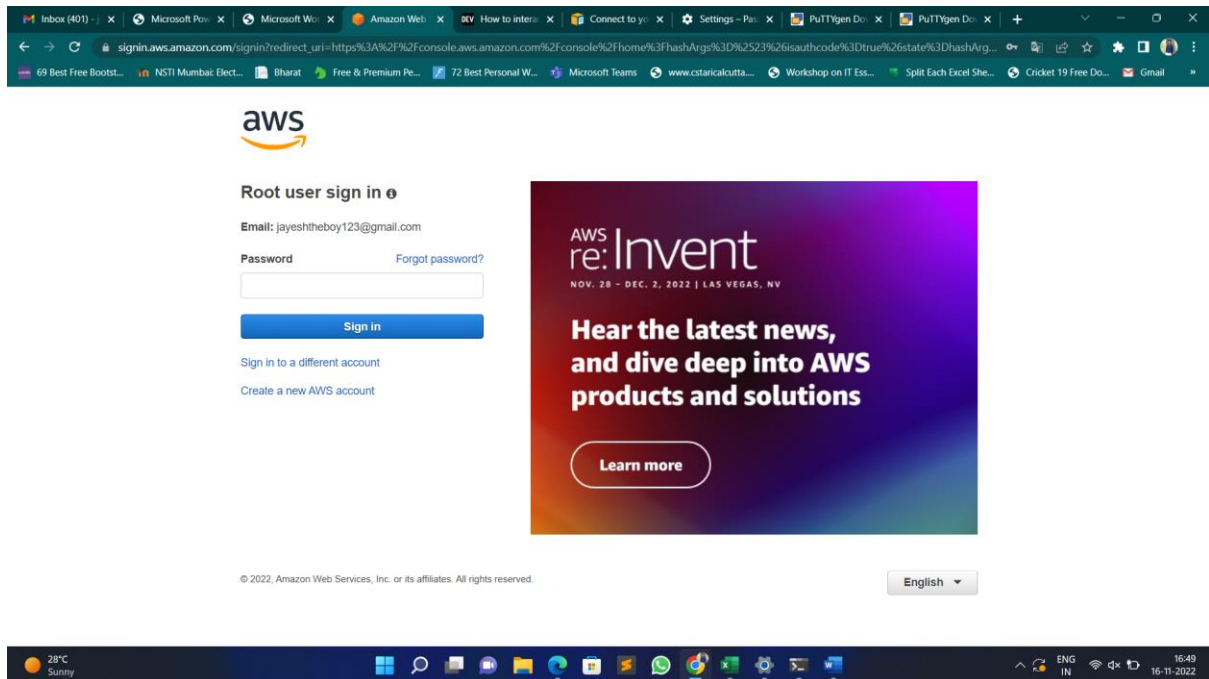
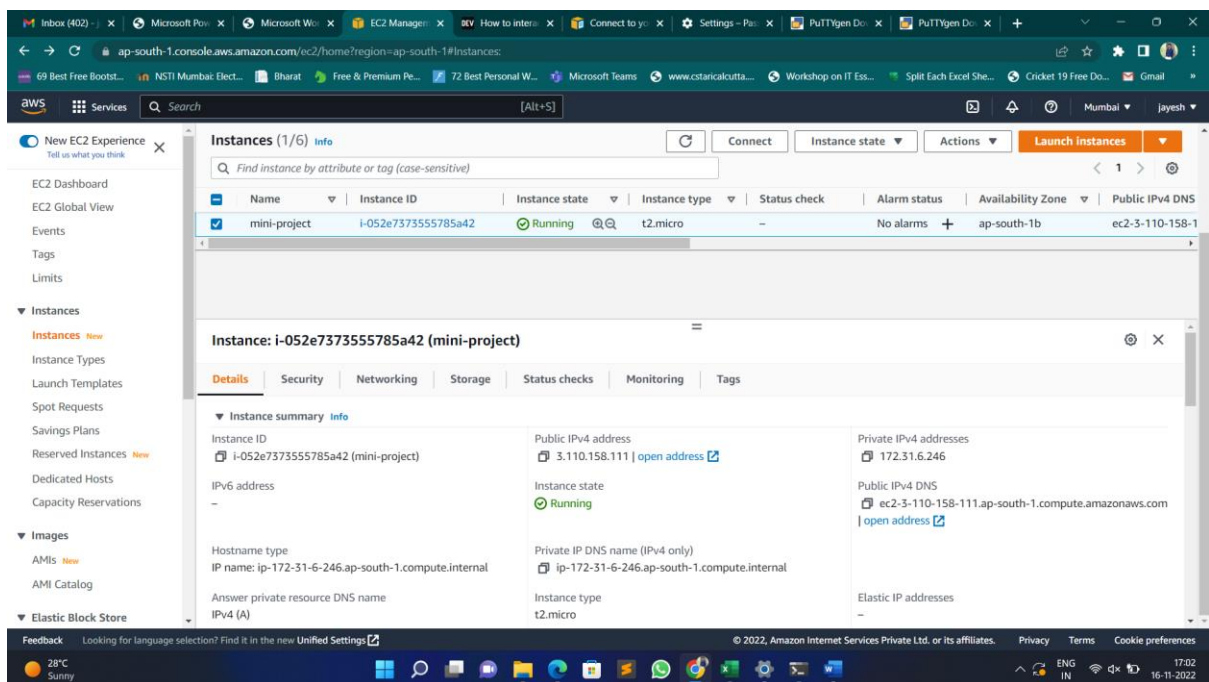


MINI PROJECT - 1

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.

```
ec2-user@ip-172-31-7-21:~$  
login as: ec2-user  
Authenticating with public key "imported-openssh-key"  
  
 _ _ _ _ _  
 _ _ _ _ _ / Amazon Linux 2 AMI  
 _ _ _ _ _  
  
https://aws.amazon.com/amazon-linux-2/  
19 package(s) needed for security, out of 27 available  
Run "sudo yum update" to apply all updates.  
[ec2-user@ip-172-31-7-21 ~]$
```

