## **1.Setup own Gitlab on EC2 instance:**

First create instance of EC2

After creating instance open folder which is containing .pem file in terminal

Then change the permissions of the file by using chmod command.

chmod <octalnumber> <filename>

Each file has a certain type of permissions. There are 3types of permissions

Read (r)

Write (w)

Execute (x)

Files can be shared with user, group, others.

Then permissions for user, group, others in group of 3.

So for example permissions with rw-rw-r shows that user and group can read and write the file but others can only read the file.

In octal representation r has value 4, w has value 6, x has value 1.

So rw has value 4+2+0 . Therefore rw-rw-r has octal number 664.

So to make read only file only read permission should be given to user but not anyone only read permission should be given to user so the value is 4. For groups and others they should not be able to see the file. So the octal number becomes 400

chmod 400 sumanth.pem

Connect to EC2 by using ssh by typing the following command

ssh -i /path/my-key-pair.pem ec2-user@public-dns-name

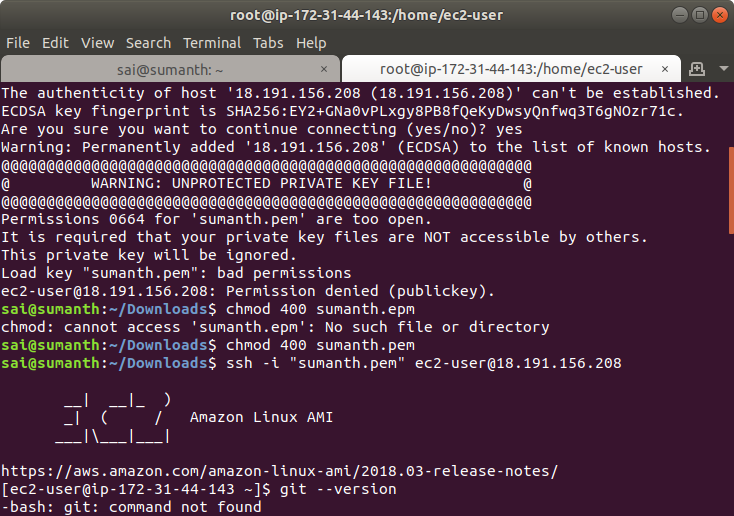
Here we are typing the command in the directory which is containing the .pem file so the command will be

ssh -i “sumanth.pem” ec2-user@18.194.157.208

After that give sudo permissions and install git

To check whether git is installed or not use command git --version

Qe-TsED7gspEGLYRf\_Wf// Api access token



To see any remote is added previously you can use the command

git remote -v

To remove any remote we can use command

git remote remove <remote-name>

### **2.Configure mail for gitlab create user &Delete once setup:**

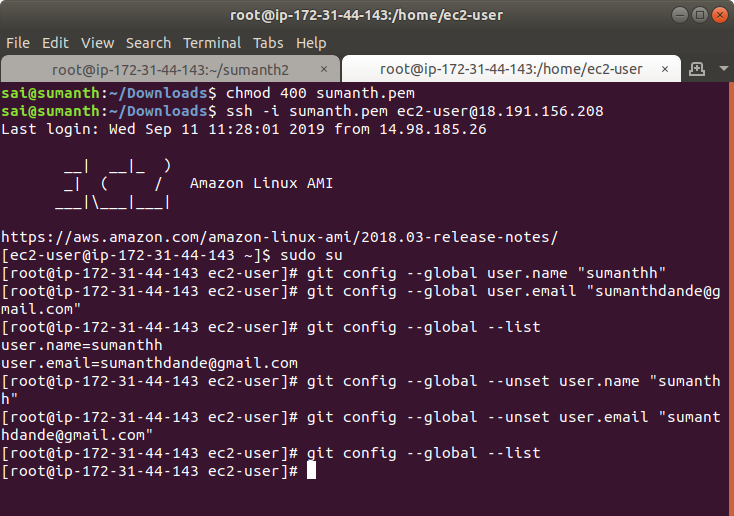
git config --global user.name “sumanthh”

git config --global user.email “[sumanthdande@gmail.com](mailto:sumanthdande@gmail.com)”

git config --global --list

git config --global --unset user.name “sumanthh”

