Git – Global Information Tracker

Git is a distributed, open-source version control system (VCS) that enables you to store code, track revision history, merge code changes, and revert to earlier code version when needed.

Git stores your source code and its full development history locally in a repository.

--->Version control system(vcs) or Source code management(scm)

Version control system(vcs)

Version control systems allow multiple developers, designers, and team members to work together on the same project.

It helps them work smarter and faster! A version control system is critical to ensure everyone has access to the latest code and modifications are tracked.

Two types:

- ➢ Centralized Version control system
- ➢ Distributed Version control system

Source code management(scm)

Source code management (SCM) is used to track modifications to a source code repository.

SCM tracks a running history of changes to a code base and helps resolve conflicts when merging updates from multiple contributors. SCM is also synonymous with Version control.
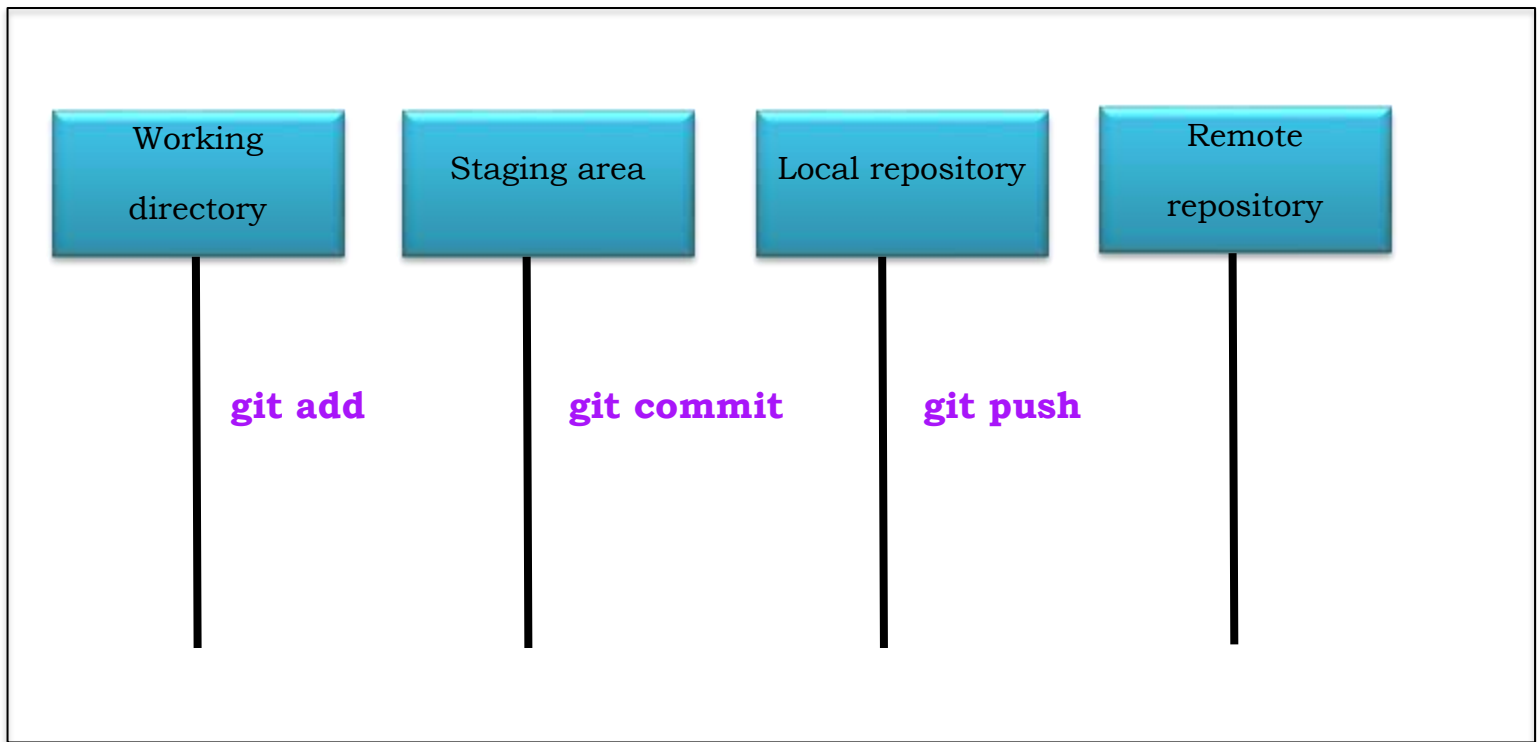
Working directory

   The Git directory is where Git stores the metadata and object database for your project.

