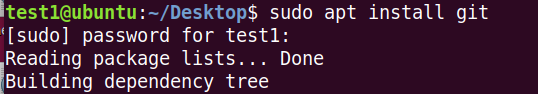
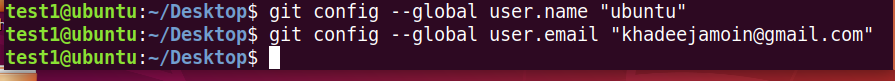
Step 1: Install Git using following command:



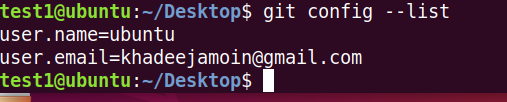
Check Git version:



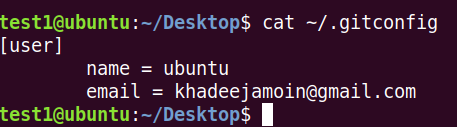
Configure Git for future use:



Step 2: To verify the configured changes:

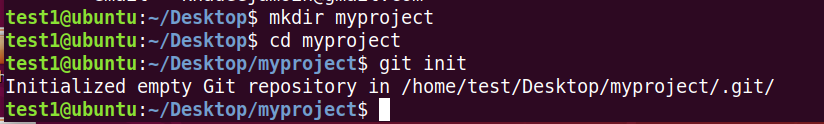


For Reading contents written in Git file:



Step 3: For Repository, first login to your GitHub account and create a new repository. Upload some files into that repository. After that, make a directory with same name as your repository in your system and change pwd to that new directory.

To initialize Git in that directory, use command: **git init**



Step 4: Now we will set up the remote, which tells git where the repository is located.

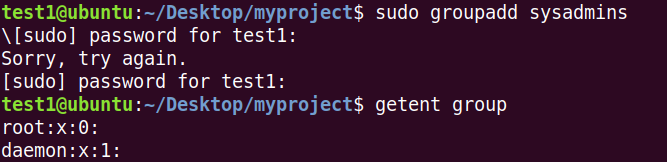
git remote add origin https://github.com/your\_username/myproject.git

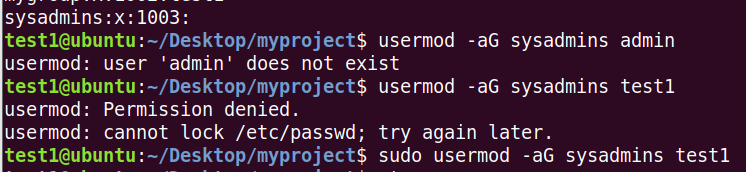


Then add a user group:

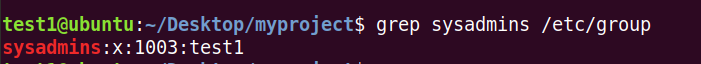
With getent command, you can see whether the group is created or not. The syntax for next command is: sudo usermod -aG <new-group-name> <user-name>

This command will add the user “test1” to the newly created group.





To see that whether the user has been added to the group, use following command.

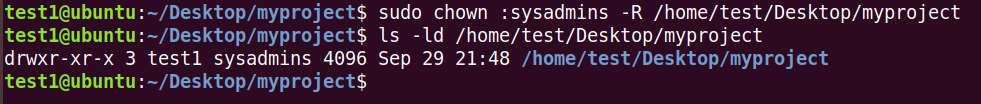


Then find out the complete path of your current directory using **pwd** command.

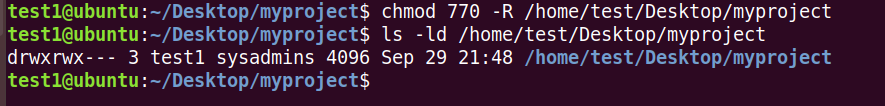


Then we have to change owner of the directory to the new user-group that we created i.e., **sysadmins**.