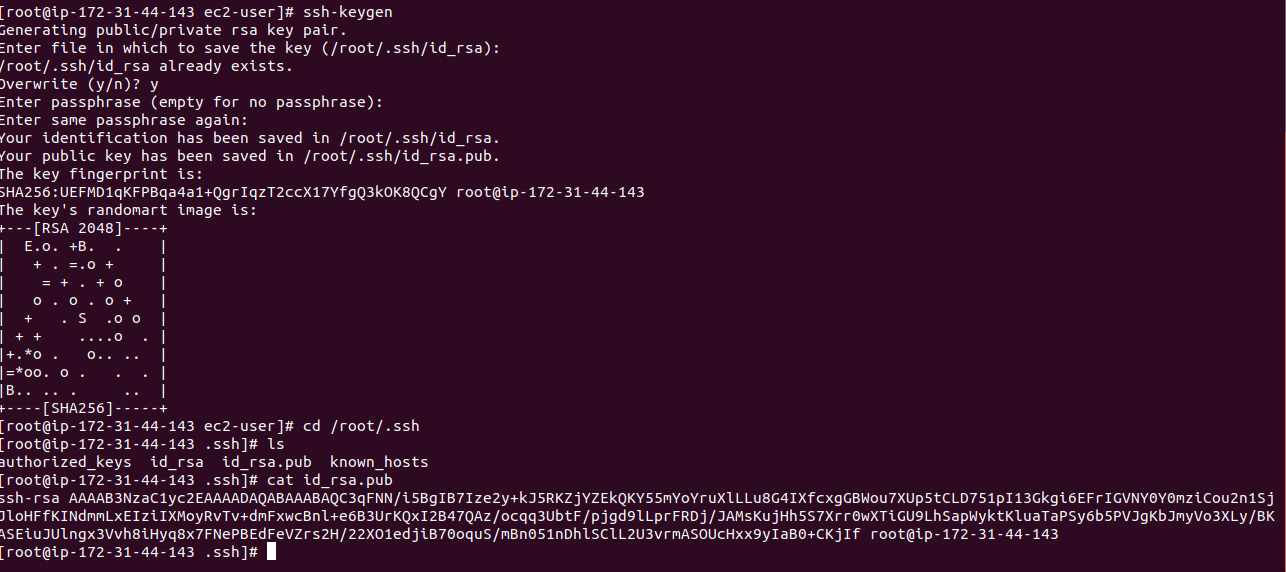
git config --global --unset user.email “[sumanthdande@gmail.com](mailto:sumanthdande@gmail.com)”



