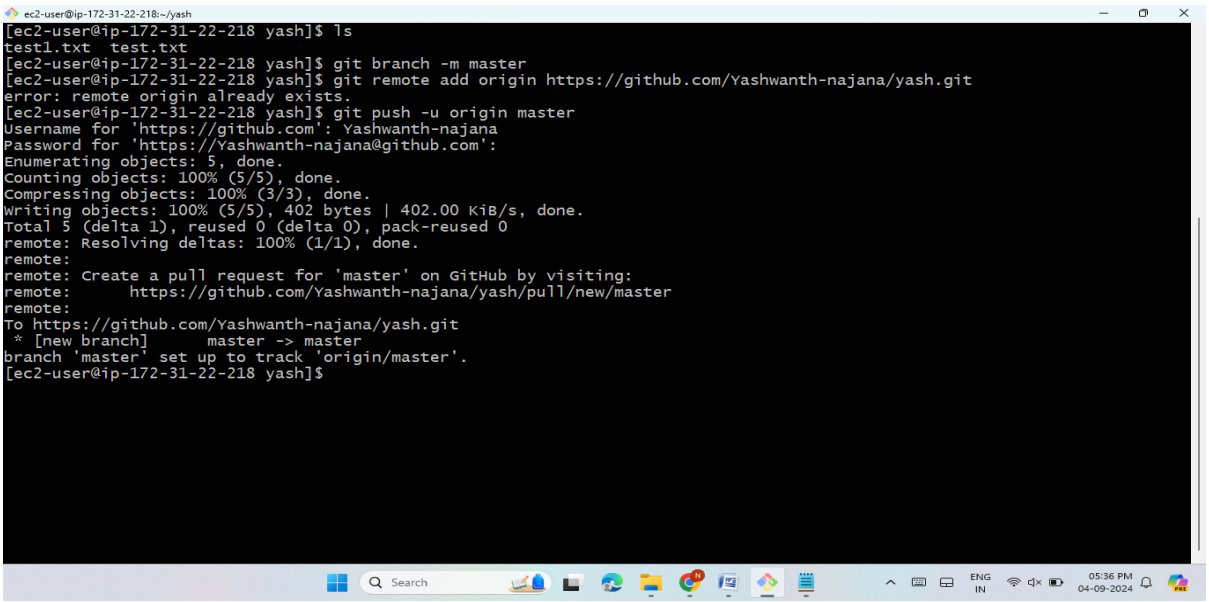
- Create one remote repo with the same name as local repo in github and do not initialize it.
- Click on new and then create new repository.
- The new repository name yash.



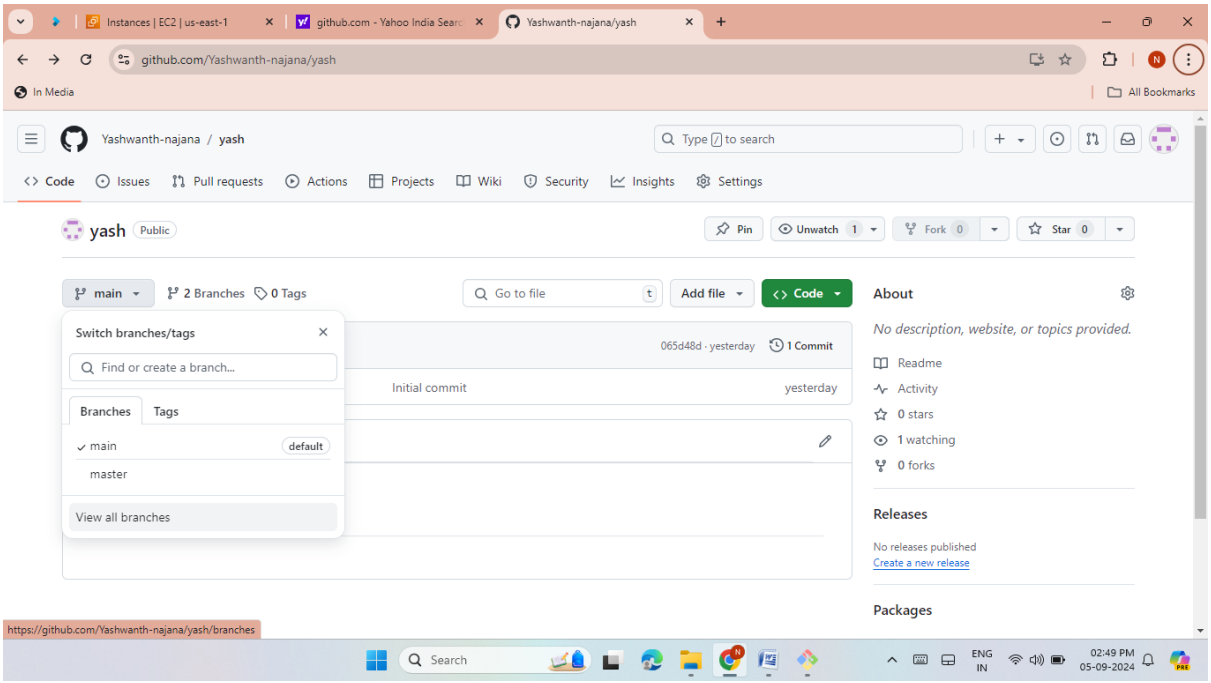
- Come to your local machine and the following commands from inside your local repo.
    - **git add <file name>**
    - **git commit –m <file name>**
    - **git status**
- It shows nothing to commit, working tree is clean.



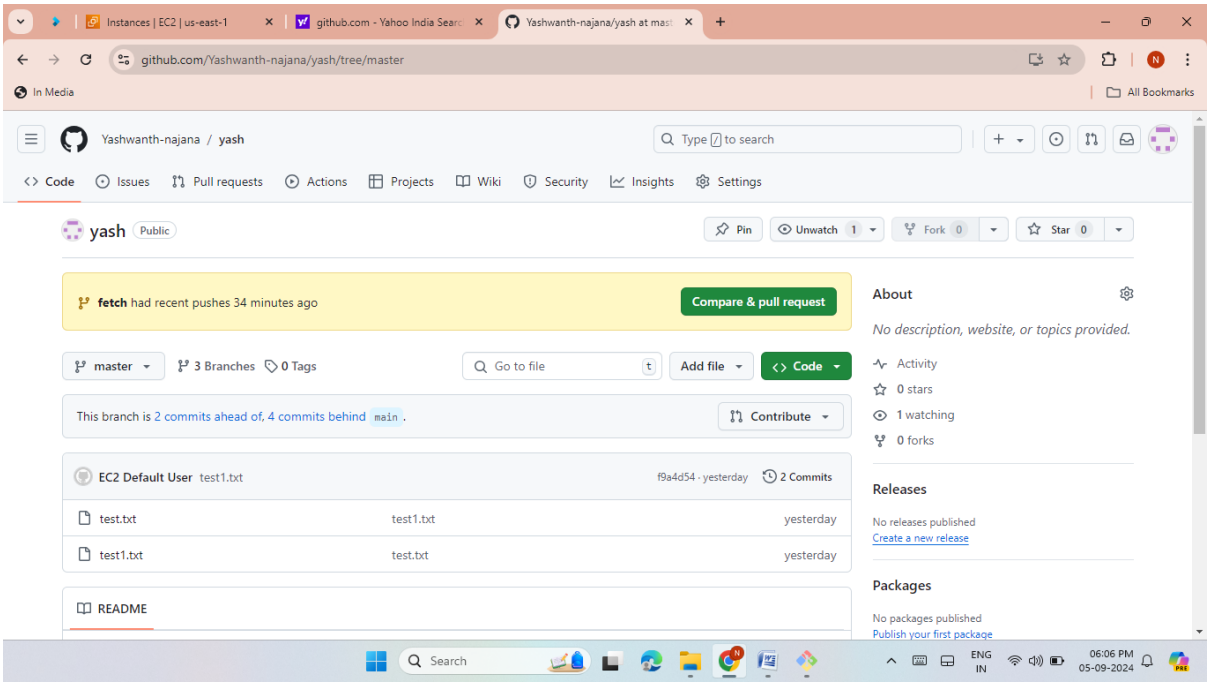
- **git branch –m master** (to change the name of branch as master branch)
- **git remote add origin<URL of your remote repo>**
- **Git push –u origin master.**(to push your local branch to remote repo)
- It will ask you for username and password, provide the username and personal access token in place of password which we created before.



