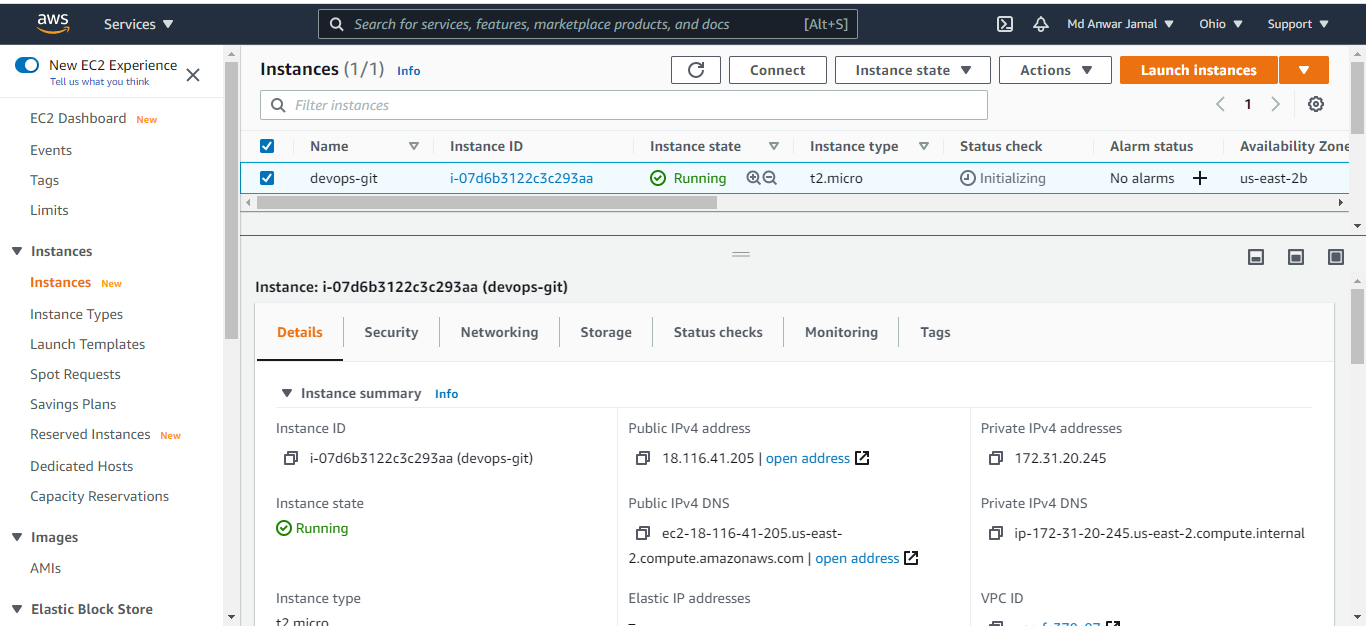
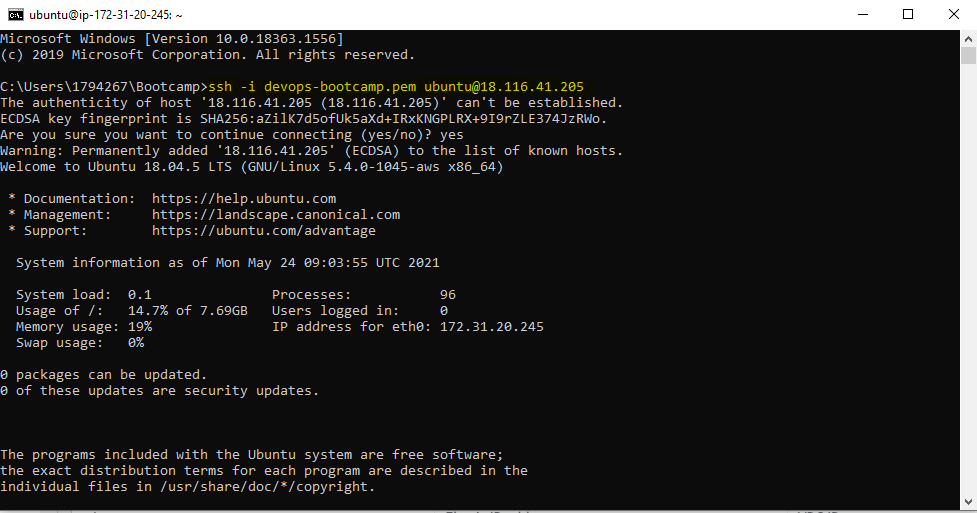
**Git Assignment for DevOps Bootcamp Batch 12**

