AWS DevOps

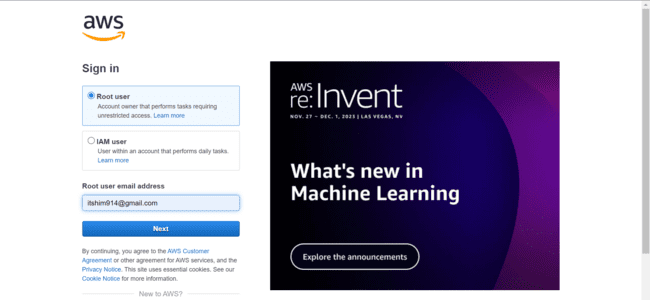
INTERNSHIP

MINI PROJECT - 2

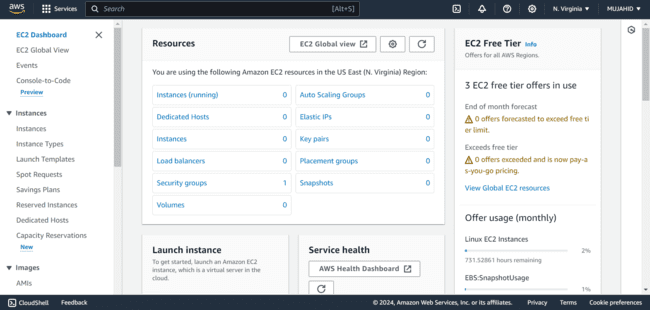
MINI PROJECT – 2

LAB – 1 [PROVISIONING EC2 INSTANCE]

* *Log into AWS management console.*

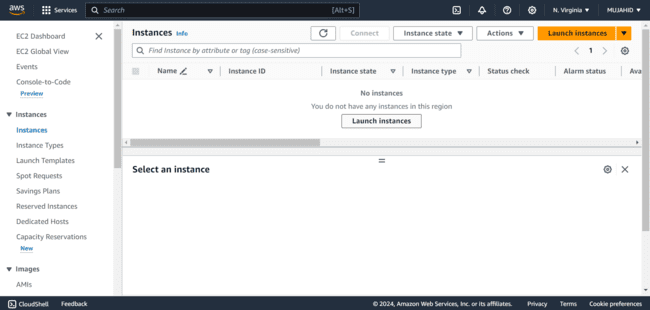


* *I logged in with my AWS credentials and entered into my AWS management console.*
* *Search for EC2 in the services search bar. Opens EC2 dashboard.*

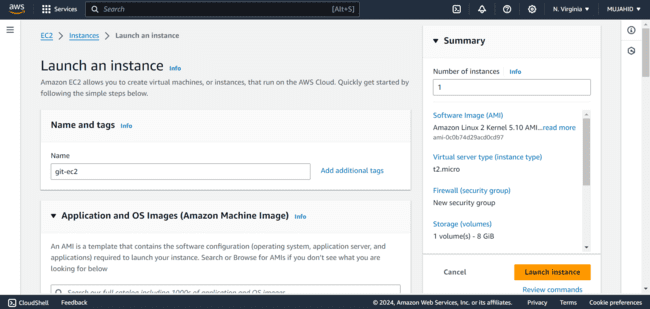


***Provisioning an Instance:***

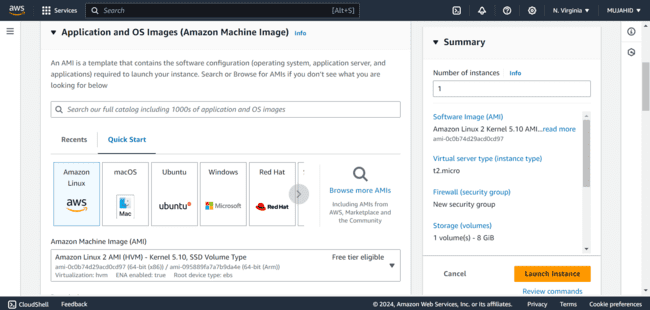
* *Click on Instances and click on Launch Instances on the top right corner.*



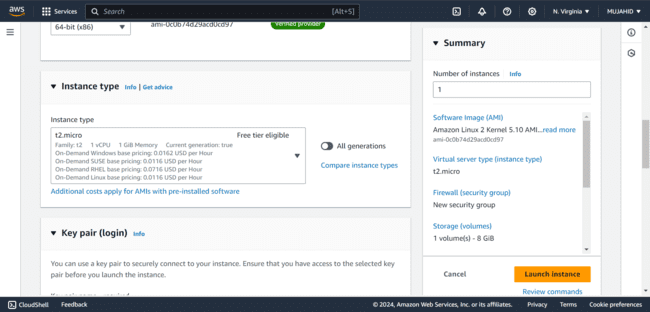
* *Give your Instance a name.*



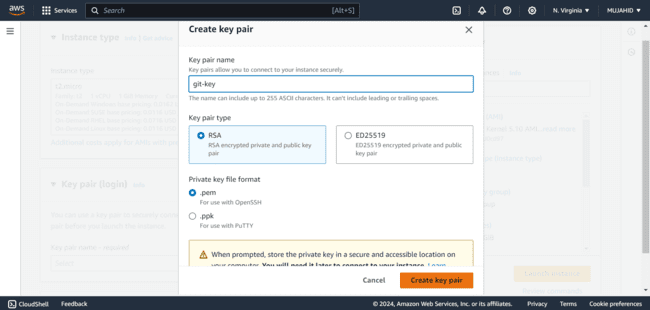
* *Select the OS/AMI, whichever is suitable.*



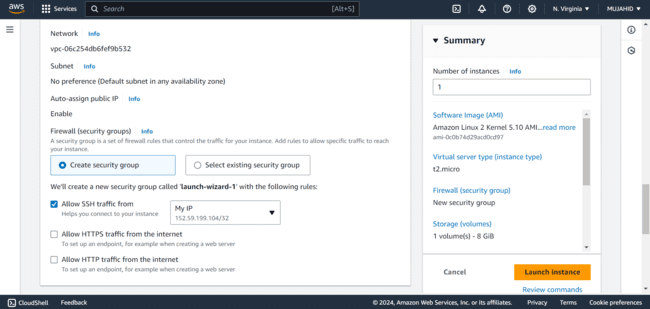
* *Select the Instance type as t2micro.*