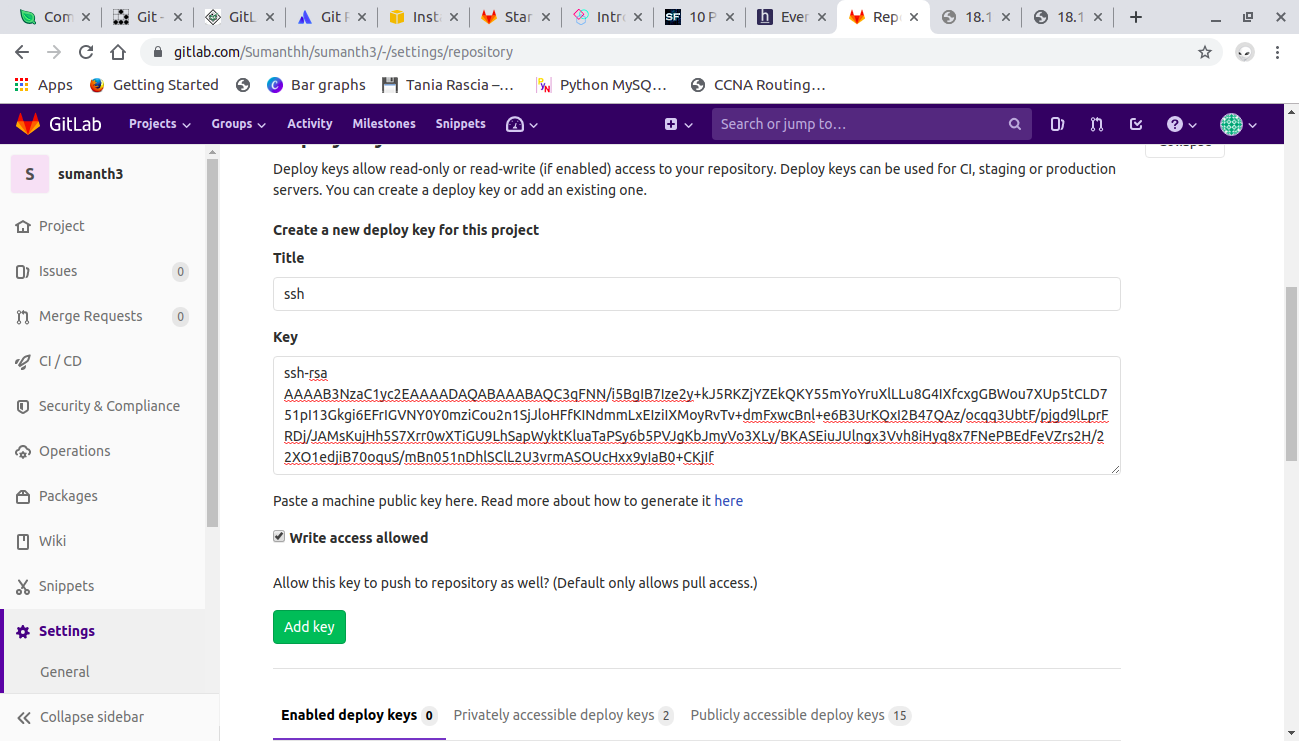
### **3.Create empty repo and clone entire repo:**

First we have to add ssh key to our project

For that use ssh-keygen



Open project settings>repository>deploy keys in gitlab



### **4.Clone specific branch and push to repo:**

Clone specific branch for that i am creating a branch by terminal and i will add some files in that

It we clone the entire repository by default it copies master branch .

So if we type

git clone [git@gitlab.com](mailto:git@gitlab.com):Sumanthh/last.git

It will clone only master branch even though the project has many branches. This happens basically because many developers will work on the code and everyone create their own branch and they need clone of only master branch.

