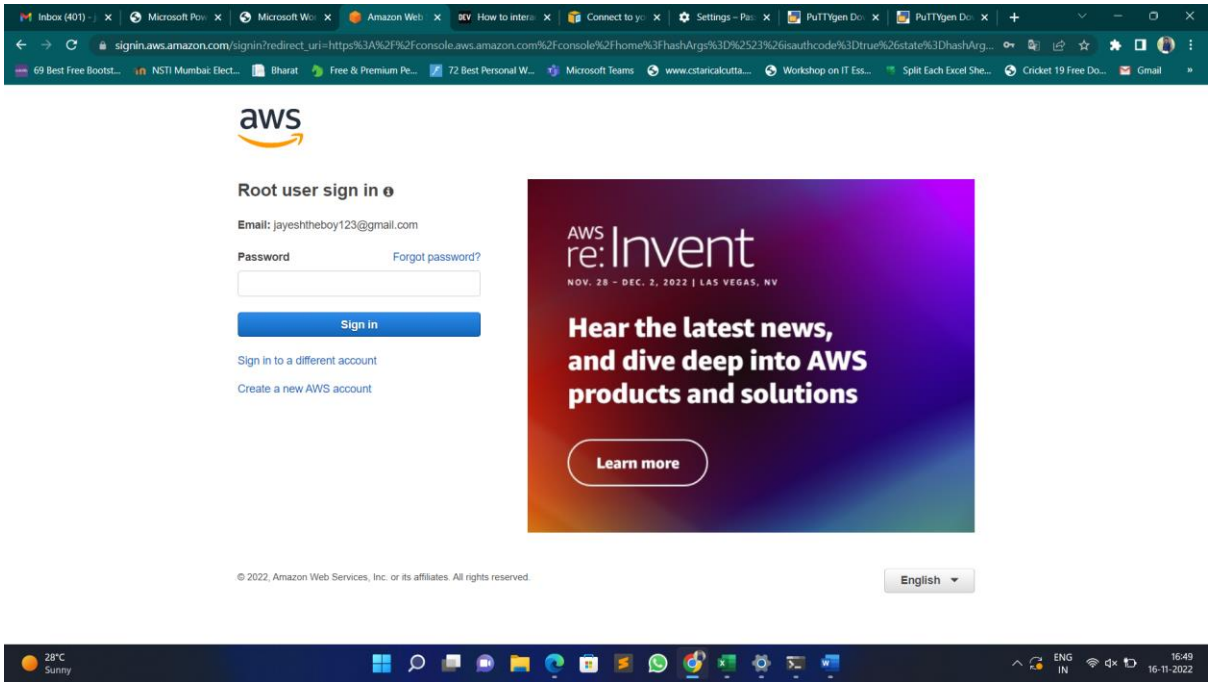
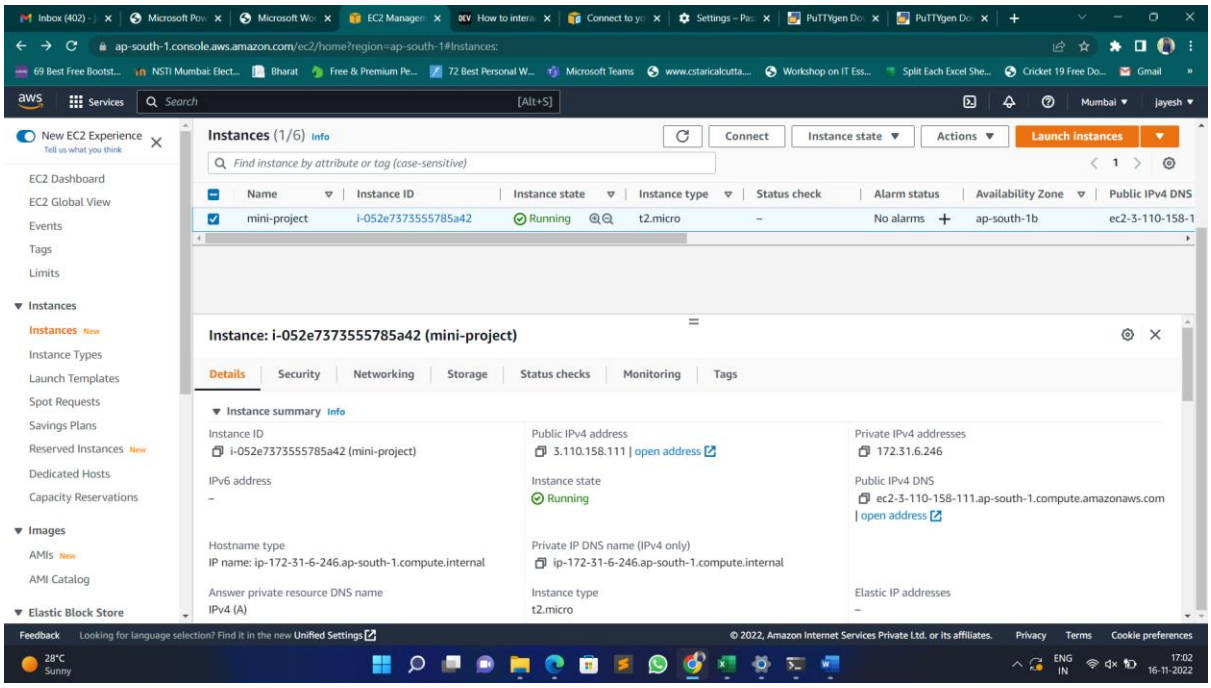
MINI PROJECT - 1