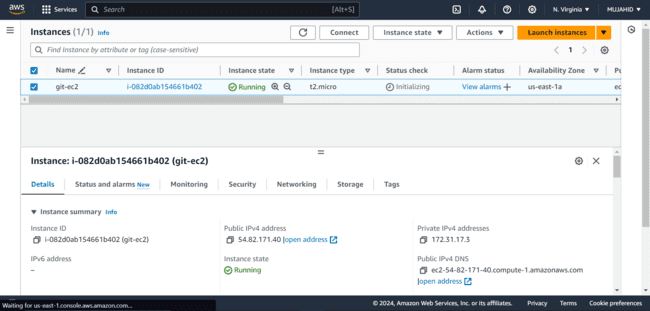
* *Create a keypair to access your Instance later.*



* *Create a new Security Group and add Inbound rule SSH allowed form My IP.*

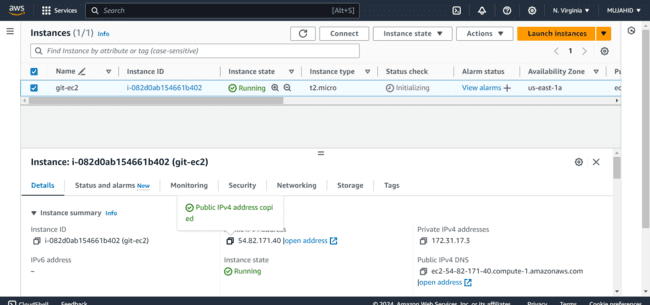


* *Check all the configurations you made and click on create Instance.*



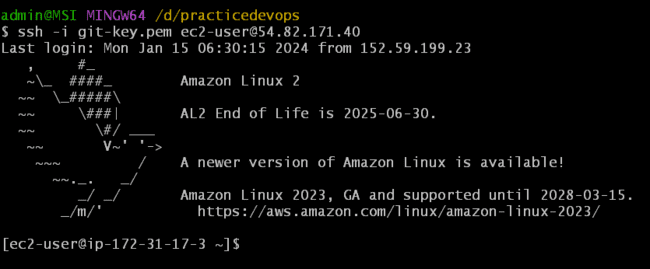
***Accessing Instance:***

* *Accessing your instance through Git bash includes copying your public IP of your Instance.*

**

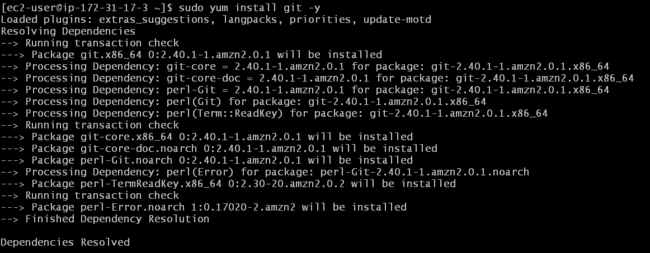
* *Get in to the path where your keypair is saved and use the below command to login*

*“ssh -i <keypair> <username>@IP”*

**

* *Install Git in your instance by using the command*

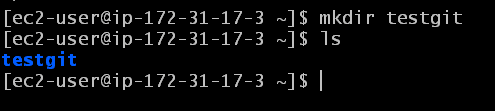
*“sudo yum install git -y”*

**

MINI PROJECT – 2

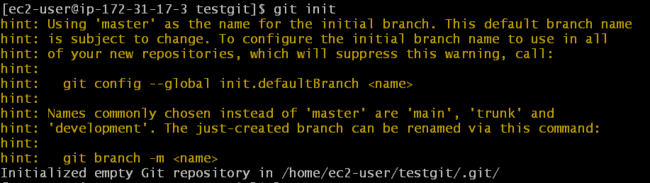
LAB – 2 [CREATING REPO IN LOCAL MACHINE]

* *After accessing the Instance through git bash, Create a directory.*

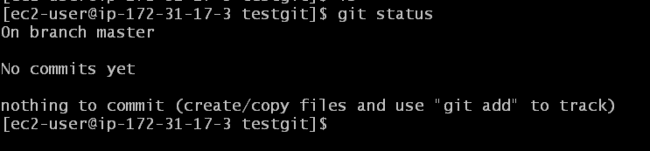
**

* *Initialize this directory using command.*

*“git init <directory>”*

**

* *Now run the command “git status” to check the status.*

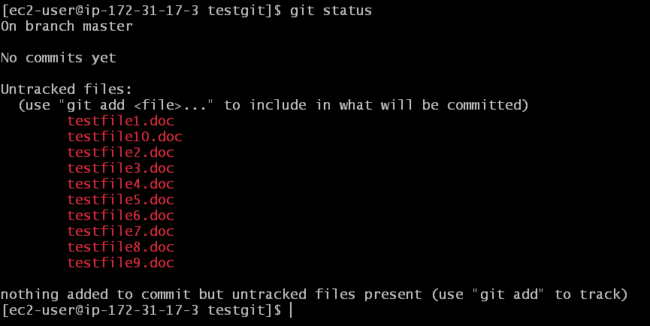
**

* *There aren’t any files in the directory so there would not be any much of output. Create some empty files using the command*

*“touch <filename>”*

**

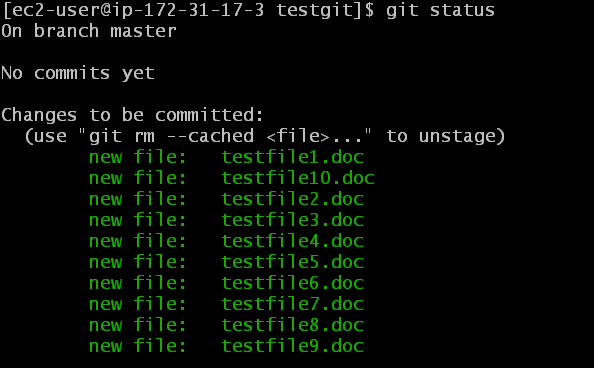
* *Now run “git status” command, there are some files present but are not tracked by git.*

**

* *For them to track by git, use command “git add <filename>”*
* *Here I added . refers to all the files in the directory.*