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

LAB 2: Create repo in local machine

```
ec2-user@ip-172-31-6-246:~/lab2
Complete!
[ec2-user@ip-172-31-6-246 ~]$ ls
[ec2-user@ip-172-31-6-246 ~]$ info

[ec2-user@ip-172-31-6-246 ~]$ mkdir lab2
[ec2-user@ip-172-31-6-246 ~]$ ls
lab2
[ec2-user@ip-172-31-6-246 ~]$ git init lab2
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/lab2/.git/
[ec2-user@ip-172-31-6-246 ~]$ cd lab2
[ec2-user@ip-172-31-6-246 lab2]$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)
[ec2-user@ip-172-31-6-246 lab2]$ touch test
[ec2-user@ip-172-31-6-246 lab2]$ git status
On branch master

No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)
[ec2-user@ip-172-31-6-246 lab2]$ git add test
[ec2-user@ip-172-31-6-246 lab2]$ git status
On branch master

No commits yet

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

[ec2-user@ip-172-31-6-246 lab2]$
```

- 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 is 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.

```
ec2-user@ip-172-31-6-246:~/lab2
[ec2-user@ip-172-31-6-246 ~]$ cd lab2
[ec2-user@ip-172-31-6-246 lab2]$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)
[ec2-user@ip-172-31-6-246 lab2]$ touch test
[ec2-user@ip-172-31-6-246 lab2]$ git status
On branch master

No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)
[ec2-user@ip-172-31-6-246 lab2]$ git add test
[ec2-user@ip-172-31-6-246 lab2]$ git status
On branch master

No commits yet

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

[ec2-user@ip-172-31-6-246 lab2]$ git commit -m "added test files"
[master (root-commit) 9fb9d01] added test files
Committer: EC2 Default User <ec2-user@ip-172-31-6-246.ap-south-1.compute.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:

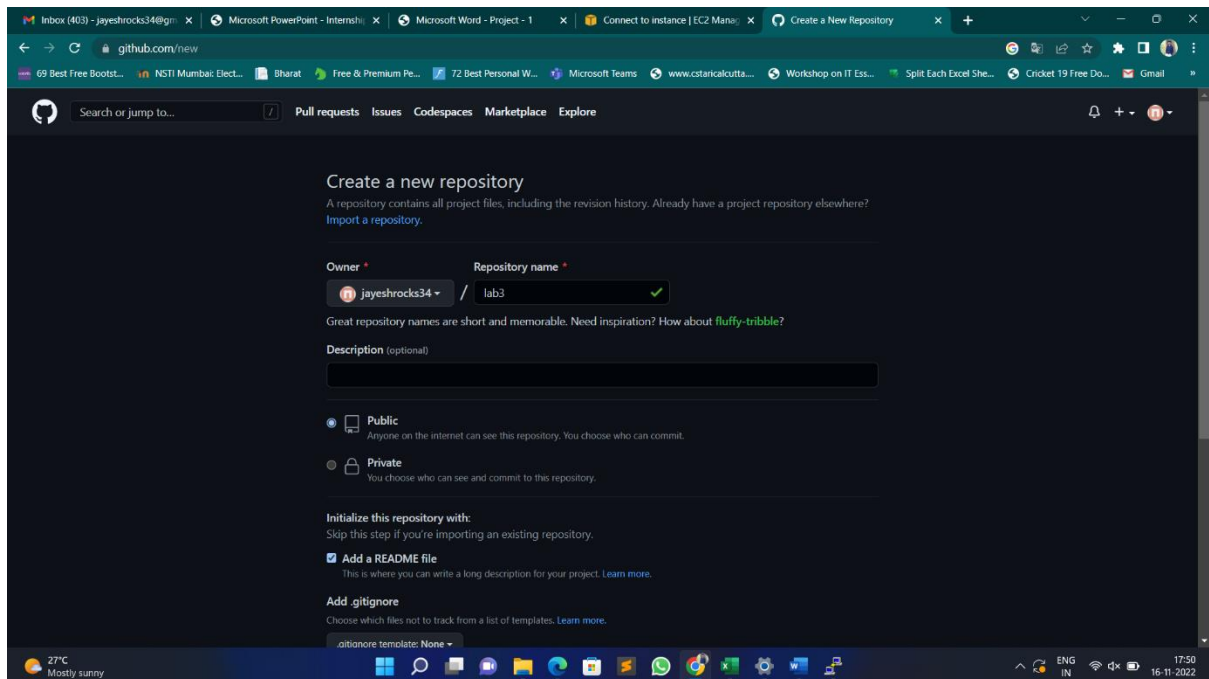
    git commit --amend --reset-author

1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 test
[ec2-user@ip-172-31-6-246 lab2]$ git status
On branch master

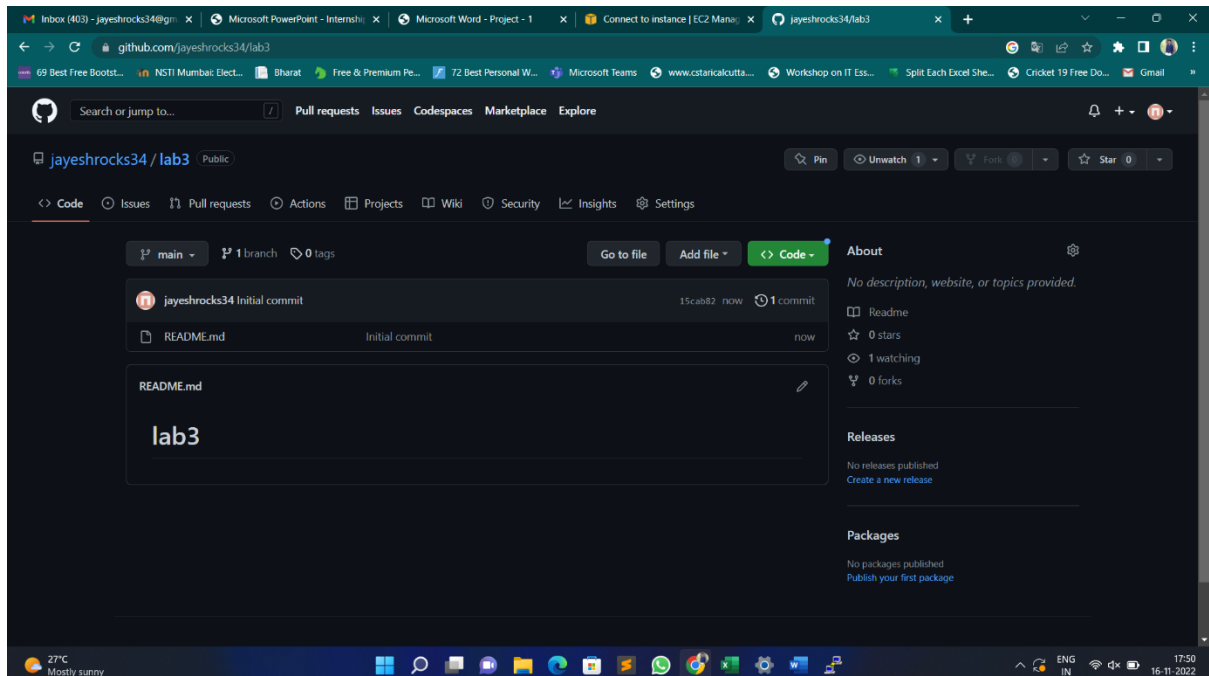
nothing to commit, working tree clean
[ec2-user@ip-172-31-6-246 lab2]$
```