- Here the master branch is created.

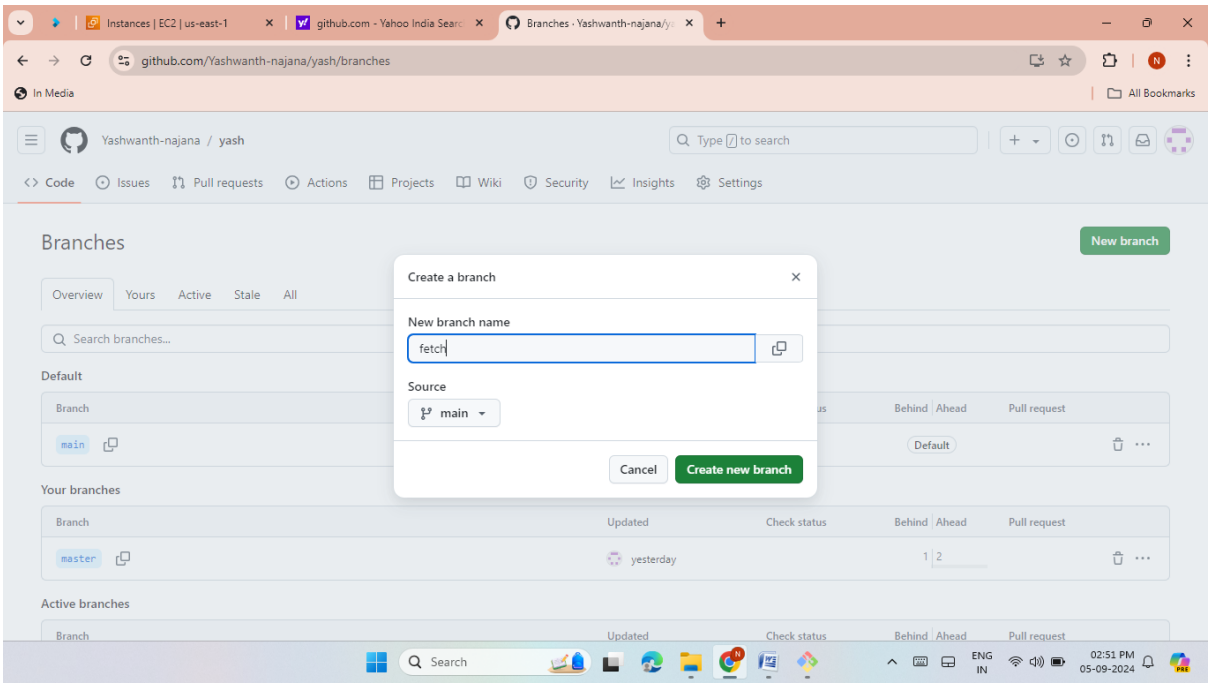


- Here the files also pushed.

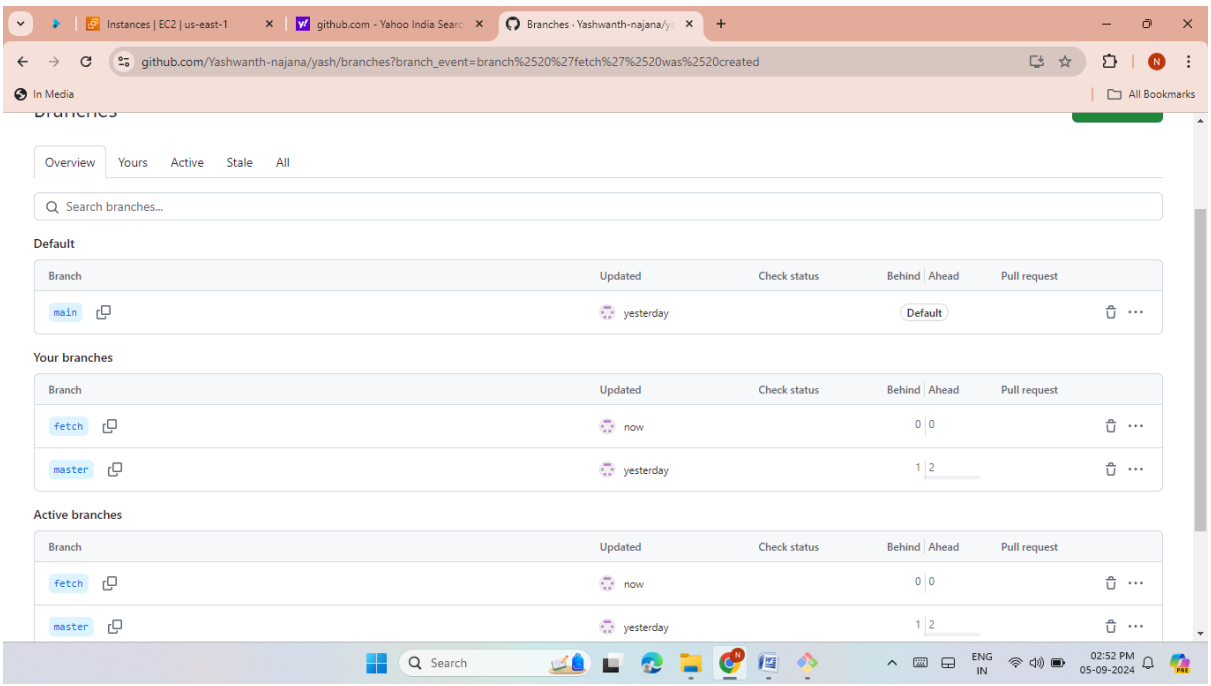


LAB 6 (CREATING A NEW BRANCH FROM YOUR MAIN BRANCH)

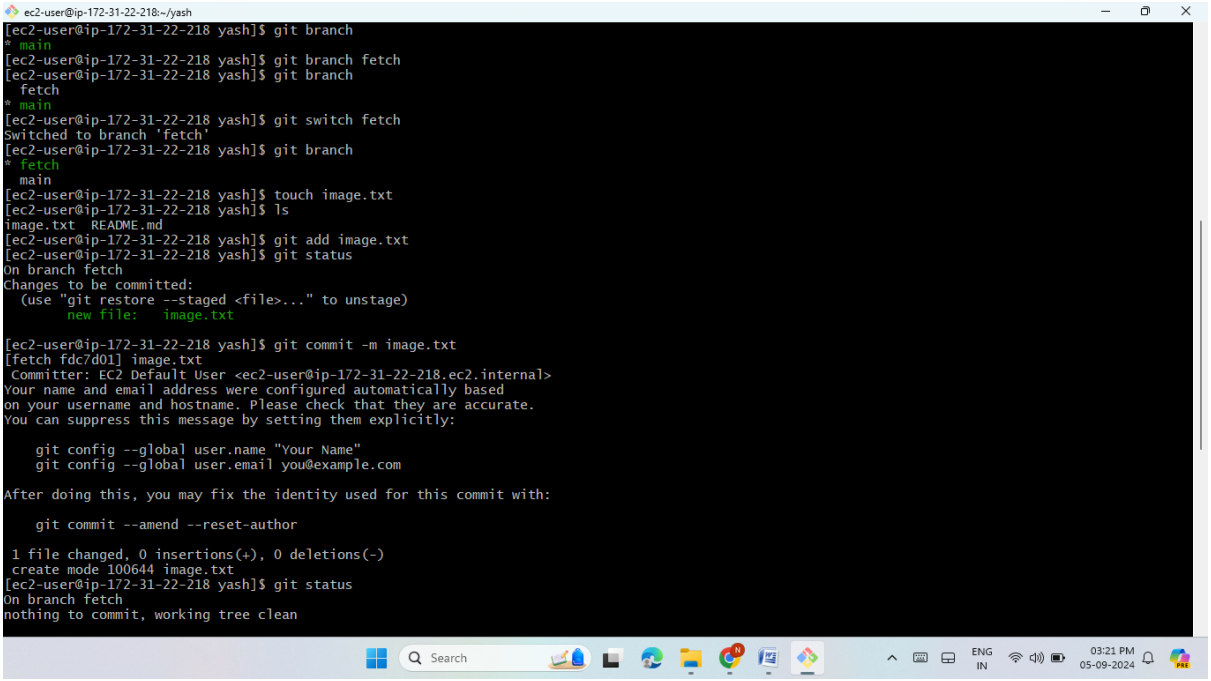
- Go to the repo at the place of main and click on the branch dropdown.
- Type the name of your new branch that you want to create and click on create button.
- Here the fetch is the new branch.
- A new branch will be created.