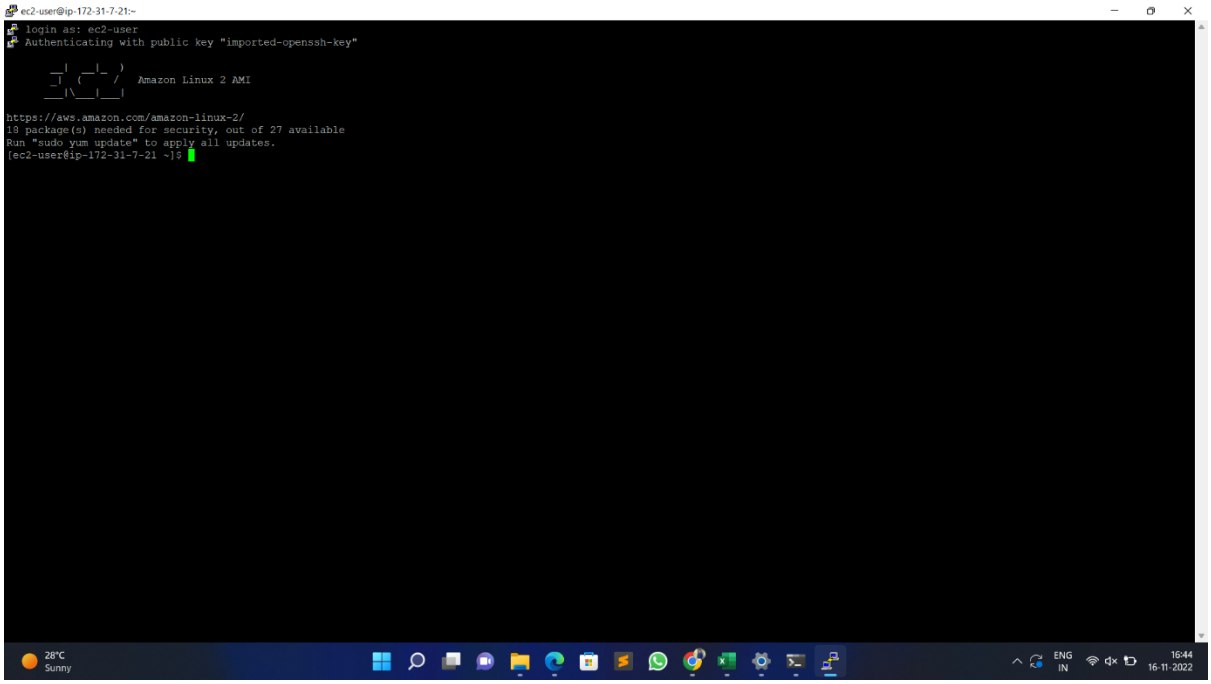
LAB 1: Creating EC2 instance



- **Logged in to AWS Console using web browser with our user credentials.**

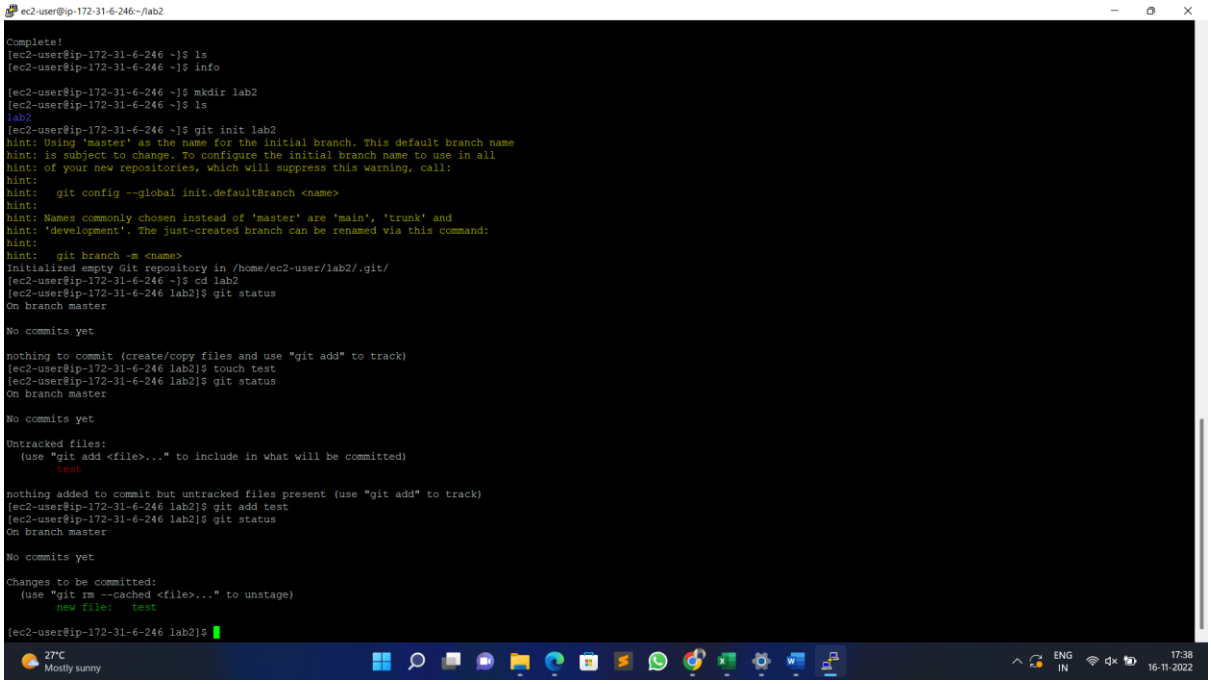


- **We launched a EC2 instance with Amazon Linux 2 AMI.**

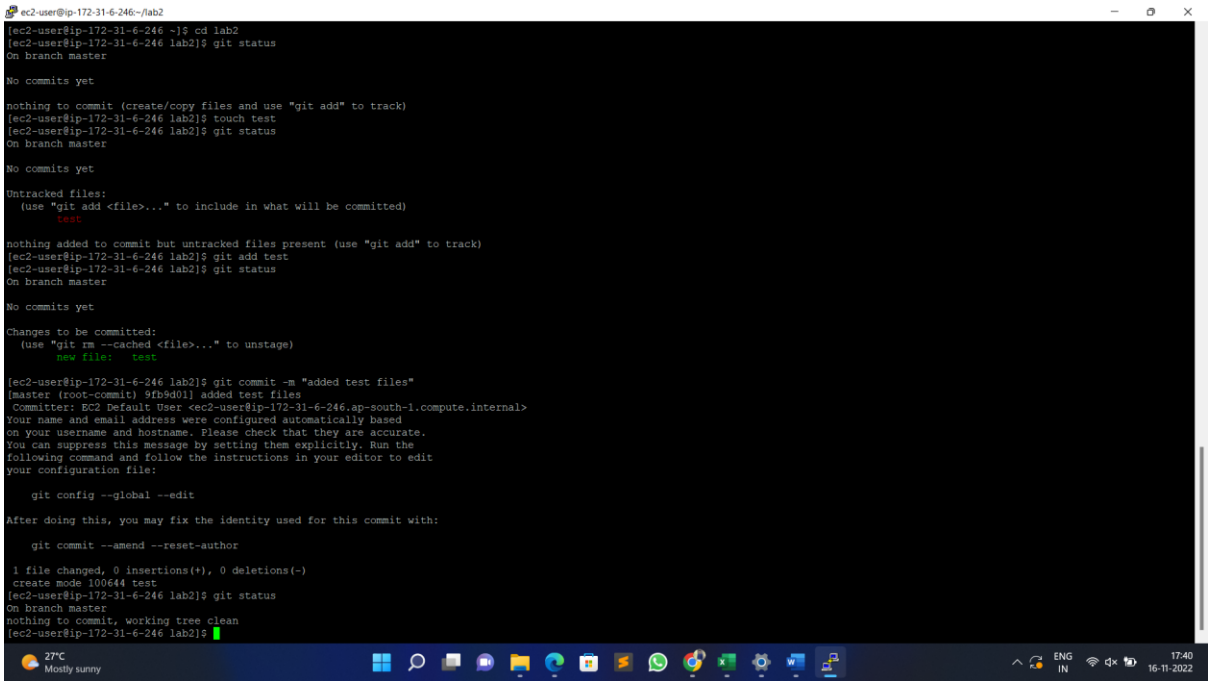


- **Connected to our Amazon Linux Server with the putty using public IPv4 address of our instance.**

LAB 2: Create repo in local machine

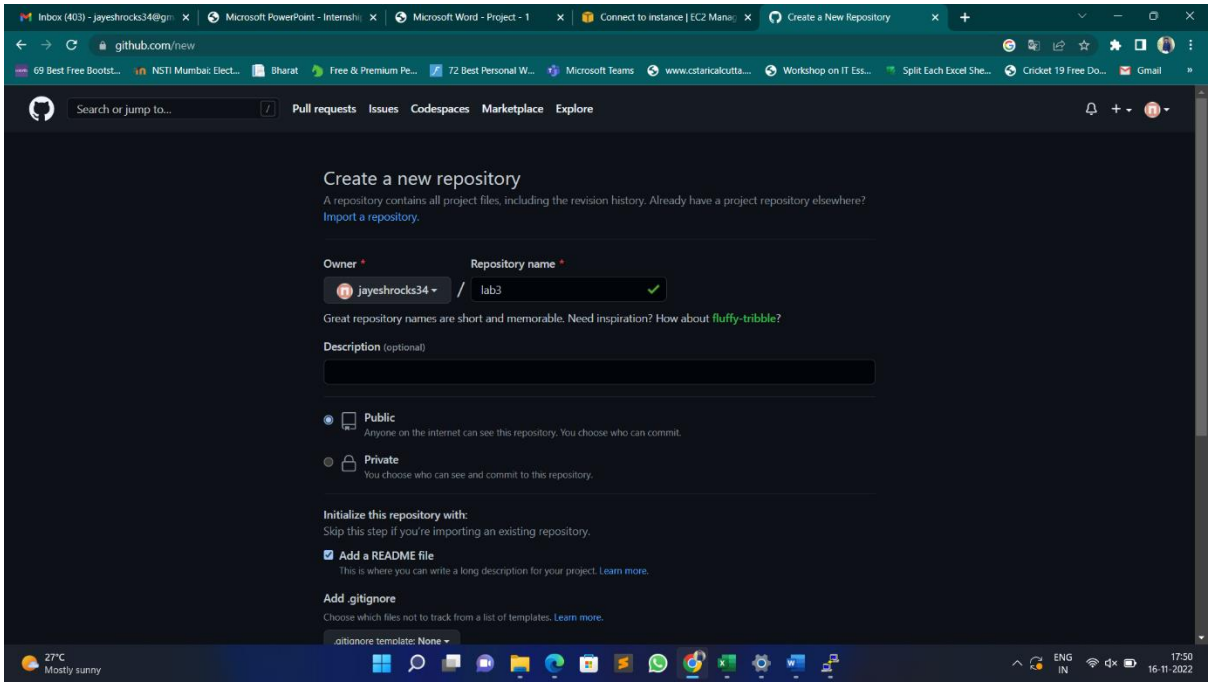


- **Created a folder in the local machine and initialized it using “git init” command.**
- **Then inside that folder we created a file using “touch” command.**
- **After creating file, we checked “git status” then it showed that the file in untracked, then for tracking file we used command “git add <filename>” then it staged our changes.**
- **After adding we checked for status then it showed the file is now tracked but we need to commit the changes.**