- 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

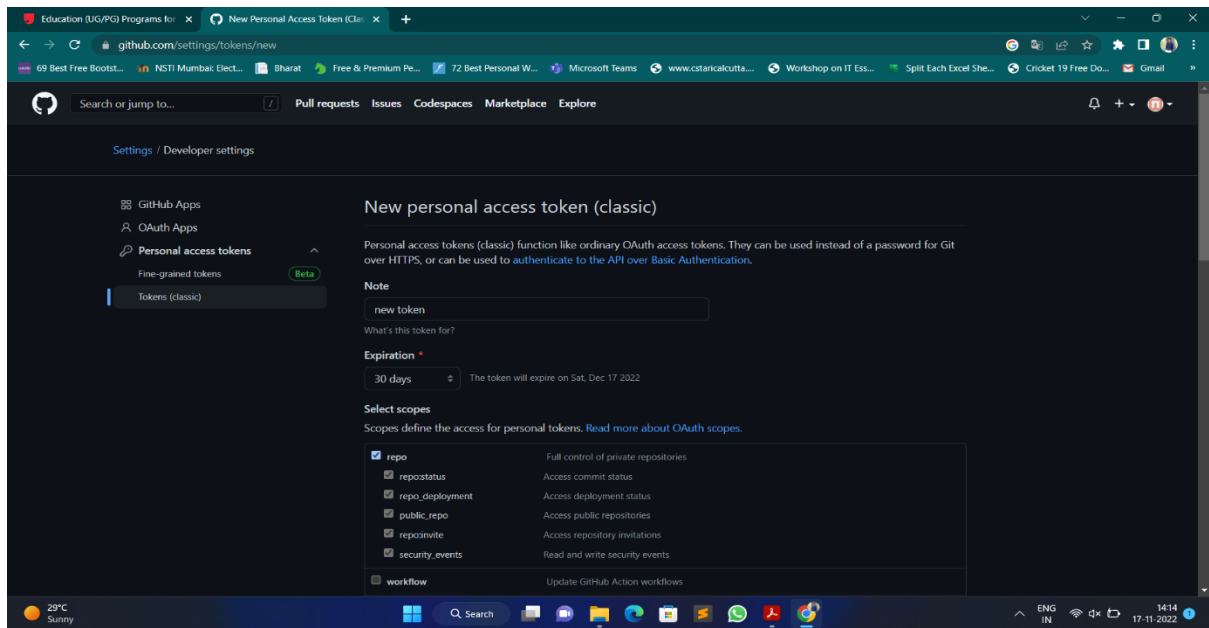


- Logged in to GitHub account using web browser'
- Then created a new Private Repository and added readme file to initialize the repository.

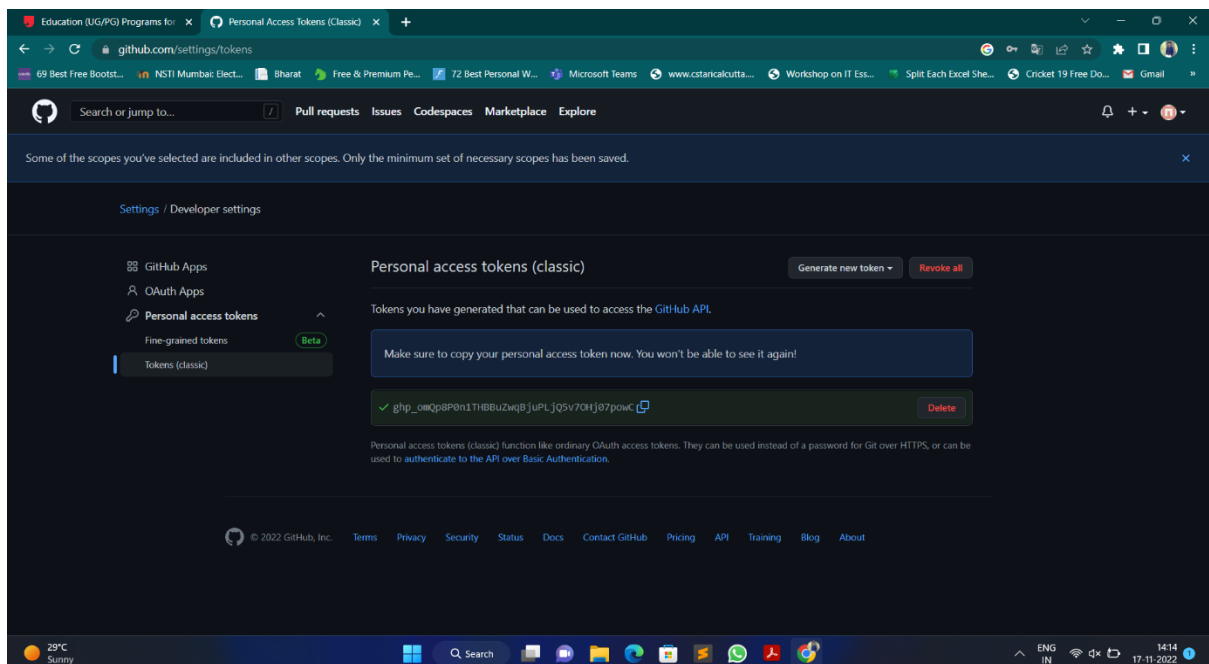


- The Repository is created successfully as can see in above image.