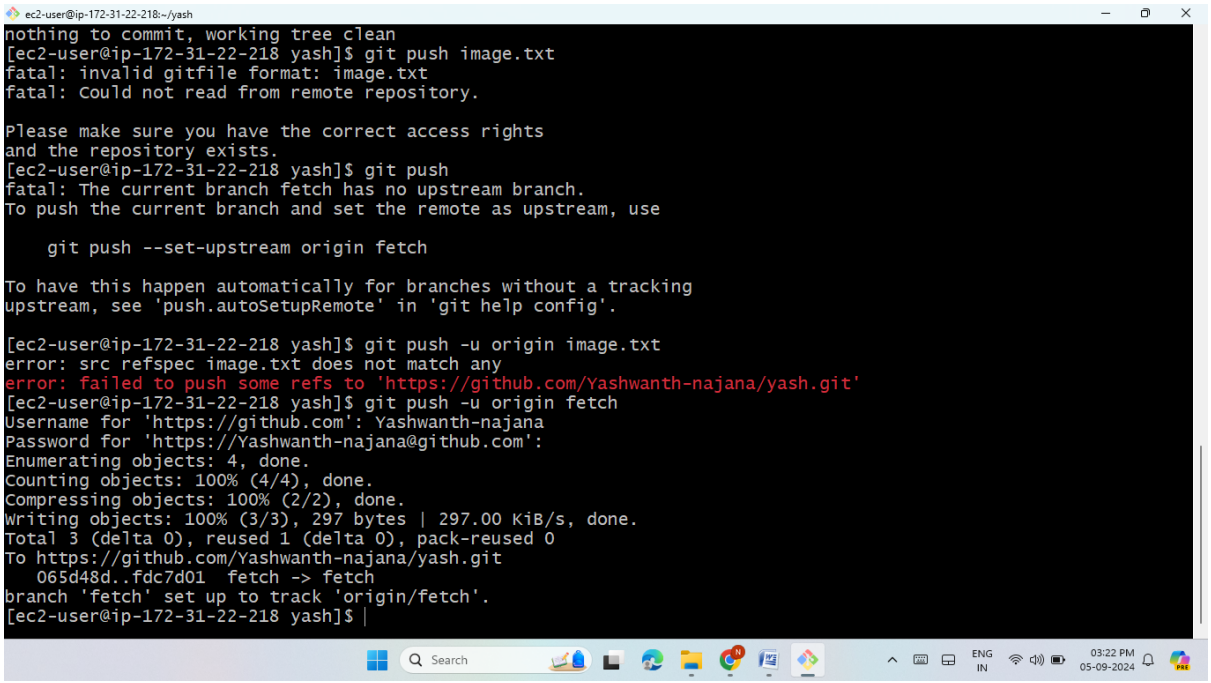


- A new branch will be created.

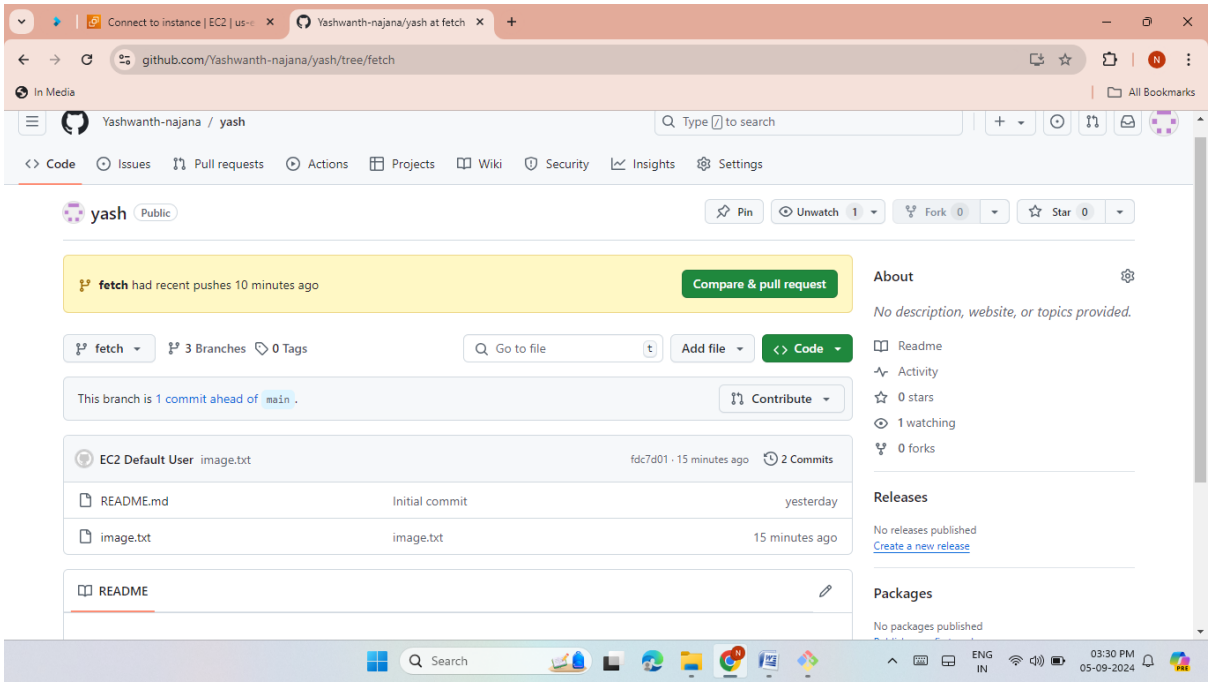


- Make some changes in this branch directly from the console.
- Here you will see that your changes are only applied to your new branch but not to the main branch.
- First clone the repository then check the files.
- Run the following commands.
    - **git branch**
    - **Git branch fetch** (to get the branch fetch)
    - **Git switch fetch** (to change the branch main to fetch)
- Create some files using the **touch** command.
    - Here I created image.txt file.
- **git add and commit** the files
- Then check the **status**, it shows nothing to commit, working tree is clean.
- **git push <file name>** (it shows error)
- **git push** (it shows error)
- **git push -u origin <branch>.** (we must use this command to push the files from local to remote)

