1)Describe the usage of the git stash command by using an example and also state the process by giving the screenshot of all the commands written in git bash.

Usage of the git statsh:-

The git stash commad is used to temporarily save changes in a Git repository that are not yet ready to be committed.

The **git stash command** enables you to switch branches without committing the current branch.

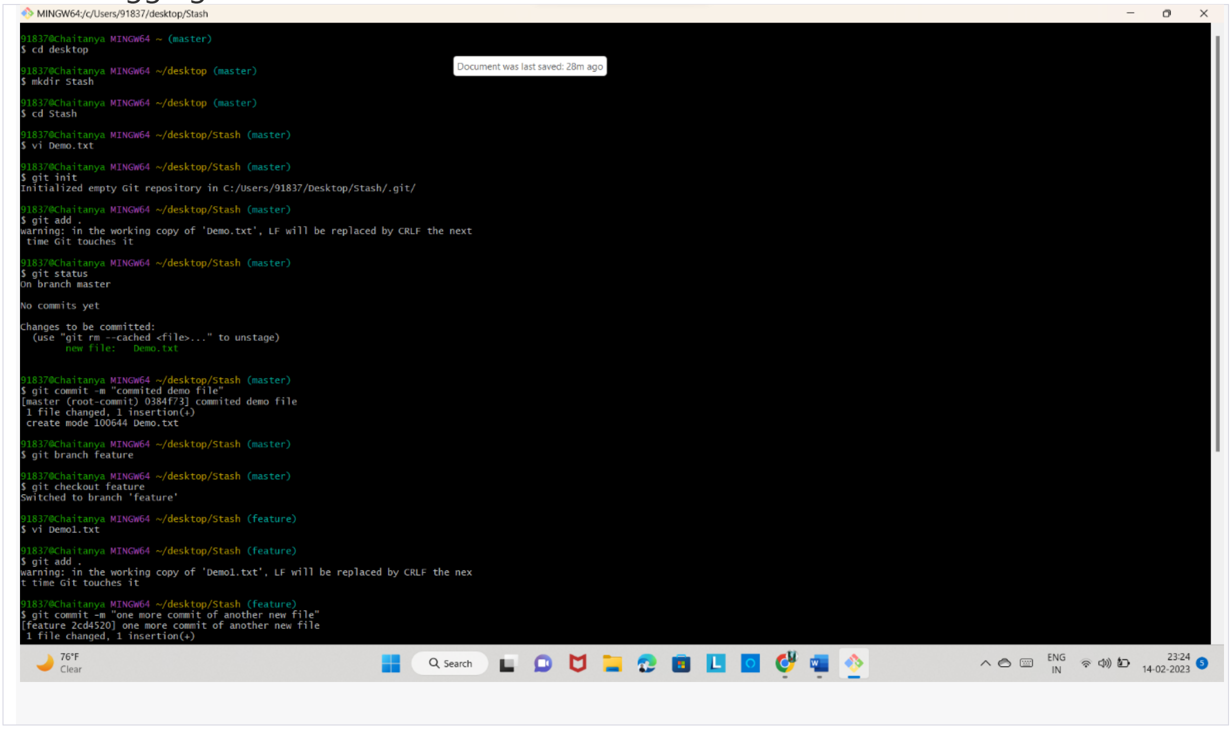
Generally, the stash's meaning is "**store something safely in a hidden place**." The sense in Git is also the same for stash; Git temporarily saves your data safely without committing.

Git stash uses **STACK** data structure.

Here are the steps to use git stash:-

Step:-1

Create a directory within a directory created a file and initialize the file,add the file into stagging area and commited the file

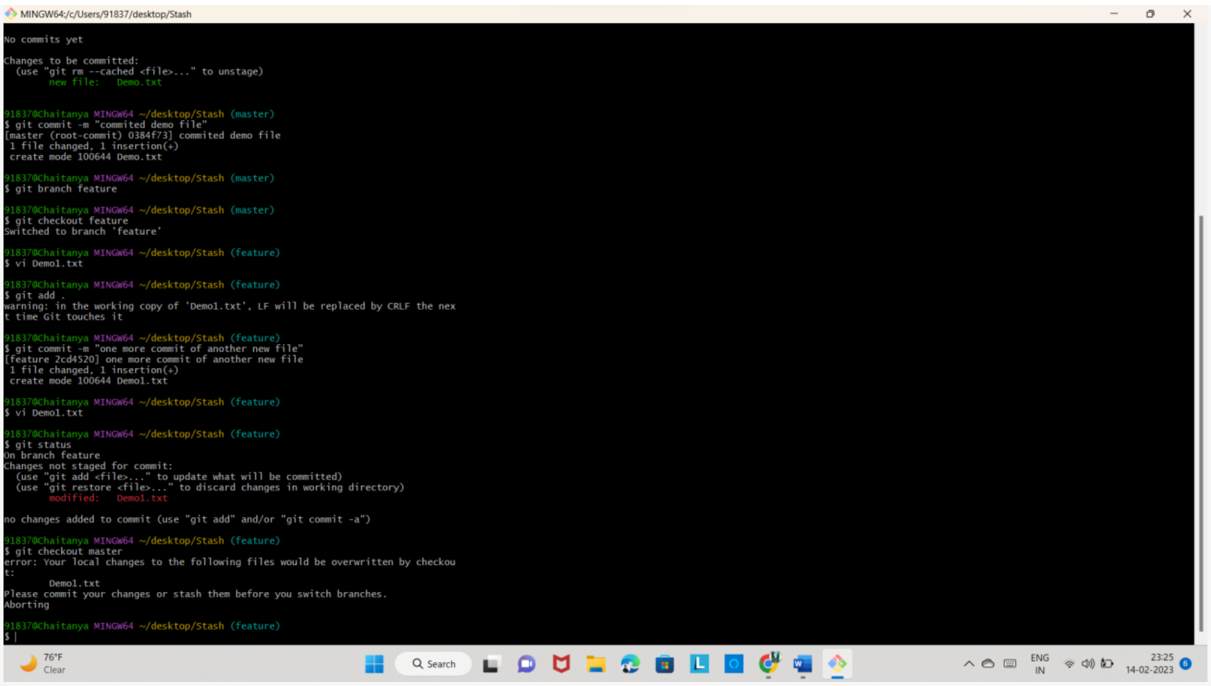


```
MINGW64:/c/Users/91837/desktop/Stash

91837@Chaitanya MINGW64 ~ (master)
$ cd desktop

91837@Chaitanya MINGW64 ~/desktop (master)
$ mkdir Stash

91837@Chaitanya MINGW64 ~/desktop (master)
$ cd Stash

91837@Chaitanya MINGW64 ~/desktop/Stash (master)
$ vi Demo.txt

91837@Chaitanya MINGW64 ~/desktop/Stash (master)
$ git init
Initialized empty Git repository in C:/Users/91837/Desktop/Stash/.git/

91837@Chaitanya MINGW64 ~/desktop/Stash (master)
$ git add .
warning: in the working copy of 'Demo.txt', LF will be replaced by CRLF the next
 time Git touches it

91837@Chaitanya MINGW64 ~/desktop/Stash (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   Demo.txt


91837@Chaitanya MINGW64 ~/desktop/Stash (master)
$ git commit -m "commited demo file"
[master (root-commit) 0384f73] commited demo file
 1 file changed, 1 insertion(+)
 create mode 100644 Demo.txt

91837@Chaitanya MINGW64 ~/desktop/Stash (master)
$ git branch feature

91837@Chaitanya MINGW64 ~/desktop/Stash (master)
$ git checkout feature
Switched to branch 'feature'

91837@Chaitanya MINGW64 ~/desktop/Stash (feature)
$ vi Demo1.txt

91837@Chaitanya MINGW64 ~/desktop/Stash (feature)
$ git add .
warning: in the working copy of 'Demo1.txt', LF will be replaced by CRLF the nex
t time Git touches it

91837@Chaitanya MINGW64 ~/desktop/Stash (feature)
$ git commit -m "one more commit of another new file"
[feature 2cd4520] one more commit of another new file
 1 file changed, 1 insertion(+)
```