**Page 1**

Created an AWS EC2 instance



SSH into EC2 instance using *devops-bootcamp.pem* file from CMD



**Page 2:**

# *To check version of git installed on EC2 instance*

**$git --version**

# Set user’s name for commits

**$git config --global user.name “Md Anwar Jamal”**

# Set user’s mail for commits

**$git config –global user.email “cse.mdanwarjamal@gmail.com”**

# Create new directory called developerA to emulate developerA’s system

**$mkdir developerA**

# Move into folder

**$cd developerA**

# This will initialize an empty git repository and **.git** directory is added into pwd

**$git init**

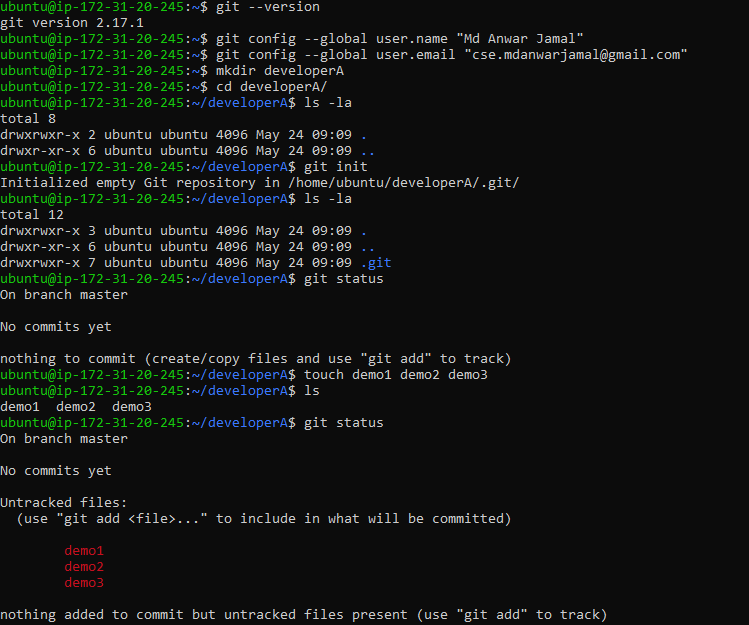
# Creates three new files demo1 demo2 demo3