Staging area

   It is temporary location
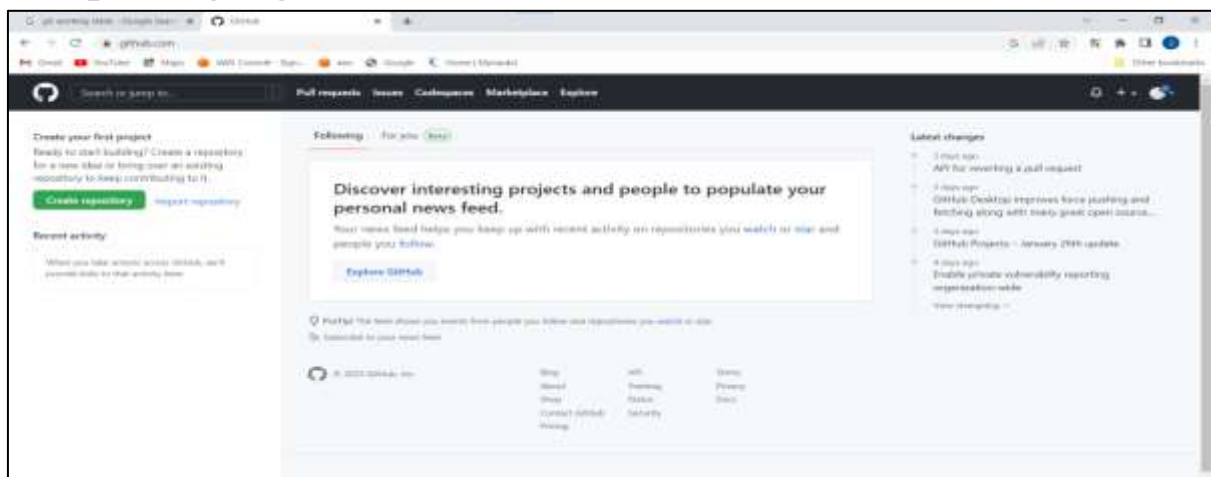
Local repository

   Local repositories reside on the computers of team members.