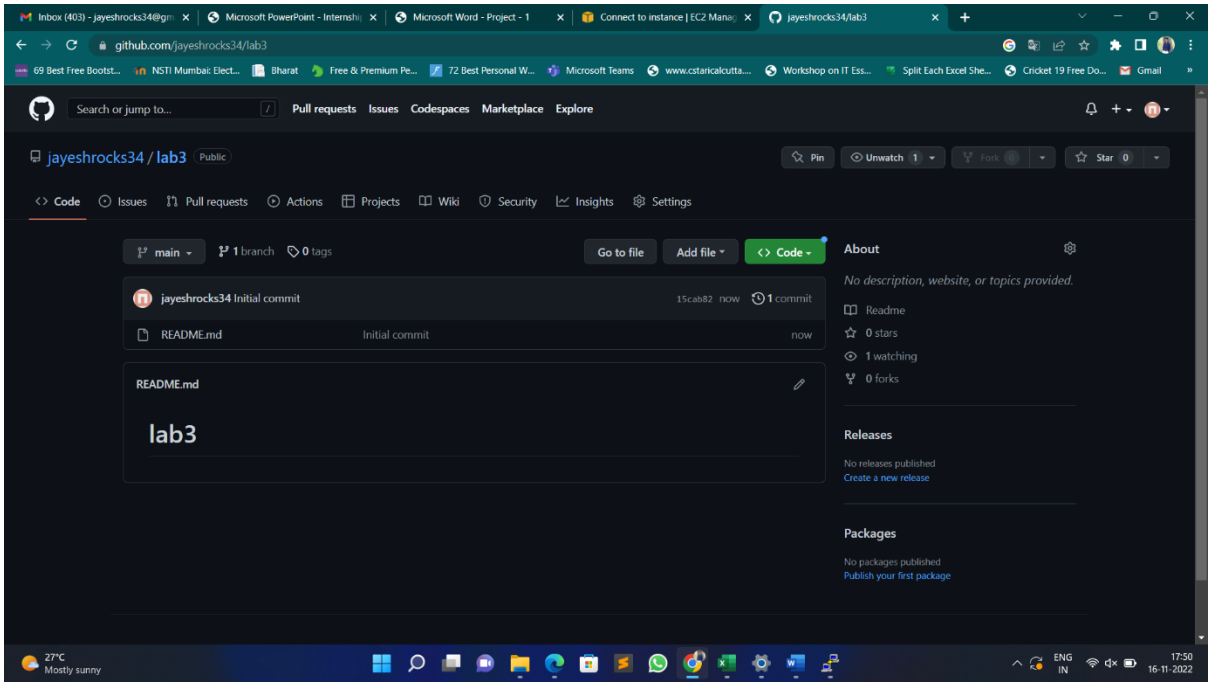


- **For committing the changes, we used ‘git commit -m “message” ‘. This command commits the changes and saves the date and time of the changes and author details of the changes done by.**

LAB 3: Creating repo in remote location – GitHub

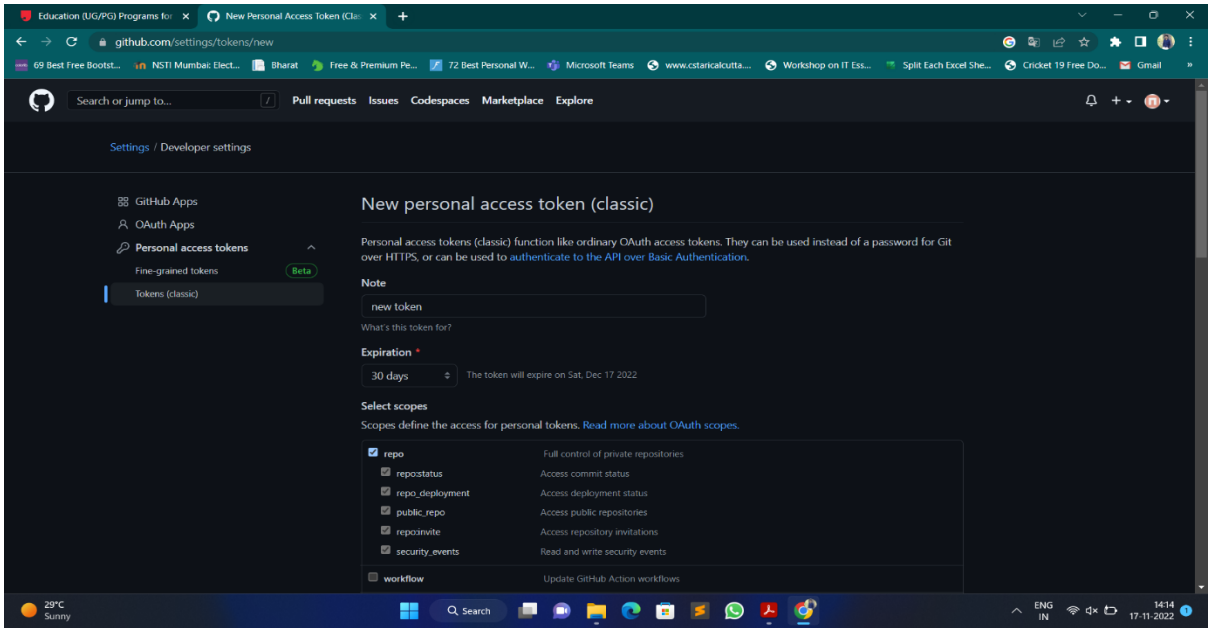


- **Logged in to GitHub account using web browser'**
- **Then created a new Private Repository and added readme file to initialize the repository.**