**$touch demo1 demo2 demo3**

# git status will show what files are untracked as well as what files are staged

**$git status**

**Page 3**



**Page 4:**

# It will add demo1 file from working space to staging area

**$ git add demo1**

# Only demo1 file gets staged rest are still untracked

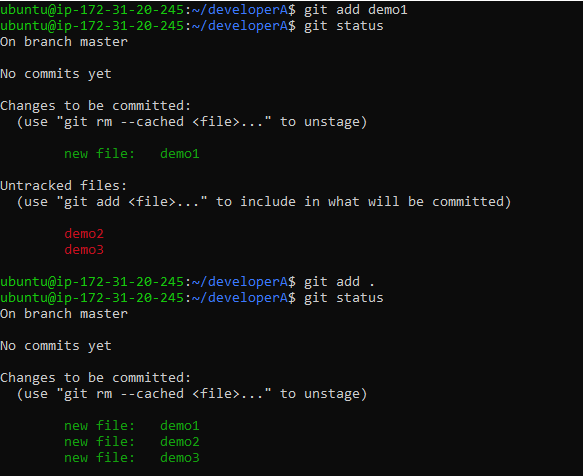
**$ git status**

# It will add all untracked files in the current directory into staging area

**$ git add .**

# All files are staged now

**$ git status**



**Page 5:**

# This will move files from staging area to Local git repository and -m to add commit message

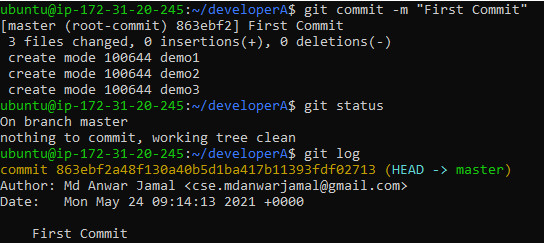
**$ git commit –m “First Commit”**

# After committing, staging area will be cleared out

**$ git status**

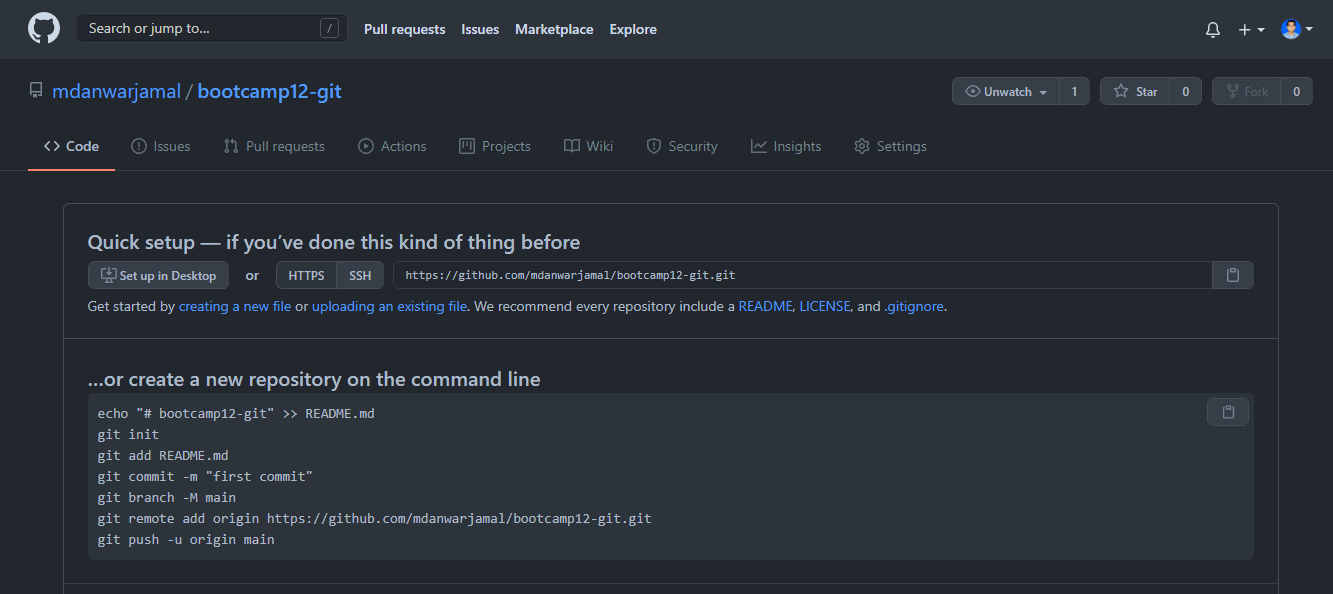
# It will show history of commits, (Commit SHA ID, Author’s Details set during $ git config, Date and Time of commit as well as Commit Message

**$ git log**



**Page 6**

Created an empty GitHub repository “**bootcamp12-git**”



**Page 7**

# To add remote repo to local repo

# **origin** is a variable that holds Remote Repo URL as value, so whenever we refer origin, we internally are referring to remote repo

**$ git remote add origin** [**https://github.com/mdanwarjamal/bootcamp12-git.git**](https://github.com/mdanwarjamal/bootcamp12-git.git )

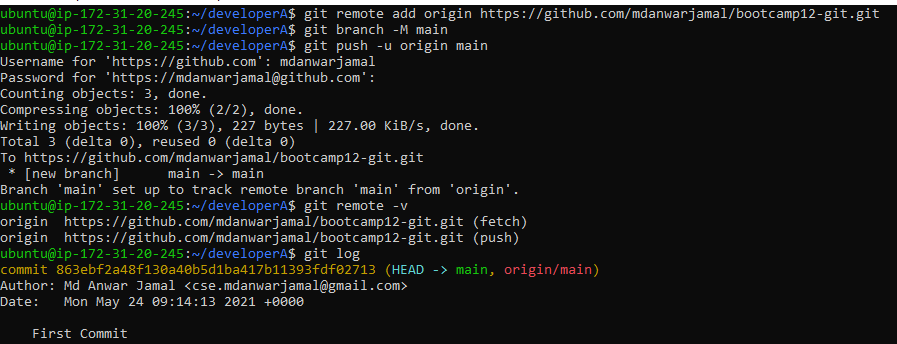
# **-M** to rename **master (**created by default when we do **git init**) branch into **main**

