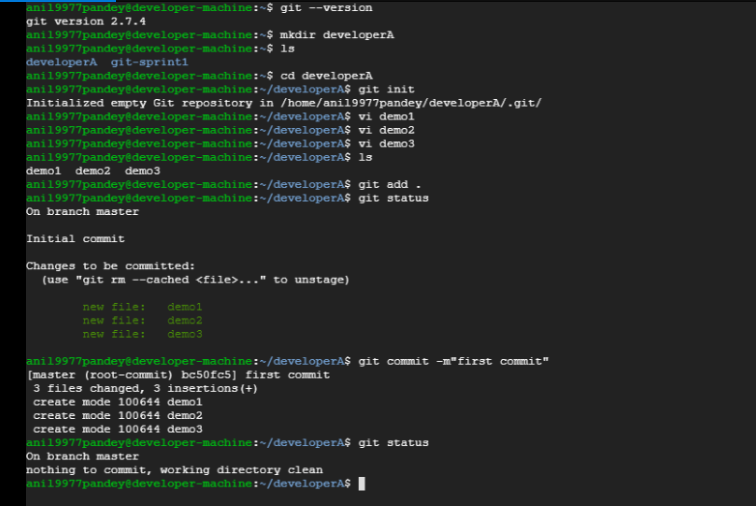
# **Bootcamp Day 1 – Git Assignment**

1. Create a local git repository developerA and create 3 files into it. (demo1,2 and 3)

* I have created a directory called developerA and created 3 files in the directory.

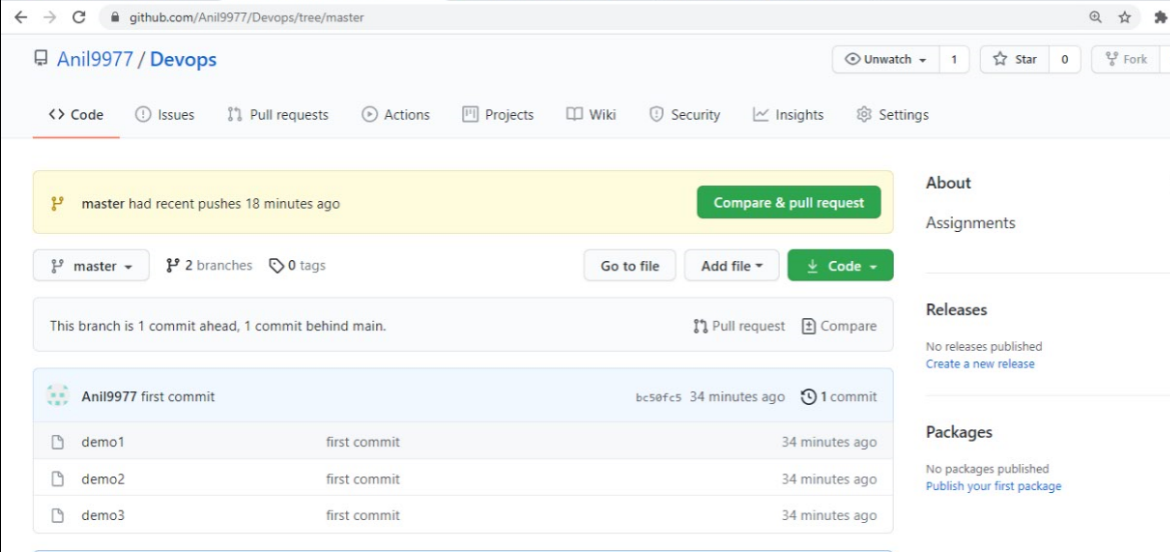
1. Add all the 3 files into staging area and commit it in the local repository

* I have used **git init** to initialize the git and configured with my email and user names.
* Added the files to staging area with **git add .**
* Committed with comments as **“first commit’**



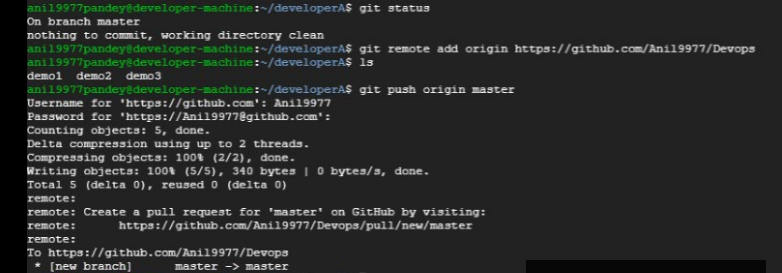
1. Create a remote repository.

* I have created a git repository named as Devops in my login.



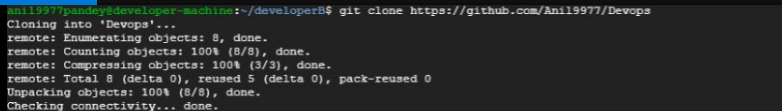
1. Add the remote repository to the local repository and push the changes to the remote repository from local repository

* I have added remote repository to local repository with **git remote add origin <https://github.com/Anil9977/Devops>**
* Pushed the content to remote repository with **git push origin master**. I have provided my credentials while it asked for.



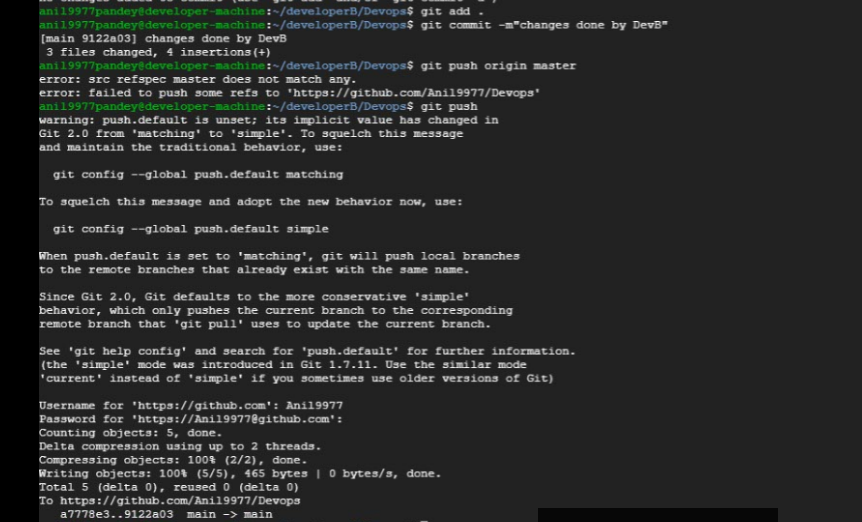
1. Clone the remote repository to another folder developerB.

* As part of this, I have copied the remote repository URL and used got clone using **git clone <URL>** to clone to the local .



1. Make and commit the changes via developer B and synchronize the changes in developer A repository

* I have added an extra line “# changes made by developerB”
* I have committed the changes with message “changes made by developerB”
* After committing, I have pushed the code to remote with “**git push origin master**”
* In DeveloperA, I have pulled these changes with “**git pull origin master**” command



1. Create a new branch for a new feature you want to add to the application(say feature1) and change the files.

* I have created a new branch called feature1 with the command “git branch feature1” and switched to branch with “git checkout feature1”
* I have changed the file demo2 by adding “Changes made in feature1 branch”line.
* I have added to staging and committed with message “changes done in feature1 branch”

1. Merge back the created branch with the master branch

* As I already committed the code, I have now switched to master branch with “**git checkout master**” and merge the branch with master with the command “**git merge feature1**

1. Delete the feature1 branch after merging.

* After merging the code, I have used the git branch -d feature1 to delete the branch.