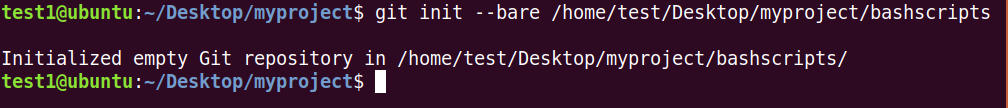
The following command changes the owner of newly created directory to sysadmins.



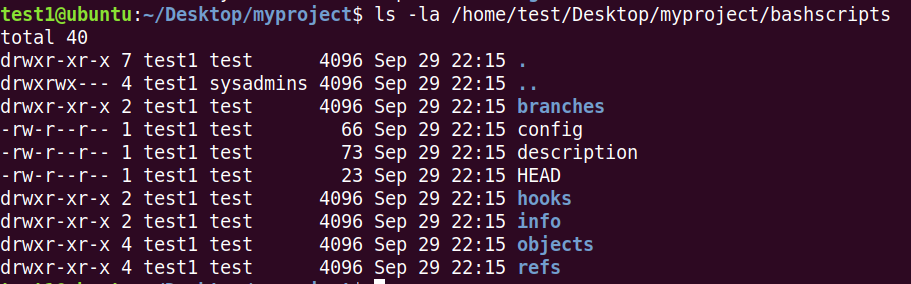
The following command changes the file permissions for the **myproject** directory. The flag **770** indecates that owner and the group will have all rights while the other users (o) will not have any rights on the directory and its files. To see the difference, compare the above screenshot with the current screenshot.



Then initialize a bare project repository.



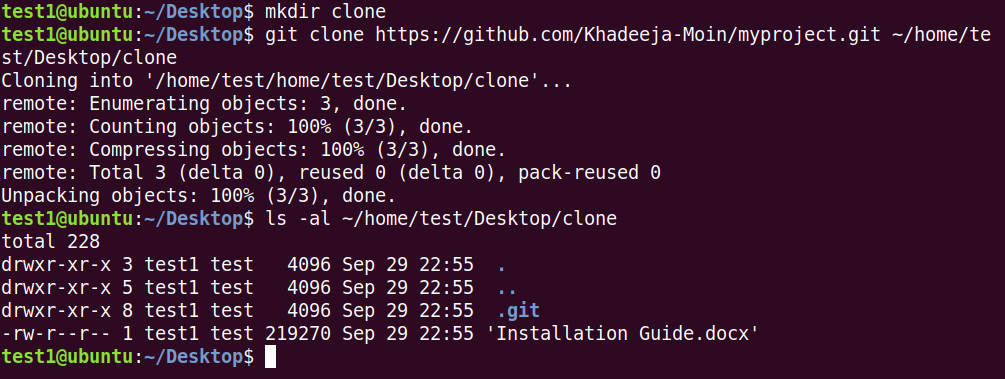
At this point, you have successfully initialized a bare Git directory which is the central storage facility for the project. Try to do a listing of the directory to see all the files and directories in there:



**Cloning Git Repository:**

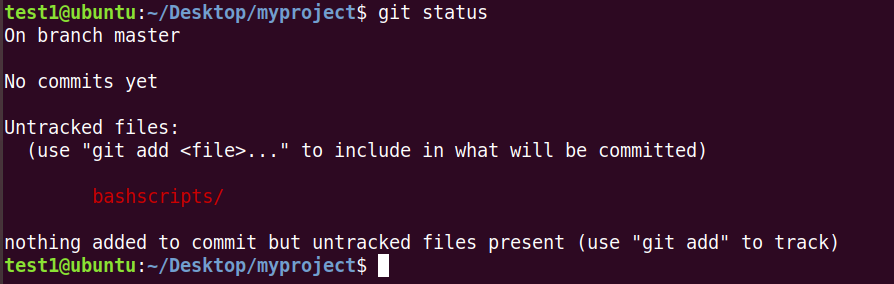
When you clone a repository, you copy the repository from GitHub.com to your local machine. Cloning a repository pulls down a full copy of all the repository data that GitHub.com has at that point in time, including all versions of every file and folder for the project.