**$ git branch –M main**

# To push files from Local git repository to already added Remote GitHub Repository.

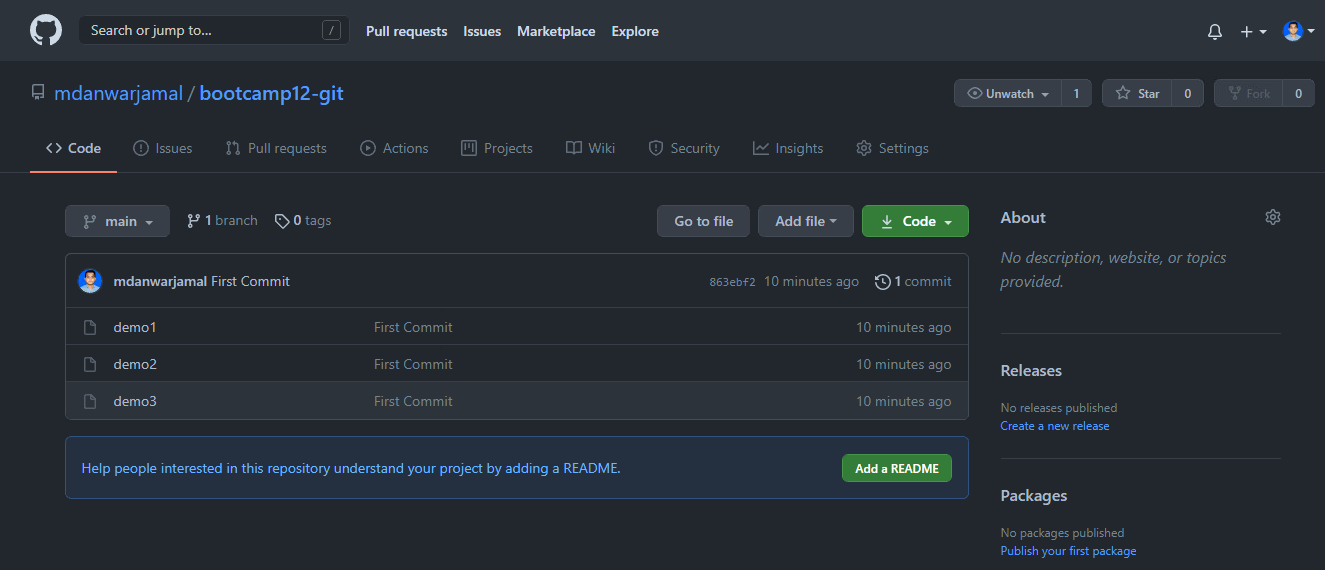
# -u is for user (It will ask for username and password)

**$git push origin –u origin main**



**Page 8**

After git push, state of GitHub Repository



**Page 9**

# Create developerB directory to emulate developer’s B system

**$ mkdir developerB**

# Move into developerB’s system

**$ cd developerB**

# It will Clone the remote repository and create **bootcamp12-git** directory with branch as **main**

**$ git clone** [**https://github.com/mdanwarjamal/bootcamp12-git.git**](https://github.com/mdanwarjamal/bootcamp12-git.git)

# Show commit history

**$ git log**

# Editing demo1 file

**$ vi demo1**

# Doing git status to check status of files

**$ git status**

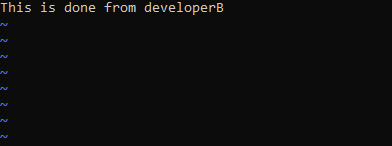
# Adding modified demo1 file into staging area