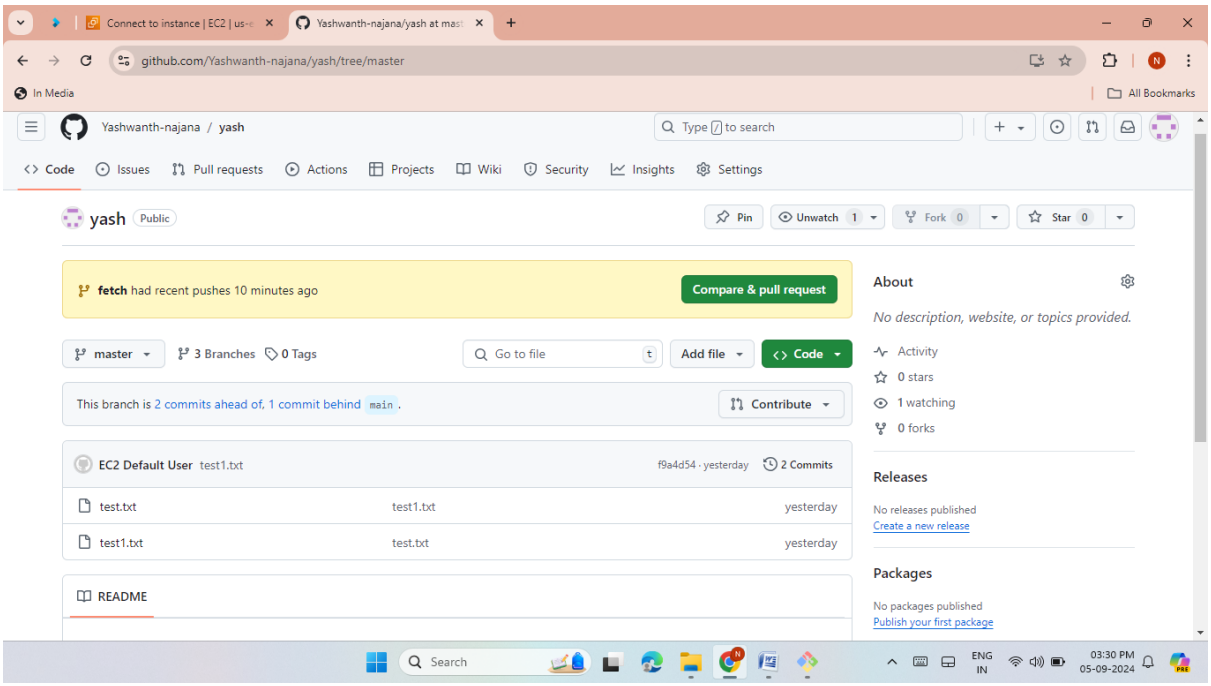




- Here the fetch branch got the file (image.txt).

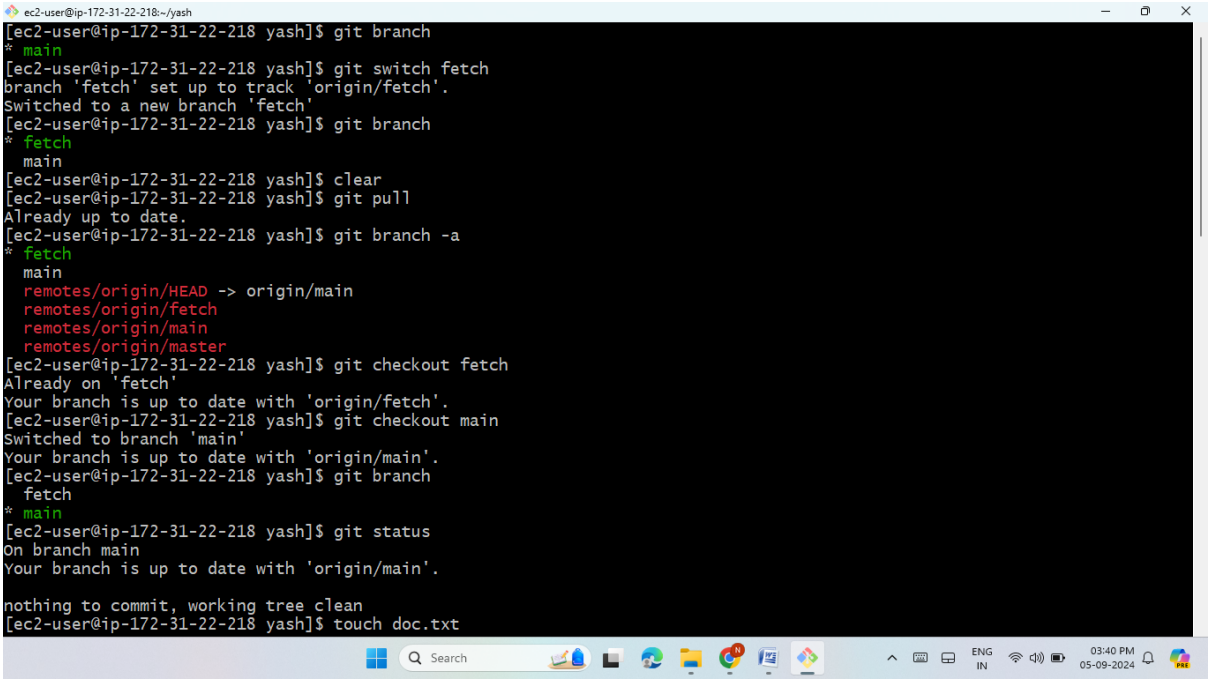


- We can check remaining branch also. Only fetch branch got the file. Remaining branch have no changes.

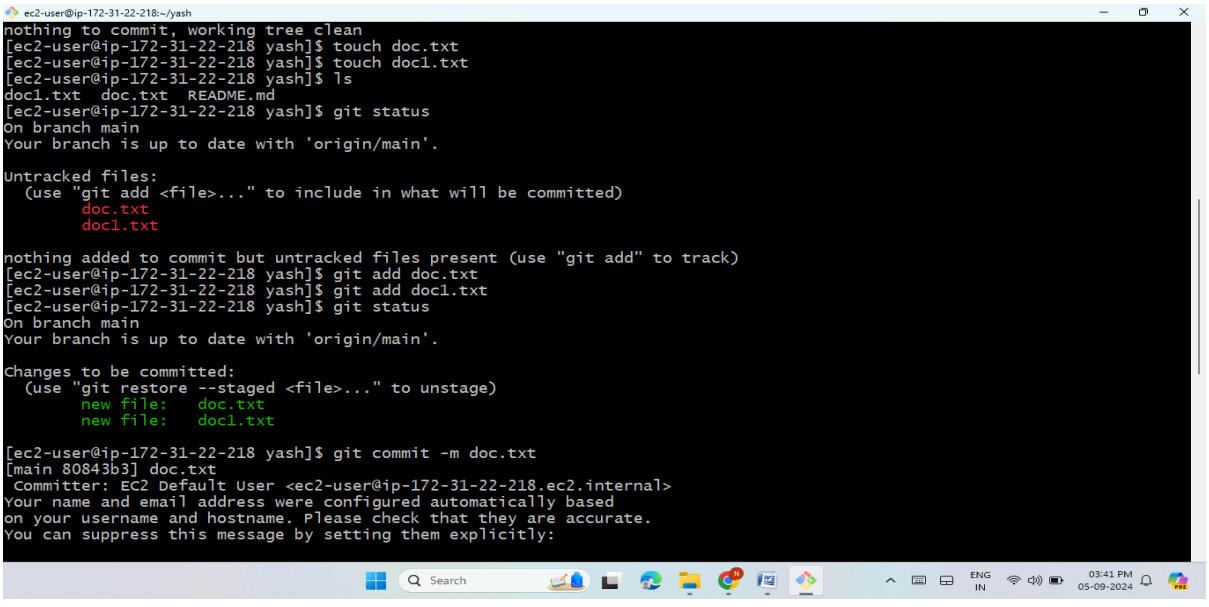


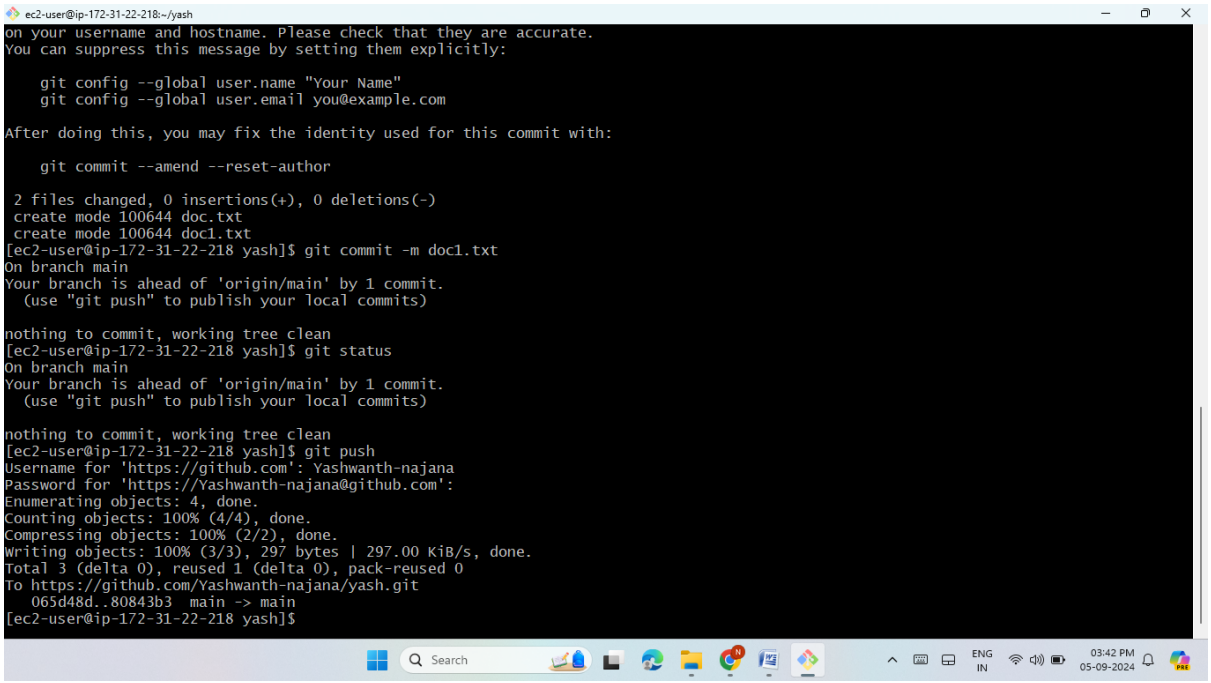
LAB 7 (PULL ALL THE BRANCHS IN YOUR LOCAL MACHINE)