Step:-2

Next, Created a branch named as feature and again created a file ,add the file into stagging area and commited the file.

Later,done some modifications in one file and the status of a file .When the file in modified stage try to switch the branch then it shows an error

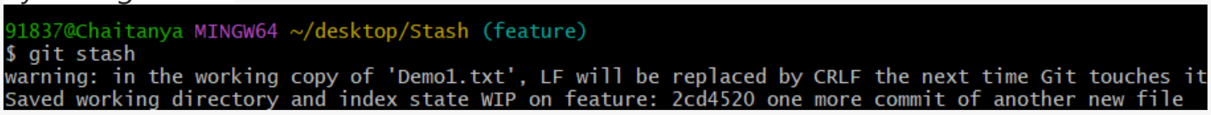


```
No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   Demo.txt


91837@Chaitanya MINGW64 ~/desktop/Stash (master)
$ git commit -m "commited demo file"
[master (root-commit) 0384f73] commited demo file
 1 file changed, 1 insertion(+)
 create mode 100644 Demo.txt

91837@Chaitanya MINGW64 ~/desktop/Stash (master)
$ git branch feature

91837@Chaitanya MINGW64 ~/desktop/Stash (master)
$ git checkout feature
Switched to branch 'feature'

91837@Chaitanya MINGW64 ~/desktop/Stash (feature)
$ vi Demo1.txt

91837@Chaitanya MINGW64 ~/desktop/Stash (feature)
$ git add .
warning: in the working copy of 'Demo1.txt', LF will be replaced by CRLF the nex
t time Git touches it

91837@Chaitanya MINGW64 ~/desktop/Stash (feature)
$ git commit -m "one more commit of another new file"
[feature 2cd4520] one more commit of another new file
 1 file changed, 1 insertion(+)
 create mode 100644 Demo1.txt

91837@Chaitanya MINGW64 ~/desktop/Stash (feature)
$ vi Demo1.txt

91837@Chaitanya MINGW64 ~/desktop/Stash (feature)
$ git status
On branch feature
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   Demo1.txt

no changes added to commit (use "git add" and/or "git commit -a")

91837@Chaitanya MINGW64 ~/desktop/Stash (feature)
$ git checkout master
error: Your local changes to the following files would be overwritten by checkou
t:
        Demo1.txt
Please commit your changes or stash them before you switch branches.
Aborting

91837@Chaitanya MINGW64 ~/desktop/Stash (feature)
$ |
```

Git Stash:-

Stashes the modified files and creates new stash

Syntax:-git stash

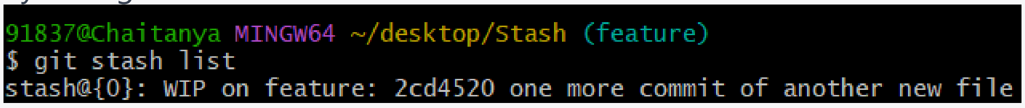


```
91837@Chaitanya MINGW64 ~/desktop/Stash (feature)
$ git stash
warning: in the working copy of 'Demo1.txt', LF will be replaced by CRLF the next time Git touches it
Saved working directory and index state WIP on feature: 2cd4520 one more commit of another new file
```

Git Stash List:-

To see all the stashes list we use

Syntax:-git stash list

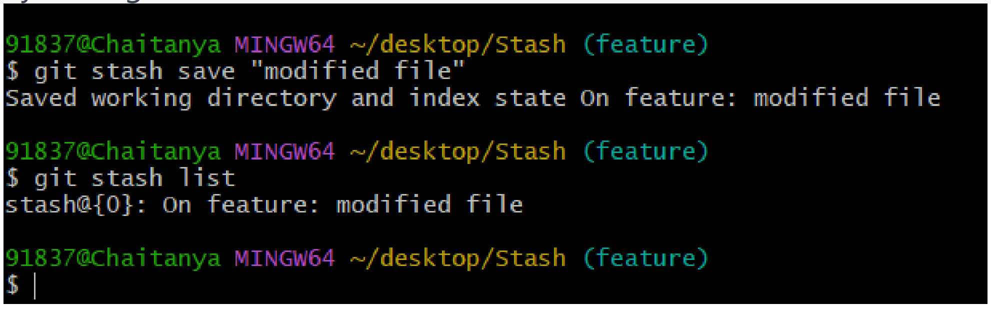


```
91837@Chaitanya MINGW64 ~/desktop/Stash (feature)
$ git stash list
stash@{0}: WIP on feature: 2cd4520 one more commit of another new file
```

Git Stash Save:-

If you want to give the name for that particular stash

Syntax:- git stash save<name>

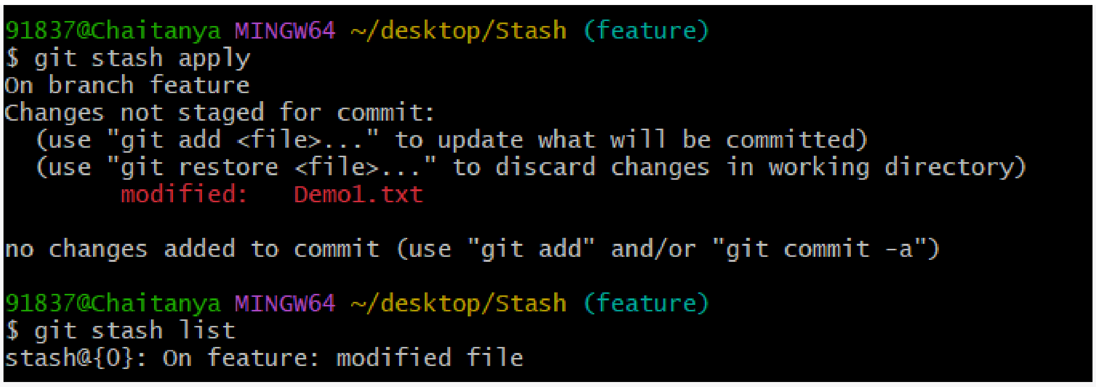


```
91837@Chaitanya MINGW64 ~/desktop/Stash (feature)
$ git stash save "modified file"
Saved working directory and index state On feature: modified file

91837@Chaitanya MINGW64 ~/desktop/Stash (feature)
$ git stash list
stash@{0}: On feature: modified file

91837@Chaitanya MINGW64 ~/desktop/Stash (feature)
$ |
```