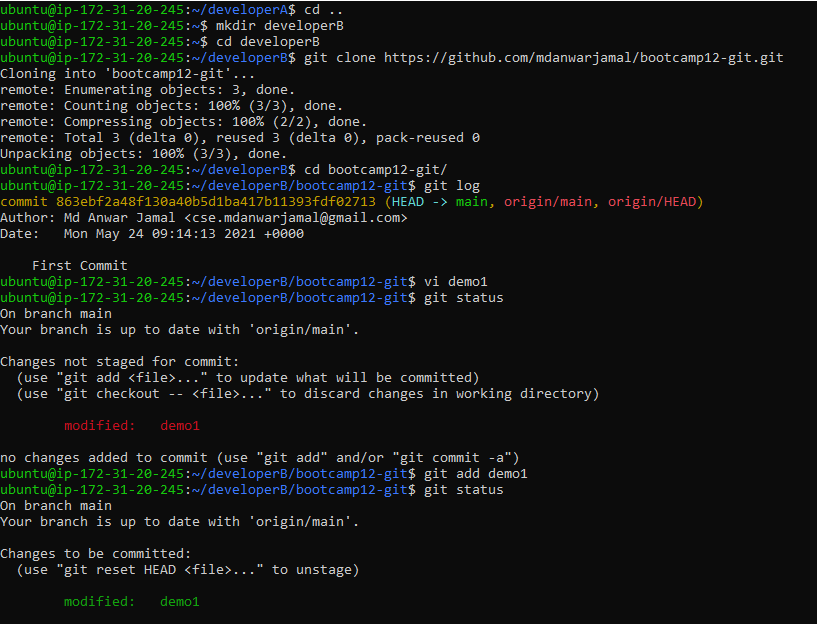
**$ git add demo1**

# Doing git status to check status of files

**$ git status**

**Page 10**





**Page 11**

***\*\*Sorry forgot to take the screenshot of below commands\*\****

**#** Doing git config so that developerB details will be reflected in commit history, these are local git config values and will not be affecting global settings

**$ git config user.name “Developer B”**

**$ git config user.email “developerB@email.com”**

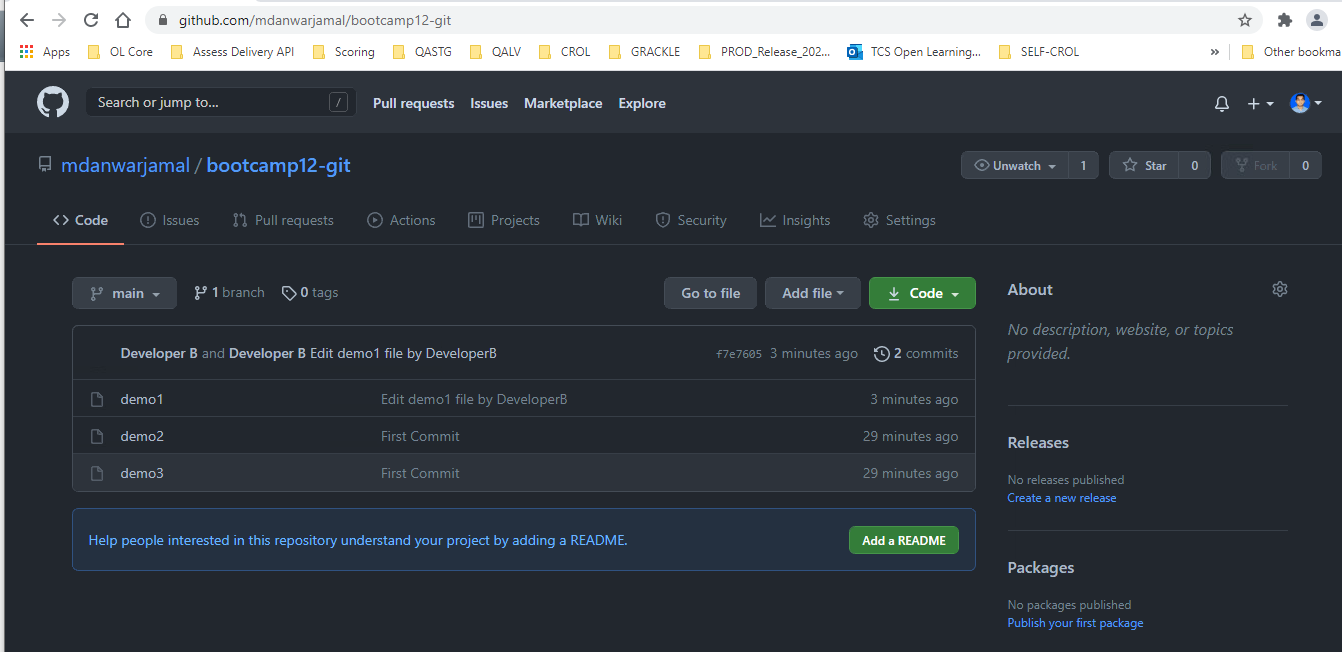
**#** Moving modified files from staging area to local repository