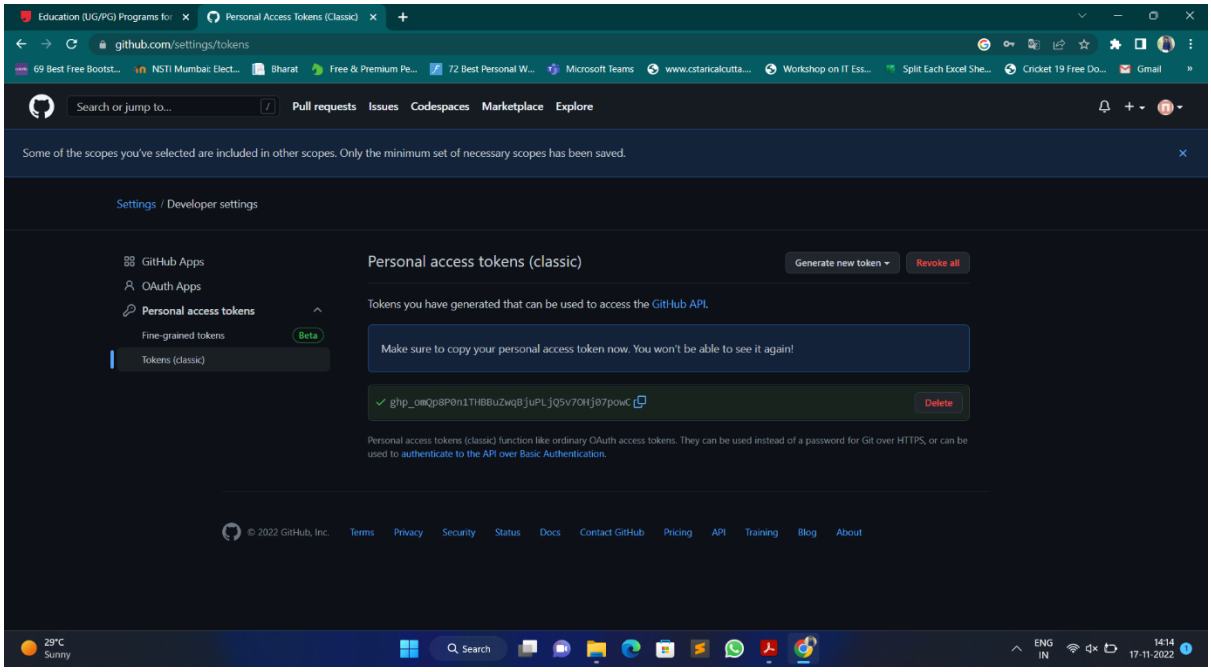


- **The Repository is created successfully as can see in above image.**

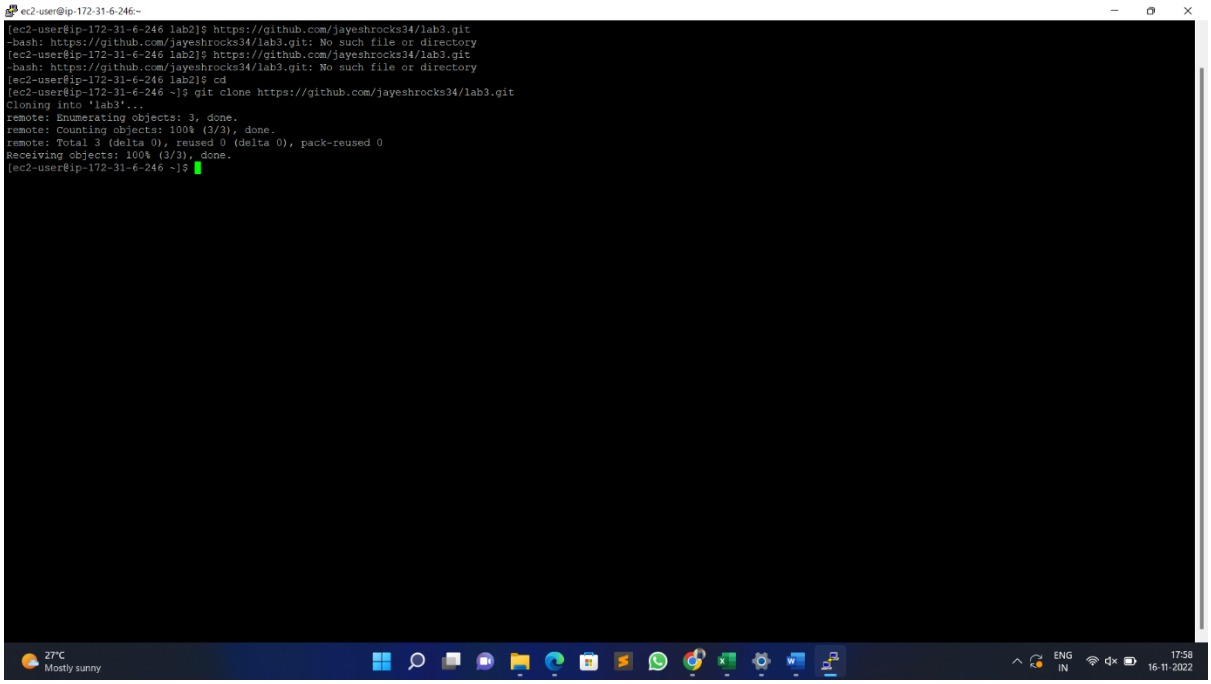
LAB 4: Working with Remote repo



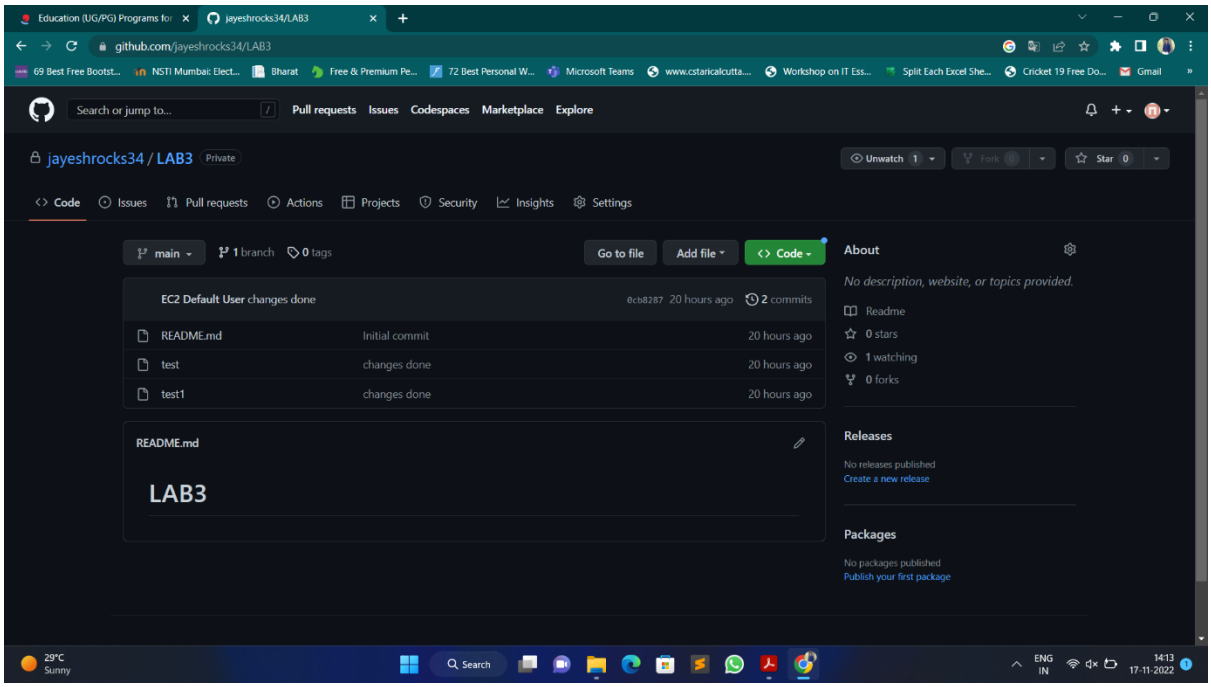
- **We need to login to our GitHub account through CLI for cloning our repository, and for that we need to enter our username and password, but your password won't work as per new changes.**
- **Then for login we need to create "Personal Access Token" to clone this repo.**
- **To create it Go to Setting → Developer Setting → Personal access tokens → Tokens(Classic) and click on 'Generate new token' then write a note if you want and select the token Expiration days as per requirement and select "repo" from scopes then scroll to end and click on "Generate token".**



- **We will get the token key just copy that key and save it somewhere for next use because you will not able to see the key again in setting, if you lost it you have to generate new again.**

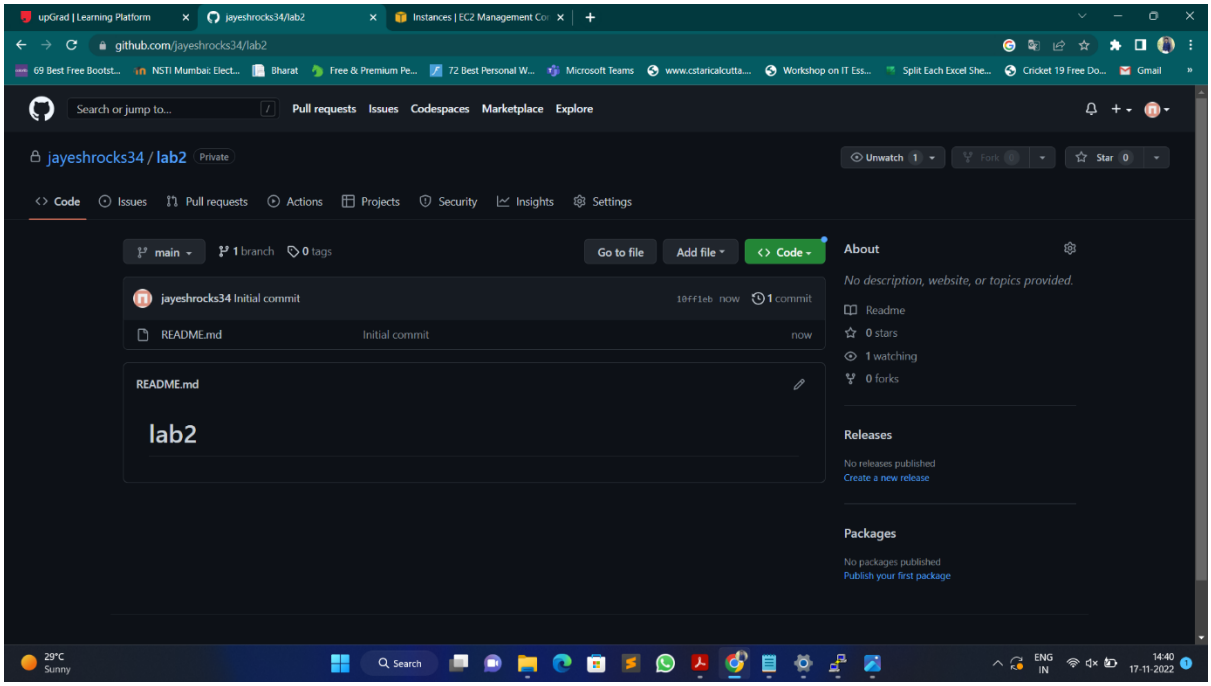


- **Then go to local machine and now try to clone repository using “git clone <URL>” command.**
- **After cloned go to repo folder and create some sample file using “touch” command.**
- **Stage these changes by running “git add <filename>” command, and commit these changes by running git commit -m "<any message> “.**
- **Then use “git push” to push all changes to remote repo, this will ask for username and password so provide username and Personal access token as password which we created above.**