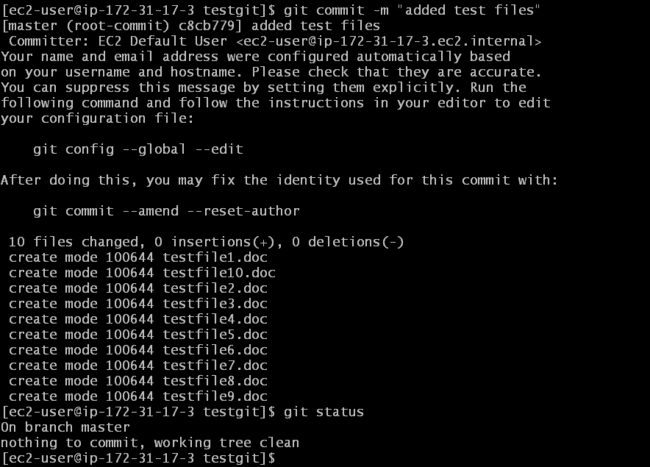
**

* *Running “git status” command again will show you the files that are being tracked by git.*

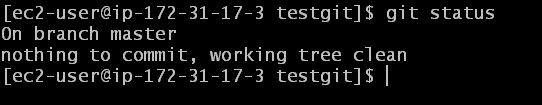
**

* *The files are now in the staging area.*
* *To commit these changes use command*

*“git commit -m “<commit description>””*

**

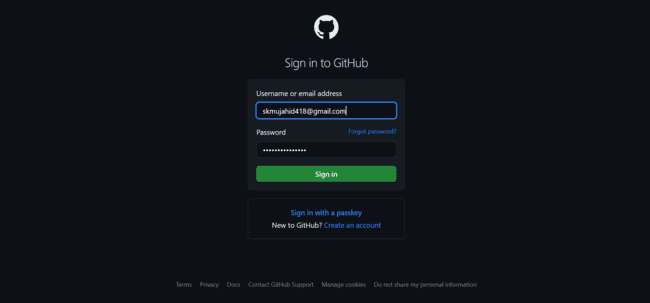
* *Run git status once again and it will show that the working tree is clean.*

**

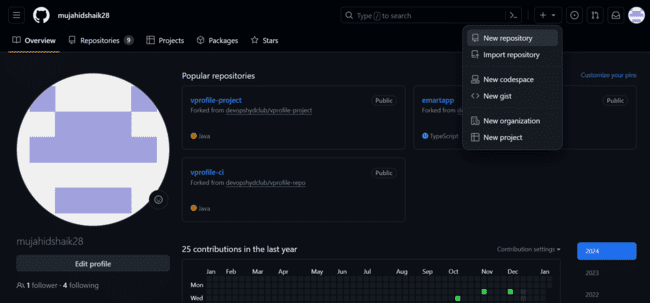
MINI PROJECT – 2

LAB – 3 [CREATING REPO IN REMOTE LOCATION]

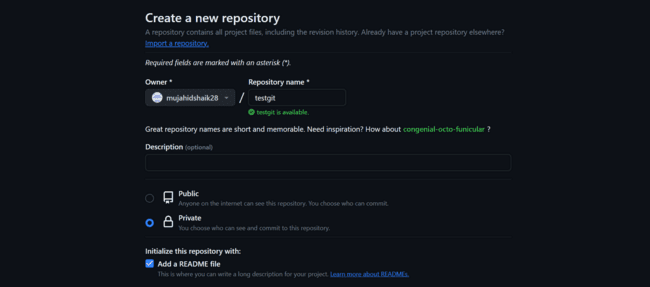
* *To create a repository in git hub.*
* *Login to your git hub account with your credentials.*



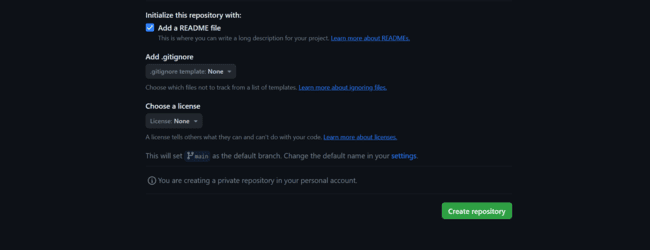
* *Click on* ***+*** *and then click on New Repository*



* *Give your repository a name and select whether it is a private or a public repository. [private is recommended]*



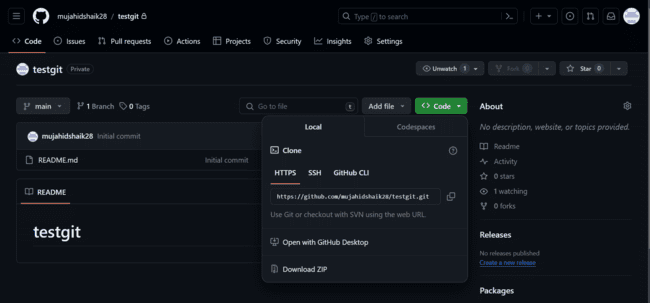
* *Initialize this repository by adding a README.md file and click on Create repository.*



MINI PROJECT – 2

LAB – 4 [WORKING WITH REMOTE REPOSITORY]

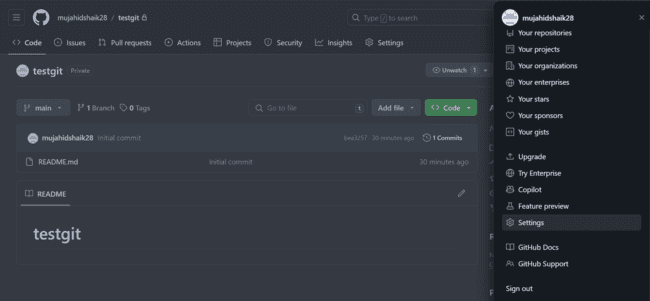
* *To clone the repository in to your Local Machine copy the clone URL of your created repository.*



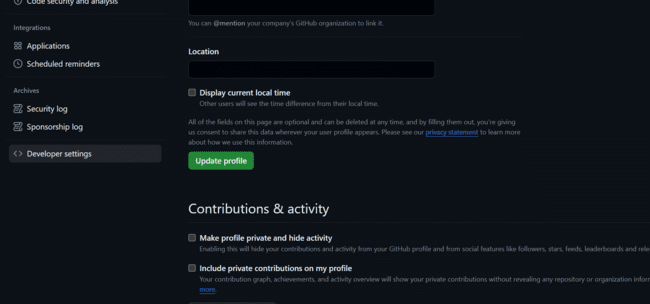
* *To clone this repository login to your Instance and using the command “git clone <CLONE URL>”*
* *In order to clone the repository, it asks for your git hub username and password.*
* *Give the username and in the password section paste a git access token.*