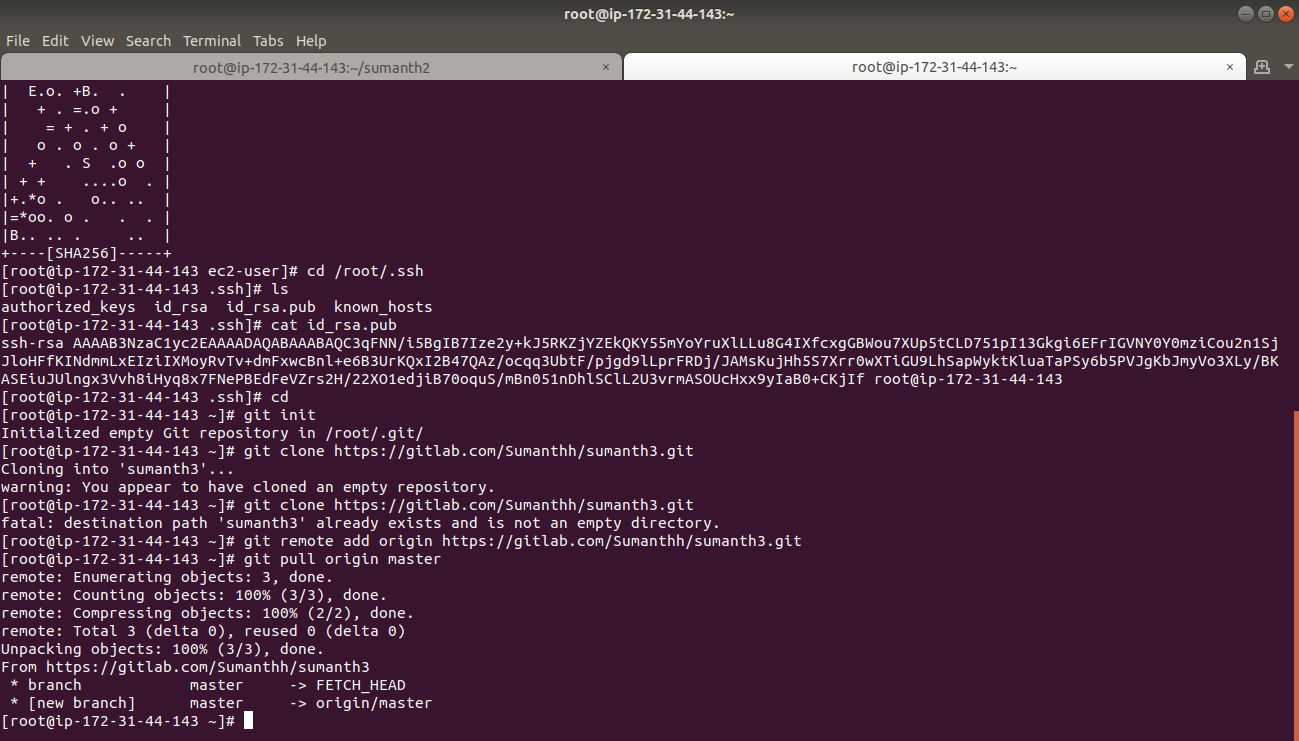
If we want to clone any specific branches we should type