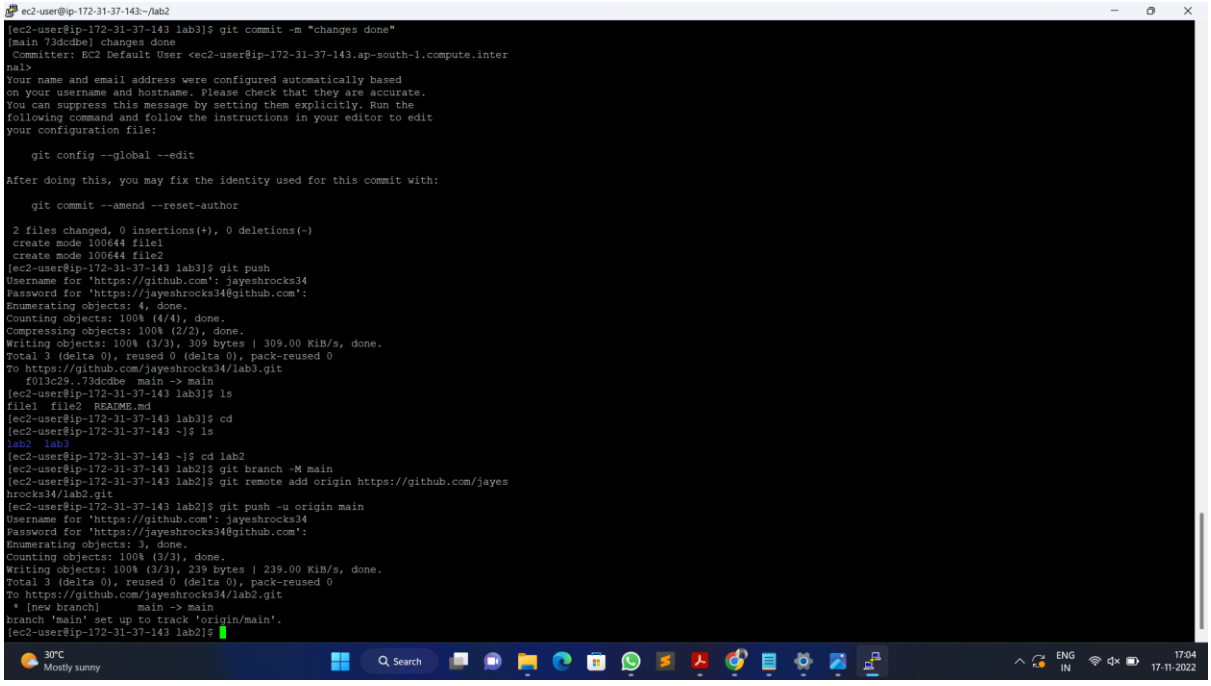


- **Then go to the remote repo and see, you will be able to find your new changes here.**

LAB 5: Pushing a locally created repo to GitHub

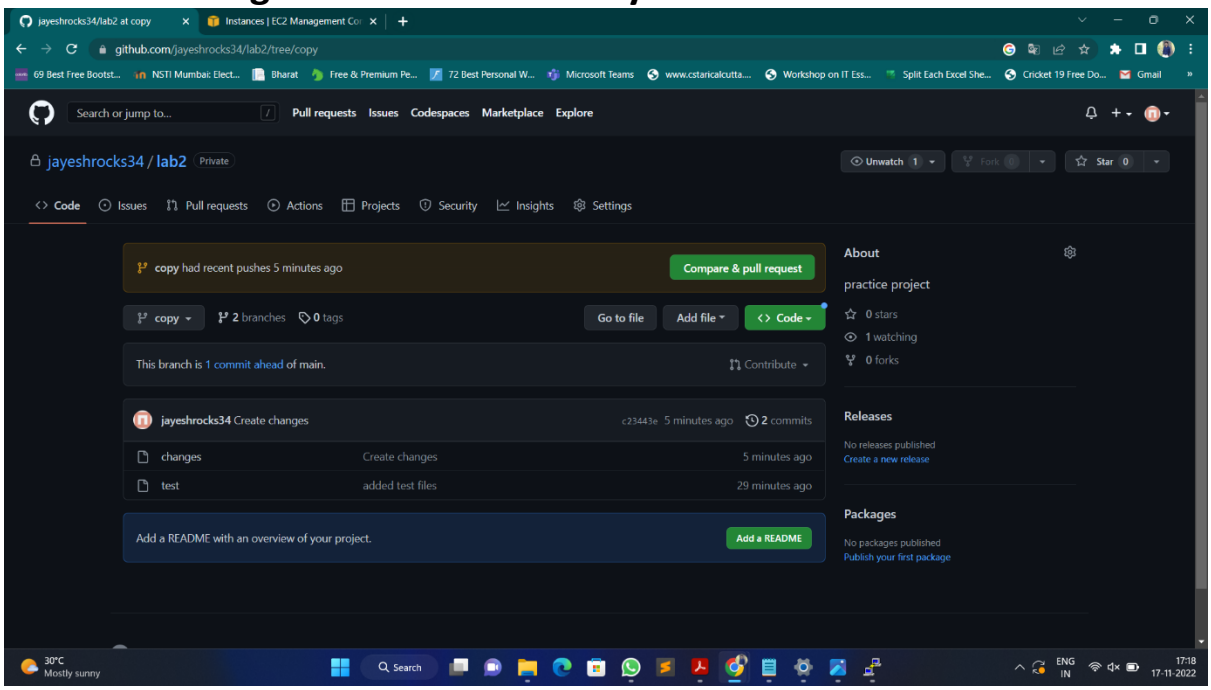


- **As we created a local repo in LAB2 with the repo name ‘lab2’, so now we have created a remote repo with the same name ‘lab2’.**

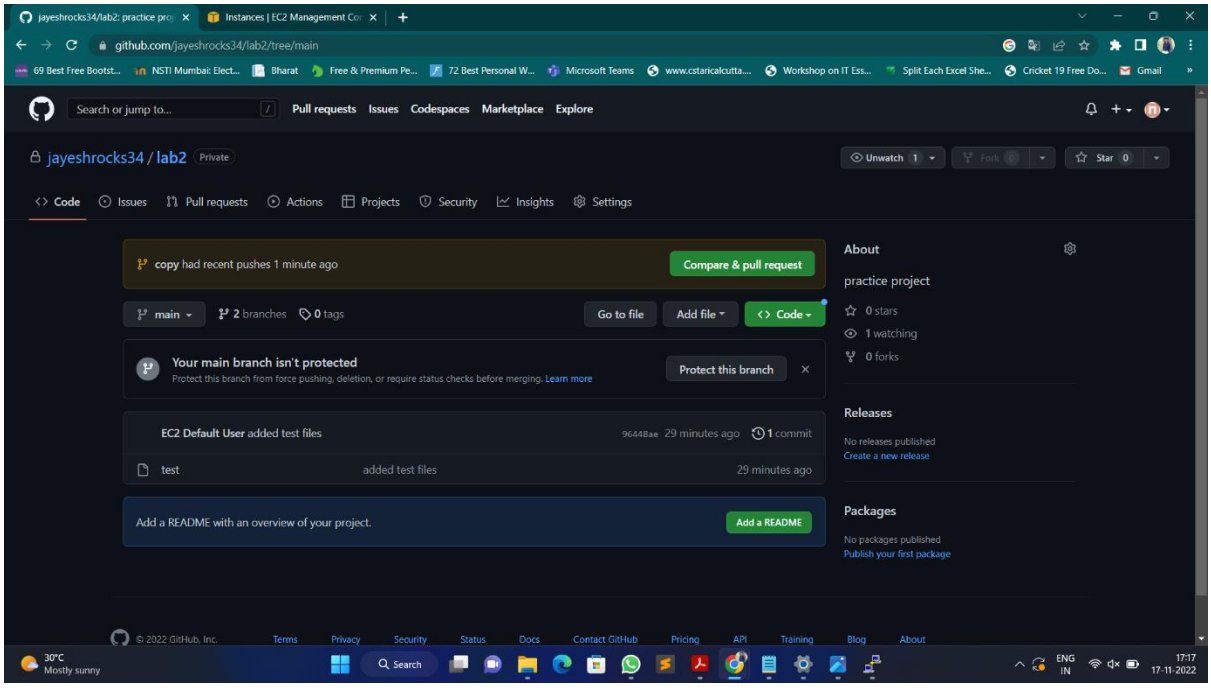


- **After creating we went to our local machine inside the ‘lab2’ repo and used “git branch -M main” command to change the name of branch as master branch is now known as main branch.**
- **Then we used “git remote add origin <URL of your remote repo>” command.**
- **Used “git push -u origin main” command to push our local branch to remote repo.**