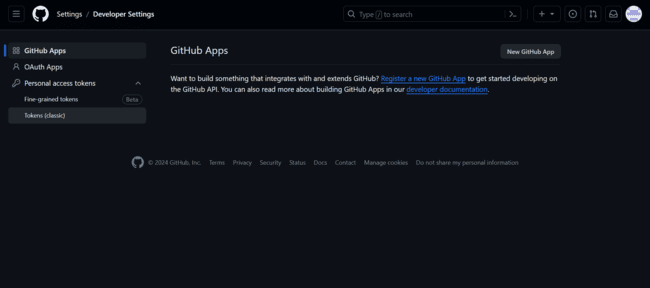
* *On the top right corner click on your profile and scroll down and click on settings.*



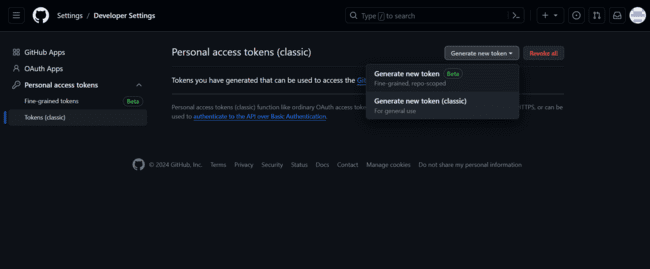
* *Click on developers settings in the left navigation panel.*



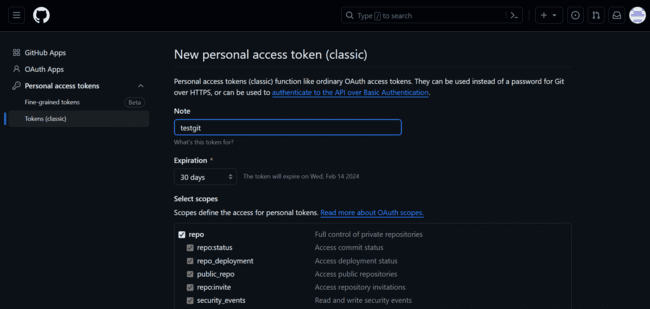
* *Click on Token [classic]*



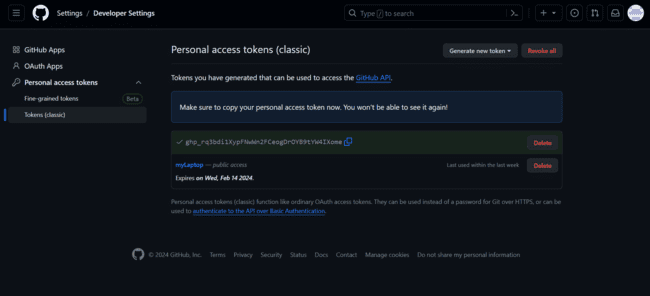
* *Click on Generate new token (classic)*



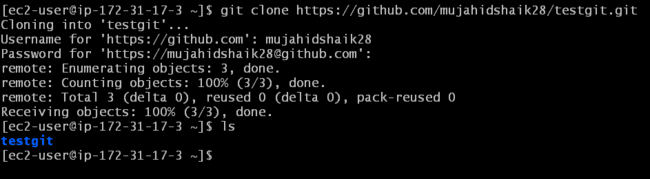
* *Give your token a Note and put check mark on repo under scopes.*



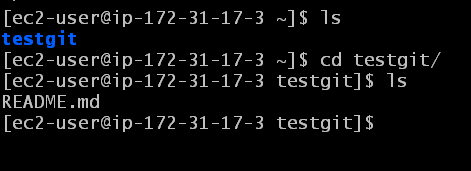
* *Copy the token and paste it somewhere else for future use.*



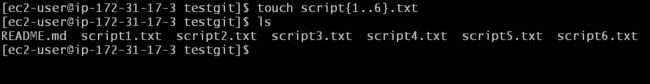
* *Back to cloning the repository, after giving username in the password section paste the token that we just created.*
* *We see that the repository is successfully cloned.*



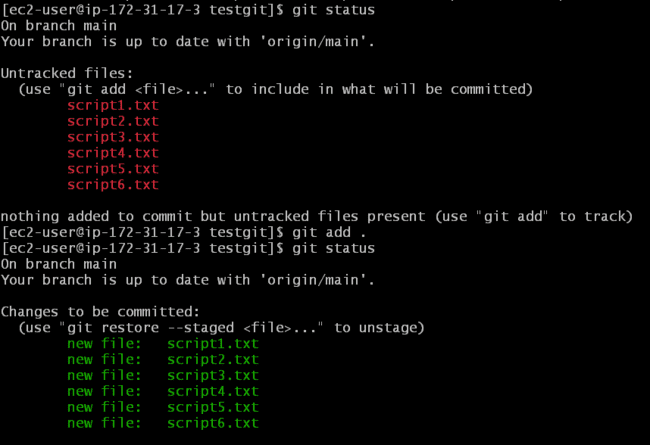
* *“cd” into the repository and check the files using command “ls”*



* *Create some text files using command “touch”*

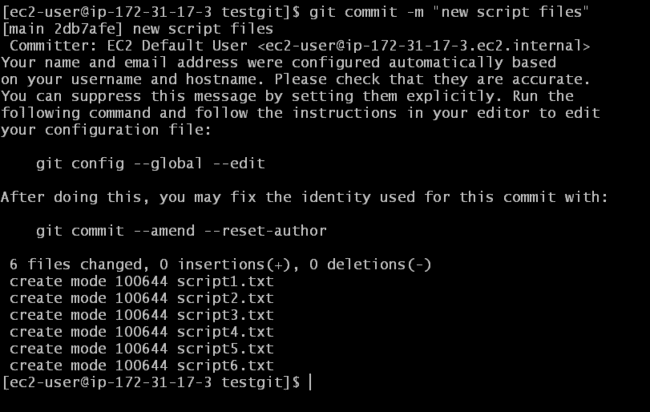


* *Check the status of the repository using command “git status”*
* *Add these files to staging area using command “git add .”*
* *Check the status again and you’ll see all the files are added to staging area.*