git clone -b <branch> [git@gitlab.com](mailto:git@gitlab.com):Sumanthh/last.git

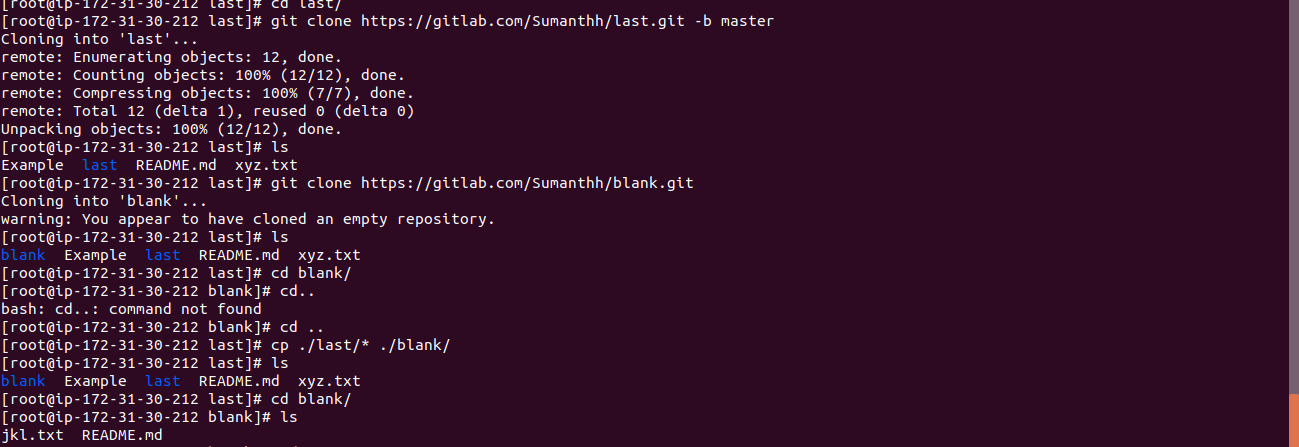
And to add a new branch type

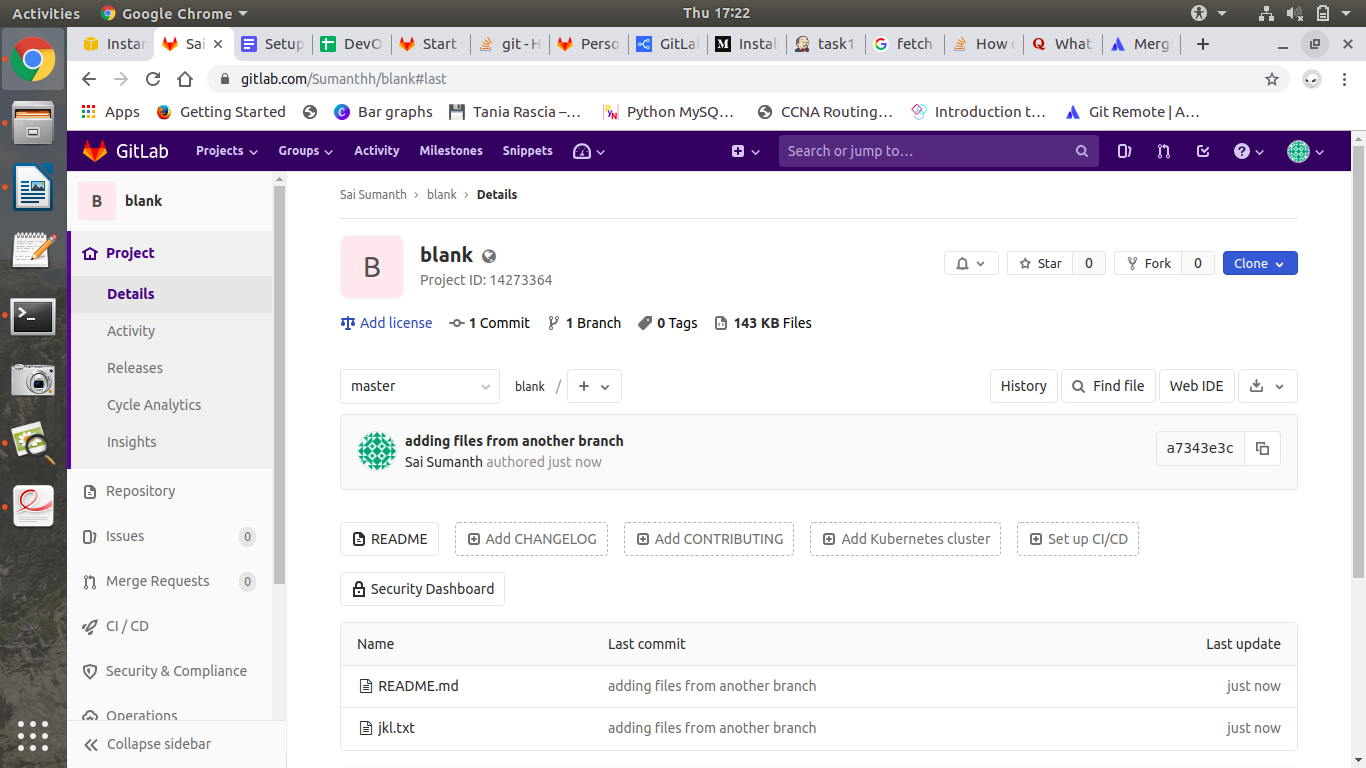
git checkout -b <branch\_name>

Add changes into the branch and then commit the changes and then push to the repo.



I created an empty repo with name “blank” and i cloned “blank” and “last” which are two different repos . And then i copied the files from “last” repo to “blank” repo and pushed it into remote.





### **5.Merge and rebase difference:**

The easiest option is to merge the master branch into the child branch using

git checkout child

git merge master

This will merge the master branch with child

This creates a new “merge commit” in the feature branch that ties together the histories of both branches.