Remote repositary
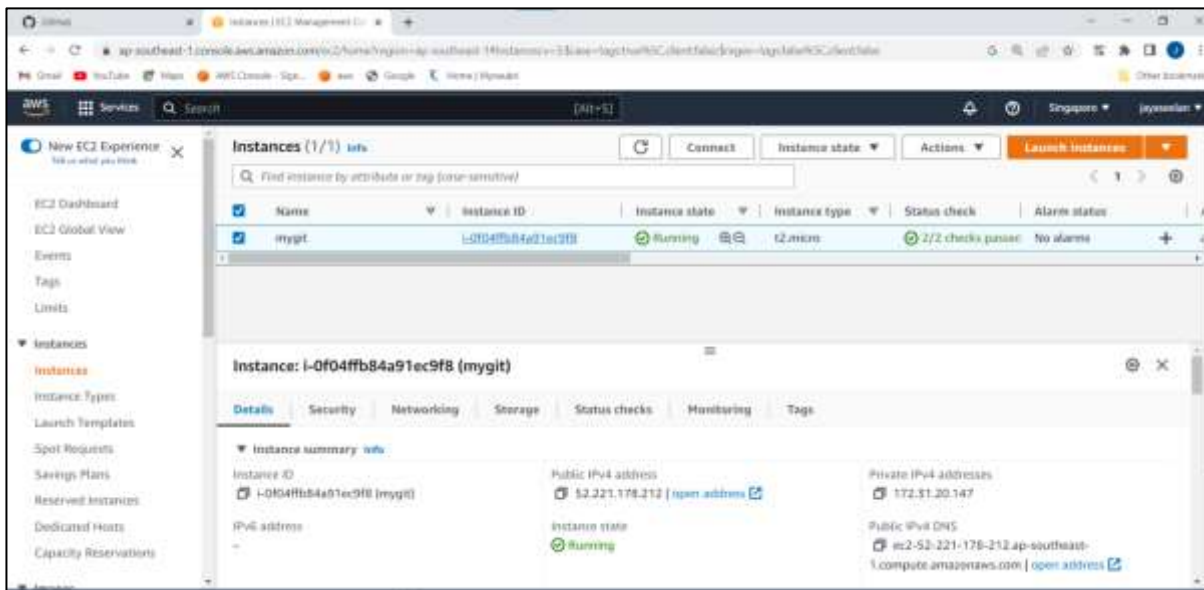
   Remote repositories are versions of your project that are hosted on the Internet or network somewhere
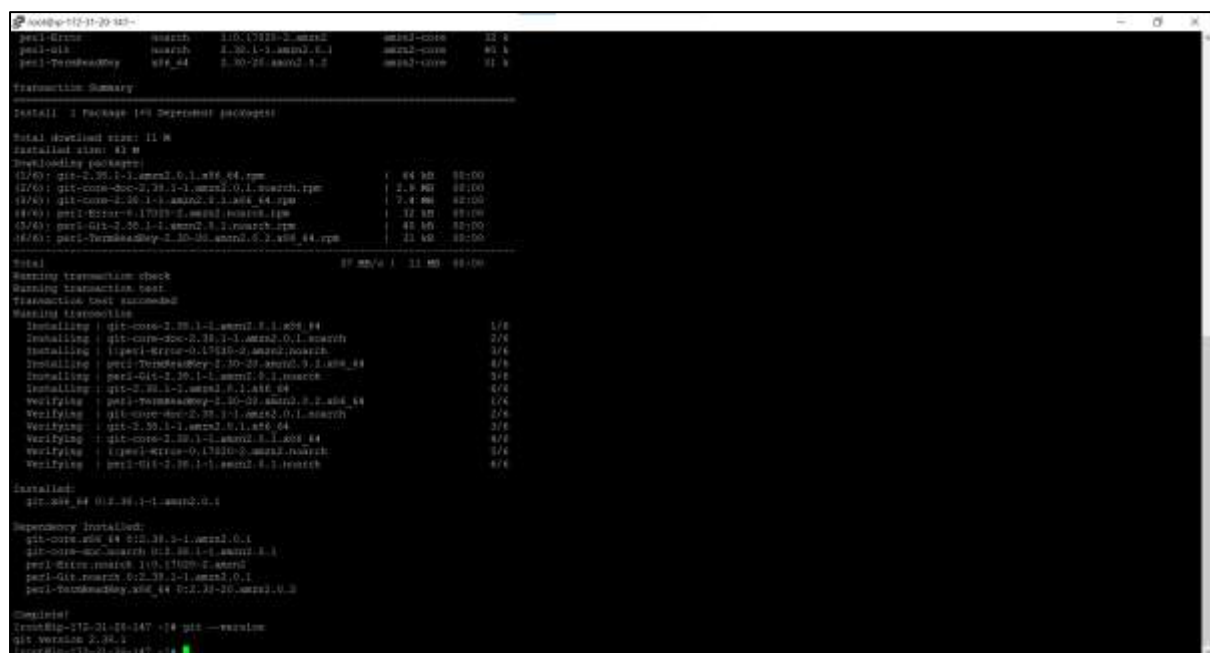
Steps to create Git

Step 1: login gitub

Step2: create linux server----->name(my git)---->security group(ssh)----->launch instance

Step 3: login linux server ----->#sudo –i

#yum install git –y

#git --version ----->(git install or not check)

Step 4: create working directory

#mkdir mygit

#cd mygit

```
Installed:
  git.x86_64 0:2.38.1-1.amzn2.0.1

Dependency Installed:
  git-core.x86_64 0:2.38.1-1.amzn2.0.1
  git-core-doc.noarch 0:2.38.1-1.amzn2.0.1
  perl-Error.noarch 1:0.17020-2.amzn2
  perl-Git.noarch 0:2.38.1-1.amzn2.0.1
  perl-TermReadKey.x86_64 0:2.30-20.amzn2.0.2

Complete!
[root@ip-172-31-20-147 ~]# git --version
git version 2.38.1
[root@ip-172-31-20-147 ~]# mkdir mygit
[root@ip-172-31-20-147 ~]# cd  mygit
[root@ip-172-31-20-147 mygit]#
```

Step 5: create local repositary

#git config --global user.name "jayaseelan4053"

#git config --global user.email "jayasee4053@gmail.com"