* *Commit these files using command*

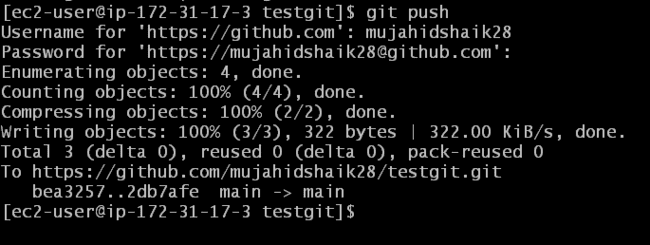
*“git commit -m <commit message>”*



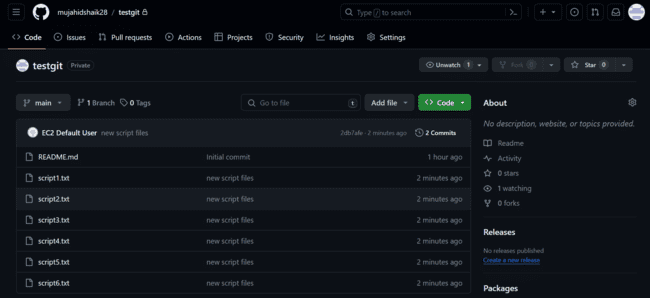
* *Check the status and you’ll see working tree is clean.*
* *Push these files to your remote repository using command*

*“git push”*

* *It asks for username and password do the same thing as we did for cloning the repository, give your username and paste the token.*



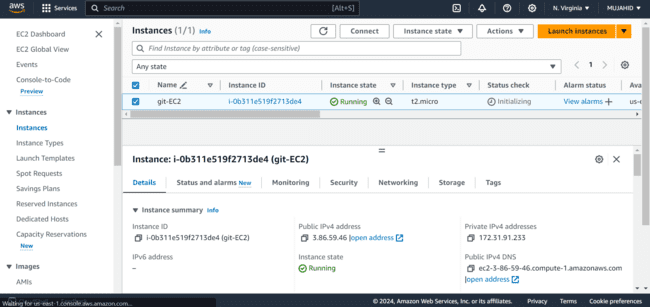
* *Open your git hub account and check the remote repository.*
* *You’ll see the files are pushed into your repository.*



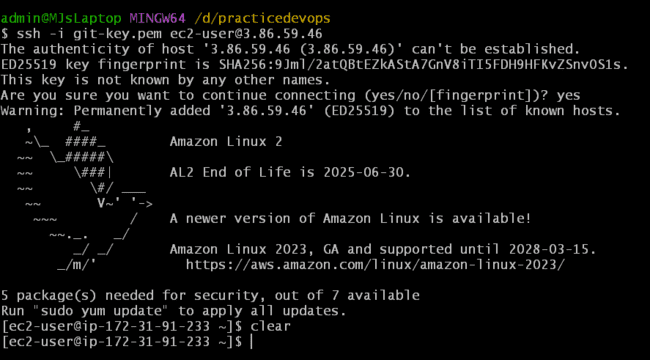
MINI PROJECT – 2

LAB – 5 [PUSHING A LOCAL REPO TO GITHUB]

* *Login to your AWS management console and Launch an Instance like we did in the Lab 1.*

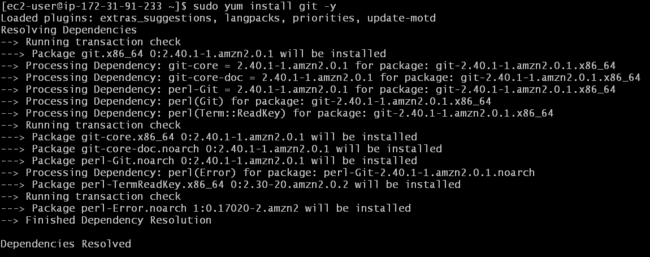


* *Access your Instance by copying public IP of the instance and executing this command in git bash by getting in to the path where keypair is saved “****ssh -i <keypair> <username>@IP****”*

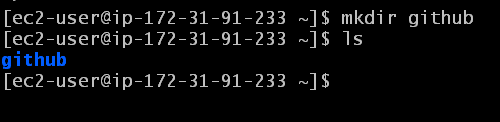


* *Install git in the instance by using command*

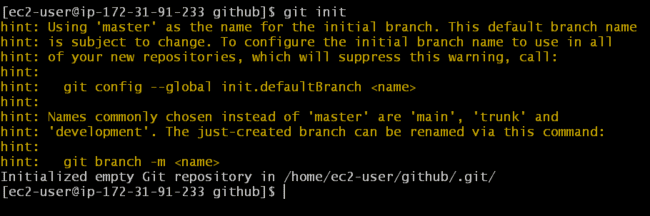
*“****sudo yum install git -y****”*



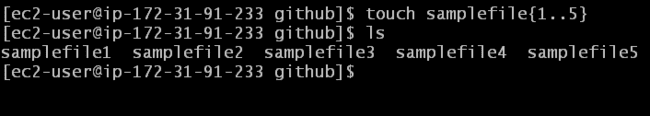
* *Create a directory in your local machine.*