**$ git commit –m “Edit demo1 file by Developer B”**

**#** Pushing file changes from Local to Remote Repo

**$ git push origin main**

State of GitHub after Developer B commits and push those changes to GitHub



**Page 12**

# Move back to developerA’s system

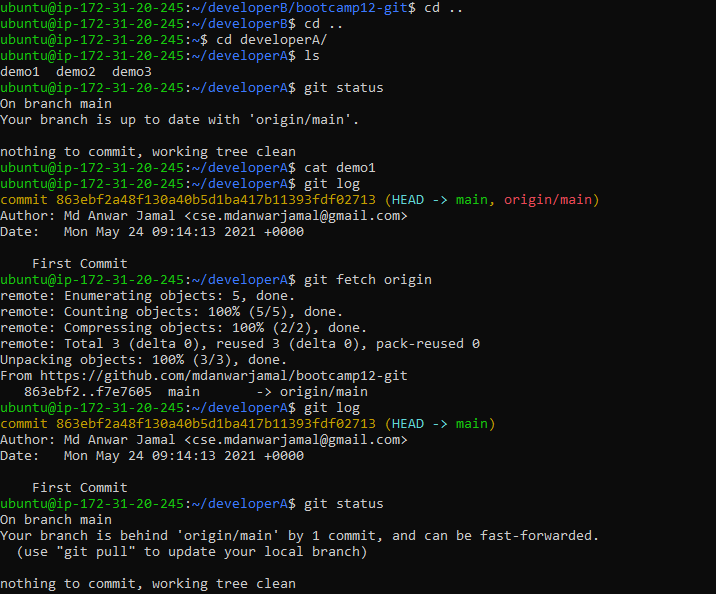
**$ cd ../developerA/**

# Check commit history

**$ git log**

# Fetching data from GitHub repo, it will not change working space. But metadata of changes in remote are fetched to local repo