#git config --global core.editor vim

#git config --global core.compression 2

#git config --global diff.tool vim.diff

&

#git config –l ----> Git config check

```
Installed:
  git.x86_64 0:2.38.1-1.amzn2.0.1

Dependency Installed:
  git-core.x86_64 0:2.38.1-1.amzn2.0.1
  git-core-doc.noarch 0:2.38.1-1.amzn2.0.1
  perl-Error.noarch 1:0.17020-2.amzn2
  perl-Git.noarch 0:2.38.1-1.amzn2.0.1
  perl-TermReadKey.x86_64 0:2.30-20.amzn2.0.2

Complete!
[root@ip-172-31-20-147 ~]# git --version
git version 2.38.1
[root@ip-172-31-20-147 ~]# mkdir mygit
[root@ip-172-31-20-147 ~]# cd  mygit
[root@ip-172-31-20-147 mygit]# git config --global user.name  "jayaseelan4053"
[root@ip-172-31-20-147 mygit]# git config --global user.email "jayasee4053@gmail.com"
[root@ip-172-31-20-147 mygit]# git config --global core.editor vim
[root@ip-172-31-20-147 mygit]# git config --global core.compression 2
[root@ip-172-31-20-147 mygit]# git config --global diff.tool vim.diff
[root@ip-172-31-20-147 mygit]# git config -l
user.name="jayaseelan4053"
user.email="jayasee4053@gmail.com"
core.editor=vim
core.compression=2
diff.tool=vim.diff
[root@ip-172-31-20-147 mygit]#
```

Step 6: create global repositary

    # git init

    #ls –larth ------>shown hidden folder(.git)

```
[root@ip-172-31-20-147 ~]# ls
mygit
[root@ip-172-31-20-147 ~]# cd mygit
[root@ip-172-31-20-147 mygit]# git init
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 /root/mygit/.git/
[root@ip-172-31-20-147 mygit]# ls
 login as: ec2-user
 Authenticating with public key "moni123"
Last login: Mon Jan 30 18:27:19 2023 from 157.51.20.176


    _|  _|_  )
    _|  (     /    Amazon Linux 2 AMI
    _|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-20-147 ~]$ sudo -i
[root@ip-172-31-20-147 ~]# ls
mygit
[root@ip-172-31-20-147 ~]# cd mygit
[root@ip-172-31-20-147 mygit]# ls -larth
total 0
dr-xr-x--- 4 root root 134 Jan 30 18:21 ..
drwxr-xr-x 7 root root 119 Jan 30 18:27 .git
drwxr-xr-x 3 root root  18 Jan 30 18:27 .
[root@ip-172-31-20-147 mygit]#
```

Step 7: create source code(file)

    #vi sample.txt ----->put content **1**

```
[root@ip-172-31-20-147 ~]# ls
mygit
[root@ip-172-31-20-147 ~]# cd mygit
[root@ip-172-31-20-147 mygit]# ls -larth
total 0
dr-xr-x--- 4 root root 134 Jan 30 18:21 ..
drwxr-xr-x 7 root root 119 Jan 30 18:27 .git
drwxr-xr-x 3 root root  18 Jan 30 18:27 .
[root@ip-172-31-20-147 mygit]# vi sample.txt
[root@ip-172-31-20-147 mygit]# cat sample.txt
1
[root@ip-172-31-20-147 mygit]#
```

Step 8:staging area add

#git add sample.txt

Step 9:push local repository

#git commit –m "1 commit"

#git log ------->(show log details)