LAB 4: Working with Remote repo



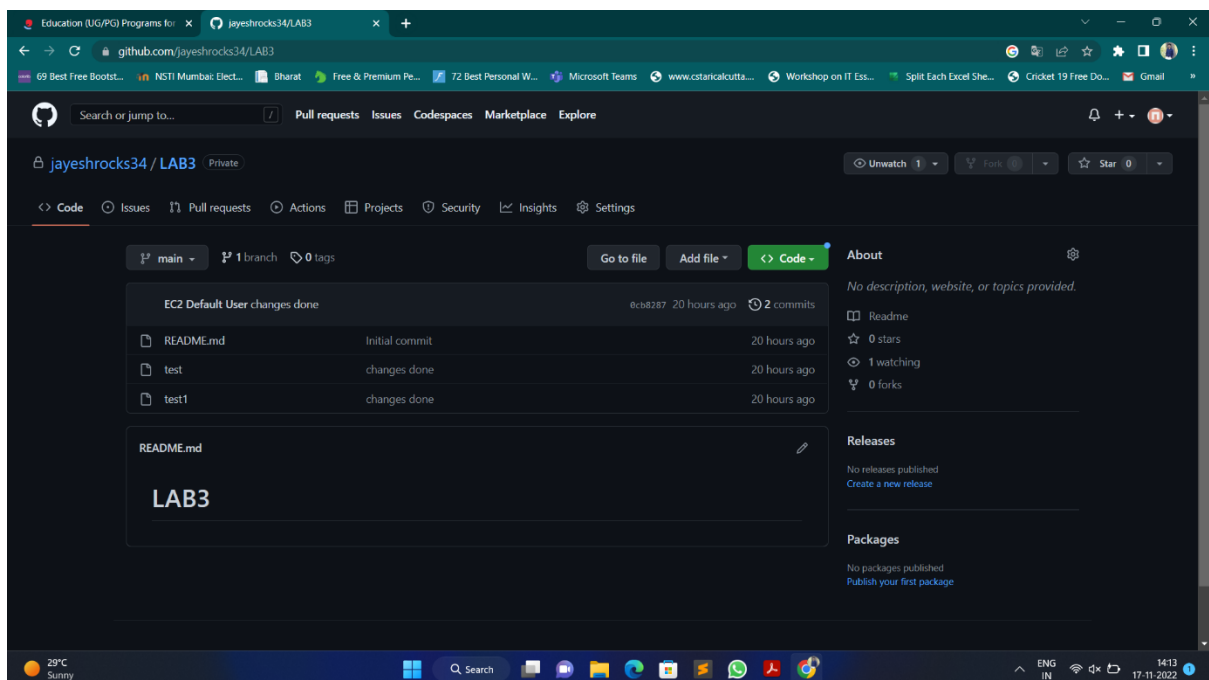
- We need to login to our GitHub account through CLI for cloning our repository, and for that we need to enter our username and password, but your password won't work as per new changes.
- Then for login we need to create "Personal Access Token" to clone this repo.
- To create it Go to Setting → Developer 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.

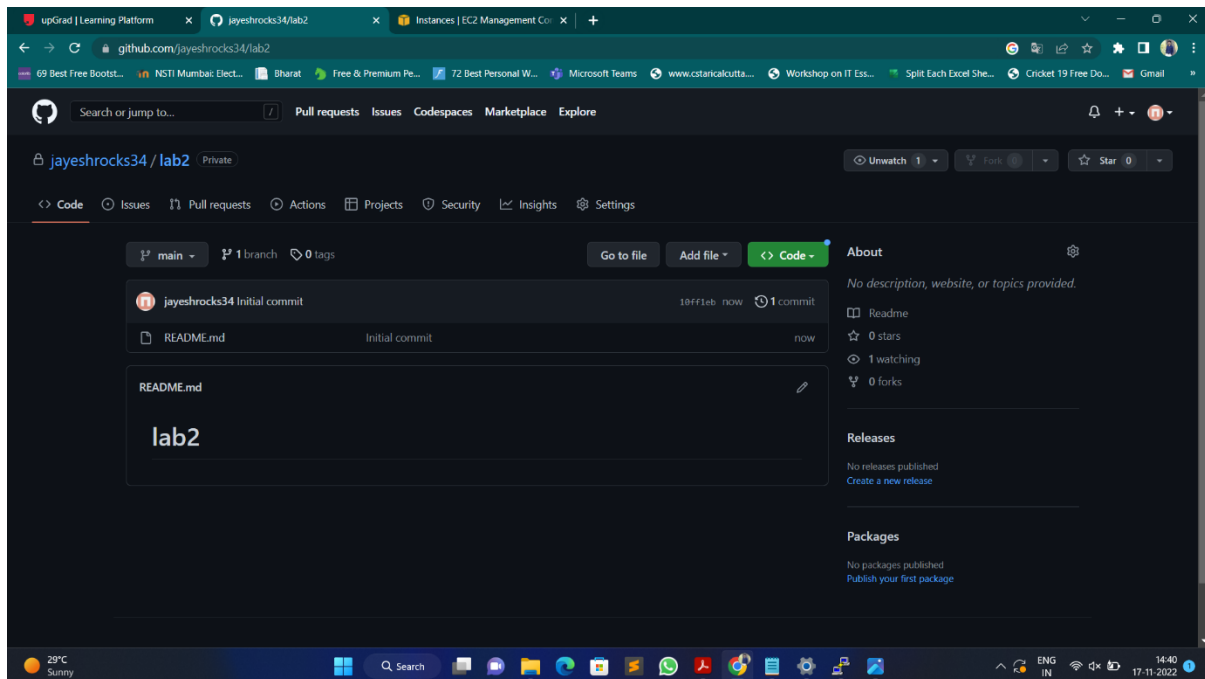
```
ec2-user@ip-172-31-6-246:~$ https://github.com/jayeshrocks34/lab3.git
-bash: https://github.com/jayeshrocks34/lab3.git: No such file or directory
ec2-user@ip-172-31-6-246 lab2]$ https://github.com/jayeshrocks34/lab3.git
-bash: https://github.com/jayeshrocks34/lab3.git: No such file or directory
ec2-user@ip-172-31-6-246 lab2]$ cd
ec2-user@ip-172-31-6-246 ~]$ git clone https://github.com/jayeshrocks34/lab3.git
Cloning into 'lab3'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
ec2-user@ip-172-31-6-246 ~]$
```

- 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.

