# GIT Assignment Part-1

● Create a new folder

● Put the following files in the folder

○ Code.txt

○ Log.txt

○ Output.txt

● Stage the Code.txt and Output.txt files

● Commit them

● And Finally push them to GitHub

Initially install git in the env that you would like to use

# Install Git On — CentOS / Fedora / RedHat / amazon Linux

sudo yum install yum-utils

sudo yum update

sudo yum install git

# Install Using Script

sudo wget https://raw.githubusercontent.com/lerndevops/labs/master/scripts/installGit.sh -P /tmp

sudo chmod 755 /tmp/installGit.sh

sudo bash /tmp/installGit.sh

# Install Git on — Ubuntu OS

sudo su -

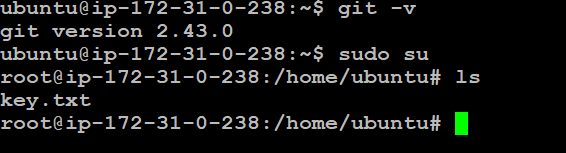
sudo apt-get install software-properties-common

sudo add-apt-repository ppa:git-core/ppa -y

sudo apt-get update

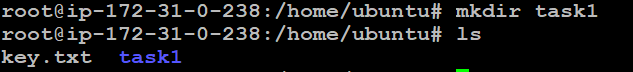
sudo apt-get install git -y

git --version ( if it writes the version then the installation is successful )



Create a new folder

mkdir task1



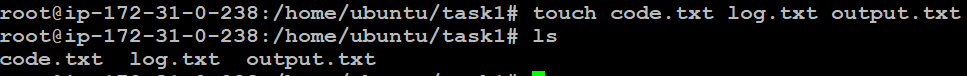
Put the following files in the folder

○ Code.txt

○ Log.txt

○ Output.txt

**touch code.txt log.txt output.txt**



Stage the Code.txt and Output.txt files

git init

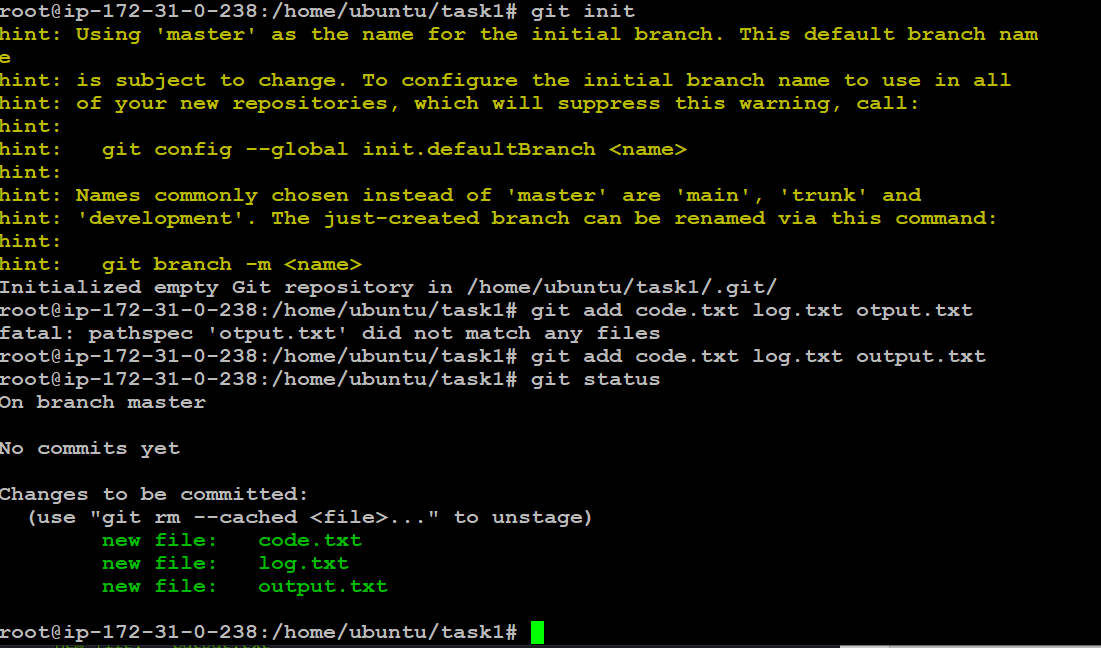
git add code.txt output.txt

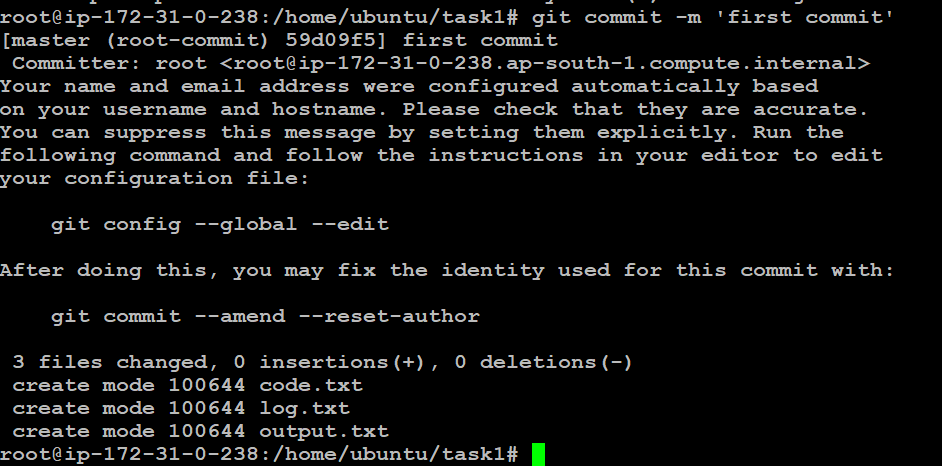
git add . **(this command will add all the files in the folder)**

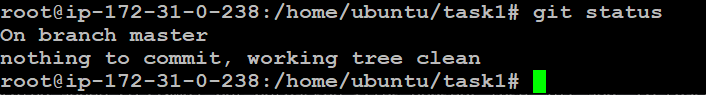
git status (to know the status of the commit)

Commit them

git commit -m “out and code”



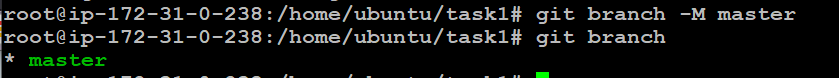




● And Finally push them to GitHub

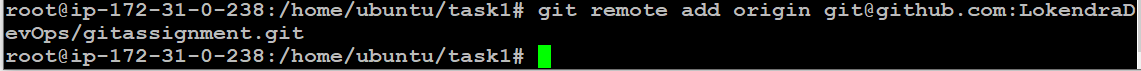
I am using HTTP URL

echo “# upgraded-funicular” >> README.md  
git init  
git add README.md  
git commit -m “first commit”