Step 10:create repositary

Gitub ---->create repository--->repository name(seelan)--->public--->create repositary
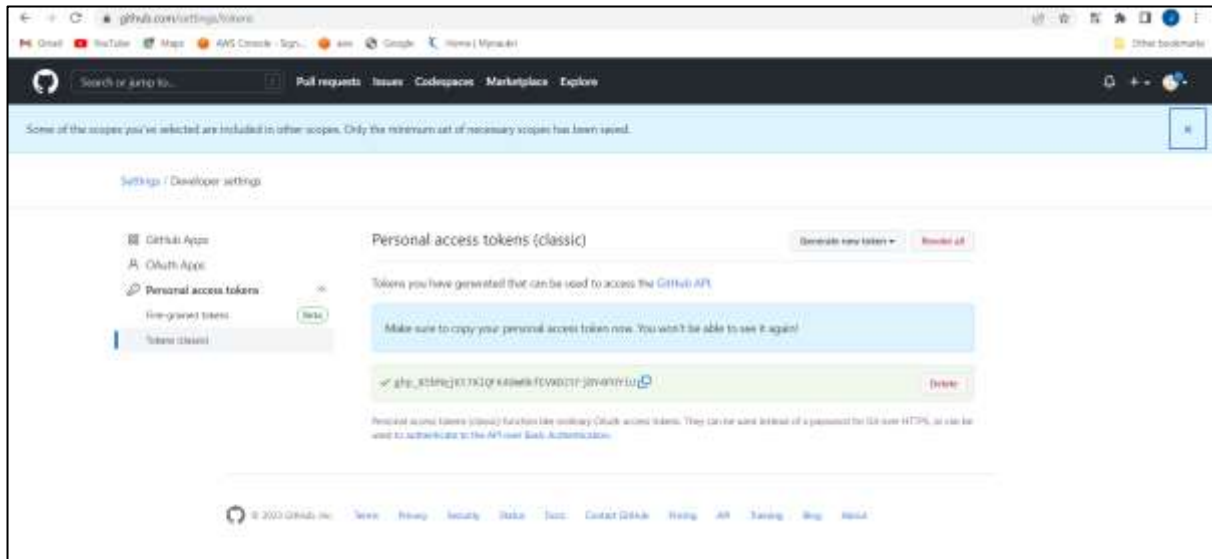
repository created..

Copy url and paste seperatly in notepad

Now mapping repository url in variable

#git remote add jayasee https://github.com/jayaseelan4053/seelan.git

Step 11:token generation

Gitub home page--->setting---->developer setting---->personal acces token--->token(classic)--->generate new token--->name(my token) --->click all check box---->generate token
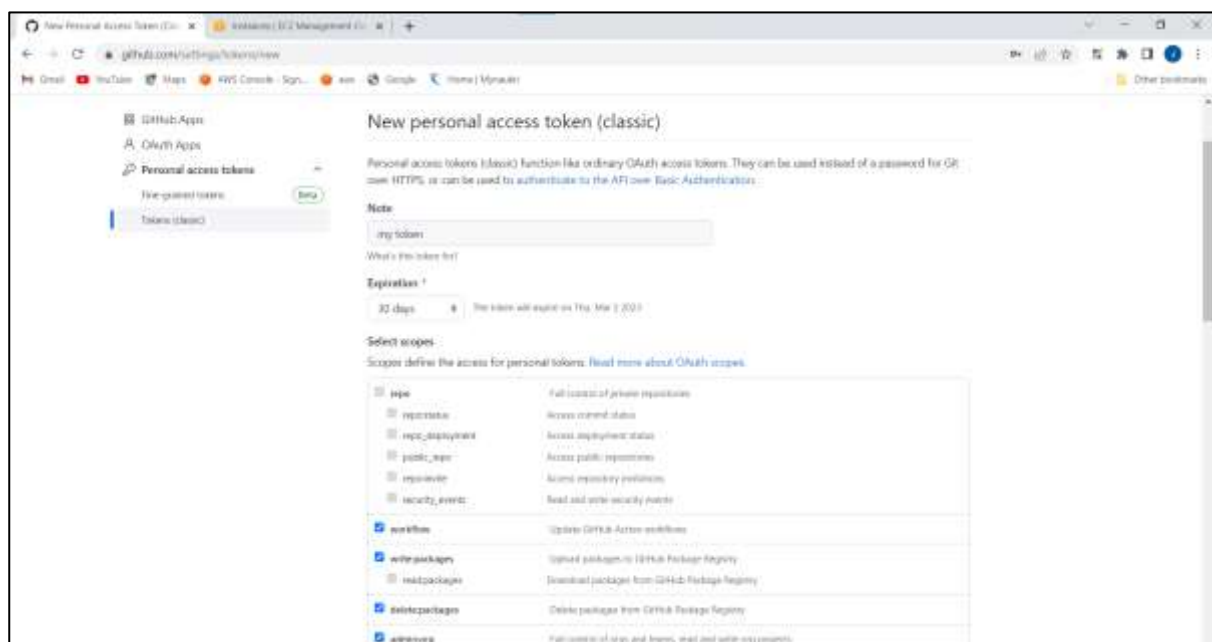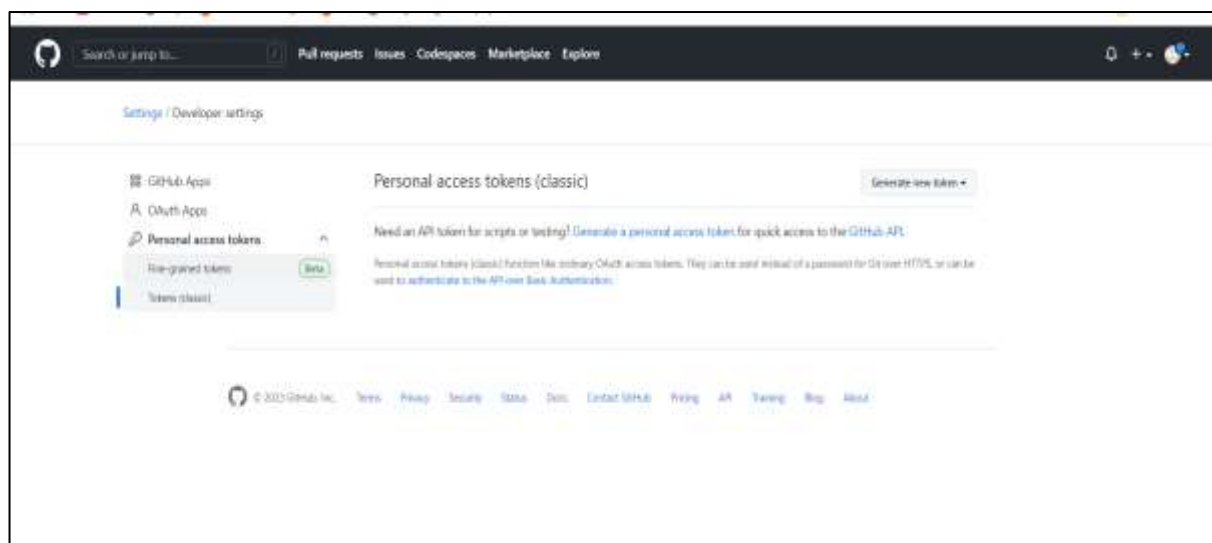
Token generated..