Git stash apply:-

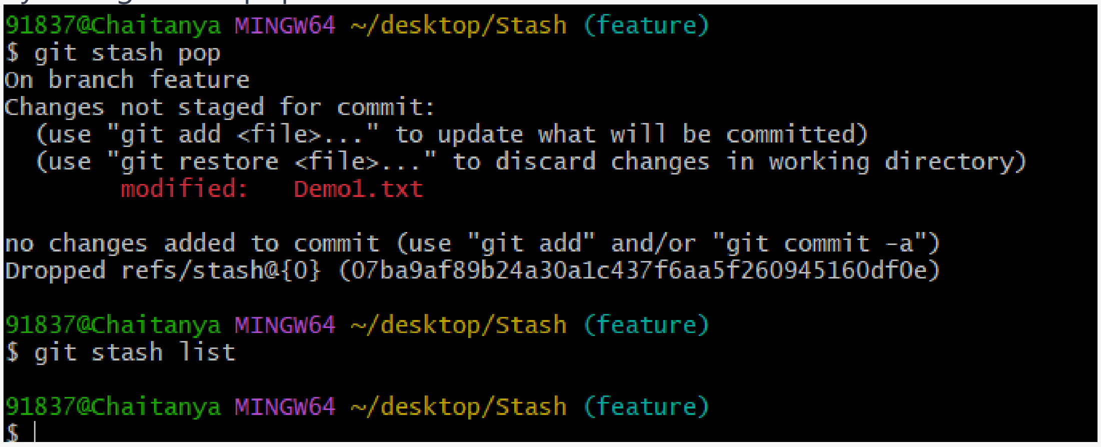
To apply the particular stash

Syntax:-git stash apply<stash-id>



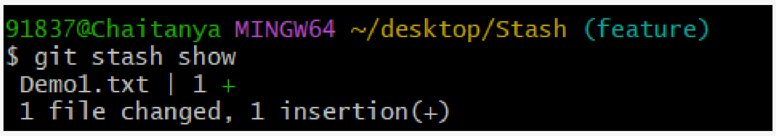
```
91837@Chaitanya MINGW64 ~/desktop/Stash (feature)
$ git stash apply
On branch feature
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   Demo1.txt

no changes added to commit (use "git add" and/or "git commit -a")

91837@Chaitanya MINGW64 ~/desktop/Stash (feature)
$ git stash list
stash@{0}: On feature: modified file
```

Git Stash Pop:-
To retrive the recent stashed data into the branch
After getting the stash data it removes automatically until and unless if there are no conflicts
Syntax:-git stash pop

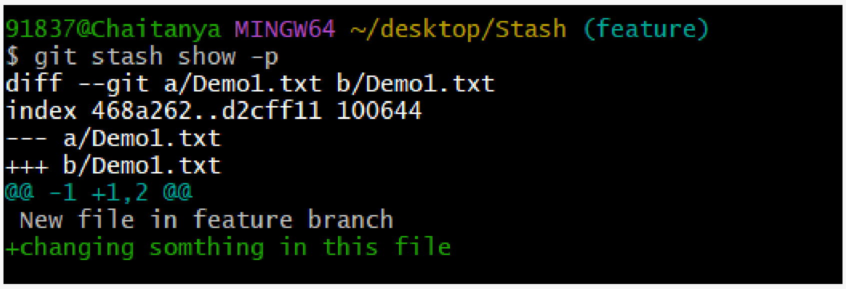


```
91837@Chaitanya MINGW64 ~/desktop/Stash (feature)
$ git stash pop
On branch feature
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   Demo1.txt

no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (07ba9af89b24a30a1c437f6aa5f260945160df0e)

91837@Chaitanya MINGW64 ~/desktop/Stash (feature)
$ git stash list

91837@Chaitanya MINGW64 ~/desktop/Stash (feature)
$ |
```

Git Stash Show:-
To check the changes of the stashed data before pulling
Syntax:-git stash show



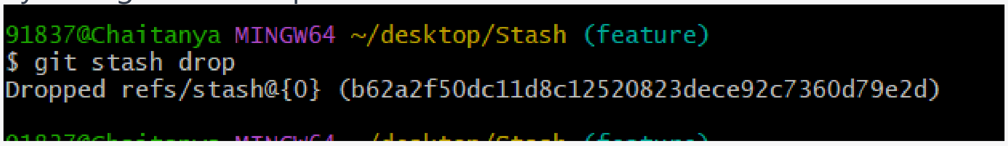
```
91837@Chaitanya MINGW64 ~/desktop/Stash (feature)
$ git stash show
 Demo1.txt | 1 +
 1 file changed, 1 insertion(+)
```



```
91837@Chaitanya MINGW64 ~/desktop/Stash (feature)
$ git stash show -p
diff --git a/Demo1.txt b/Demo1.txt
index 468a262..d2cff11 100644
--- a/Demo1.txt
+++ b/Demo1.txt
@@ -1 +1,2 @@
 New file in feature branch
+changing somthing in this file
```

Git Stash Drop:-
To delete particular stash details

Syntax:-git stash drop<stash-id>

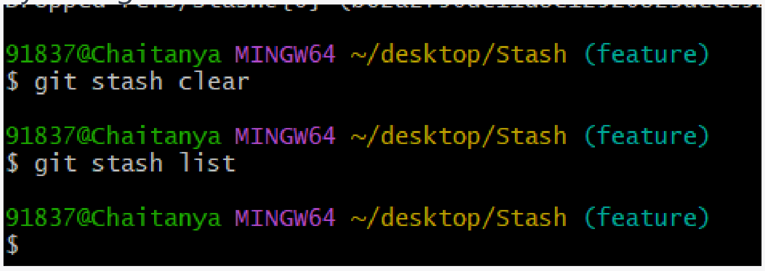


```
91837@Chaitanya MINGW64 ~/desktop/Stash (feature)
$ git stash drop
Dropped refs/stash@{0} (b62a2f50dc11d8c12520823dece92c7360d79e2d)
```

Git Clear:-

Clear the complete stash details

Syntax:-git stash clear



```
91837@Chaitanya MINGW64 ~/desktop/Stash (feature)
$ git stash clear

91837@Chaitanya MINGW64 ~/desktop/Stash (feature)
$ git stash list

91837@Chaitanya MINGW64 ~/desktop/Stash (feature)
$
```

2)By using a sample example of your choice, use the git fetch command and also use the git merge command and describe the whole process through a screenshot with all the commands and their output in git bash.