- Go to your local machine where you have the copy of your remote branch (lab -4).
- Run the command "**git pull**" to pull all the new changes such as branches form the remote location.
- Run "**git branch -a**" to list down all the branches.
- The branch which starts with remotes/→ are remote branches.
- The branch without remote /→ they are available on your local copy as well.
- Checkout to the feature branch or the branch that you crated in (lab-6)
- **git checkout<branch name>** (main)
- Make sure that are on the new branch by running git status or git branch command. (notice the * mark)

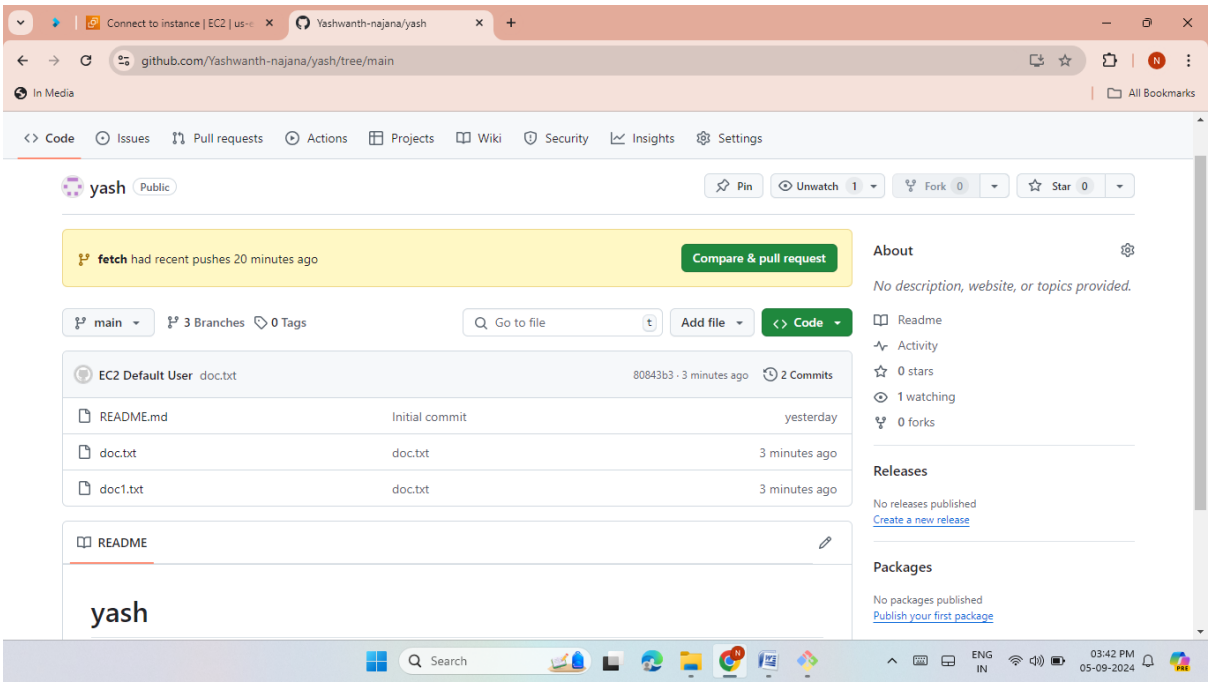


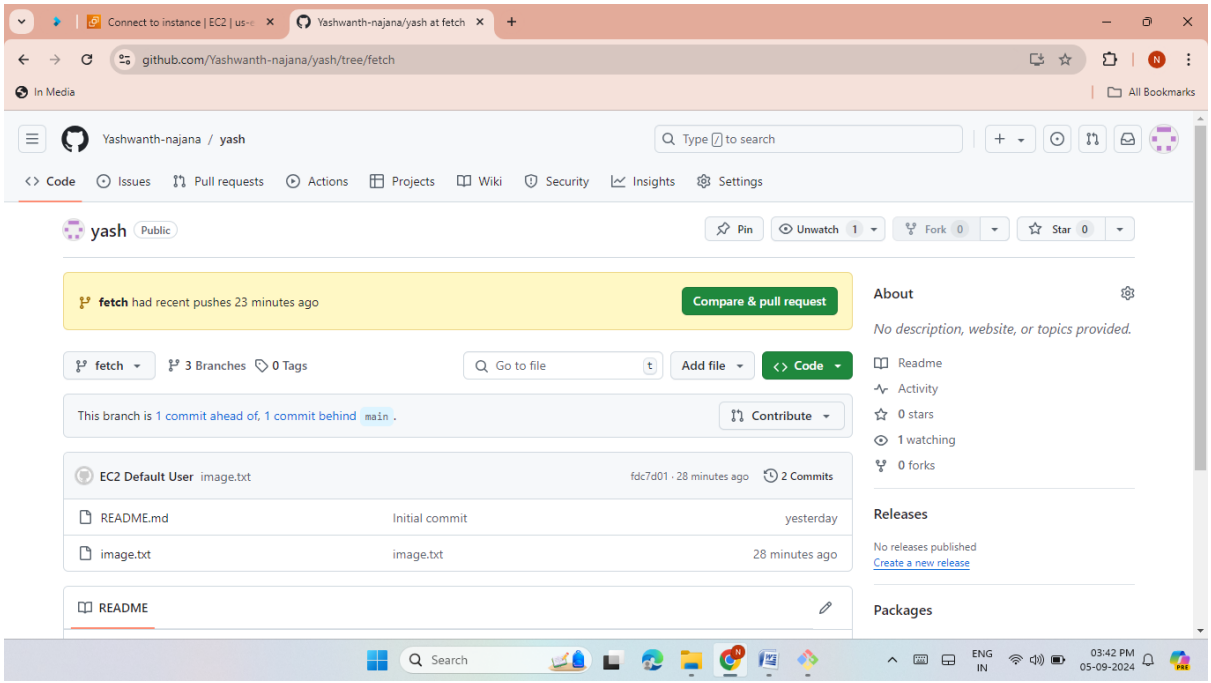
- Make some changes in this branch such as adding the files **" touch files3 file4".(doc.txt,doc1.txt)**
- **git status**
- **git add<filename>** (doc.txt,doc1.txt)
- **git commit –m "<msg>"** (doc.txt,doc1.txt)
- Run git status to show working tree is clean.
- **Git push**
- It will ask you for username and password , provide the username and personal access token in place of password which we created before.