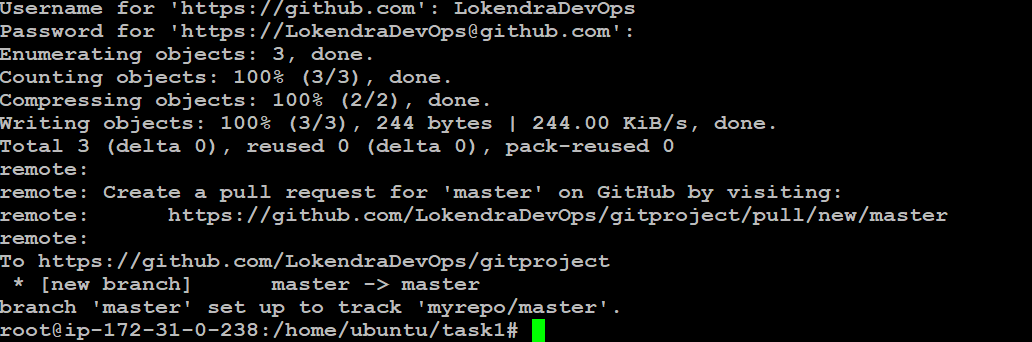
  
git branch -M master  
git remote add origin \_ username   
git push -u origin master

For SSH

echo “# upgraded-funicular” >> README.md  
git init  
git add README.md  
git commit -m “first commit”  
git branch -M master  
git remote add origin \_ username   
git push -u origin master



add username and password as generated GitHub Personal access tokens



**git — version ( if it writes the version then the installation is successful )**

# GIT Assignment Part-2

Do the following tasks:

● Create a git working directory with feature1.txt and feature2.txt in the master branch

● Create 3 branches develop, feature1 and feature2

● In develop branch create develop.txt, do not stage or commit it

● Stash this file, and checkout to feature1 branch

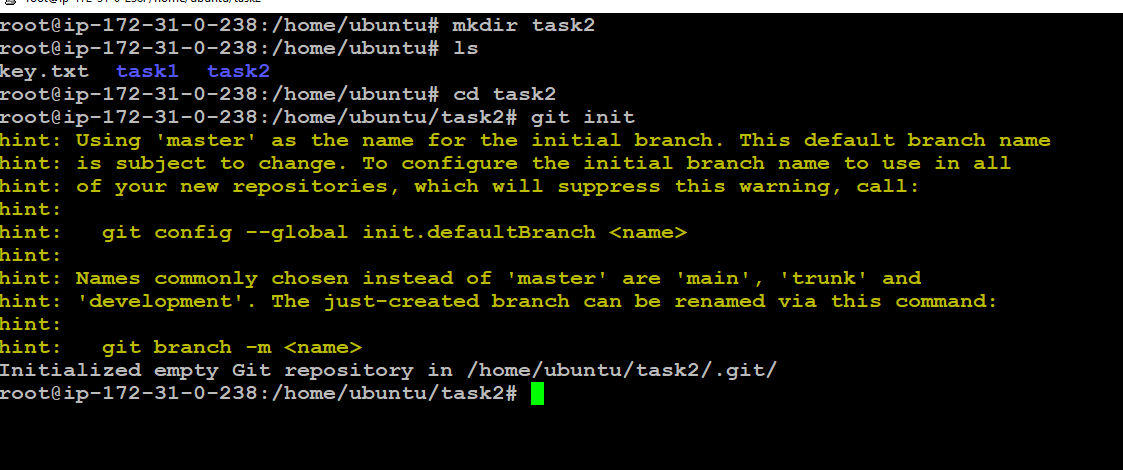
● Create new.txt file in feature1 branch, stage and commit this file

● Checkout to develop, unstash this file and commit Please submit all the git commands used to do the above steps

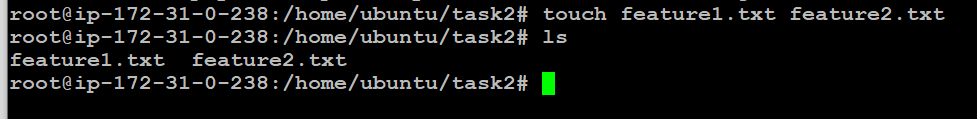
● Create a git working directory with feature1.txt and feature2.txt in the master branch

mkdir task2  
ls  
task2

git init

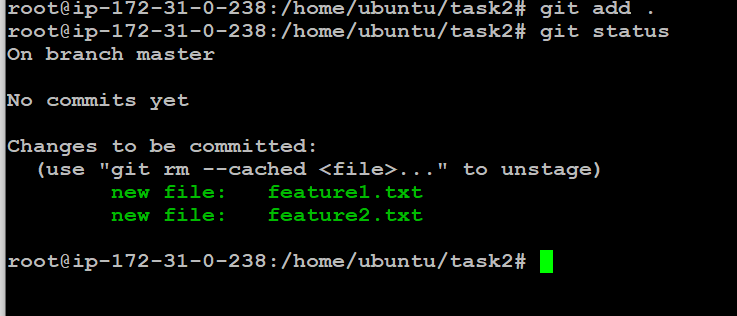


touch feature1.txt feature2.txt

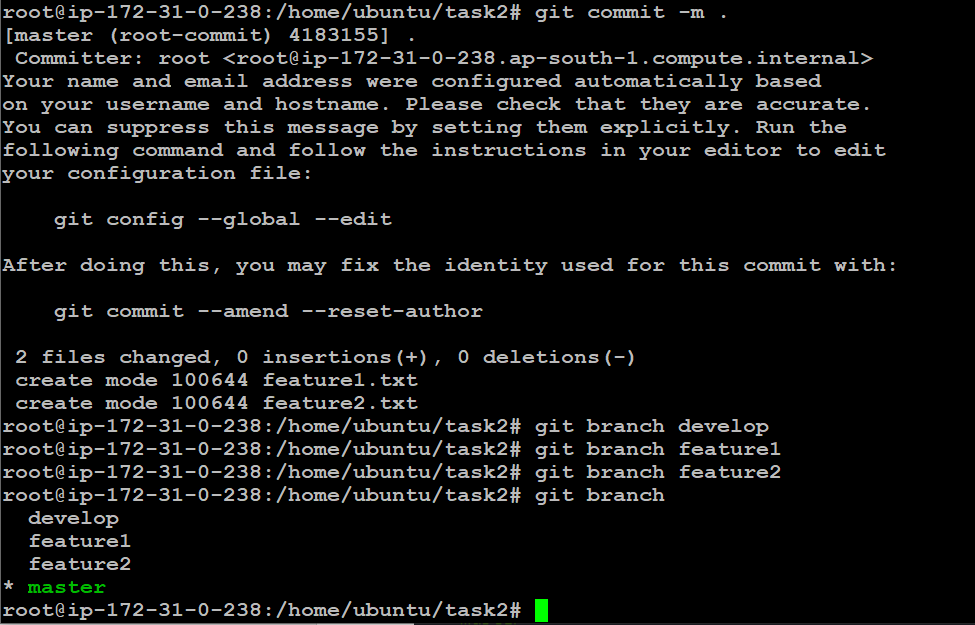


● Create 3 branches develop, feature1 and feature2

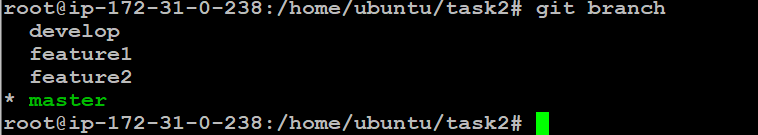
For creating branches first, we need to commit any other branches.