Git Fetch:-

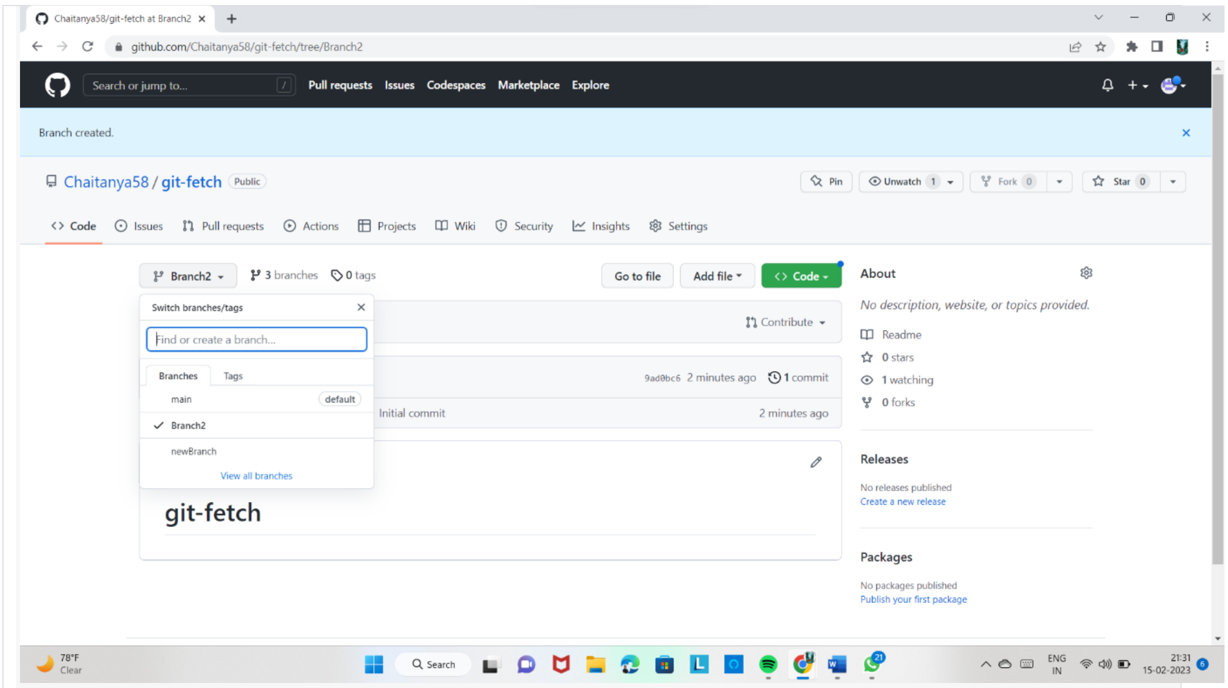
The git fetch command downloads commits, files, and refs from a remote repository into your local repository.Fetching allows us to download changes from remote repository.But those changes will not be automatically integrated to our working files.

Syntax:- git fetch<remote>

This command fetches branches and history from a specific remote repository.it only updates the remote tracking branchesEg**:- git fetch origin**
 git fetch origin would fetch all changes from the origin remote repository

We can also fetch a specific branch from a remote using the following command.
Syntax**:-git fetch<remote><branch>**
Eg:-git fetch origin master
Retrive the latest information from the master branch on the origin remote repository.

Process of git fetch command:-
1.Check the status of your local repository with the command git status.this will show you the current state of your local repository.
2.Create a new repository named as git-fetch in github
3.In that repository create two branches named as newBranch, branch2


Chaitanya58/git-fetch at Branch2
github.com/Chaitanya58/git-fetch/tree/Branch2
Search or jump to...
Pull requests Issues Codespaces Marketplace Explore
Chaitanya58 / git-fetch Public
Pin Unwatch 1 Fork 0 Star 0
Code Issues Pull requests Actions Projects Wiki Security Insights Settings
Branch2 3 branches 0 tags
Go to file Add file Code
Switch branches/tags
Find or create a branch...
Branches Tags
main default
Branch2
newBranch
View all branches
Contribute
About
Readme
0 stars
1 watching
0 forks
Releases
Create a new release
Packages
Publish your first package

Branch created.

9ad0bc6 2 minutes ago 1 commit

Initial commit 2 minutes ago

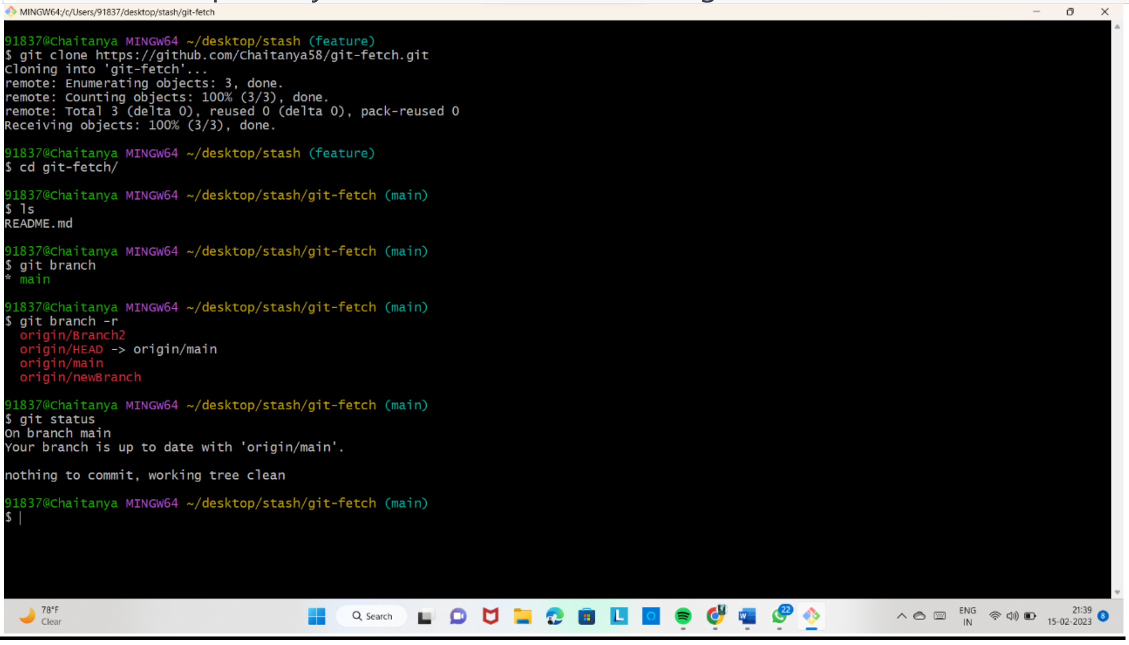
git-fetch

No description, website, or topics provided.