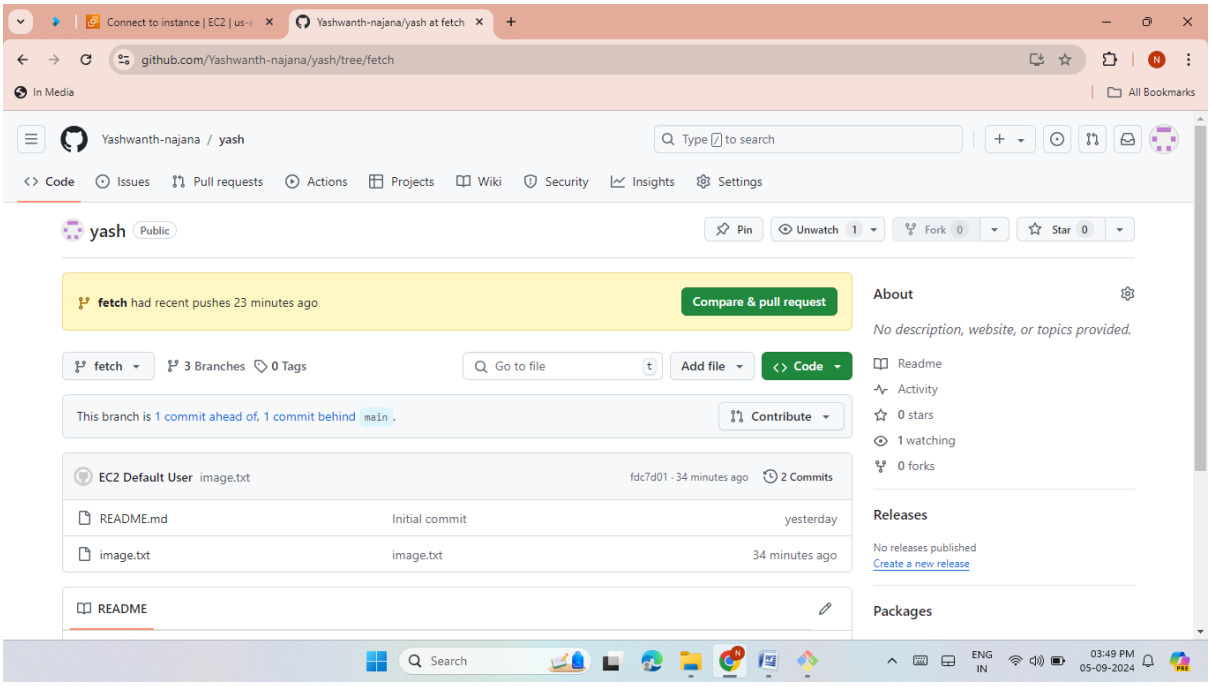
- Go to your github portal once again.
- Here see that the new changes are only available in your feature(main) branch but not in the fetch branch. The files are doc.txt,doc1.txt.



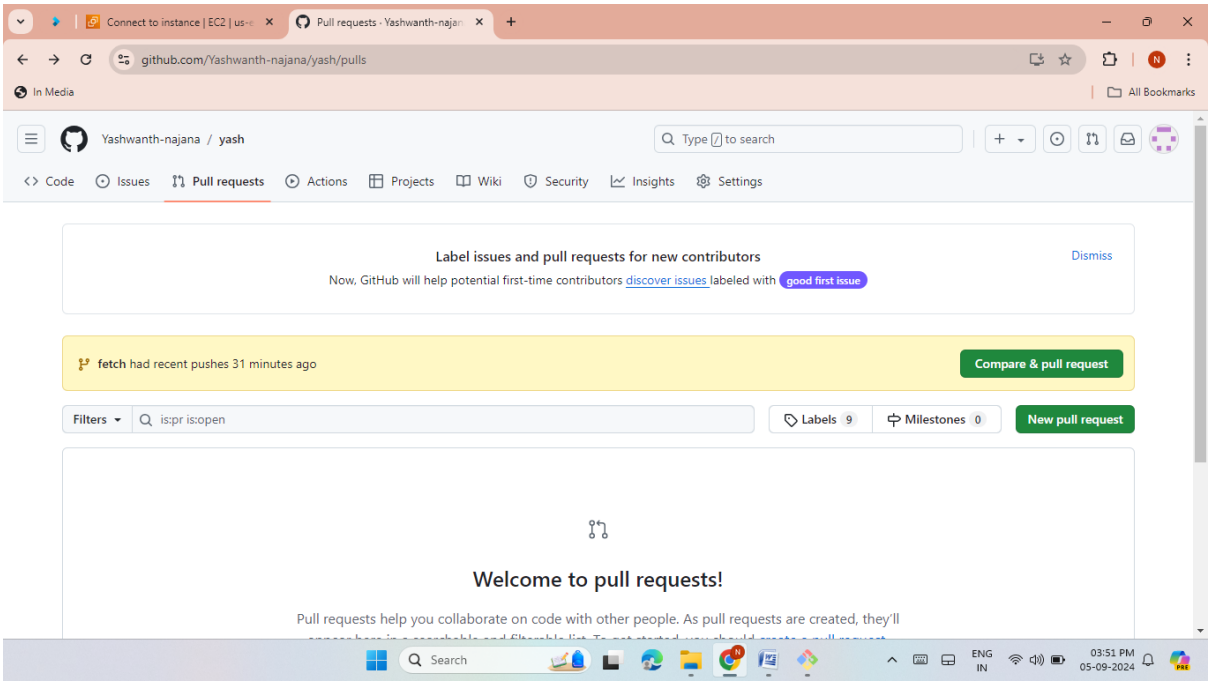


LAB 8 : (MERGE OUR FEATURE BRANCH WITH MAIN BRANCH)

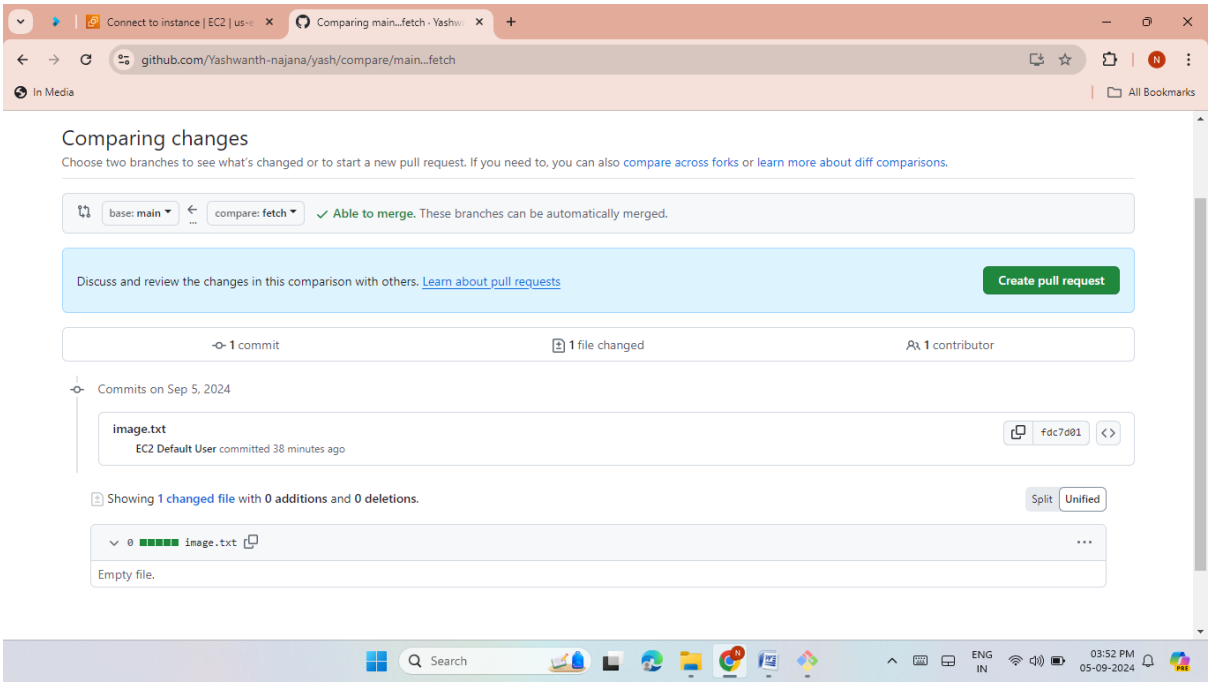
- Go to your git hub repository.
- Check the changes In your feature branch.(fetch branch)
- It have image.txt file only.