git branch develop  
git branch feature1  
git branch feature2



git branch (gives the list of branches)

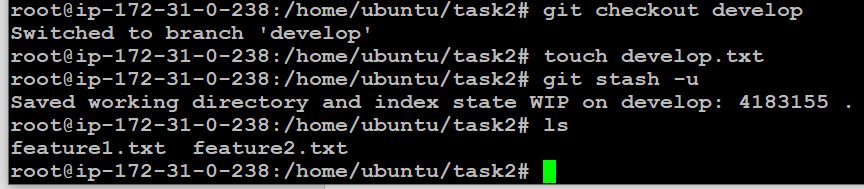


● In develop branch create develop.txt, do not stage or commit it

git checkout develop (It is used for switching the branches)

touch develop.txt

git stash -u (untracked files will be stashed)



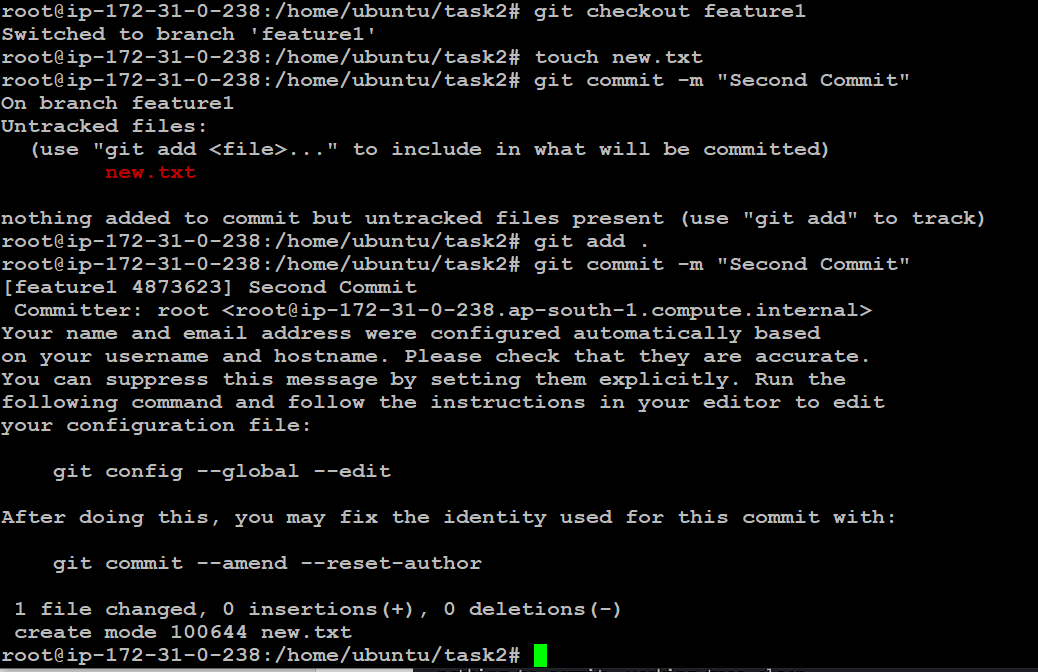
● Checkout to feature1 branch

● Create new.txt file in feature1 branch, stage and commit this file

git checkout feature1

touch new.txt

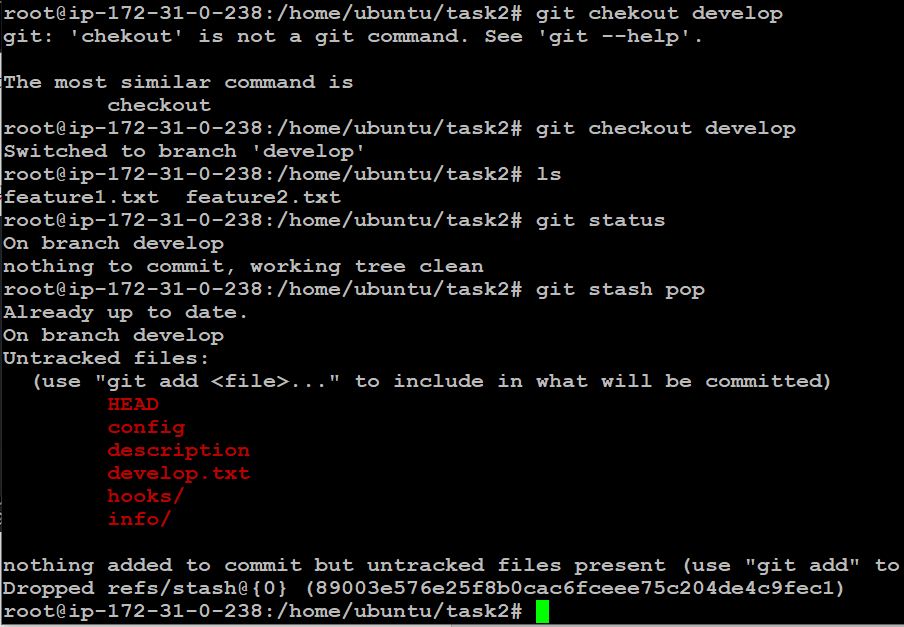
git commit -m “committing new”