No releases published

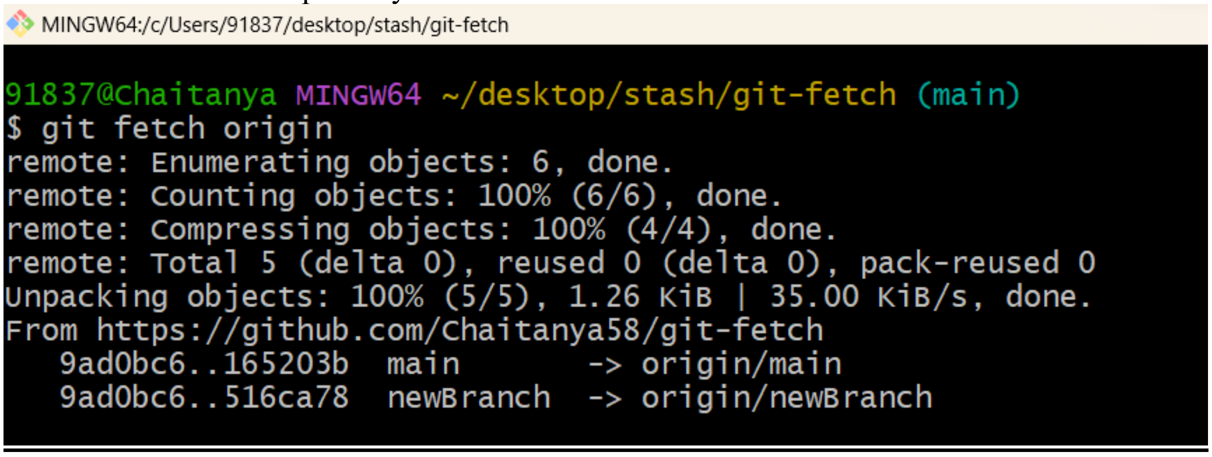
No packages published

5.clone the repository and add some files in the git branches



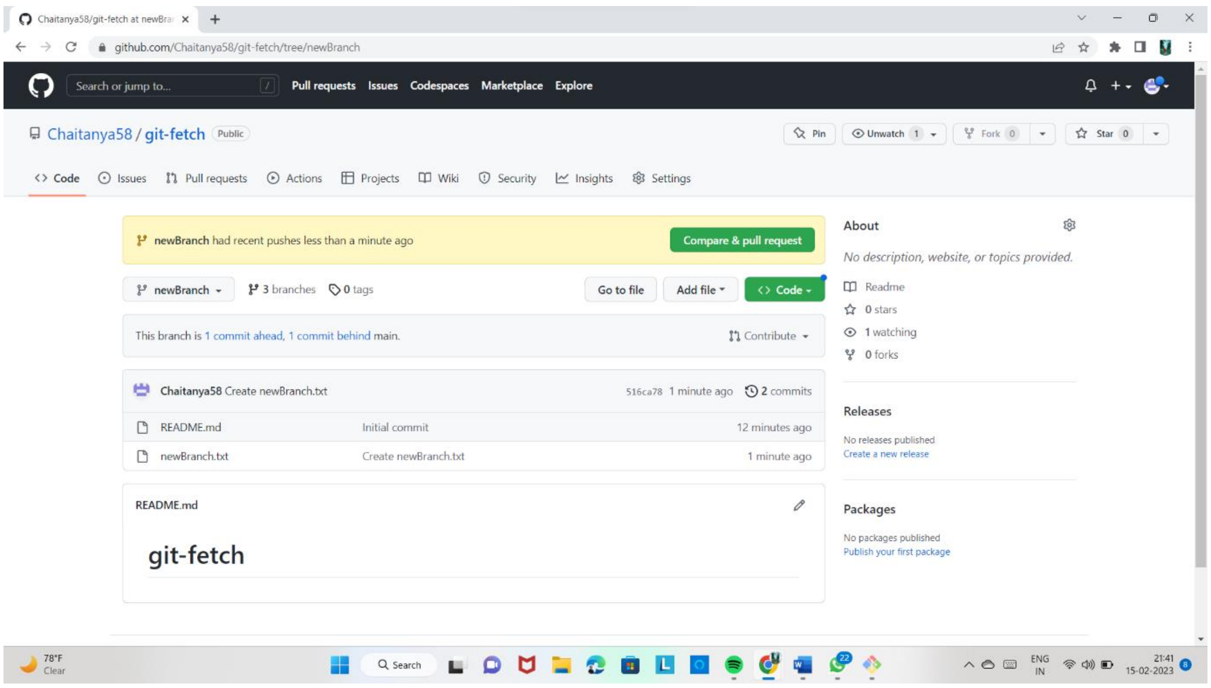
```
91837@Chaitanya MINGW64 ~/desktop/stash (feature)
$ git clone https://github.com/Chaitanya58/git-fetch.git
Cloning into 'git-fetch'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.

91837@Chaitanya MINGW64 ~/desktop/stash (feature)
$ cd git-fetch/

91837@Chaitanya MINGW64 ~/desktop/stash/git-fetch (main)
$ ls
README.md

91837@Chaitanya MINGW64 ~/desktop/stash/git-fetch (main)
$ git branch
* main

91837@Chaitanya MINGW64 ~/desktop/stash/git-fetch (main)
$ git branch -r
  origin/Branch2
  origin/HEAD -> origin/main
  origin/main
  origin/newBranch

91837@Chaitanya MINGW64 ~/desktop/stash/git-fetch (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

91837@Chaitanya MINGW64 ~/desktop/stash/git-fetch (main)
$ |
```

6.Fetches the remote repository



MINGW64:/c/Users/91837/desktop/stash/git-fetch

```
91837@Chaitanya MINGW64 ~/desktop/stash/git-fetch (main)
$ git fetch origin
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (5/5), 1.26 KiB | 35.00 KiB/s, done.
From https://github.com/Chaitanya58/git-fetch
   9ad0bc6..165203b  main       -> origin/main
   9ad0bc6..516ca78  newBranch  -> origin/newBranch
```



github.com/Chaitanya58/git-fetch/tree/newBranch

Chaitanya58 / git-fetch Public

newBranch had recent pushes less than a minute ago

newBranch · 3 branches · 0 tags

This branch is 1 commit ahead, 1 commit behind main.

Chaitanya58 Create newBranch.txt · 516ca78 1 minute ago · 2 commits

README.md	Initial commit	12 minutes ago
newBranch.txt	Create newBranch.txt	1 minute ago

README.md

git-fetch

About

No description, website, or topics provided.