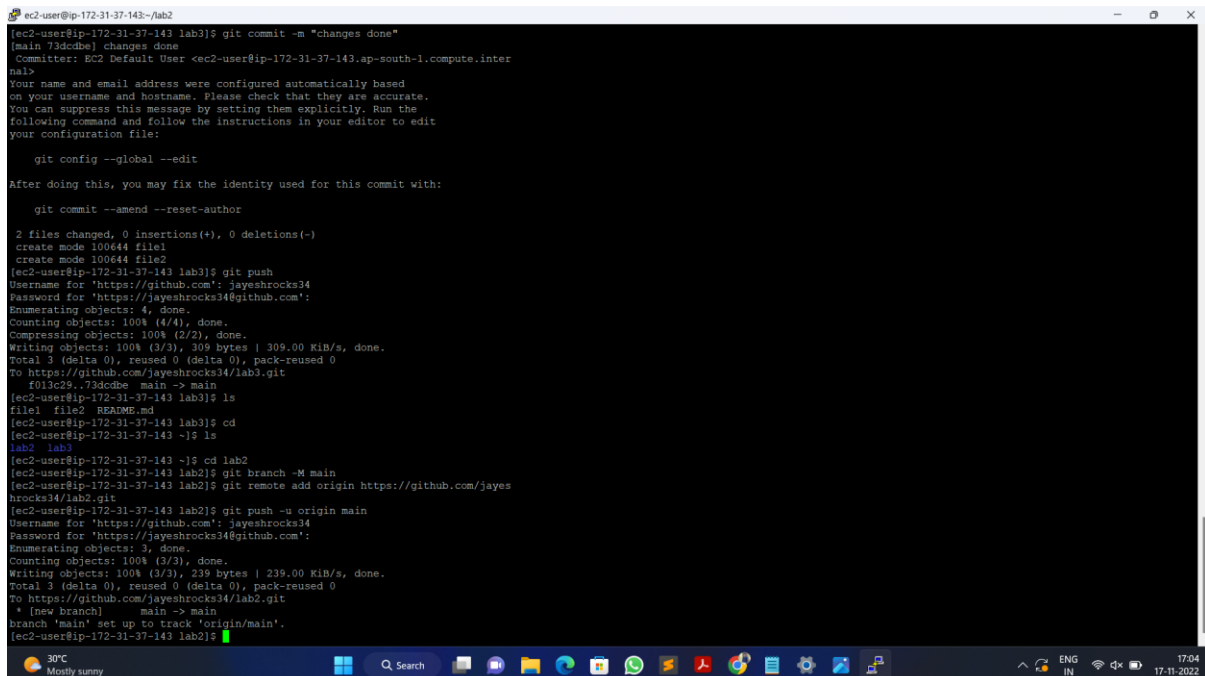


- Then go to the remote repo and see, you will be able to find your new changes here.

LAB 5: Pushing a locally created repo to GitHub

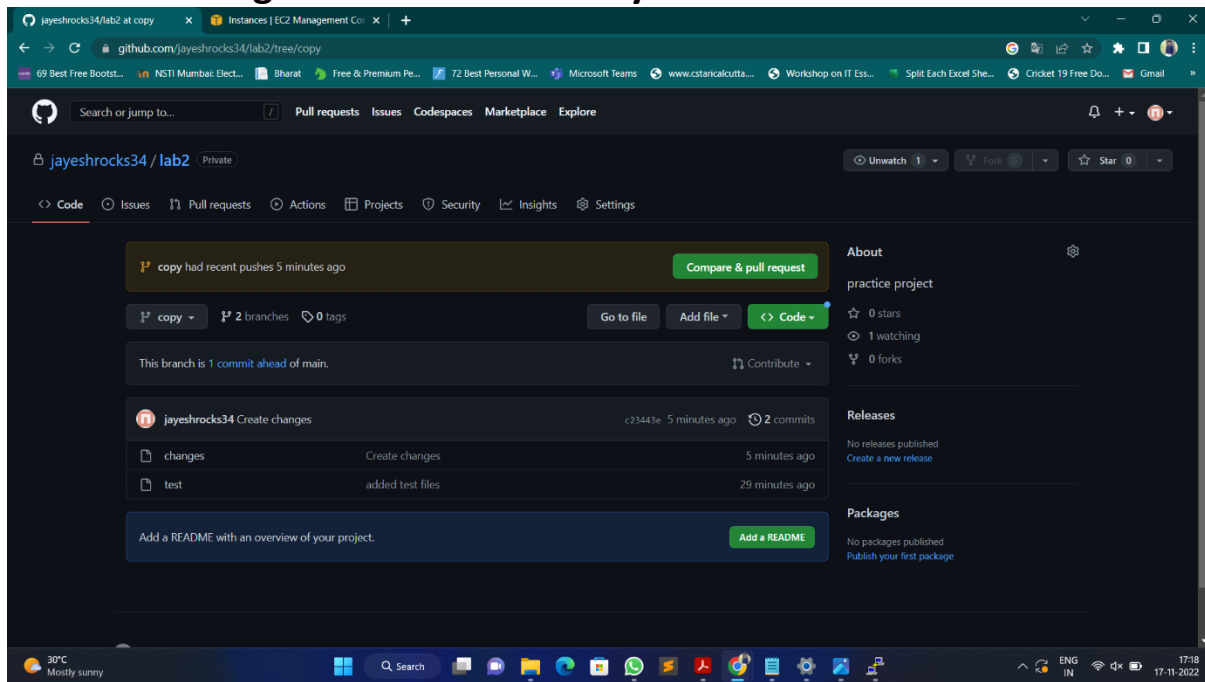


- As we created a local repo in LAB2 with the repo name 'lab2', so now we have created a remote repo with the same name 'lab2'.