Copy token url separately notepad

Check git branch ---->#git branch-->(it shown master) because master is main branch

Step 11:push remote repository ➔ **repository variable**

   #git push jayasee master

Ask username and password

Username:jayaseelan4053

Password(token url): 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

Now go gitub in ur repository it will shown ur file
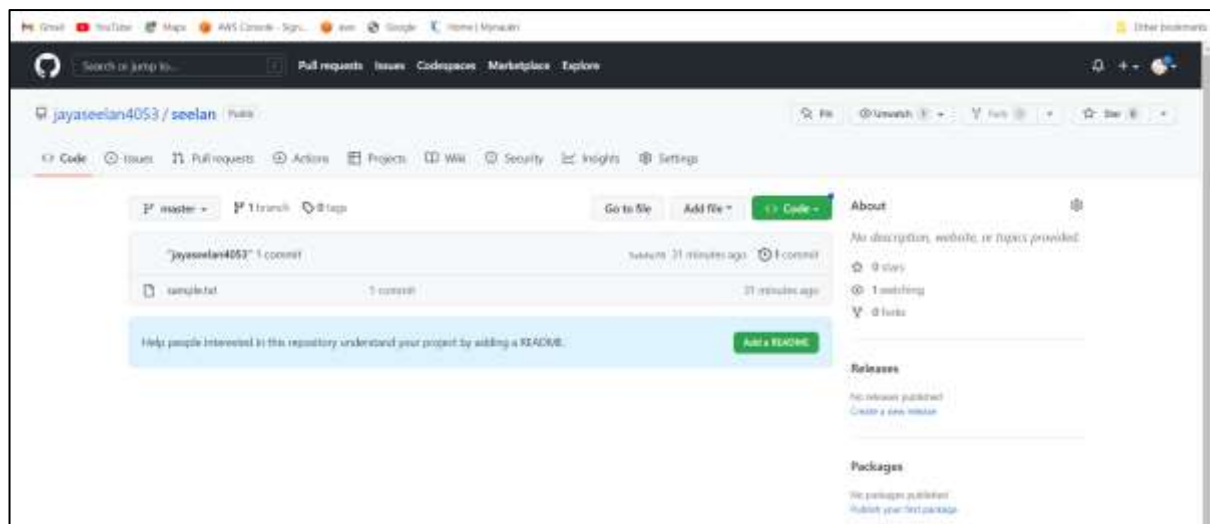
Click 1 commit it will show green color that mean this is running