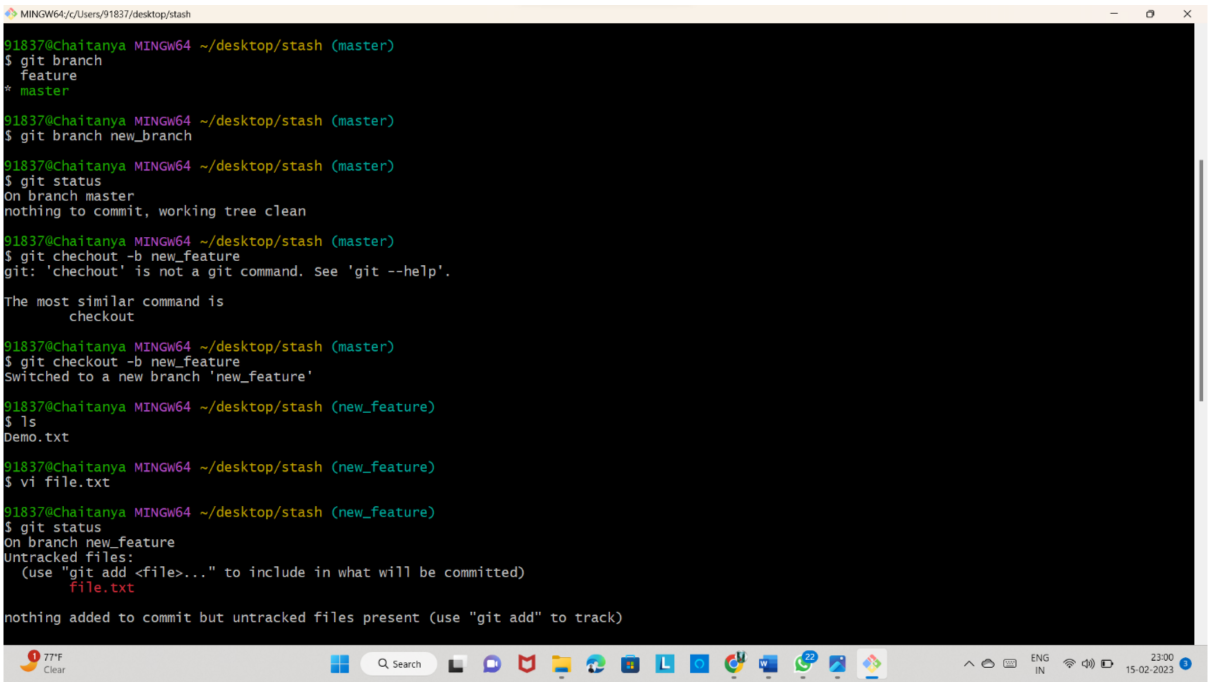
Readme · 0 stars · 1 watching · 0 forks

Releases: No releases published

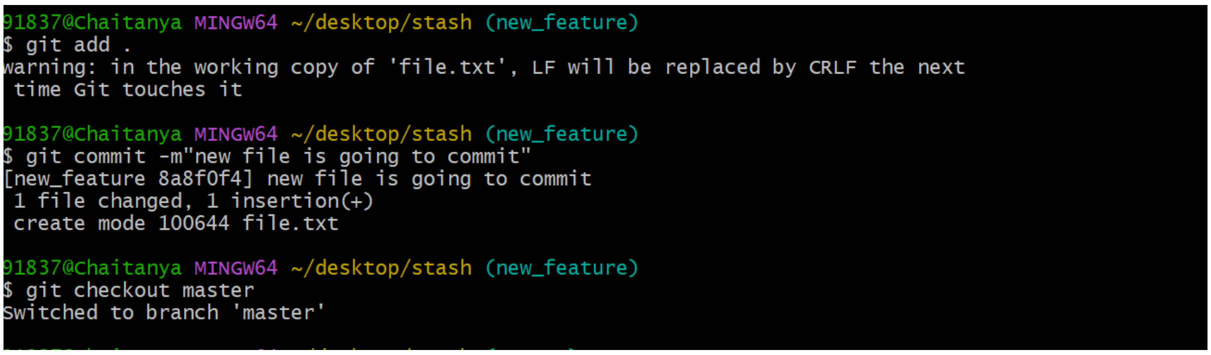
Packages: No packages published

Git Merge:-

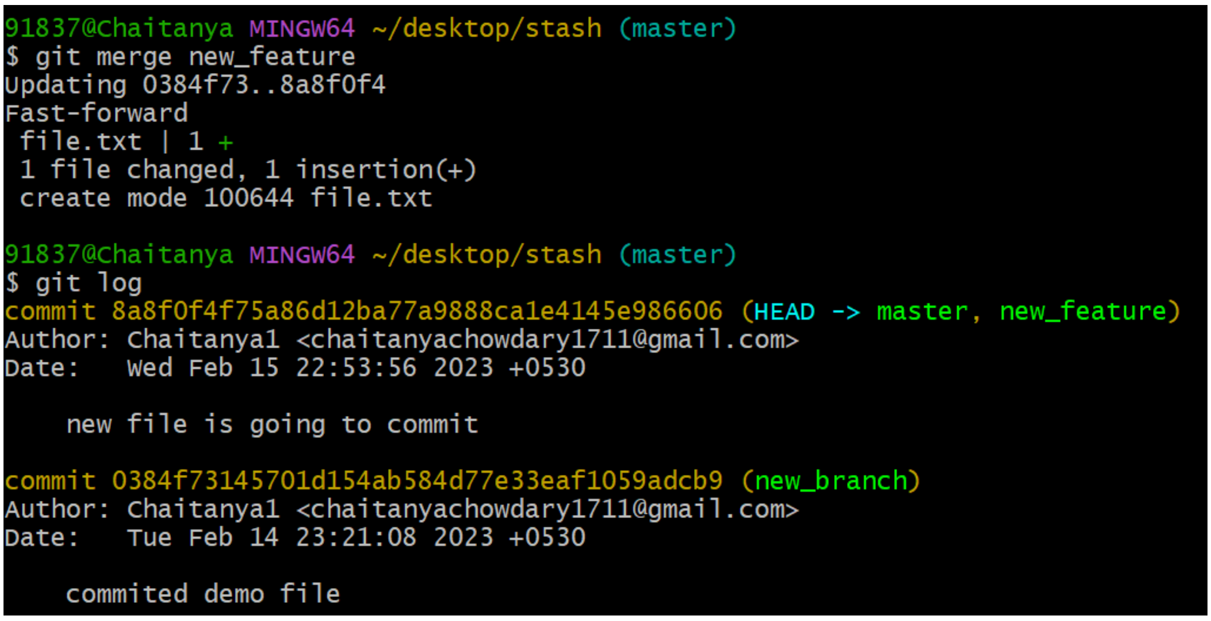
Git merging is basically to merge multiple sequences of commits, stored in multiple branches.
When you merge one branch into the another,Git takes the changes that were made on the source and applies them to the destination branch
Syntax:-**git merge <filename>**



```
91837@Chaitanya MINGW64 ~/desktop/stash (master)
$ git branch
  feature
* master

91837@Chaitanya MINGW64 ~/desktop/stash (master)
$ git branch new_branch

91837@Chaitanya MINGW64 ~/desktop/stash (master)
$ git status
On branch master
nothing to commit, working tree clean

91837@Chaitanya MINGW64 ~/desktop/stash (master)
$ git checkout -b new_feature
git: 'checkout' is not a git command. See 'git --help'.

The most similar command is
        checkout

91837@Chaitanya MINGW64 ~/desktop/stash (master)
$ git checkout -b new_feature
Switched to a new branch 'new_feature'

91837@Chaitanya MINGW64 ~/desktop/stash (new_feature)
$ ls
Demo.txt

91837@Chaitanya MINGW64 ~/desktop/stash (new_feature)
$ vi file.txt

91837@Chaitanya MINGW64 ~/desktop/stash (new_feature)
$ git status
On branch new_feature
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        file.txt

nothing added to commit but untracked files present (use "git add" to track)
```



```
91837@Chaitanya MINGW64 ~/desktop/stash (new_feature)
$ git add .
warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next
 time Git touches it

91837@Chaitanya MINGW64 ~/desktop/stash (new_feature)
$ git commit -m"new file is going to commit"
[new_feature 8a8f0f4] new file is going to commit
 1 file changed, 1 insertion(+)
 create mode 100644 file.txt

91837@Chaitanya MINGW64 ~/desktop/stash (new_feature)
$ git checkout master
Switched to branch 'master'
```