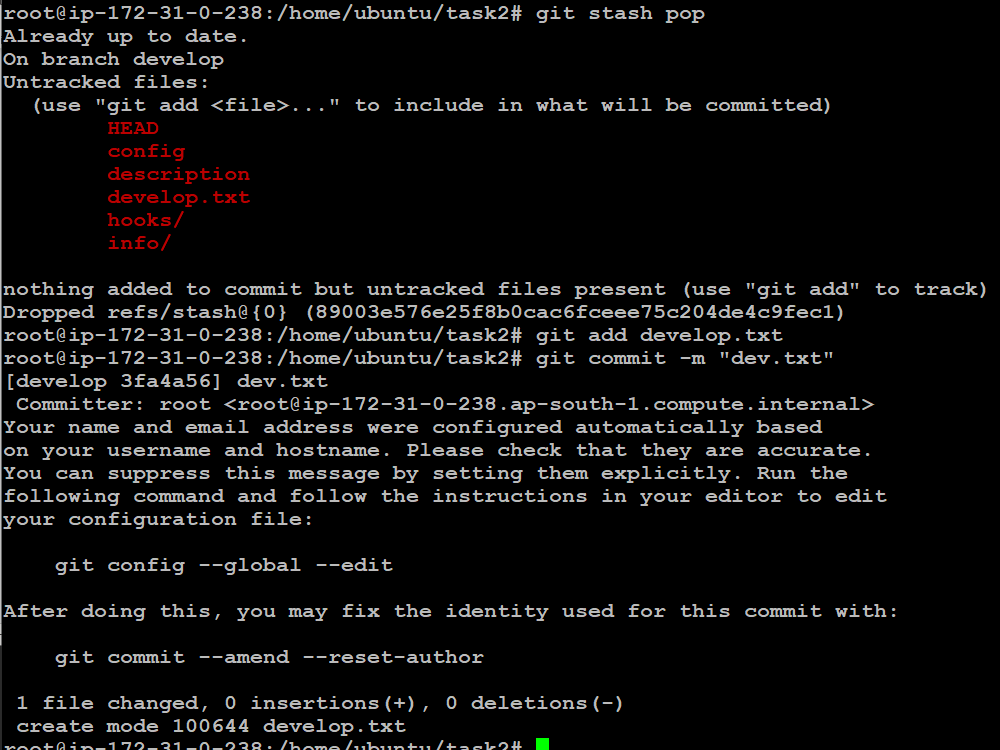


● Checkout to develop, unstash this file and commit Please submit all the git commands used to do the above steps

git checkout develop

git stash pop (It is used to stash all the files which are in unstash state)





# GIT Assignment Part-3

You have been asked to:

● Create a git working directory, with the following branches

○ Develop

○ F1

○ f2

● In the master branch, commit main.txt file

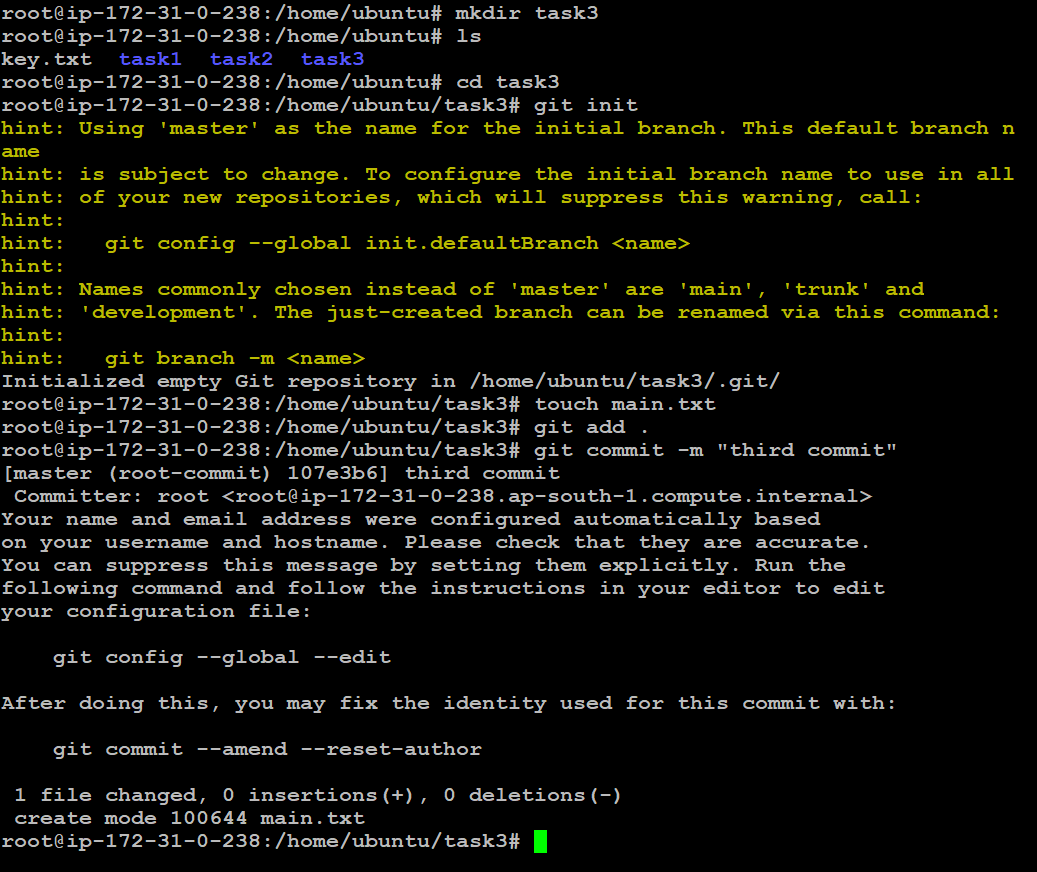
● Put develop.txt in develop branch, f1.txt and f2.txt in f1 and f2 respectively