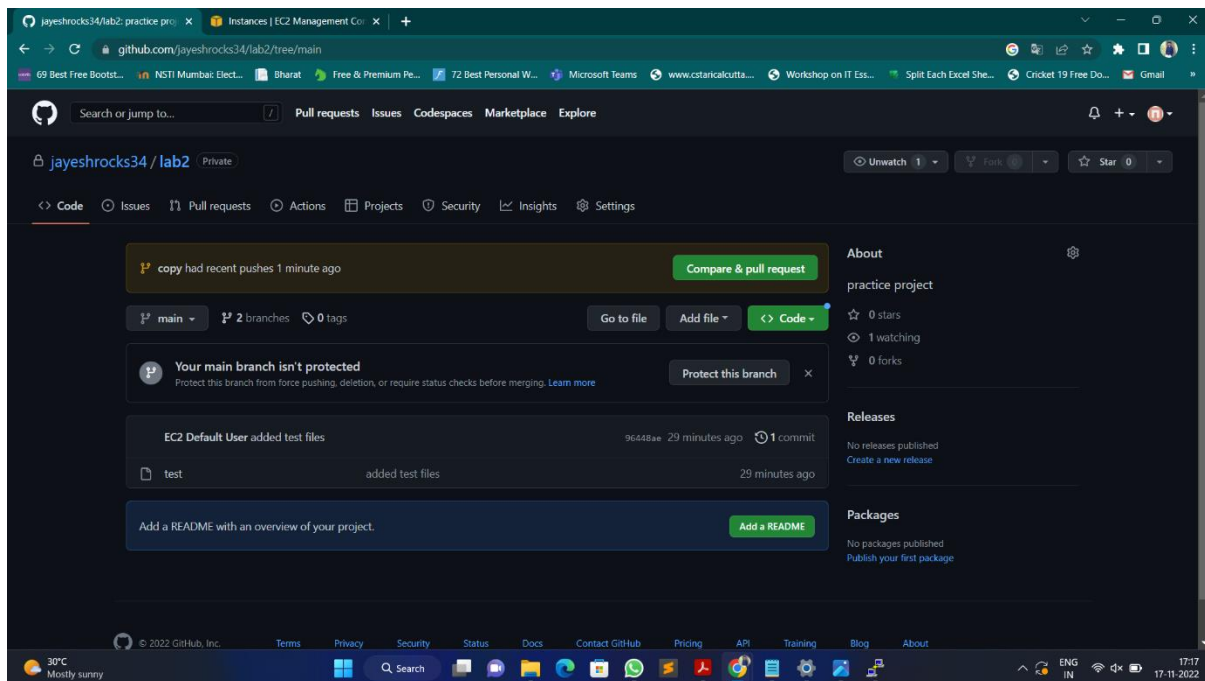


- After creating we went to our local machine inside the 'lab2' repo and used "git branch -M main" command to change the name of branch as master branch is now known as main branch.
- Then we used "git remote add origin <URL of your remote repo>" command.
- Used "git push -u origin main" command to push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

```
ec2-user@ip-172-31-37-143:~/lab2
[ec2-user@ip-172-31-37-143 ~]$ ls
lab2  lab3
[ec2-user@ip-172-31-37-143 ~]$ git pull
fatal: not a git repository (or any of the parent directories): .git
[ec2-user@ip-172-31-37-143 ~]$ cd lab3
[ec2-user@ip-172-31-37-143 lab3]$ git pull
Username for 'https://github.com': jayeshrocks34
Password for 'https://jayeshrocks34@github.com':
Already up to date.
[ec2-user@ip-172-31-37-143 lab3]$ ls
file1  file2  README.md
[ec2-user@ip-172-31-37-143 lab3]$ cd
[ec2-user@ip-172-31-37-143 ~]$ ls
lab2  lab3
[ec2-user@ip-172-31-37-143 ~]$ cd lab2
[ec2-user@ip-172-31-37-143 lab2]$ git pull
Username for 'https://github.com': jayeshrocks34
Password for 'https://jayeshrocks34@github.com':
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (6/6), 1.33 KiB | 1.33 MiB/s, done.
From https://github.com/jayeshrocks34/lab2
* [new branch]      copy      -> origin/copy
Already up to date.
[ec2-user@ip-172-31-37-143 lab2]$ git branch -a
* main
  remotes/origin/copy
  remotes/origin/main
[ec2-user@ip-172-31-37-143 lab2]$
```

- 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.

```
ec2-user@ip-172-31-37-143:~/lab2
* [new branch]      copy      -> origin/copy
Already up to date.
[ec2-user@ip-172-31-37-143 lab2]$ git branch -a
* main
  remotes/origin/copy
  remotes/origin/main
[ec2-user@ip-172-31-37-143 lab2]$ git checkout copy
branch 'copy' set up to track 'origin/copy'.
Switched to a new branch 'copy'
[ec2-user@ip-172-31-37-143 lab2]$ git status
On branch copy
Your branch is up to date with 'origin/copy'.

nothing to commit, working tree clean
[ec2-user@ip-172-31-37-143 lab2]$ git branch -a
* copy
  main
  remotes/origin/copy
  remotes/origin/main
[ec2-user@ip-172-31-37-143 lab2]$ git checkout copy
Already on 'copy'
Your branch is up to date with 'origin/copy'.
[ec2-user@ip-172-31-37-143 lab2]$ git branch
* copy
  main
[ec2-user@ip-172-31-37-143 lab2]$ git status
On branch copy
Your branch is up to date with 'origin/copy'.

nothing to commit, working tree clean
[ec2-user@ip-172-31-37-143 lab2]$
```

- 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.

```
ec2-user@ip-172-31-37-143:~/lab2
* copy
main
[ec2-user@ip-172-31-37-143 lab2]$ git status
On branch copy
Your branch is up to date with 'origin/copy'.

nothing to commit, working tree clean
[ec2-user@ip-172-31-37-143 lab2]$ touch file3 file4
[ec2-user@ip-172-31-37-143 lab2]$ ls
changes changes1 file3 file4 test
[ec2-user@ip-172-31-37-143 lab2]$ git status
On branch copy
Your branch is up to date with 'origin/copy'.

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

nothing added to commit but untracked files present (use "git add" to track)
[ec2-user@ip-172-31-37-143 lab2]$ git add file3 file4
[ec2-user@ip-172-31-37-143 lab2]$ git status
On branch copy
Your branch is up to date with 'origin/copy'.

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

[ec2-user@ip-172-31-37-143 lab2]$
```