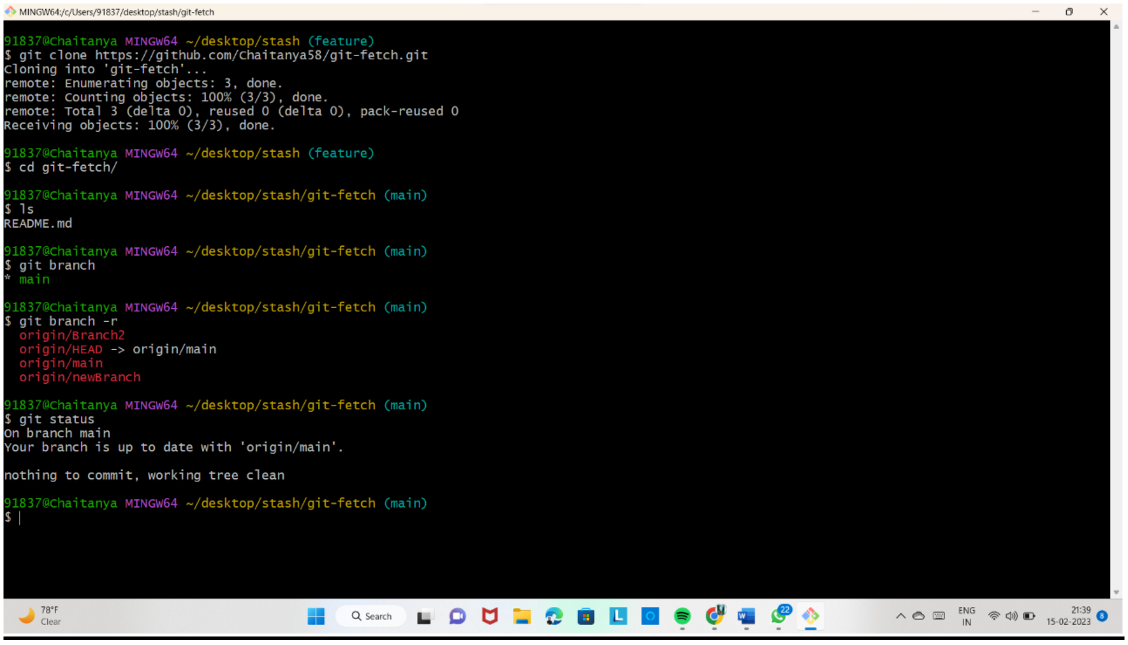
```
91837@Chaitanya MINGW64 ~/desktop/stash (master)
$ git merge new_feature
Updating 0384f73..8a8f0f4
Fast-forward
 file.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 file.txt

91837@Chaitanya MINGW64 ~/desktop/stash (master)
$ git log
commit 8a8f0f4f75a86d12ba77a9888ca1e4145e986606 (HEAD -> master, new_feature)
Author: Chaitanya1 <chaitanyachowdary1711@gmail.com>
Date:   Wed Feb 15 22:53:56 2023 +0530

    new file is going to commit

commit 0384f73145701d154ab584d77e33eaf1059adcb9 (new_branch)
Author: Chaitanya1 <chaitanyachowdary1711@gmail.com>
Date:   Tue Feb 14 23:21:08 2023 +0530

    commited demo file
```

3)State the difference between git fetch and git pull by doing a practical example in your git bash and attach a screenshot of all the processes.

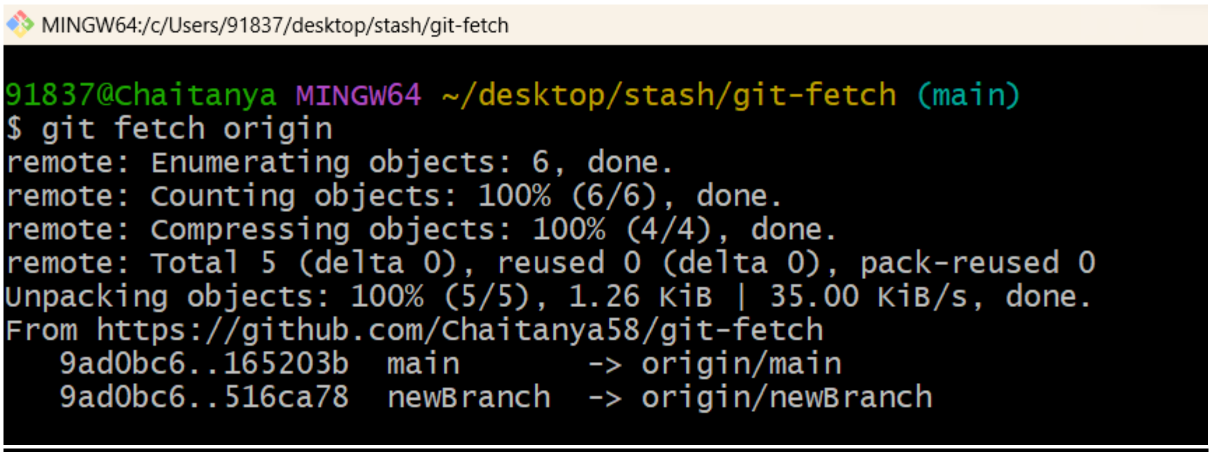
Git Fetch:-Git fetch downloads the changes from a remote repository to your local repository, but it does not apply those changes to your current working branch.

Instead, it updates your remote tracking branch to reflect any changes that have occurred on the remote repository.