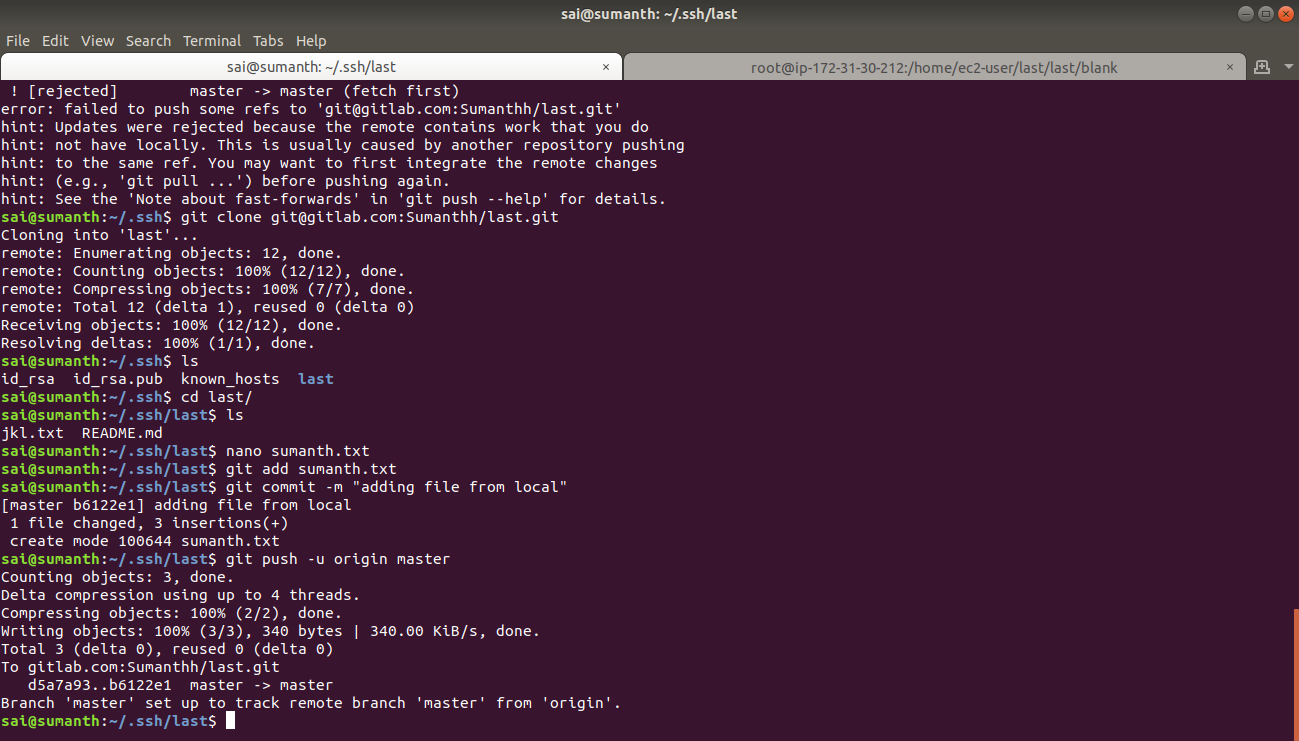
git merge apply all unique commits from branch A into branch B in one commit with final result