LAB 6: Creating a new branch from your main branch

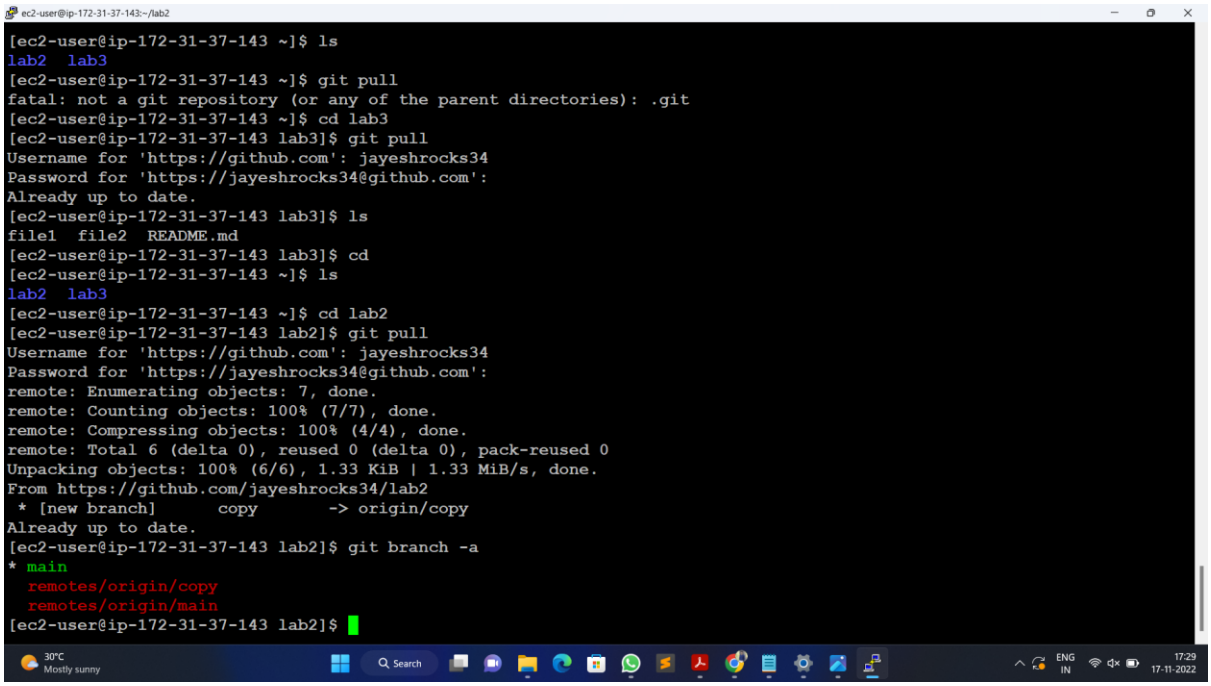


- **Went to the repository on GitHub at the place of main and clicked on the branch dropdown and typed the name “copy” and clicked on create button.**
- **A new branch of name “copy” is created.**
- **Then added a new file named “changes” to the copy branch as can see in above image.**

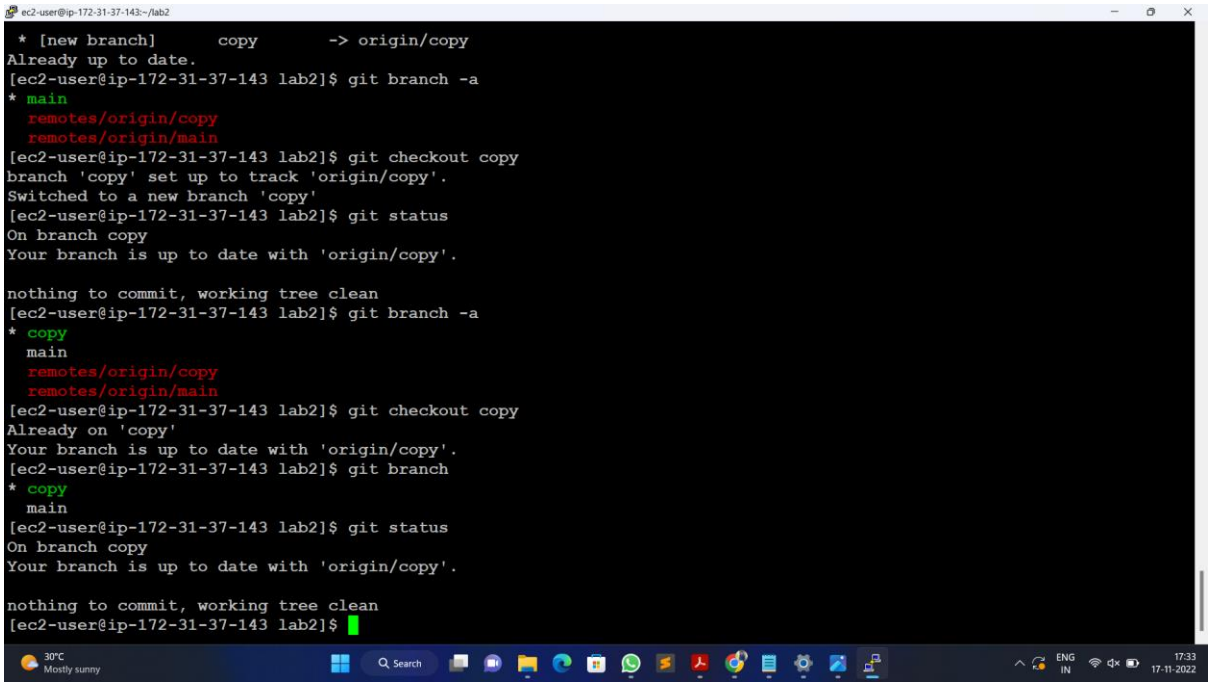


- **Now we saw in our main branch but the changes we made in our new branch are not done in the main branch.**
- **The changes are applied to our new branch only because for getting changes in main branch we need to pull and merge it.**

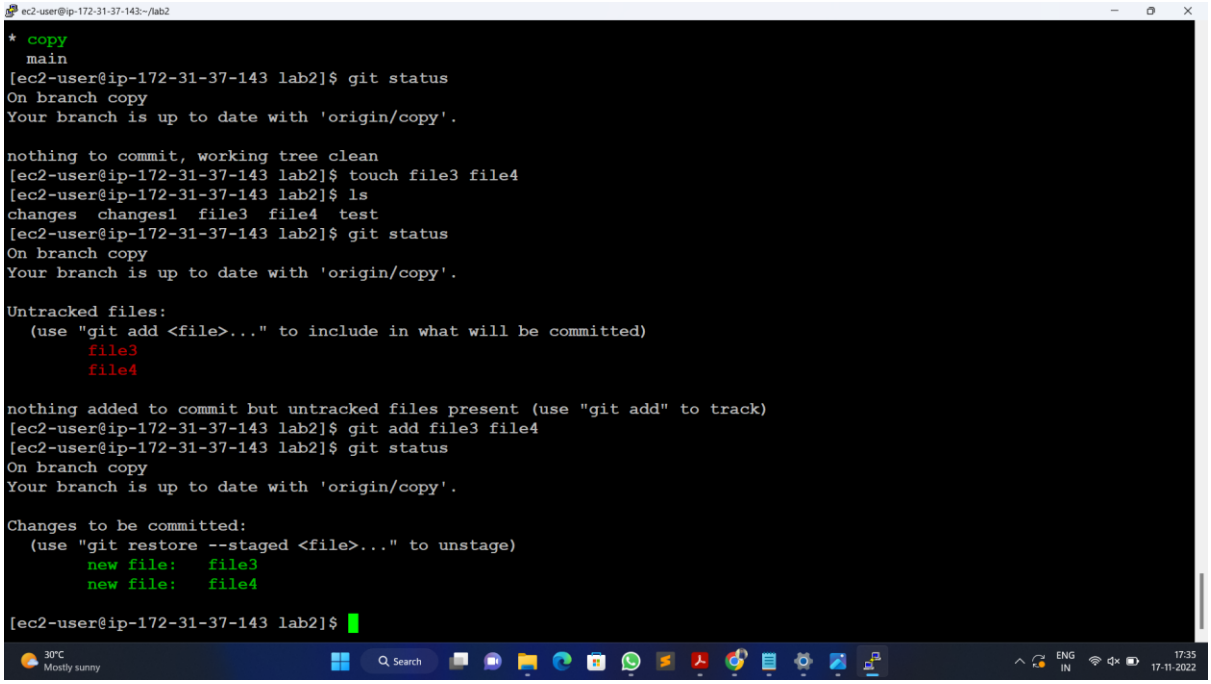
LAB 7: Pull all the branches in your local machine