* *Initialize it using command “git init”*

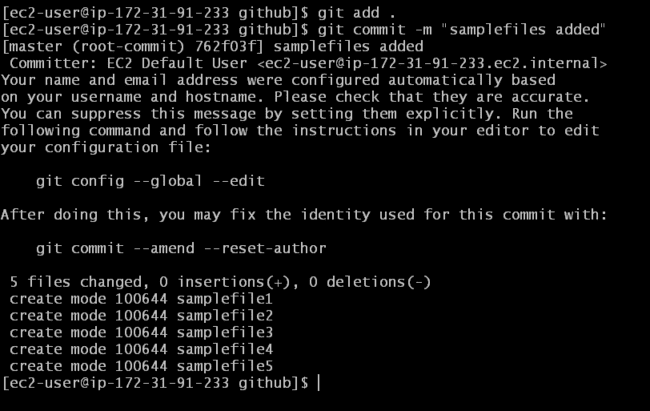


* *Create some files using command “touch”*

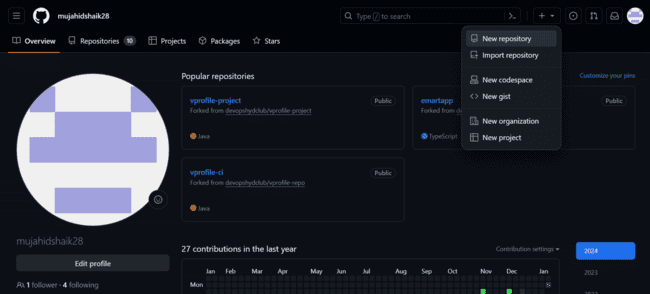


* *Tracks these files and commit them using commands “git add .”*

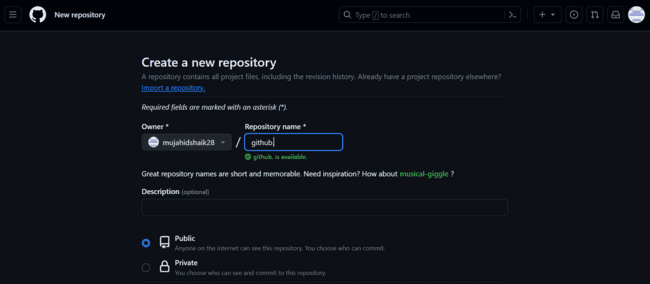
*And “git commit -m “<commit message>””*



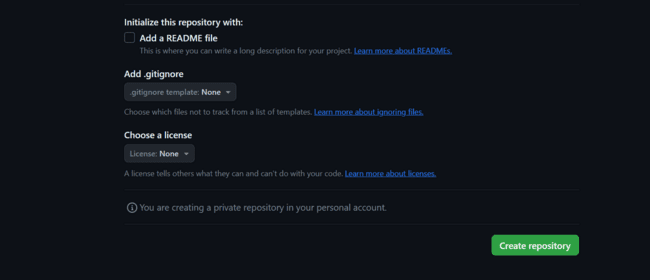
* *Open your git hub remote account.*
* *Create a remote repository with same name as local repository [github].*
* *Click on* ***+*** *and then click on New Repository.*



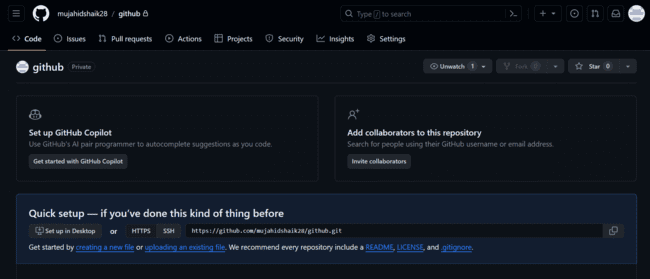
* *Give your remote repository same name as local repository.*



* *Create the repository without initializing it.*
* *Click on create repository.*



* *Remote repository is now created successfully.*

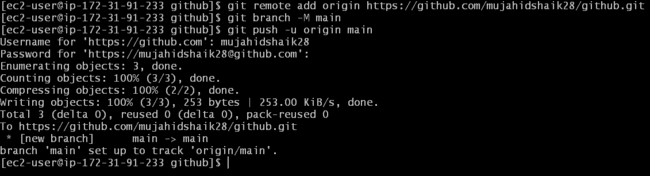


* *Execute all the commands shown in the repository.*

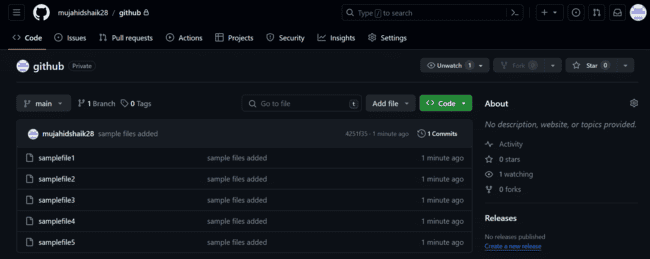
*“git branch -M main”*

“*git remote add origin <URL>*”

“*git push -u origin main*”



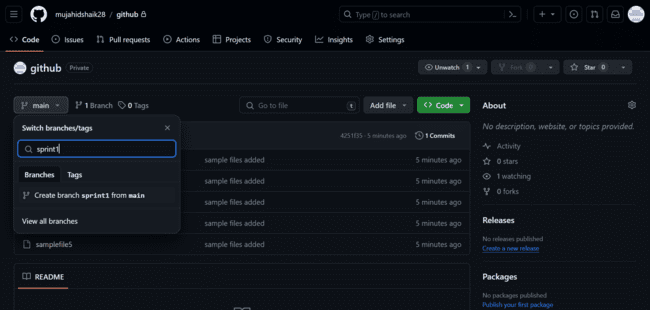
* *Check the remote repository in git hub.*
* *The files that we created in our local repository are successfully pushed to our remote repository.*



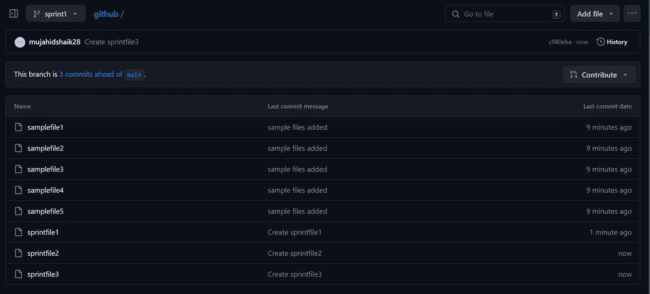
MINI PROJECT – 2

LAB – 6 [CREATING A NEW BRANCH FROM MAIN]

* *Open your Git hub remote repository.*
* *Click on Main, give your new branch a name and then click on “create branch* ***sprint1*** *from* ***main****”.*



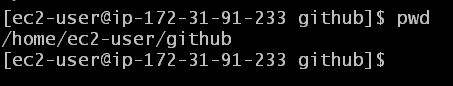
* *We see a new branch with the given name is created [sprint].*
* *Make changes in this branch directly from console.*