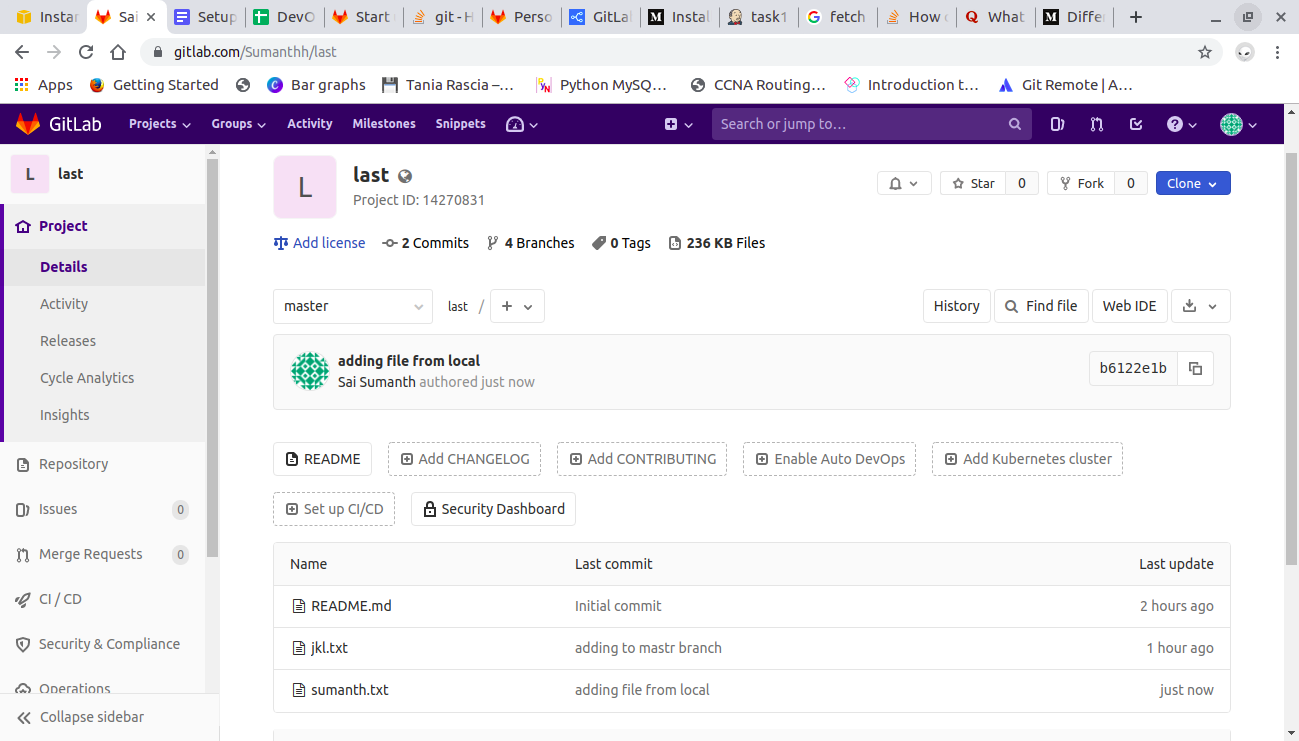
git merge doesn’t rewrite commit history, just adds one new commit

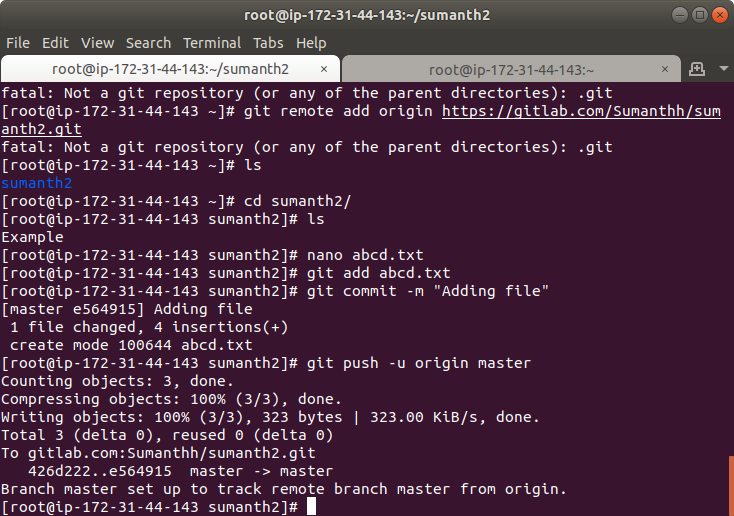
git rebase gets all unique commits from both branches and applies them one by one