● Push all these branches to github

● On local delete f2 branch

● Delete the same branch on github as well

● In the master branch, commit main.txt file

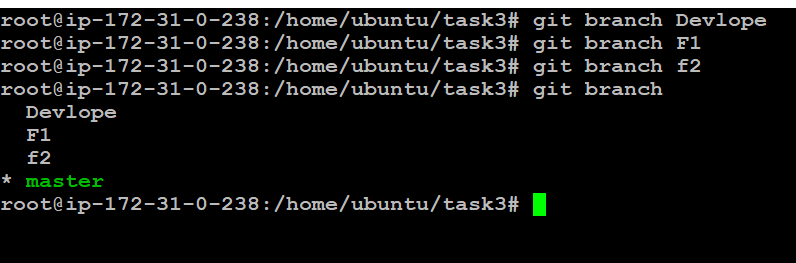


● Create a git working directory, with the following branches

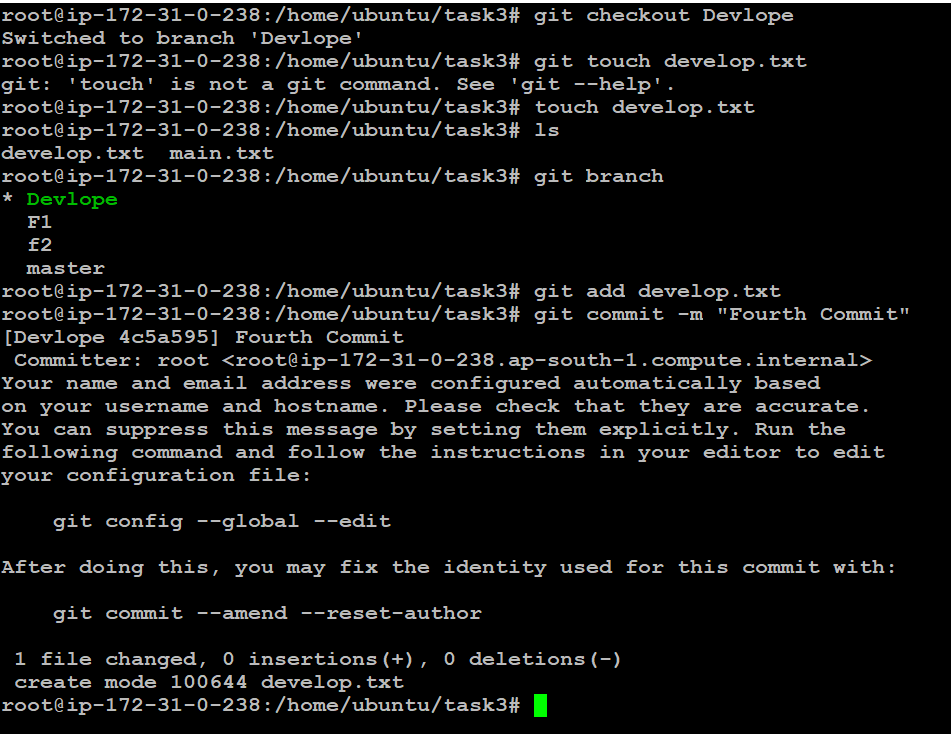
○ Develop

○ F1

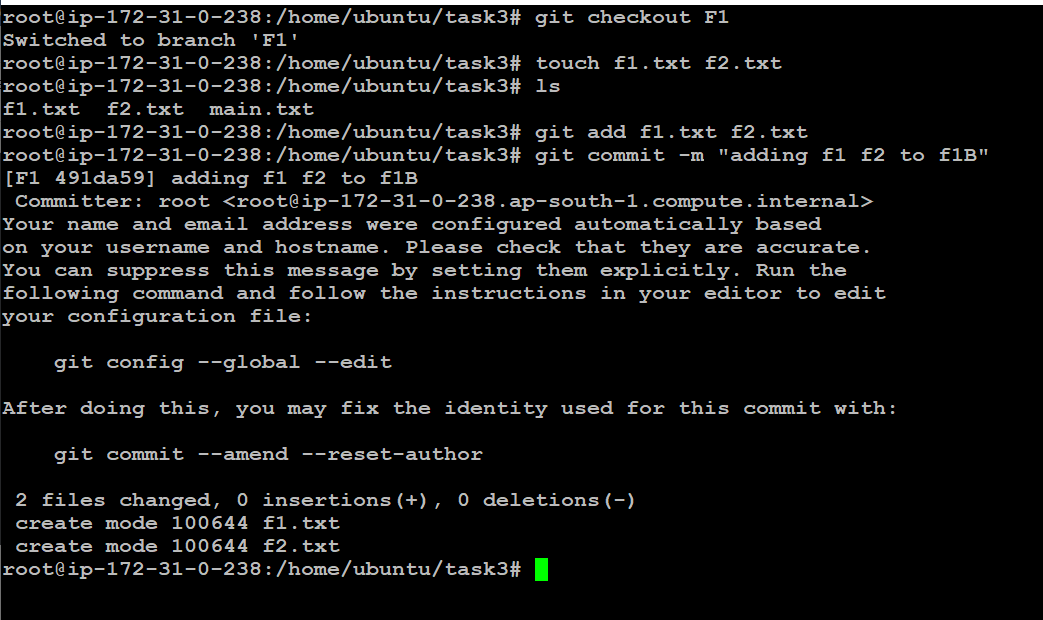
○ f2



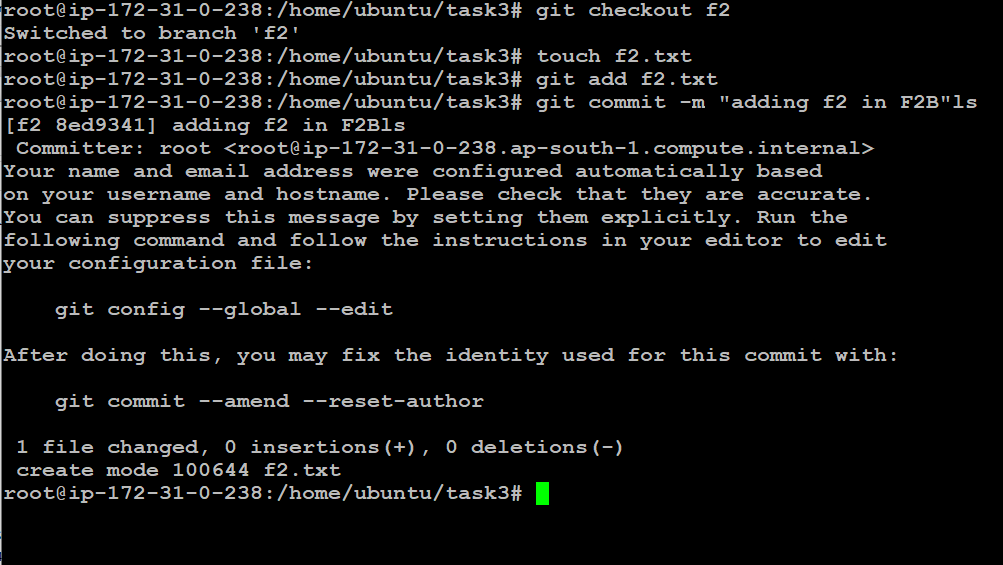
● Put develop.txt in develop branch, f1.txt and f2.txt in f1 and f2 respectively