Here, we have cloned Git repository to another directory at Desktop named as **clone**. For this, you have to give the URL of your actual repository that you created on GitHub. We can see our uploaded file **“Installation Guide.docx”** in the cloned repository.



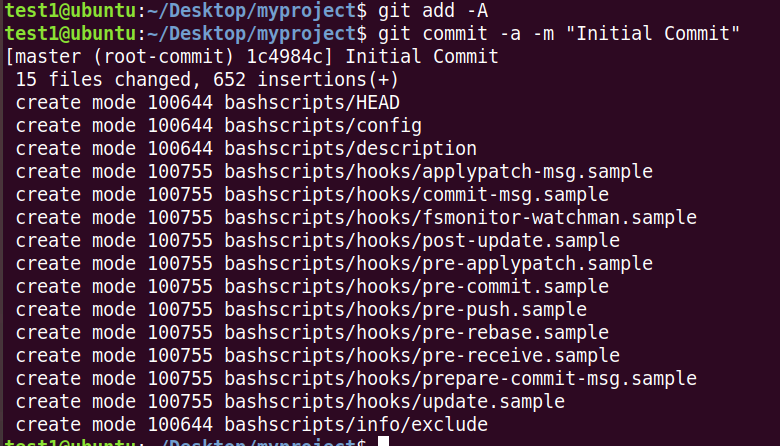
**Git Status:**

This command shows the status of your git repository. You must be in the directory in which you initiated git to see the output of this command. The directory with the same name as your git repository.

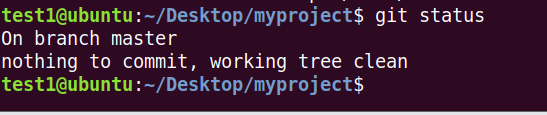


**Git Stage Changes and Commit**

Next, stage all the changes using the add command with the -A switch and do the initial commit. The -a flag instructs the command to automatically stage files that have been modified, and -m is used to specify a commit message:

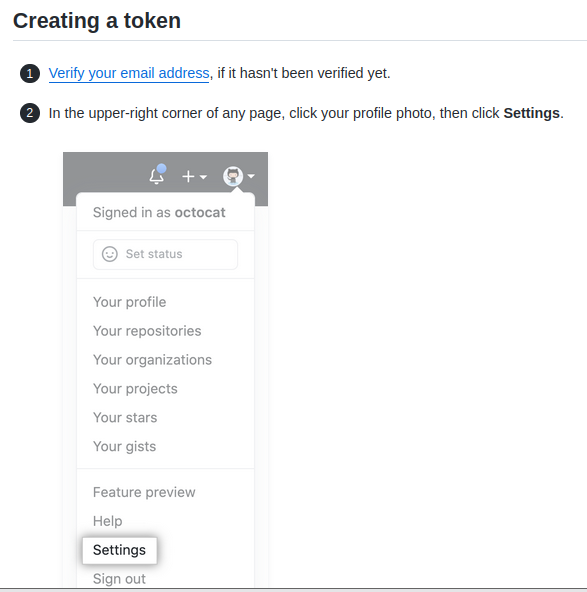


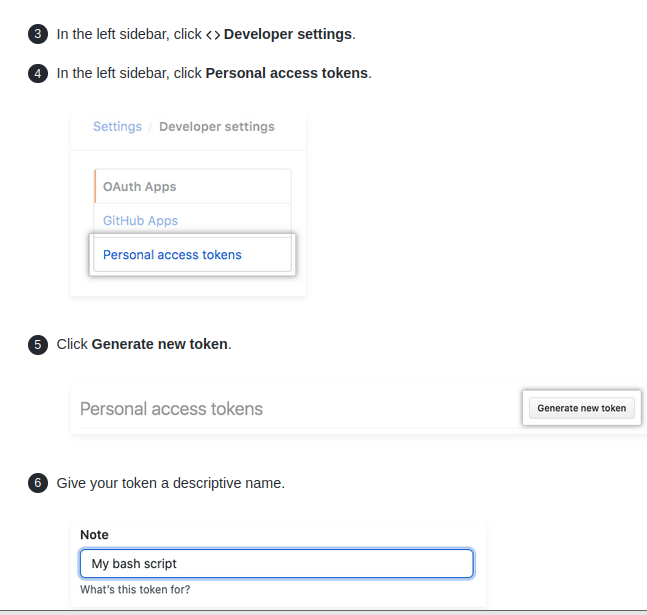
After this if we check git status then, we will see the following output.

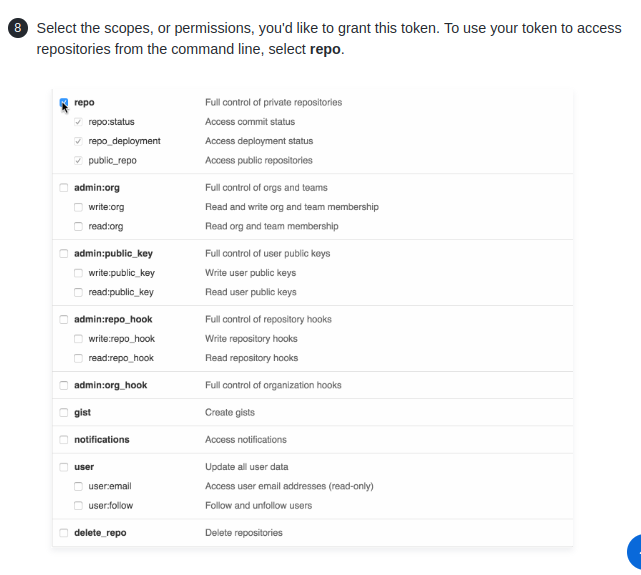


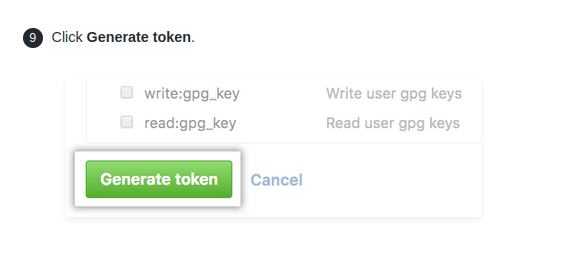
**Publish Local Commits to Central Git Repository**

For this, you have to create a PAT from your Git Repository.

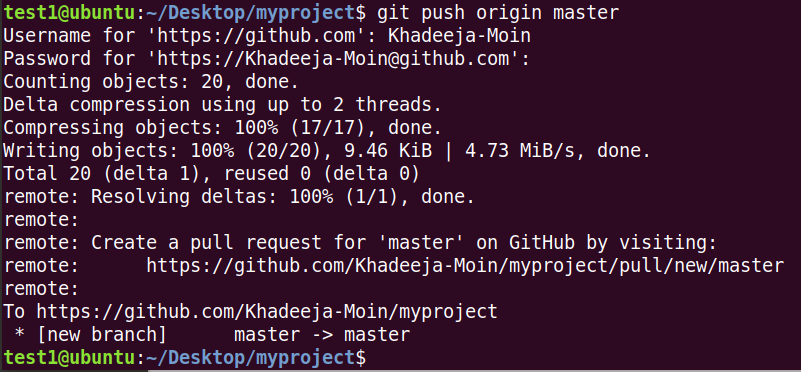








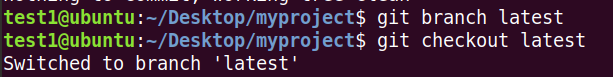
To execute the command: git push origin master

First provide your username. Then instead of password, enter the Personal Access Token you just generated.