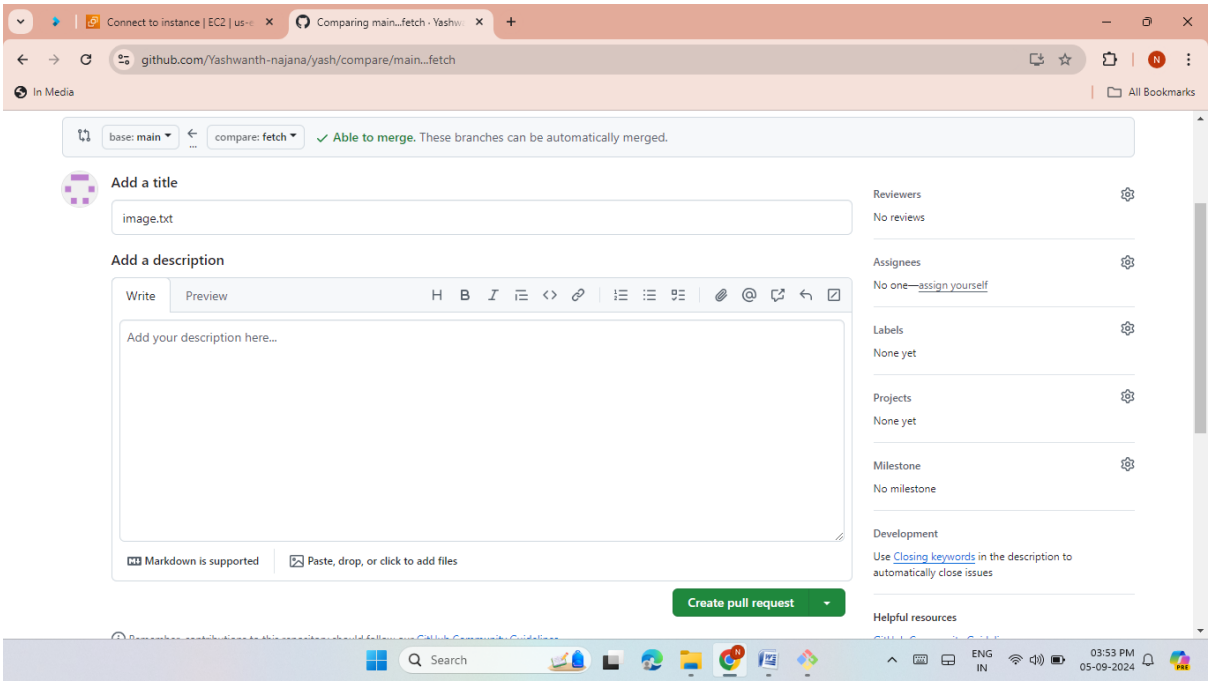
- Go to pull request tab and click on create pull request.

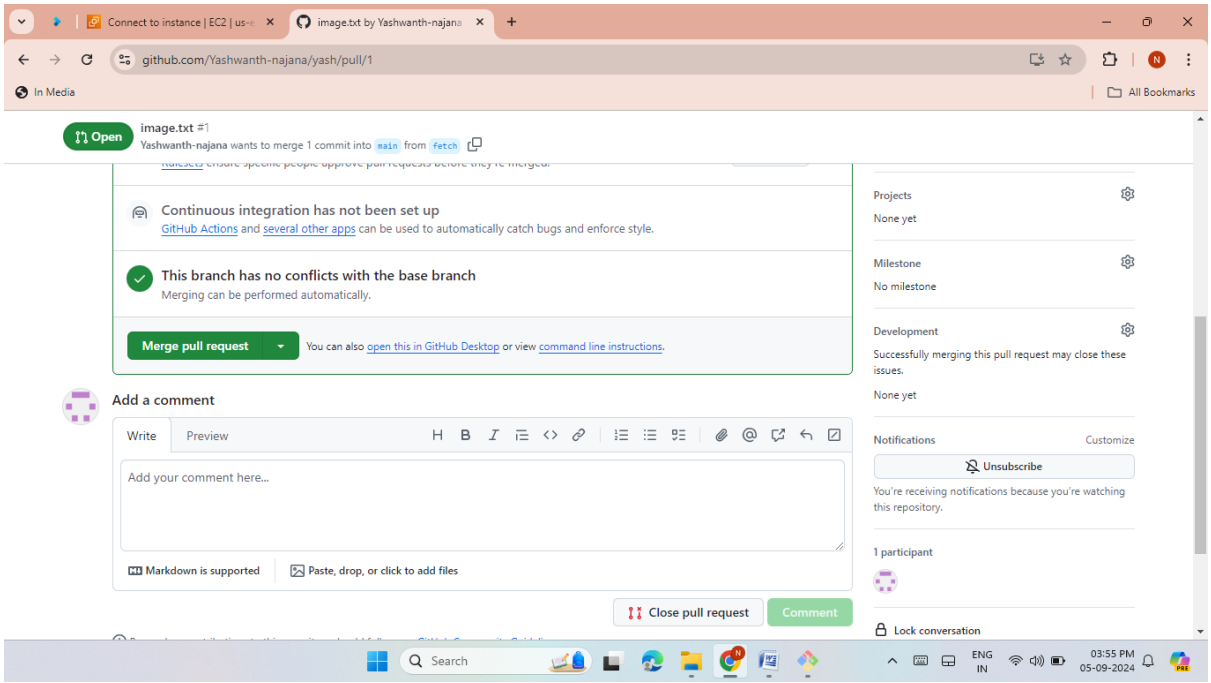


- Put your main branch (where u want to merge/destination) and the feature branch(fetch branch) in the respective blocks.

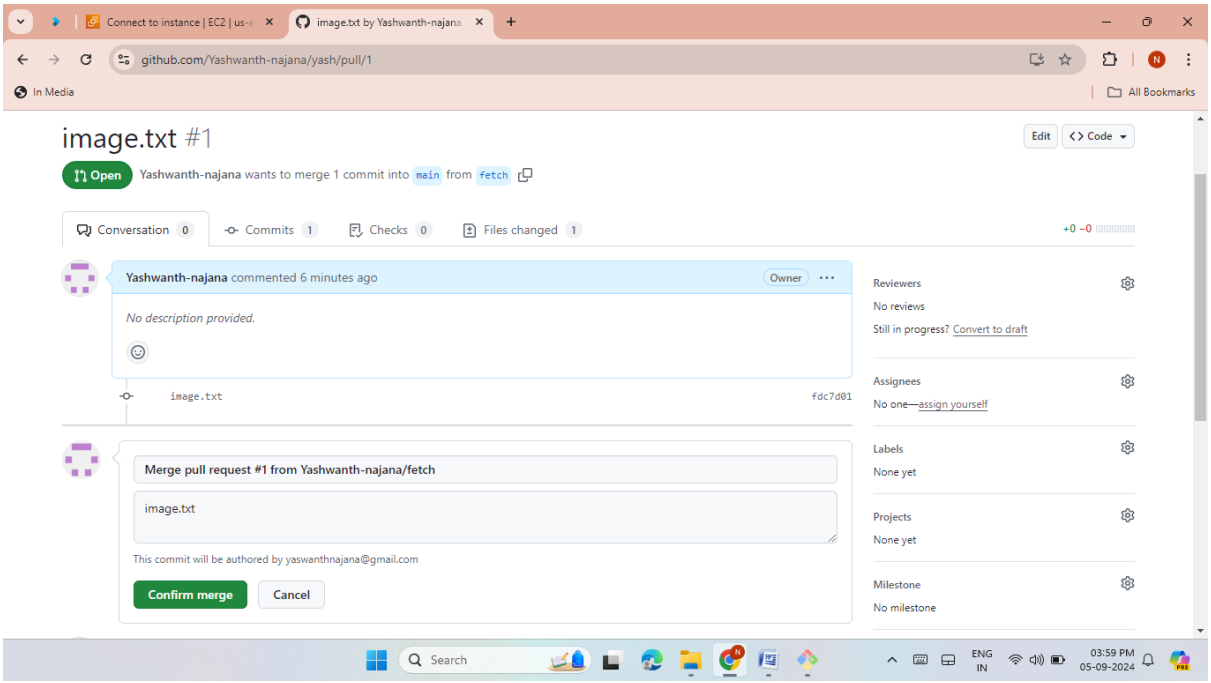


➢ Click on create pull request and it will ask for a comment, just click again on create pull request.