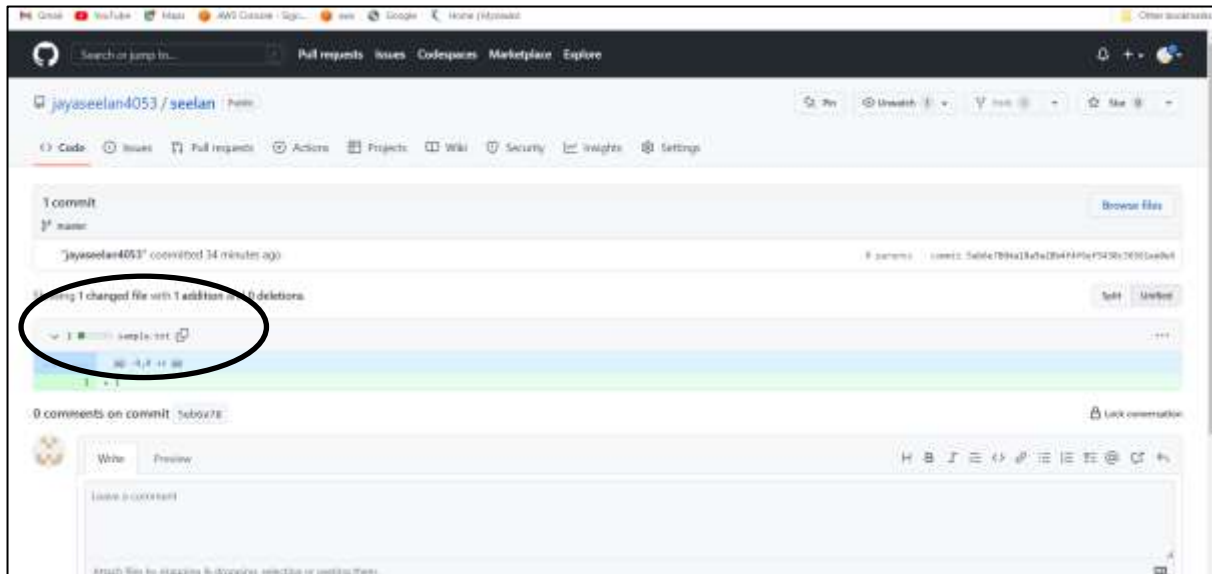
Step 11:add content for same file --->2 and 3

 # vi sample.txt --->add 2 & 3

 #git commit –am "2 commit"------>add and local repository push

 #git log ---->git status check(it will show 2 commit because 2 line add)

#git push jayasee master ----push remote repository

Give username and password(token)

Now go gitub in ur repository it will shown ur file

It will shown 2 commit
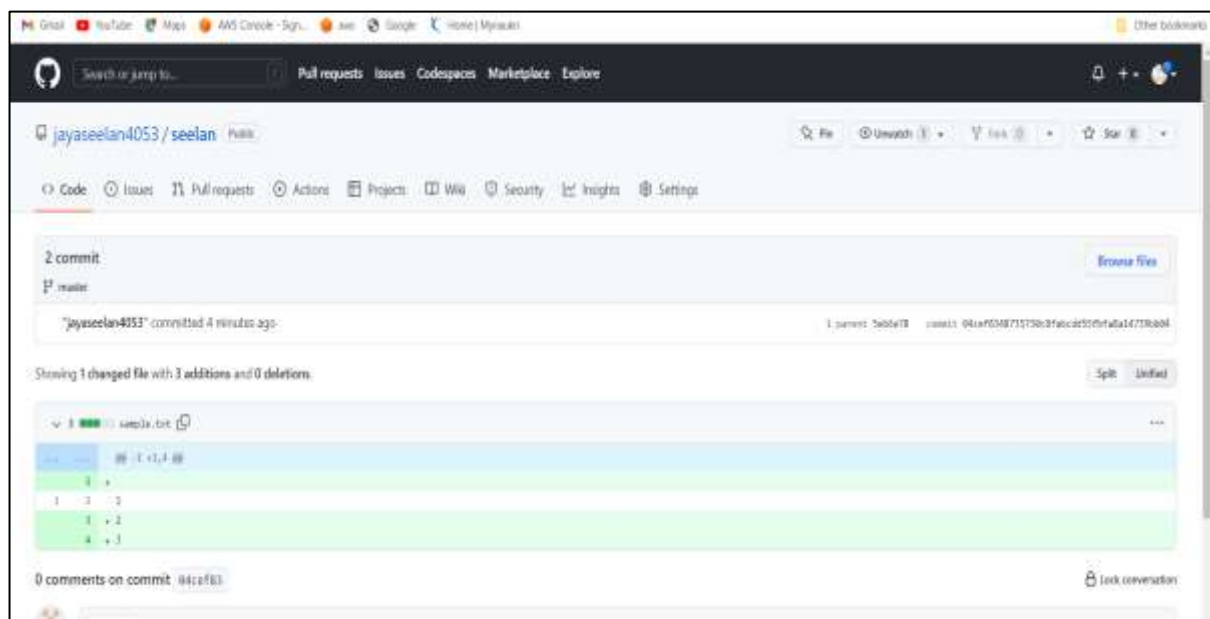
Step 12: Now remove one line in ur file