git rebase rewrite commit history but doesn’t create extra commit for merging

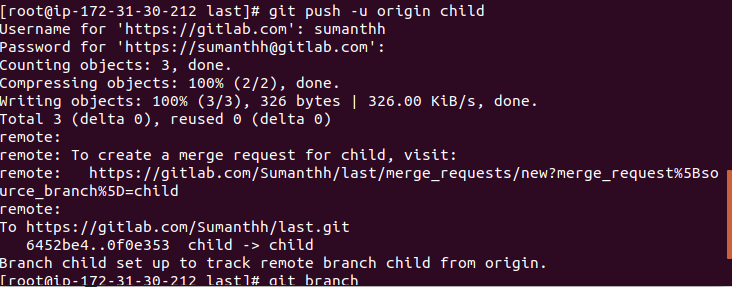
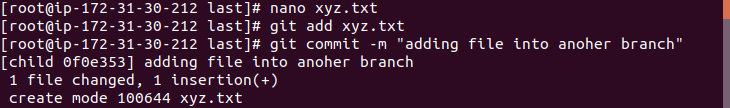
### **7.On local create git and push to remote:**







### **8.Add new files to origin remote to specific branch:**



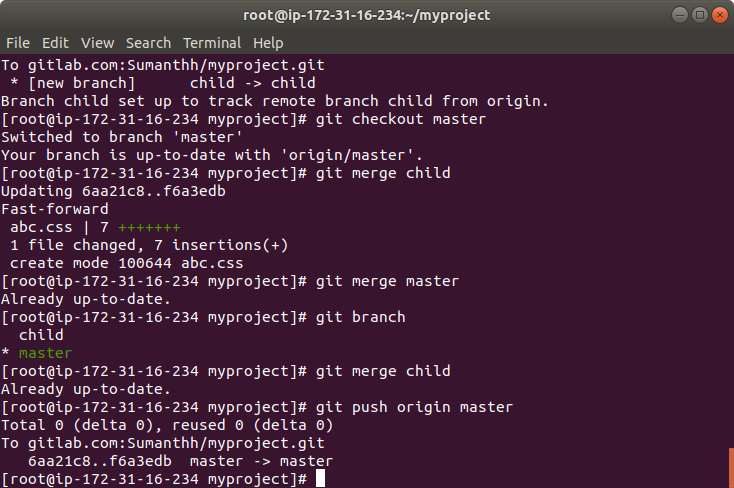
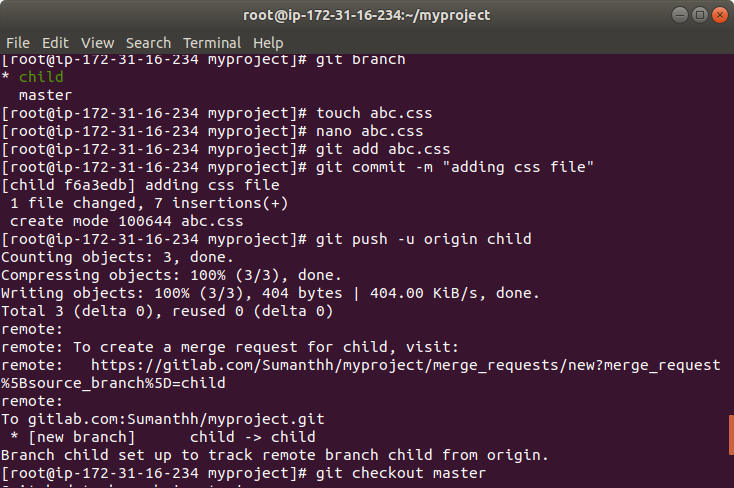
### **9.Create two branches and merge them:**

Use command

git merge <branch> this command merges the <branch> with current branch

git merge child

Here i am working on master branch and i have to merge child branch with master branch.



### **10.Merge the code to master branch and push to remote:**

It is similar to pushing to specific branch.

In this we will push to master branch

git push -u origin master