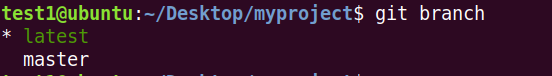
**Create Git Branch**

To create git branch, use command: git branch <branch-name>

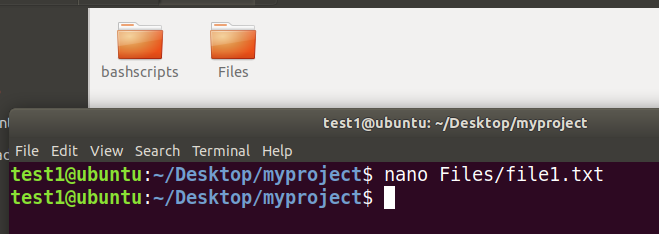
To switch to the branch: git checkout <branch-name>



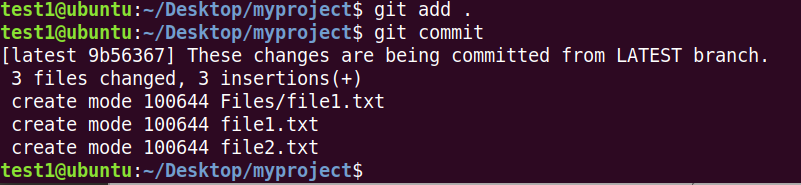
To see all branches and currently active branch, use following command:

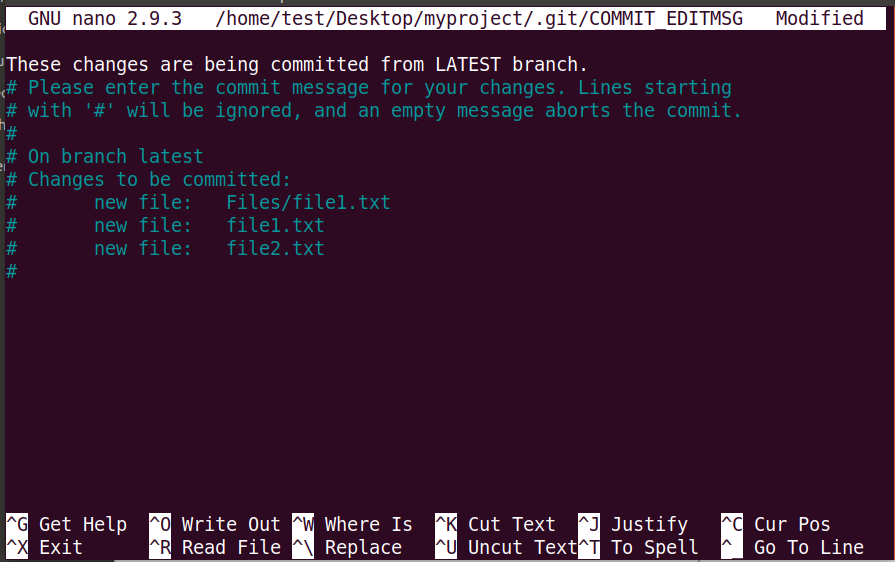


First create a new directory in **“myproject”** folder. Then create a new file in that directory using any editor and write something in that file.