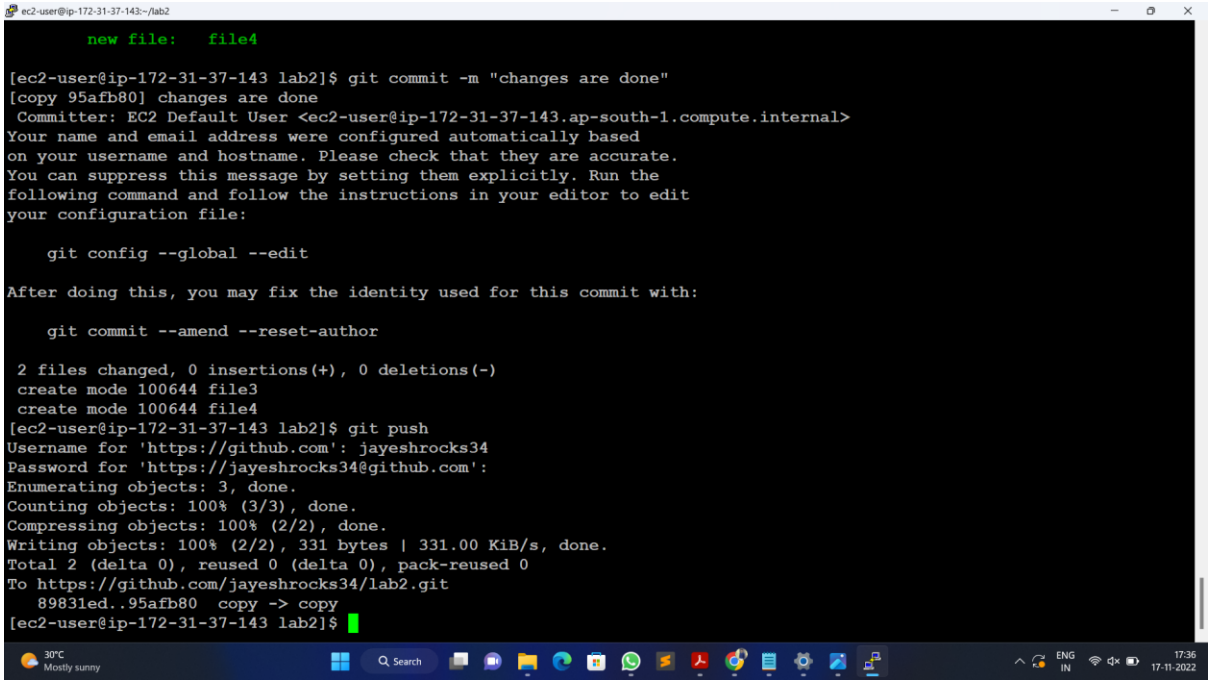
- **Now from our local machine we have run the command “git pull” to pull all the new changes such as branches from the remote location to our local machine.**
- **Then we run “git branch -a” to see the list of our branches.**
- **Here we saw that we are in our main branch.**



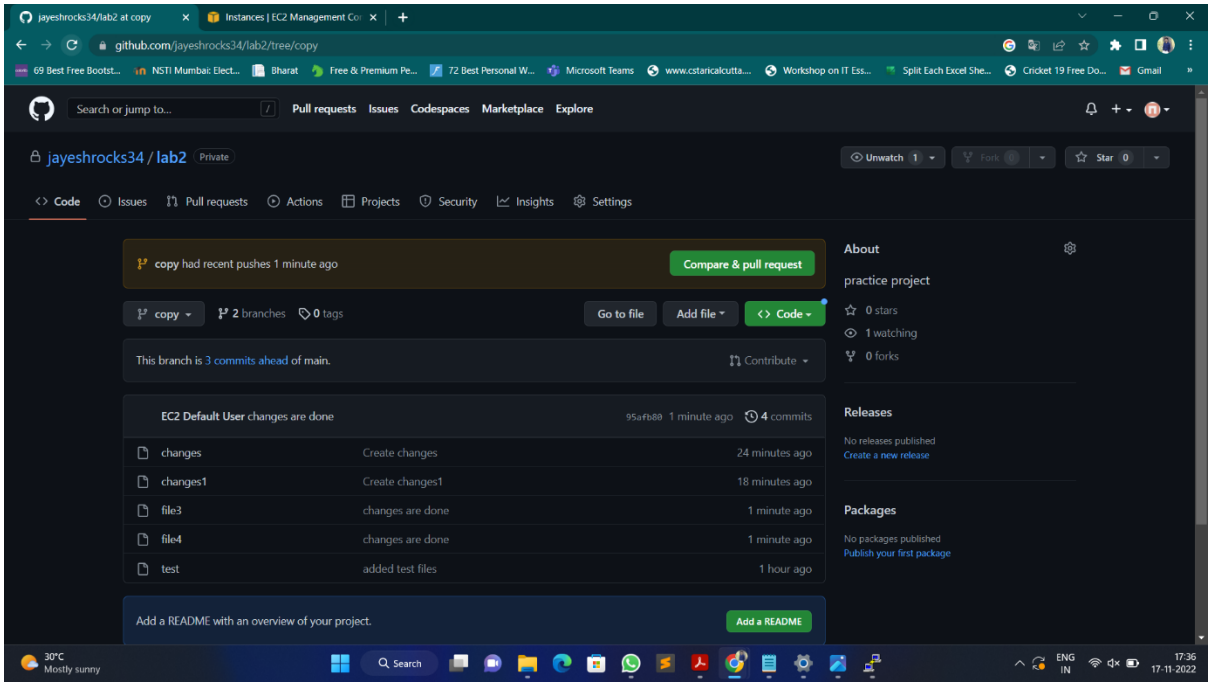
- **Now we have to go to our new branch named “copy”, for that we have to use “git checkout copy” command.**
- **Now for checking on which branch we are working we use “git status” or “git branch command. Here the * mark shows that which branch we are working.**



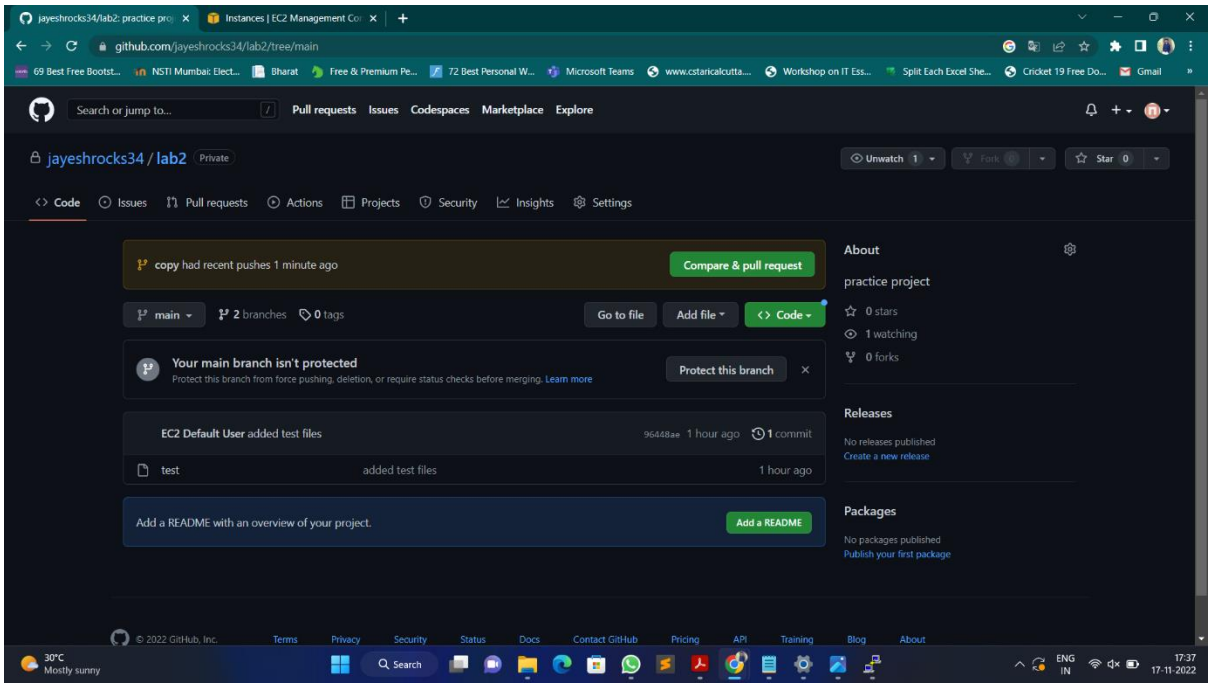
- **So now in copy branch we have added some new file using “touch file3 file4” command.**
- **For tracking the files, we used “git add”.**



- **Now for committing the changes we used ‘git commit -m “message” ‘command.**
- **The we pushed all our changes to remote repo using “git push”.**