**$ git fetch origin**



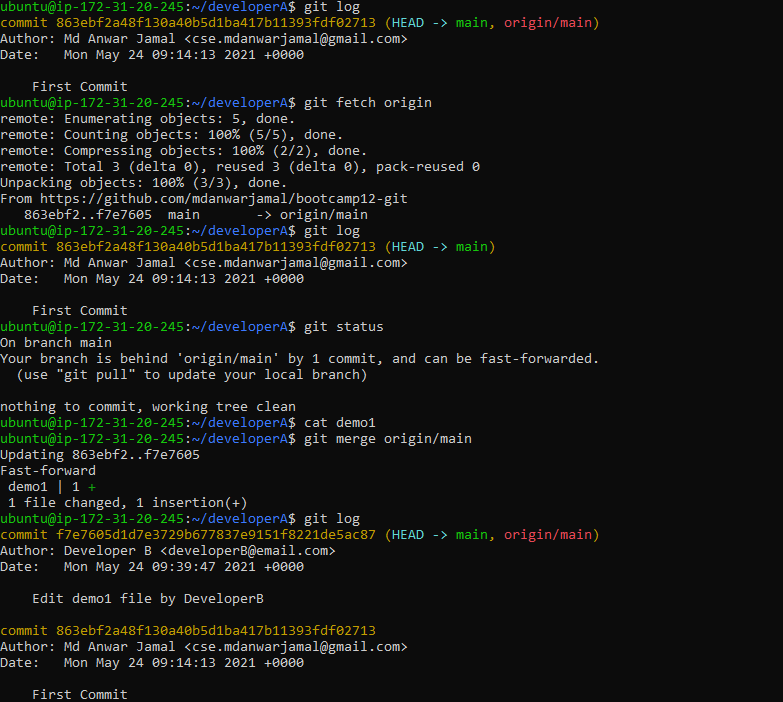
**Page 13**

# Fetching data from GitHub repo, it will not change working space. But metadata of changes in remote are fetched

**$ git fetch origin**

# Merging fetched changes of remote repo into local workspace area as well

**$ git merge origin/main**



**Page 14**

# List all local branch

**$ git branch**

# Create new feature1 branch

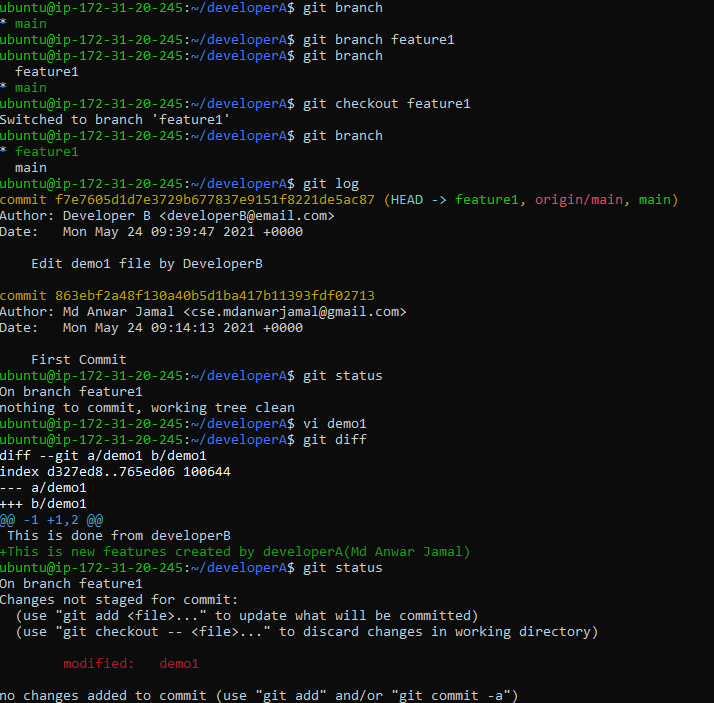
**$ git branch feature1**  
# Switch branch from main to feature1