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

```
ec2-user@ip-172-31-37-143:~/lab2
        new file:   file4

[ec2-user@ip-172-31-37-143 lab2]$ git commit -m "changes are done"
[copy 95afb80] changes are done
Committer: EC2 Default User <ec2-user@ip-172-31-37-143.ap-south-1.compute.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:

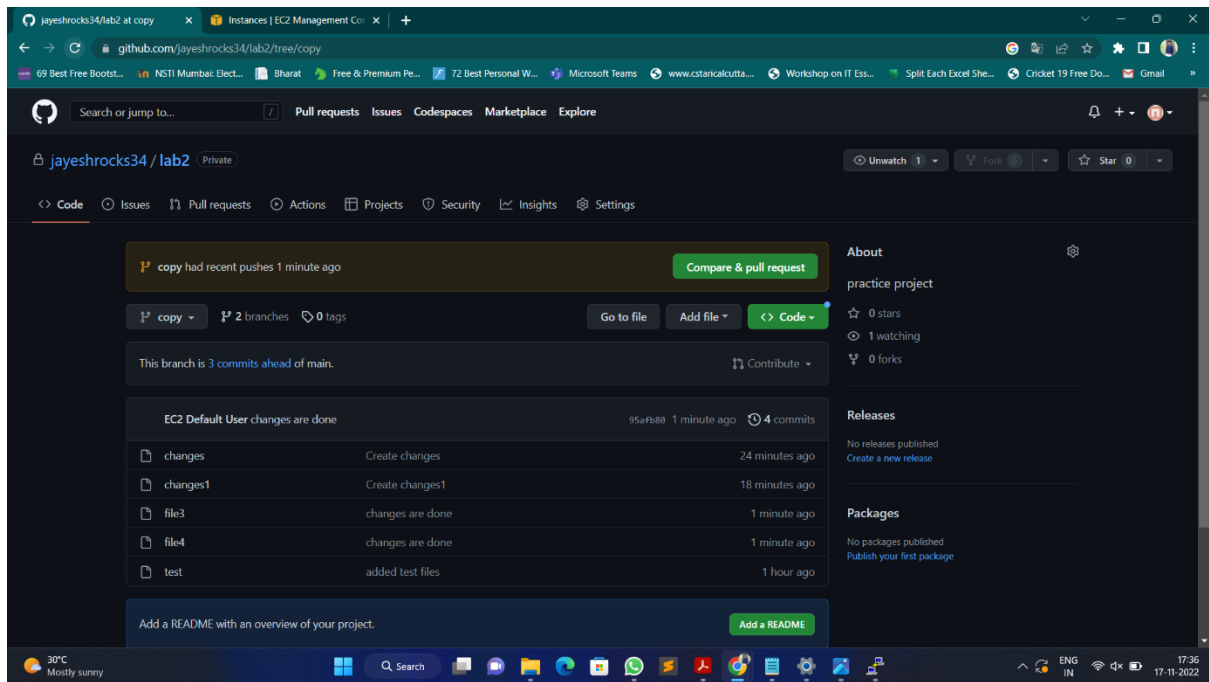
    git config --global --edit

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

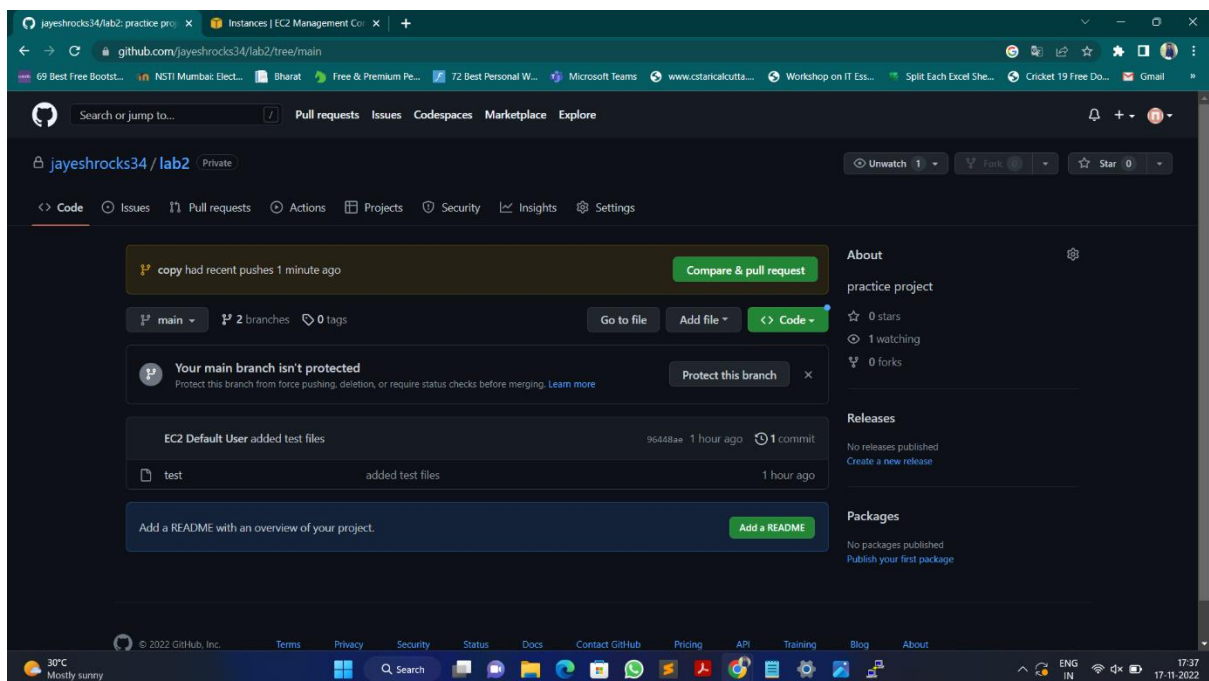
    git commit --amend --reset-author

2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file3
create mode 100644 file4
[ec2-user@ip-172-31-37-143 lab2]$ git push
Username for 'https://github.com': jayeshrocks34
Password for 'https://jayeshrocks34@github.com':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 331 bytes | 331.00 KiB/s, done.
Total 2 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/jayeshrocks34/lab2.git
   89831ed..95afb80  copy -> copy
[ec2-user@ip-172-31-37-143 lab2]$
```