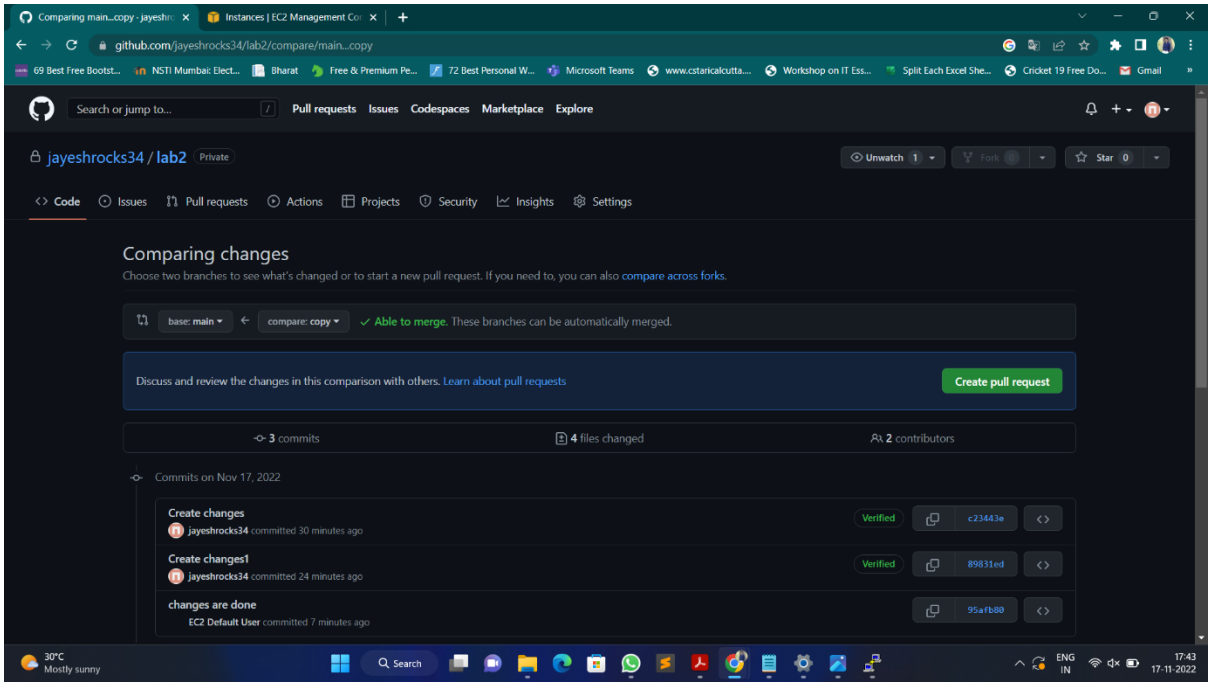


- **We checked on browser the files are successfully pushed as we can see in the above image.**
- **The files are pushed to our new branch.**

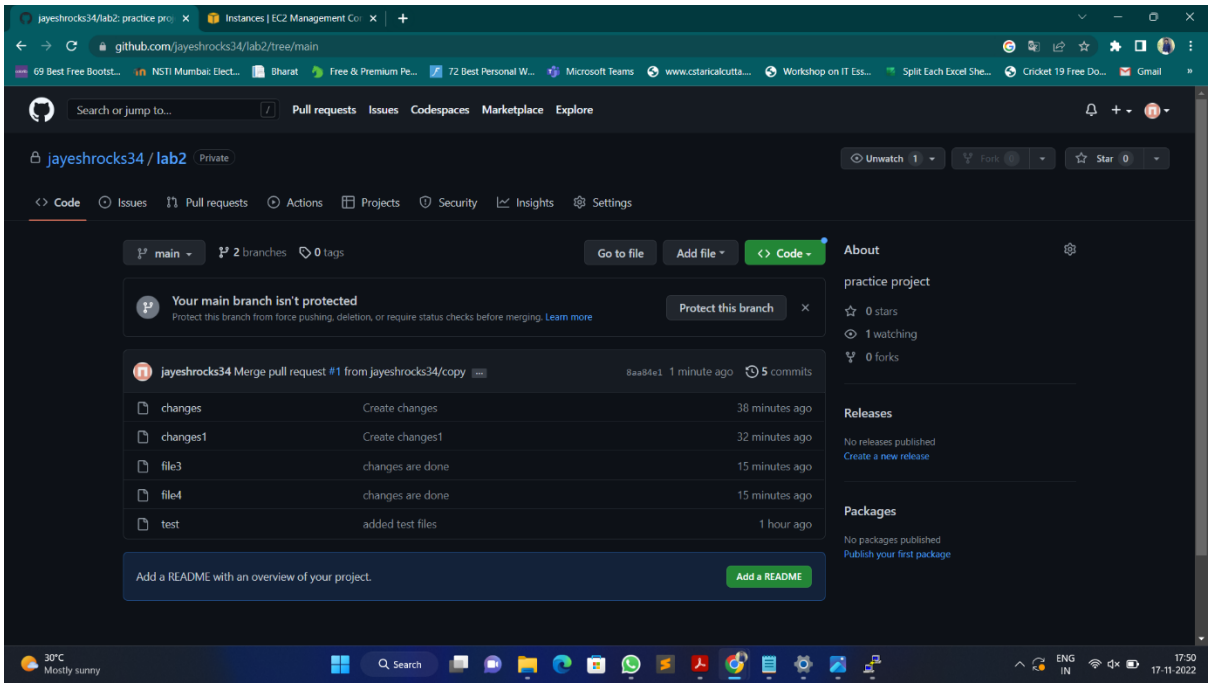


- **The changes are not applied to the main branch.**

LAB 8: Merge our feature branch with main branch

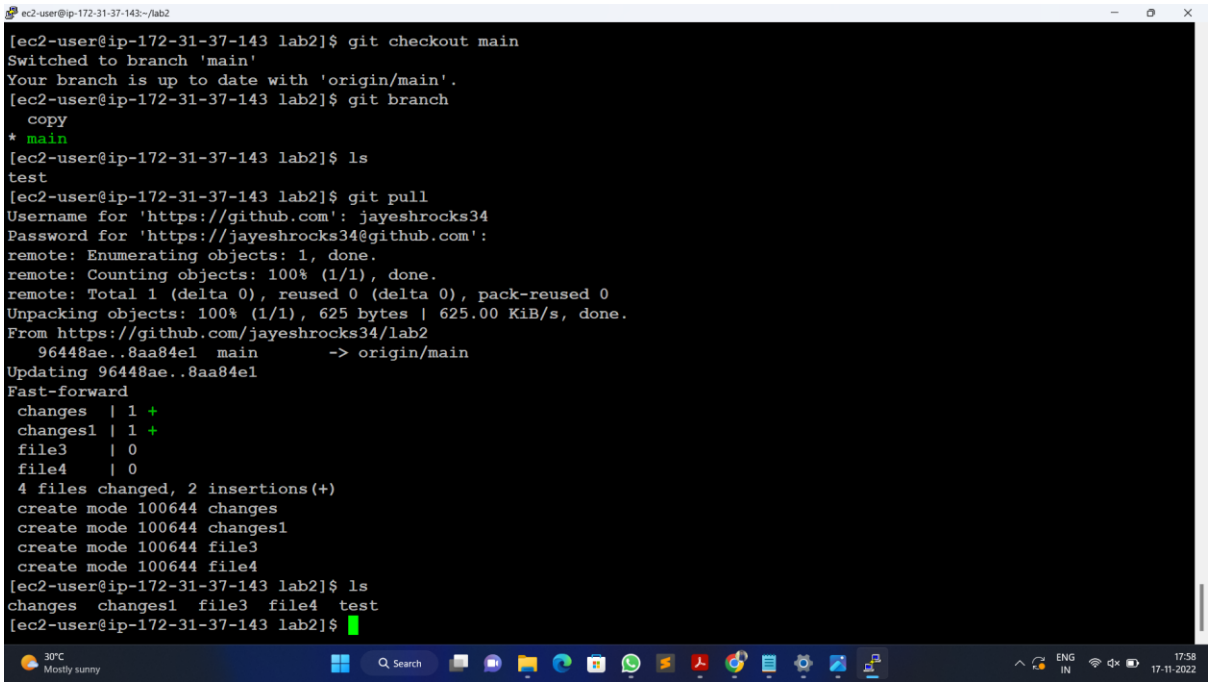


- **Now we have to merge our feature branch with main branch, for that go to the pull request tab and click on create Pull Request (PR).**
- **Click on create pull request and it will ask for a comment, just click again on create pull request Click on review changes and then merge**
- **Then the changes will merge with main branch.**



- **As we can see in above image all files are merged with main branch.**

LAB 9



- **Now from local machine we change to main branch using “git checkout main” command.**
- **Now run the command "git pull" to pull all the new changes such as branches from the remote location.**
- **Here we see that the new changes are only available in your main branch.**