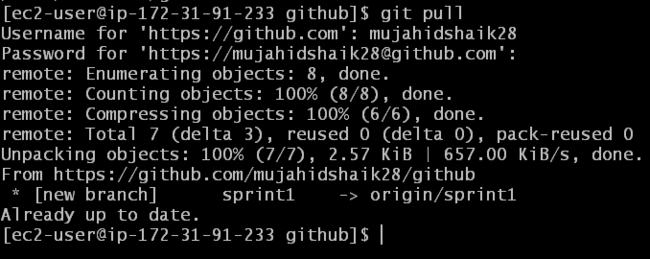
MINI PROJECT – 2

LAB – 7 [PULLING ALL BRANCHES IN LOCAL MACHINE]

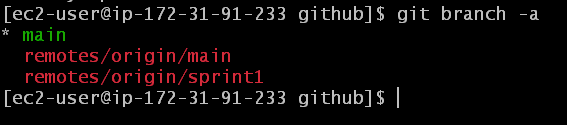
* *Get in to your local machine where you created the previous directory.*



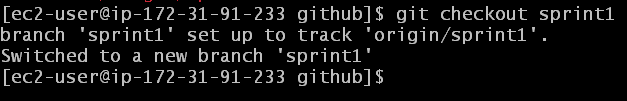
* *Execute command “git pull”*



* *All the branches are pulled into your local machine*
* *To check, use command “git branch -a”*



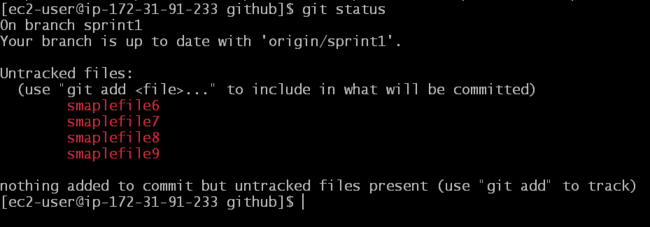
* *Switch to the other branch you created [sprint1].*



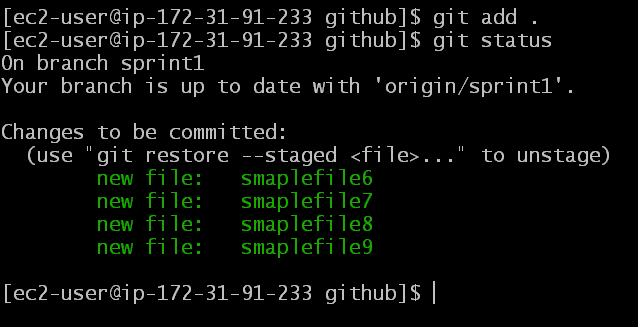
* *Except the files that are already present, create some new files using “touch” command.*



* *Check the status of the Repo with “git status” command.*

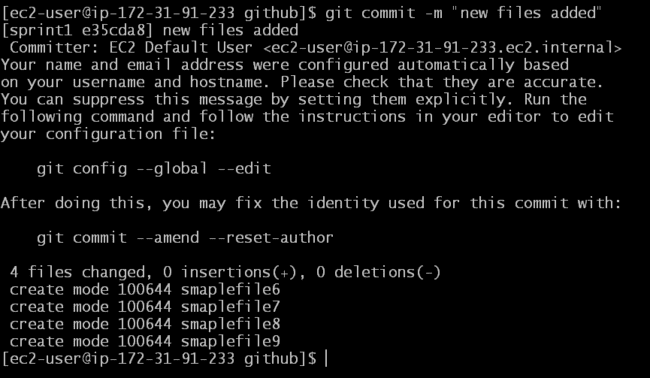


* *Add these files using command “git add .”*



* *After adding, commit them using command*

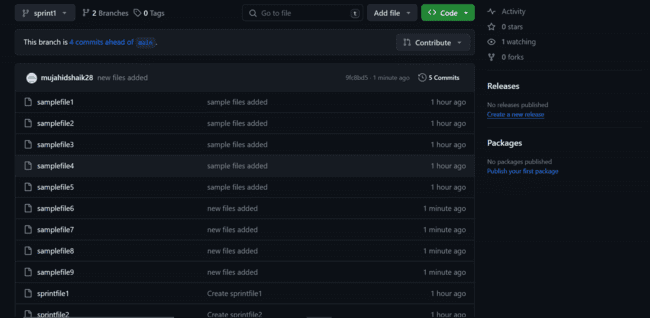
*“git commit -m “<commit message>””*



* *Make sure all the files in this branch are added and committed.*
* *Push these files to remote repository using command “git push”*



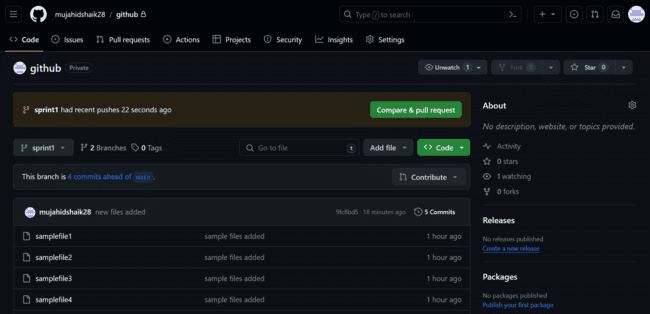
* *Check the remote repository, new files that are created in local repository are now pushed in to remote repository.*



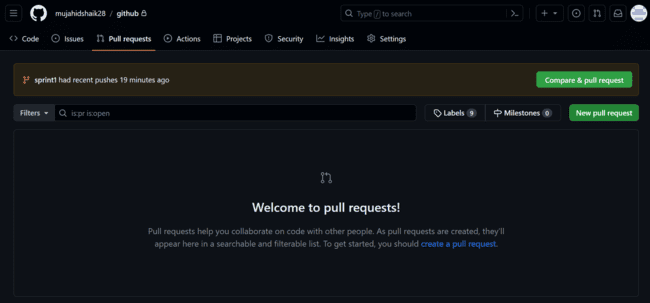
MINI PROJECT – 2

LAB – 8 [MERGING NEW BRANCH WITH MAIN BRANCH]

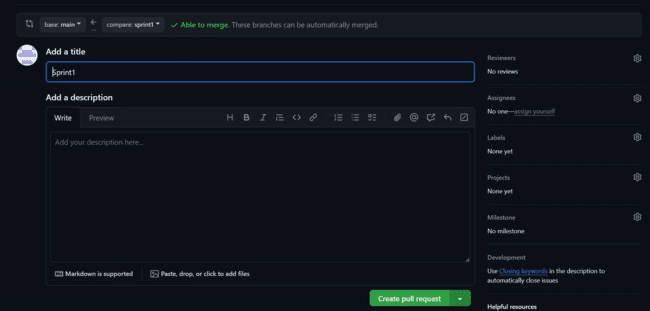
* *Get in to your remote repository in git hub account.*
* *Click on Pull requests tab.*