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

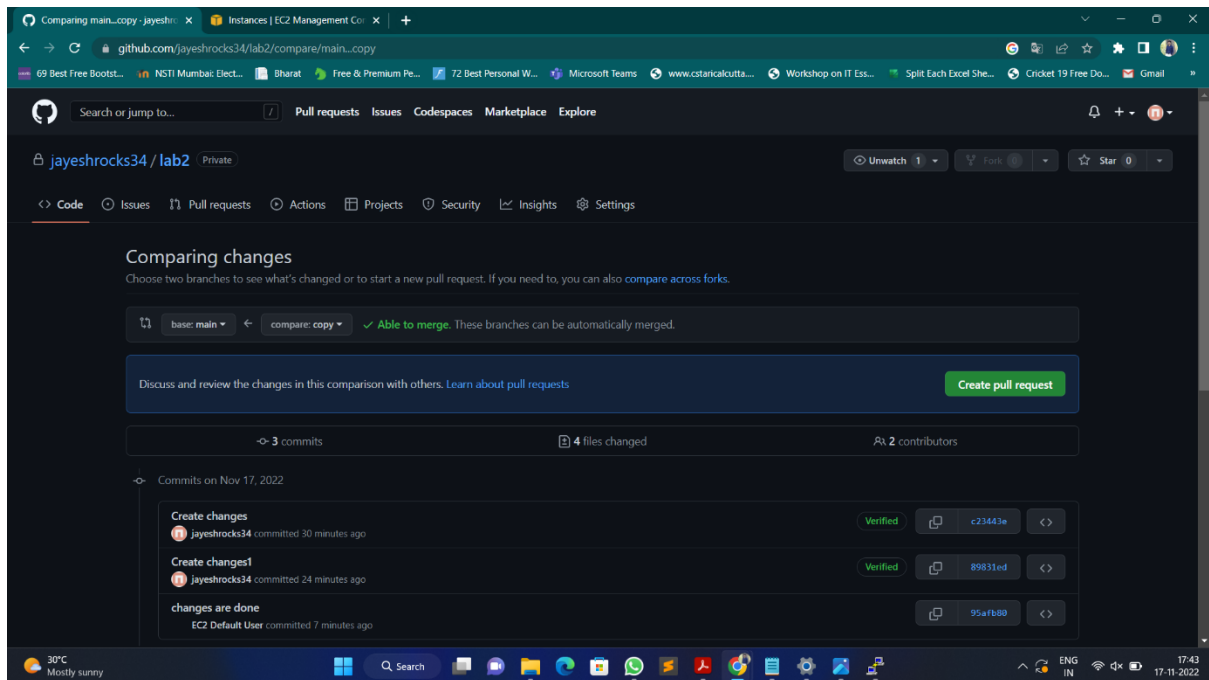


- We checked on browser the files are successfully pushed as we can see in the above image.
- The files are pushed to our new branch.

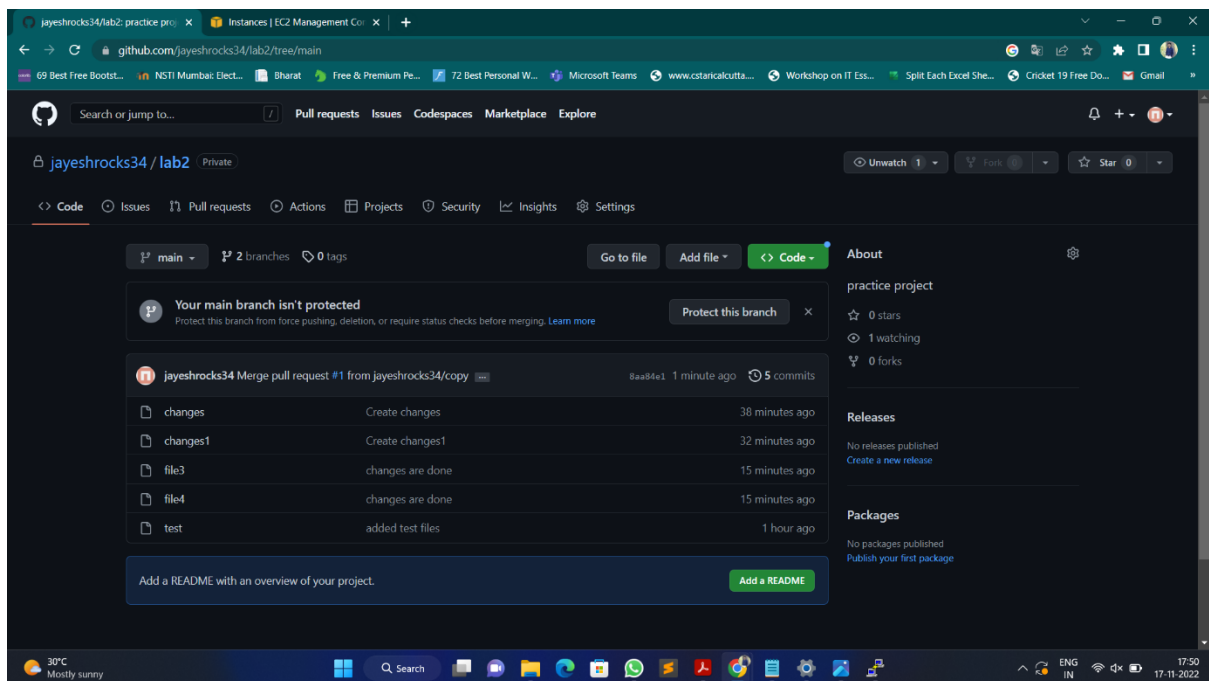


- The changes are not applied to the main branch.

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

```
ec2-user@ip-172-31-37-143:~/lab2
[ec2-user@ip-172-31-37-143 lab2]$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
[ec2-user@ip-172-31-37-143 lab2]$ git branch
  copy
* main
[ec2-user@ip-172-31-37-143 lab2]$ ls
test
[ec2-user@ip-172-31-37-143 lab2]$ git pull
Username for 'https://github.com': jayeshrocks34
Password for 'https://jayeshrocks34@github.com':
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), 625 bytes | 625.00 KiB/s, done.
From https://github.com/jayeshrocks34/lab2
   96448ae..8aa84e1  main       -> origin/main
Updating 96448ae..8aa84e1
Fast-forward
 changes | 1 +
 changes1 | 1 +
 file3   | 0
 file4   | 0
4 files changed, 2 insertions(+)
create mode 100644 changes
create mode 100644 changes1
create mode 100644 file3
create mode 100644 file4
[ec2-user@ip-172-31-37-143 lab2]$ ls
changes changes1 file3 file4 test
[ec2-user@ip-172-31-37-143 lab2]$
```

- 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.