   #vi sample.txt ---->remove 2nd line

#git commit –am "3 commit"------>add and local repository push

 #git log ---->git status check (it will show 3 commit)

#git push jayasee master ----push remote repository

Give username and password(token)

Now go gitub in ur repository it will shown ur file

It will shown 3 commit
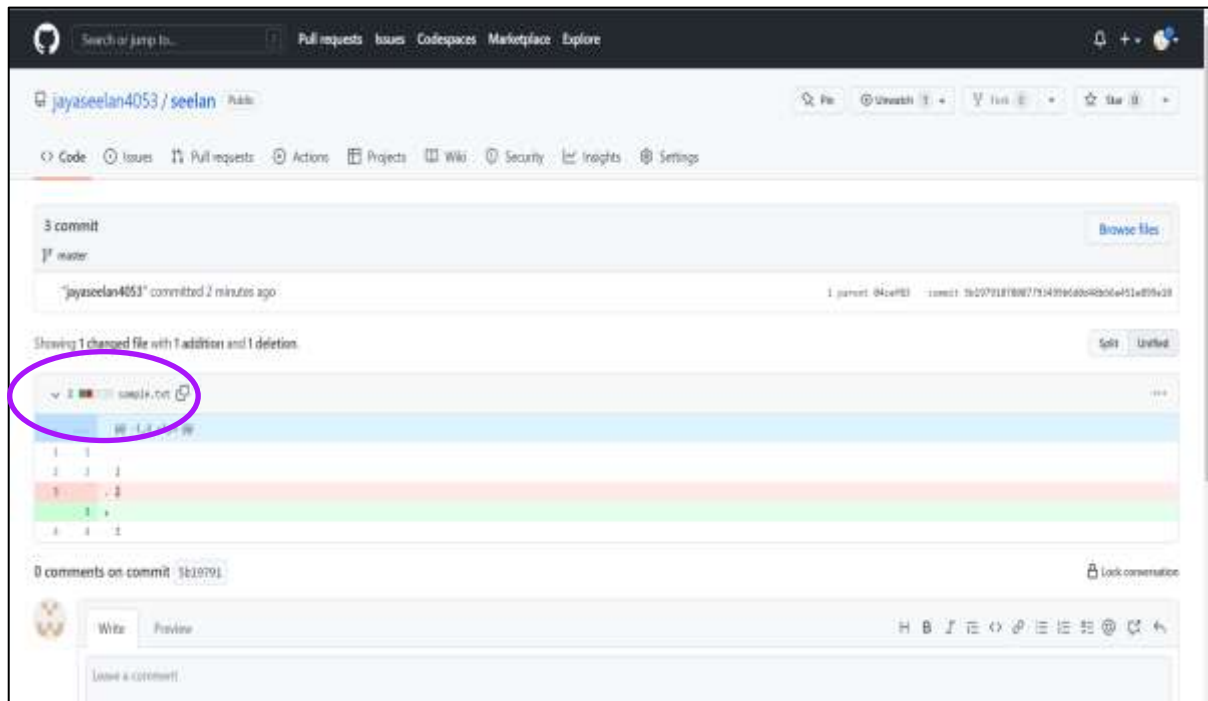
One line in file deleted so it will be show red color.