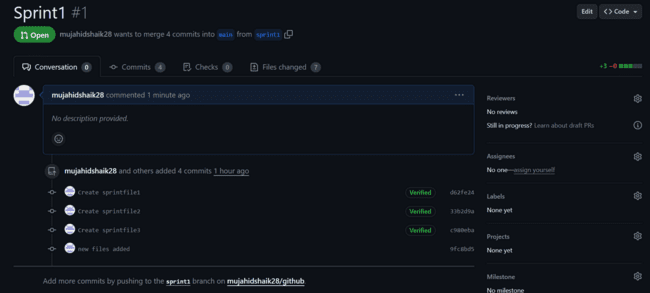
* *Click on compare and Pull request.*



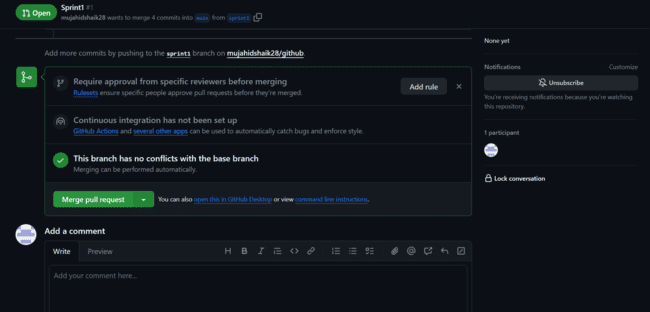
* *Make sure the base is set to Main and Compare is set to the Feature repository.*



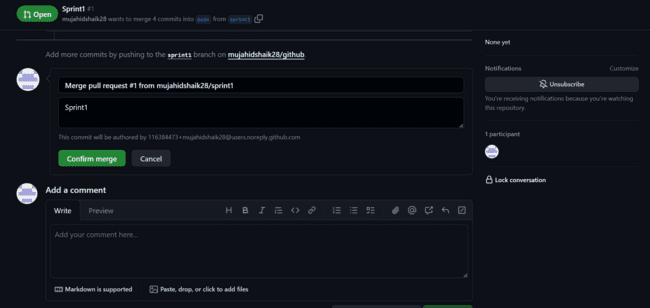
* *A new pull request is created from the branch sprint1 to Main.*



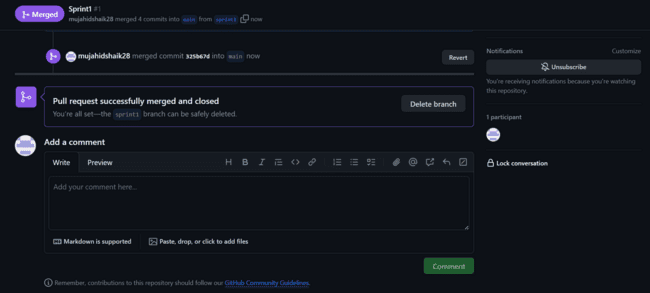
* *Click on Merge Pull request.*
* *You can add a comment if you want.*



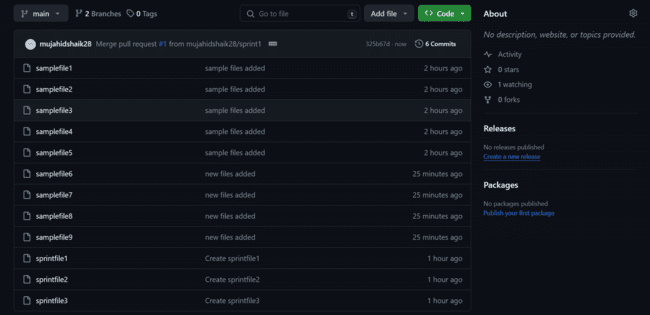
* *Asks for confirmation to merge.*
* *Click on Confirm Merge.*



* *Now the Merge is Successful.*
* *Go in to the Main branch and review.*



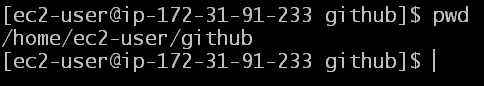
* *All the files in the branch Sprint1 are now merged in Main.*



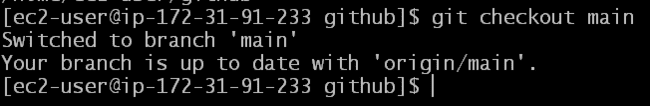
MINI PROJECT – 2

LAB – 9 [PULLING NEW CHANGES IN LOCAL REPO]

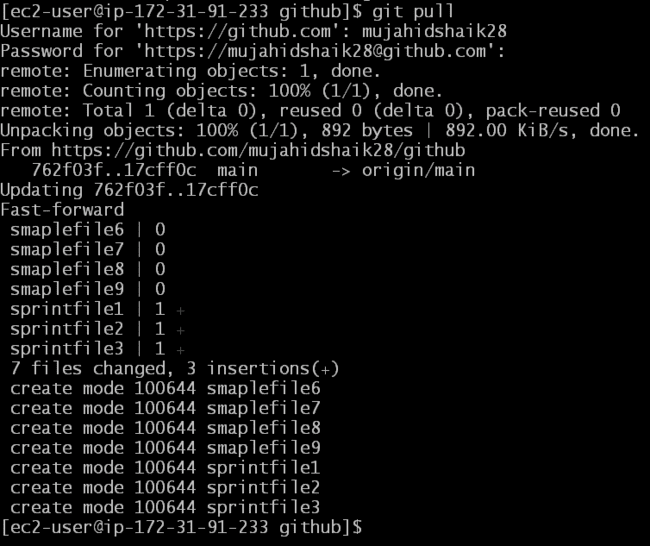
* *Get In to the path where you created your local repository.*



* *Checkout to the Main branch from Feature branch using command “git checkout main”*



* *Pull all the changes done in Remote repository in to your Local repository using command “git pull”*



* *All the files are successfully pulled in to Local repository.*
* *Check using command “ll” or “ls”*