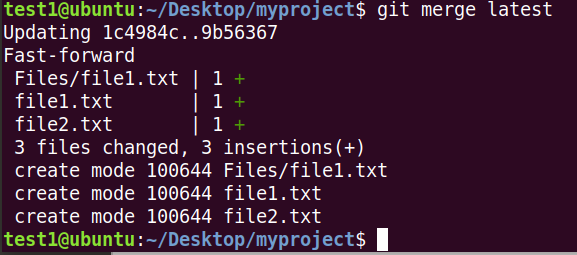
**To Add and Commit changes in Git:**



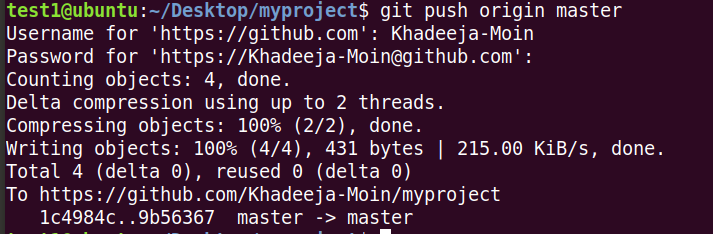


**Merge Changes from one Branch to Another**



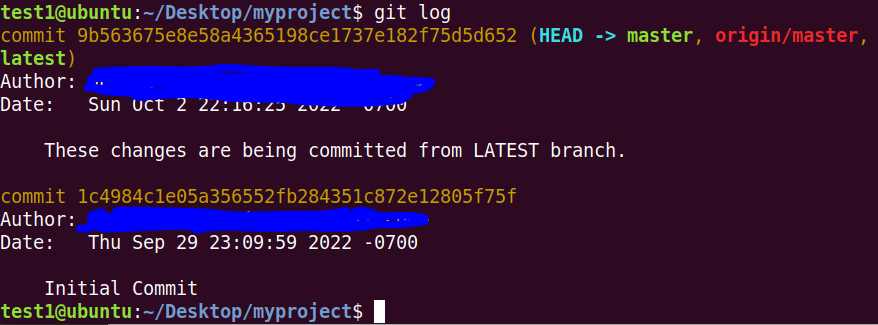


The above commands will merge your changes to the master branch but it will not reflect on your GitHub account until you push these changes. To push changes, use: git push origin master

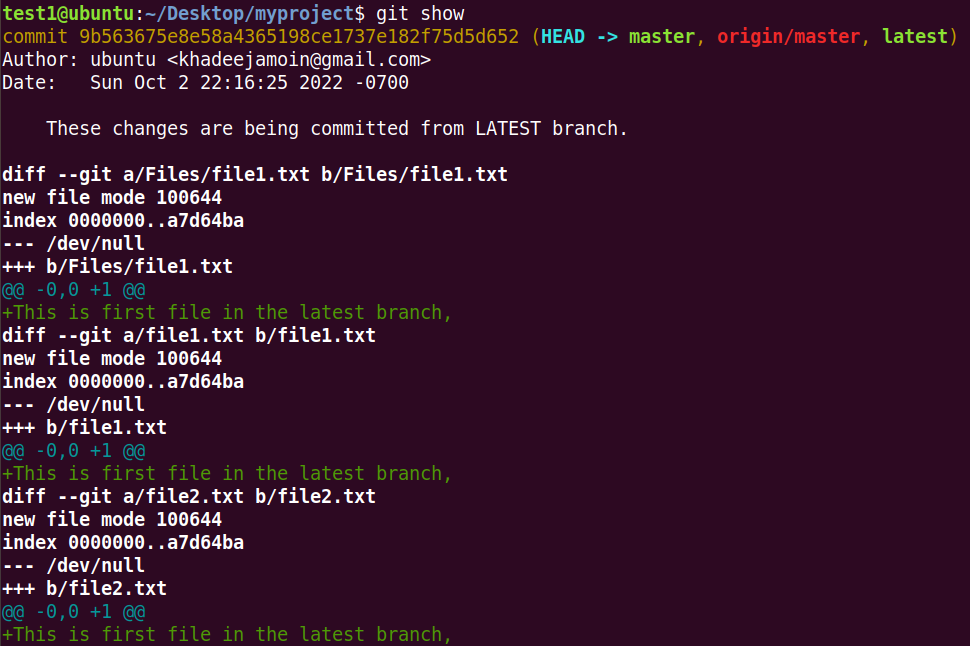


Now, reload your GitHub page and changes will be reflected.

**Git Commit Log**



**Git show**



**To show the difference between two branches**



**Task:** Explore the “PULL” and command to delete particular branch with the help of lab session.