**$ git checkout feature**

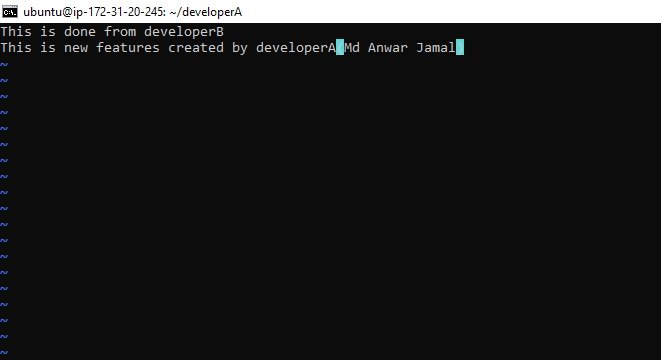
# Edit demo1 file to add features

**$ vi demo1**

# Display difference between local and working space version of files **$ git diff**



**Page 15**



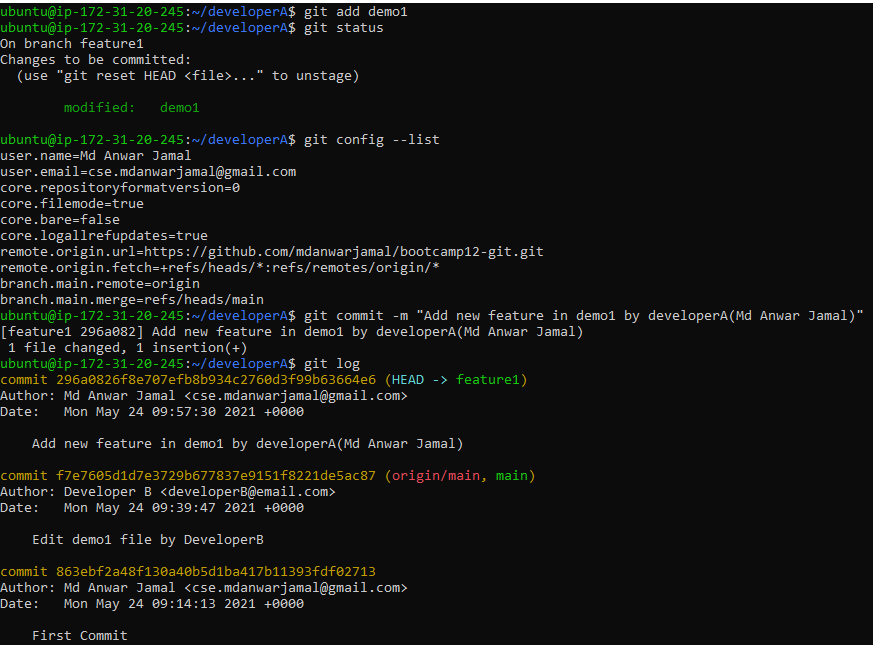
**Page 16**

# Adding files to staging area

**$git add demo1**

# Committing changes for new feature

**$git commit –m “Add new feature in demo1 by develeoperA(Md Anwar Jamal)”**



**Page 17**

**#** Switch to main branch

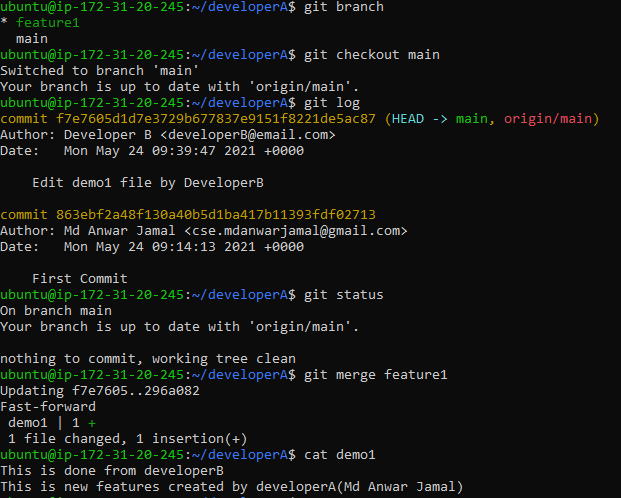
#We have to be in the **destination branch** where other branch needs to be merged

**$ git checkout main**

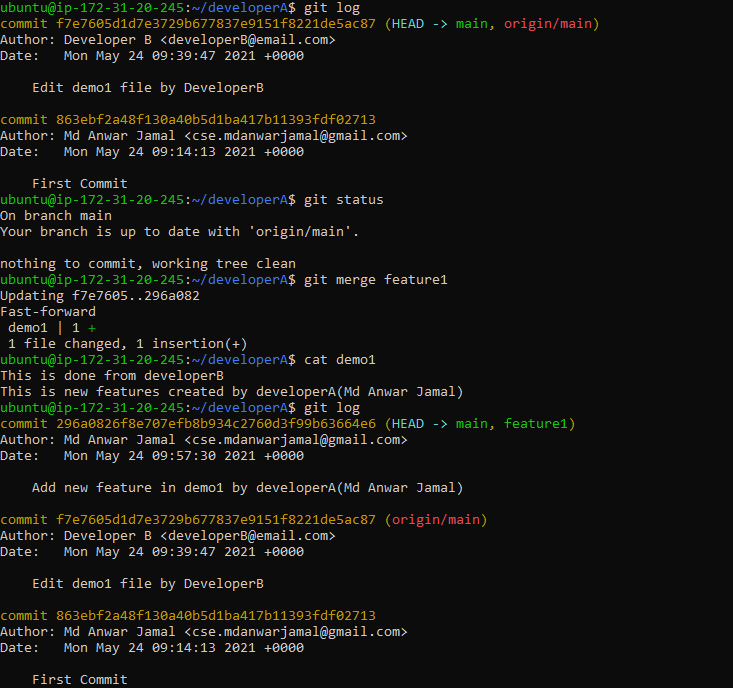
**#** Merging **feature1** branch into **main** branch

# Fast-forward means it is applying those changes in sequence and HEAD of master moves over new commits

**$ git merge feature1**



**Page 18**



**Page 19**

**#** List all local branches

**$ git branch**

**#** Delete feature1 branch

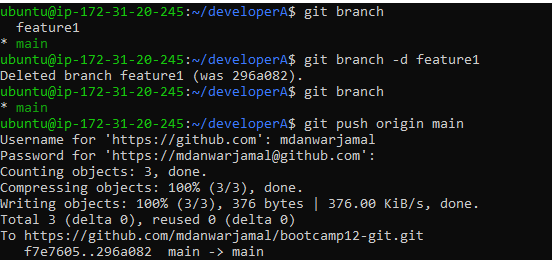
# **-d** denoted merged branch

# **-D** denotes unmerged branch

**$ git branch –d feature1**

**#** Push code to github repository

**$ git push origin main**



**Page 20**

After git merge, state of GitHub Repo