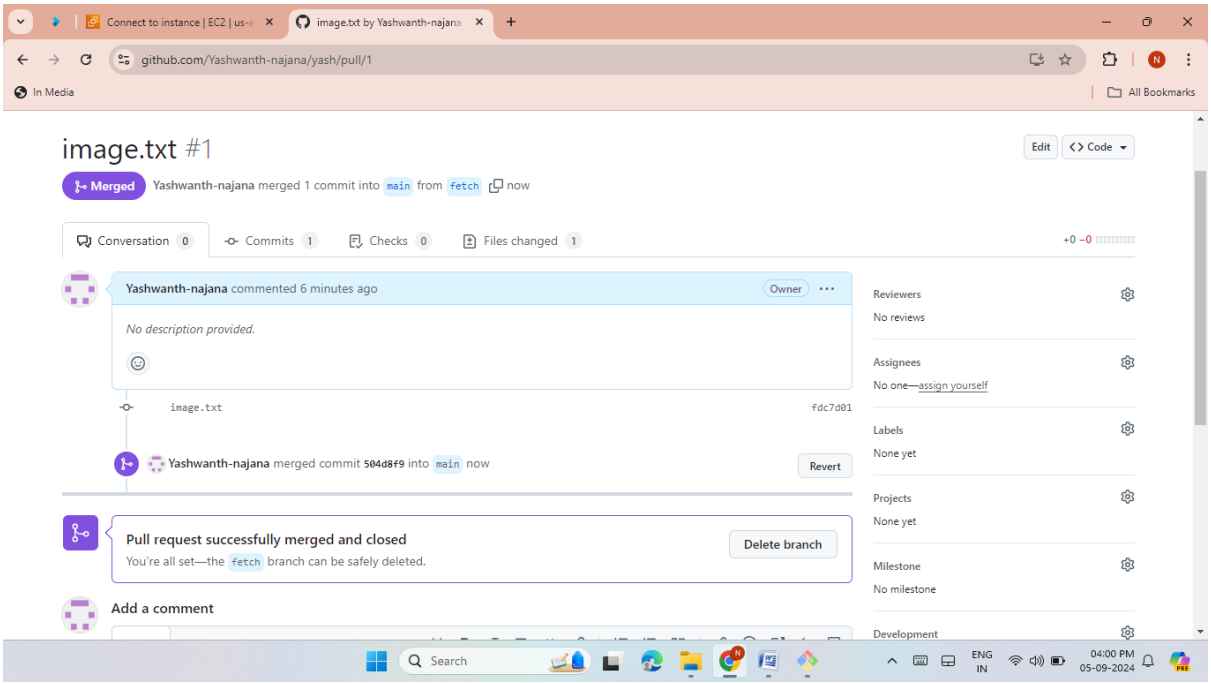




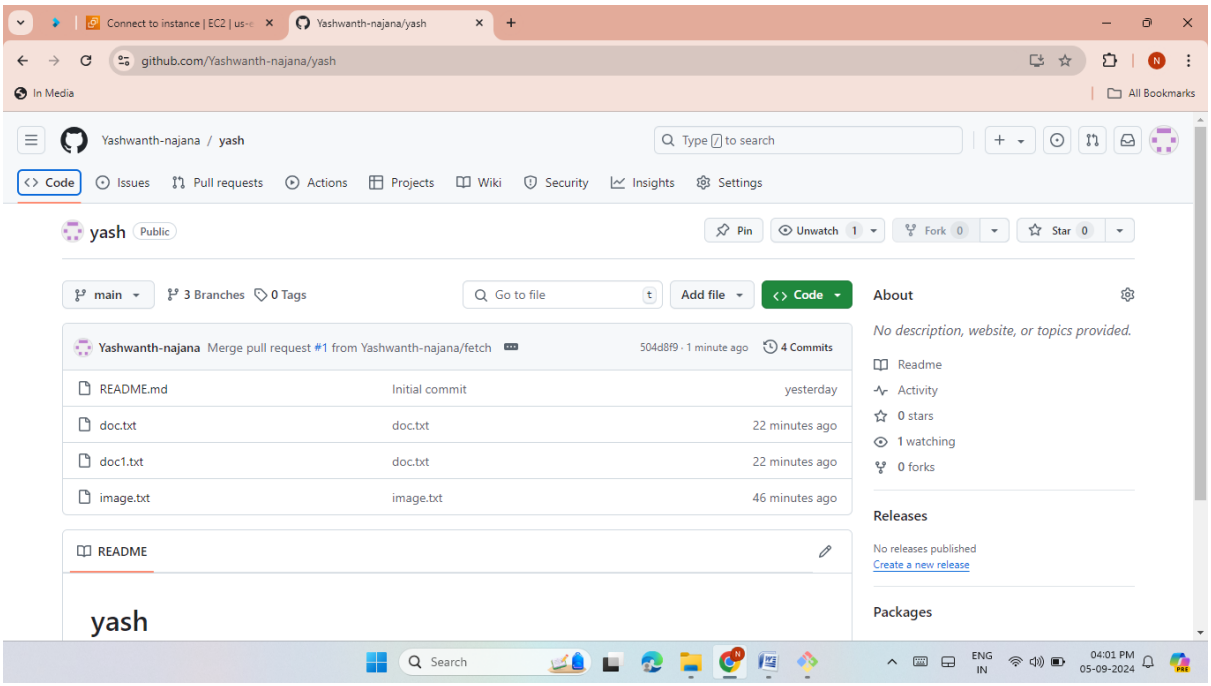
➢ Click on confirm merge.



➢ Once done, you can delete this feature branch if required or you can simply ignore it.

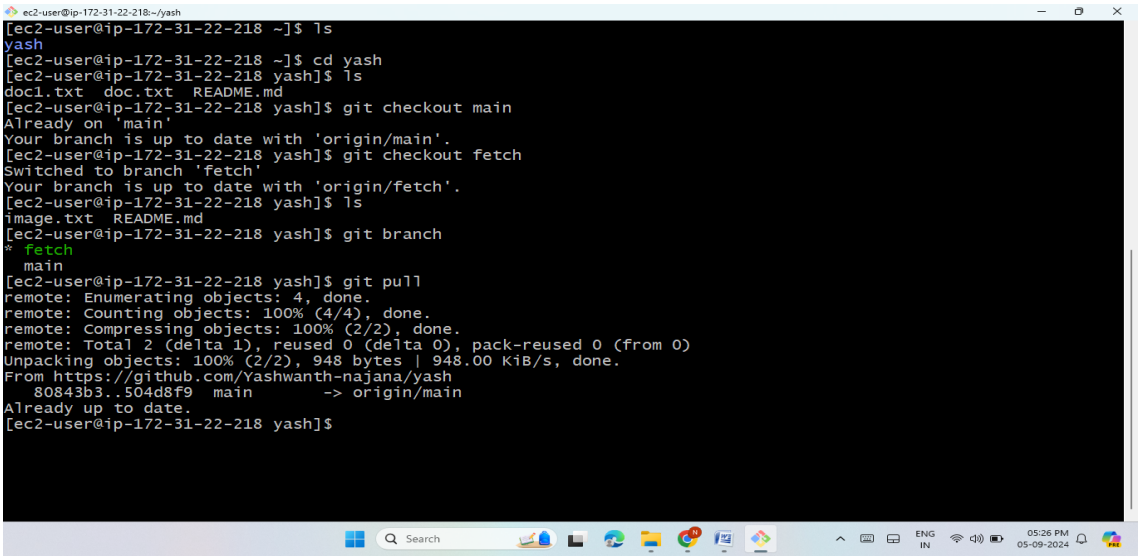


- Go to your code in main branch and see the changes are now visible here.
- Here the fetch branch files are added (image.txt).



LAB 9 : (GO TO LOCAL MACHINE)

- Go to your local machine where you have the copy of your remote repo.
- Checkout to the fetch branch
- **git checkout** < branch name>
- Now run the command "**git pull**" to pull all the new changes such as branches form the remote location.



- Here see that the changes are only available in your fetch branch.