develop.txt in develop branch



f1.txt in f1 Branch



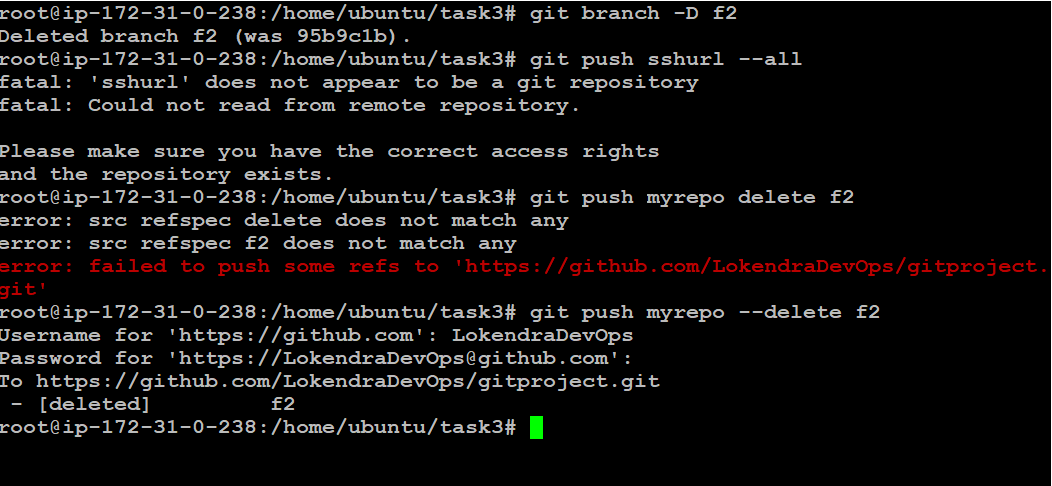
f2.txt in f2 Branch

● Push all these branches to github

git remote add sshurl <URL>

ssh-keygen

past the ssh keygen in git ssh and GPG key



● On local delete f2 branch

● Delete the same branch on github as well

git checkout master

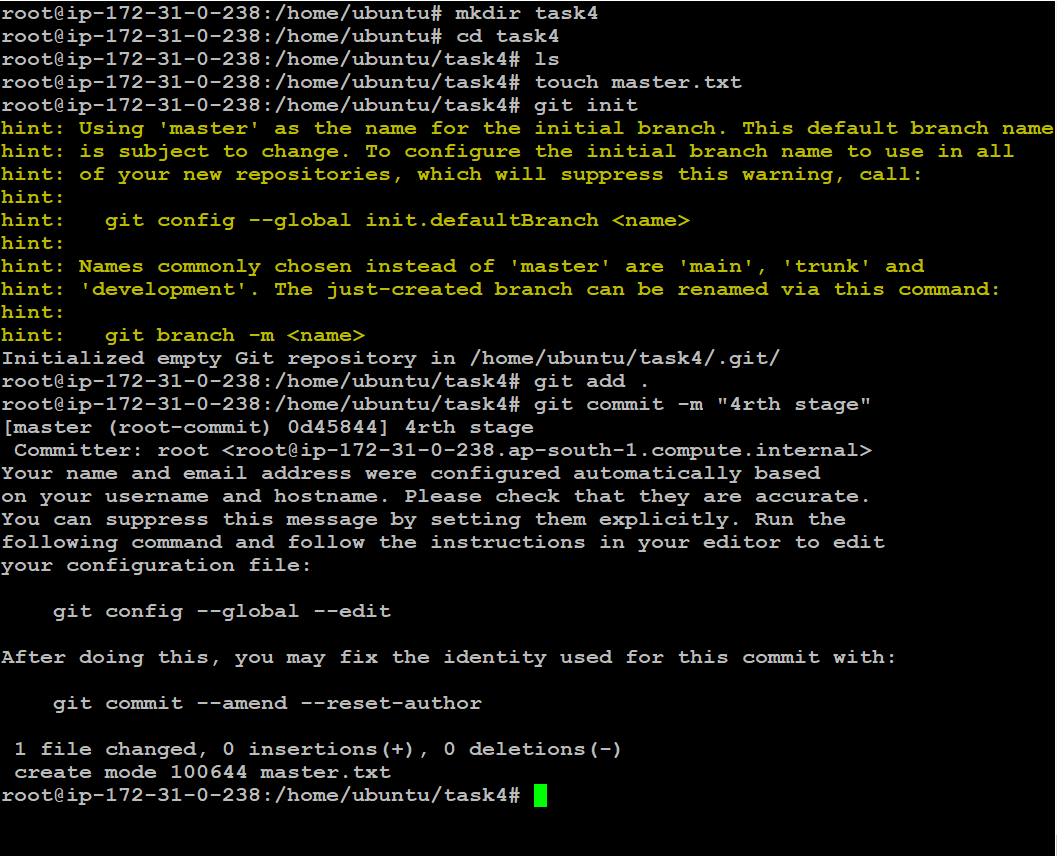
git branch -D f2

git push sshurl — delete f2

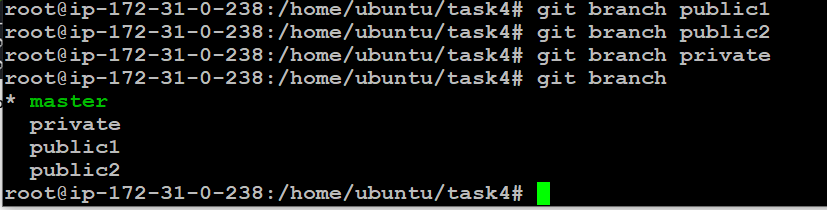
# GIT Assignment Part-4

You have been asked to:  
● Put master.txt on master branch, stage and commit  
● Create 3 branches: public1, public2 and private  
● Put public1.txt on public 1 branch, stage and commit  
● Merge public 1 on master branch  
● Merge public 2 on master branch  
● Edit master.txt on private branch, stage and commit  
● Now update branch public 1 and public 2 with new master code in private  
● Also update new master code on master  
● Finally update all the code on the private branch

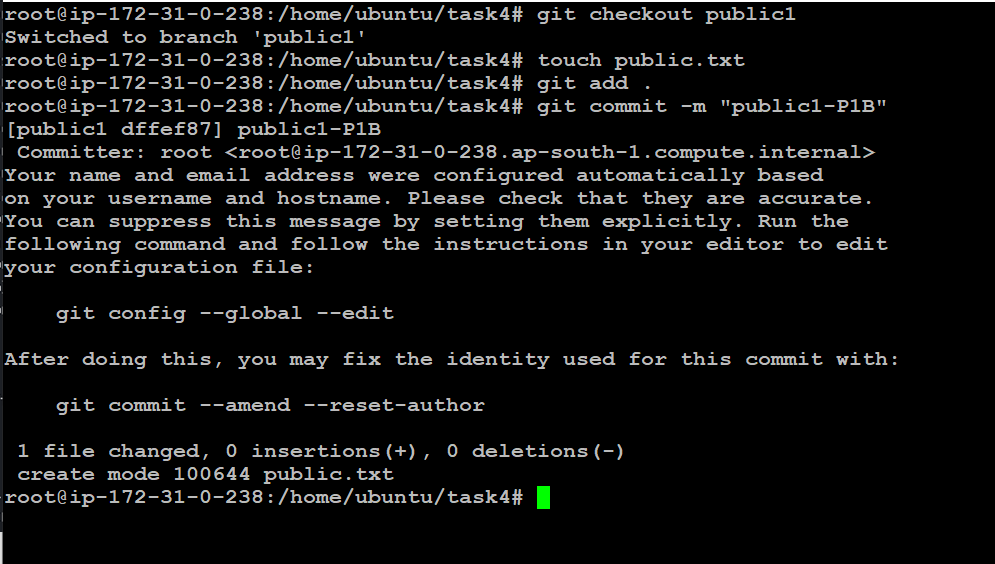
● Put master.txt on master branch, stage and commit



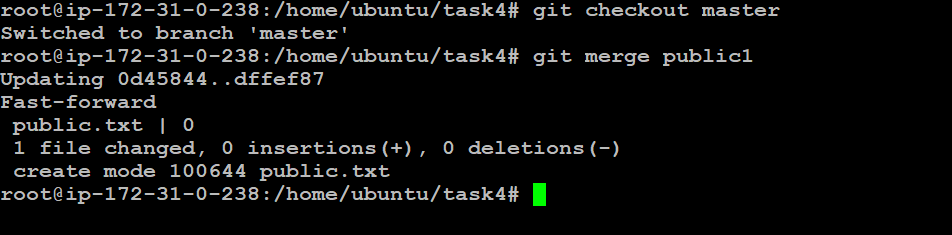
● Create 3 branches: public1, public2 and private



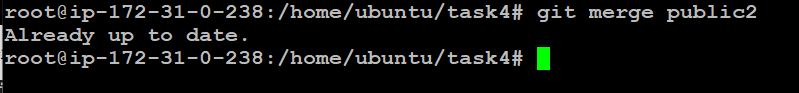
● Put public1.txt on public 1 branch, stage and commit



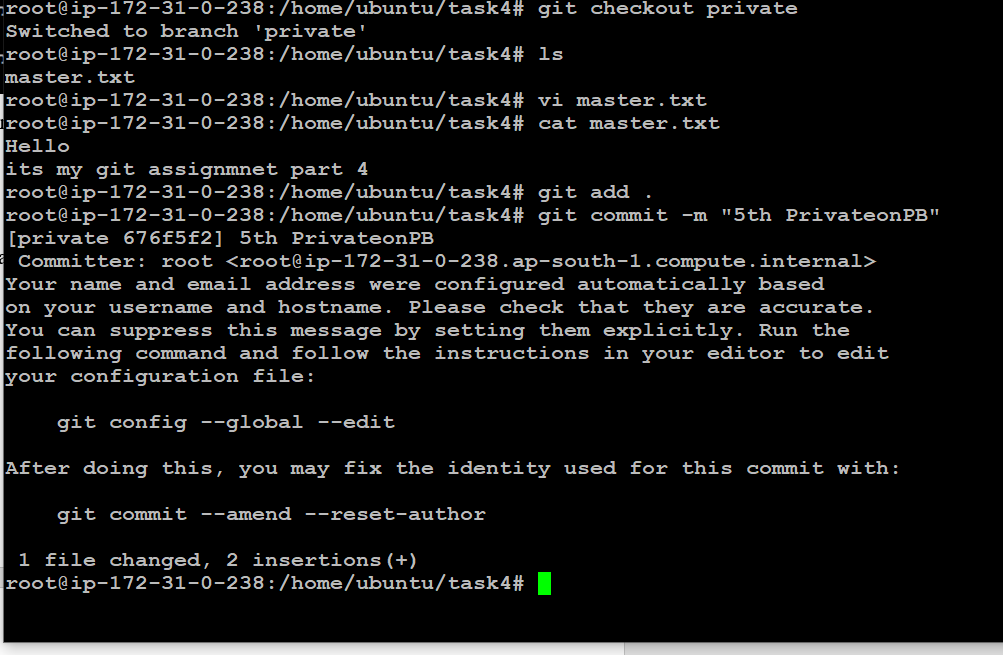
● Merge public 1 on master branch



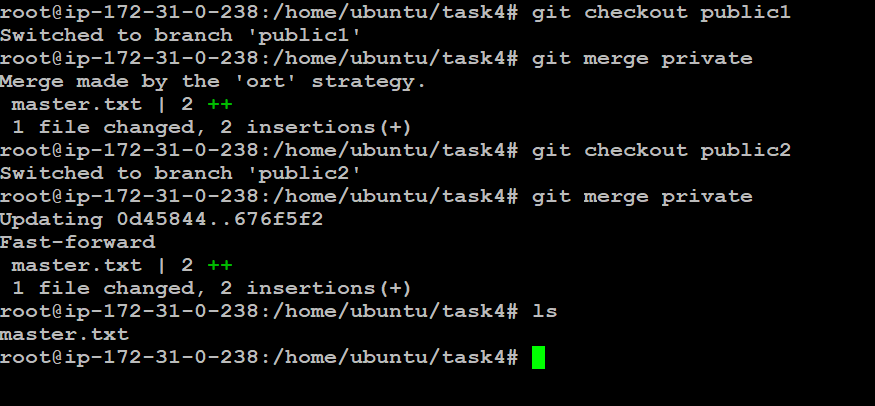
● Merge public 2 on master branch



● Edit master.txt on private branch, stage and commit

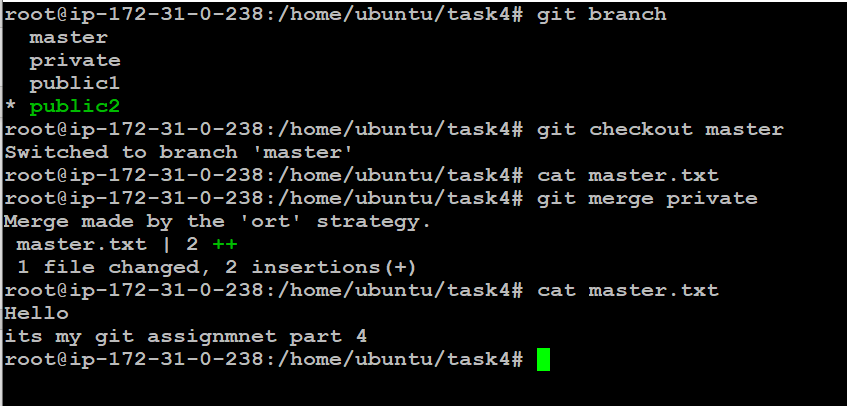


● Now update branch public 1 and public 2 with new master code in private



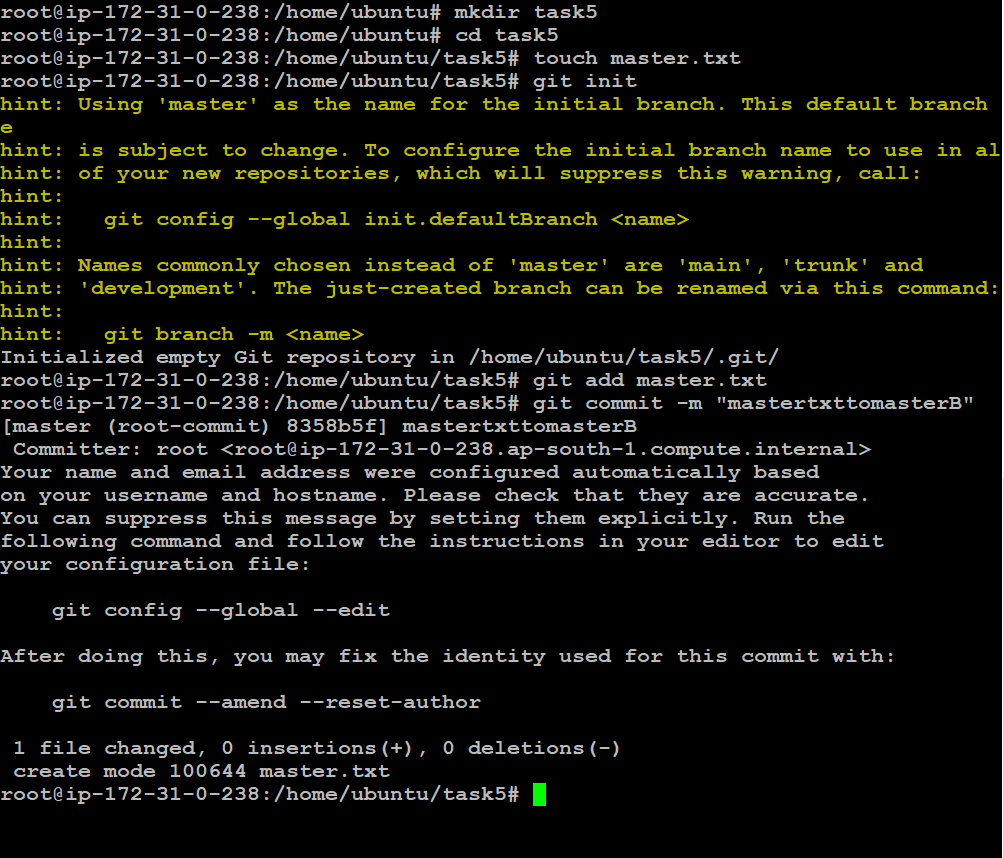
● Also update new master code on master

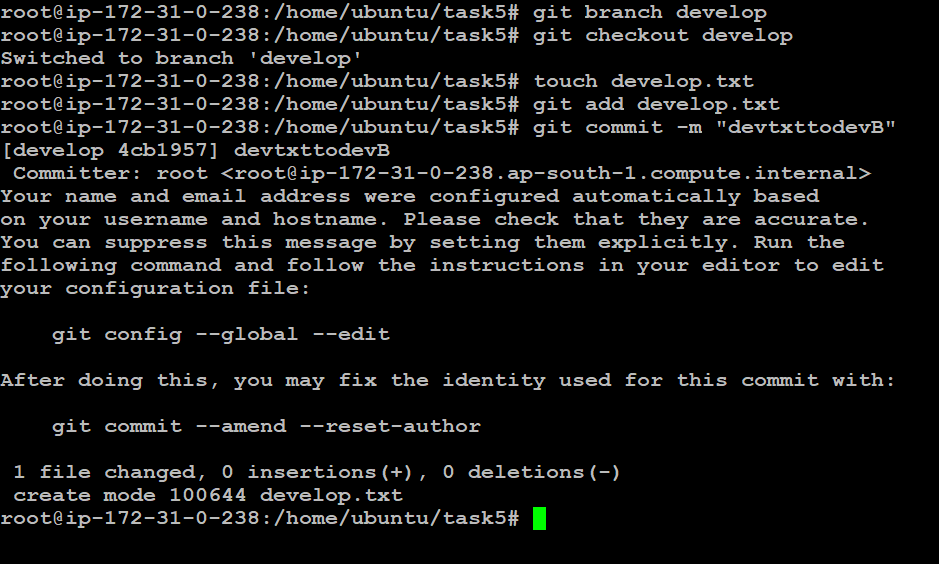
● Finally update all the code on the private branch

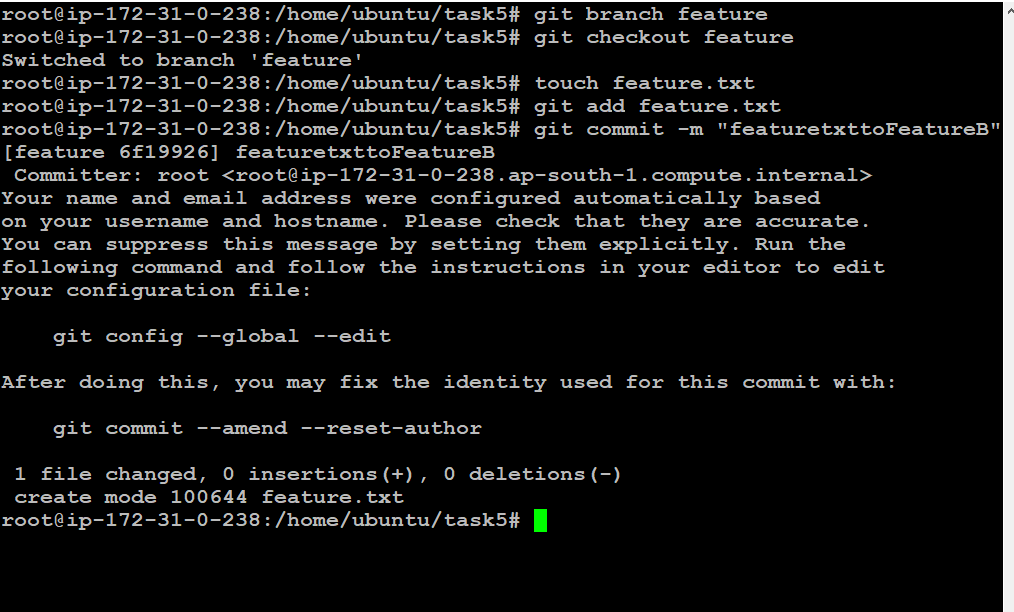


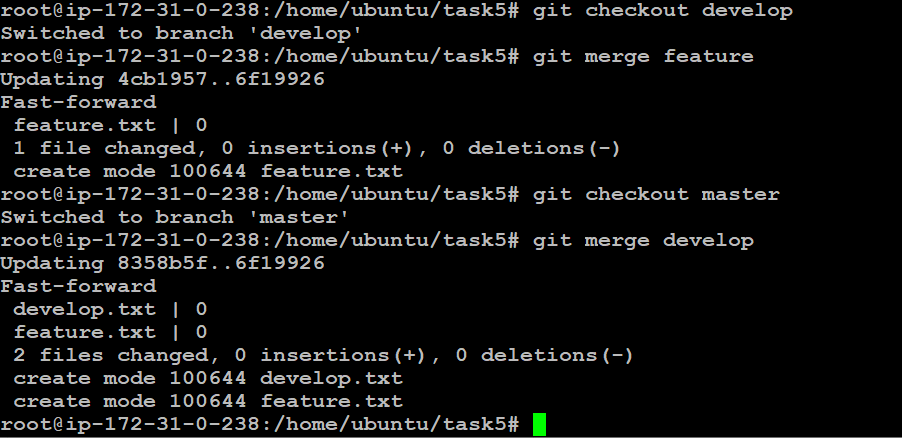
# GIT Assignment — Part 5

You have been asked to:  
● Create a gitflow workflow architecture on git  
● Create all the required branches  
● Starting from the feature branch, push the branch to the master, following the  
architecture  
● Push a urgent.txt on master using hotfix









● Push a urgent.txt on master using hotfix