Git fetch is useful when you want to check for changes in a remote repository without merging them into your current working branch.



```
MINGW64:/c/Users/91837/desktop/stash/git-fetch

91837@Chaitanya MINGW64 ~/desktop/stash (feature)
$ git clone https://github.com/Chaitanya58/git-fetch.git
Cloning into 'git-fetch'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.

91837@Chaitanya MINGW64 ~/desktop/stash (feature)
$ cd git-fetch/

91837@Chaitanya MINGW64 ~/desktop/stash/git-fetch (main)
$ ls
README.md

91837@Chaitanya MINGW64 ~/desktop/stash/git-fetch (main)
$ git branch
* main

91837@Chaitanya MINGW64 ~/desktop/stash/git-fetch (main)
$ git branch -r
  origin/Branch2
  origin/HEAD -> origin/main
  origin/main
  origin/newBranch

91837@Chaitanya MINGW64 ~/desktop/stash/git-fetch (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

91837@Chaitanya MINGW64 ~/desktop/stash/git-fetch (main)
$ |
```



```
MINGW64:/c/Users/91837/desktop/stash/git-fetch

91837@Chaitanya MINGW64 ~/desktop/stash/git-fetch (main)
$ git fetch origin
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (5/5), 1.26 KiB | 35.00 KiB/s, done.
From https://github.com/Chaitanya58/git-fetch
   9ad0bc6..165203b  main       -> origin/main
   9ad0bc6..516ca78  newBranch  -> origin/newBranch
```

Git Pull:-

git pull, on the other hand, does both a git fetch and a git merge in one step.

It downloads the changes from a remote repository to your local repository and immediately applies those changes to your current working branch. git pull is useful when you want to update your local branch to the latest changes in a remote branch and immediately see those changes in your working copy.

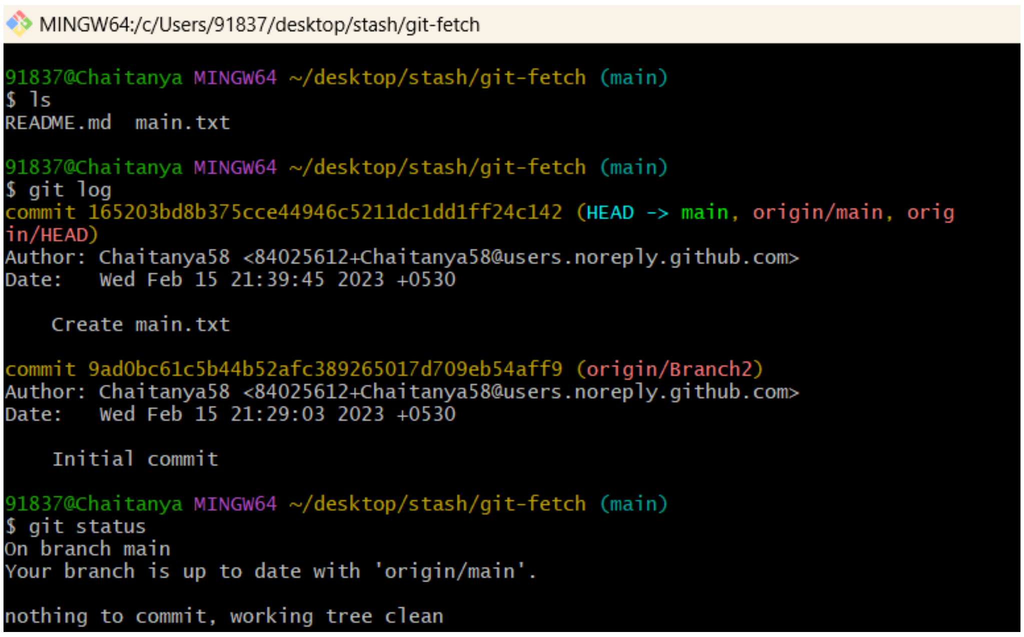
Steps for git pull:-

Step1:-

At first, check the status of the git repository and check what files are present in.

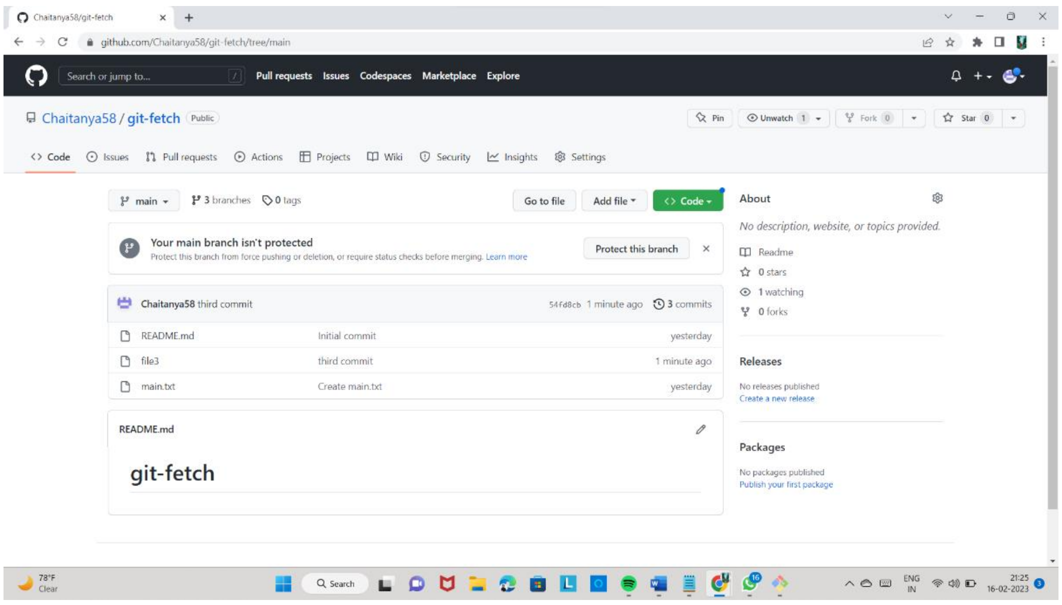
Step2:-

Add a file in git repository and commit the file



MINGW64:/c/Users/91837/desktop/stash/git-fetch

```
91837@Chaitanya MINGW64 ~/desktop/stash/git-fetch (main)
$ ls
README.md  main.txt

91837@Chaitanya MINGW64 ~/desktop/stash/git-fetch (main)
$ git log
commit 165203bd8b375cce44946c5211dc1dd1ff24c142 (HEAD -> main, origin/main, orig
in/HEAD)
Author: Chaitanya58 <84025612+Chaitanya58@users.noreply.github.com>
Date:   Wed Feb 15 21:39:45 2023 +0530

    Create main.txt

commit 9ad0bc61c5b44b52afc389265017d709eb54aff9 (origin/Branch2)
Author: Chaitanya58 <84025612+Chaitanya58@users.noreply.github.com>
Date:   Wed Feb 15 21:29:03 2023 +0530

    Initial commit

91837@Chaitanya MINGW64 ~/desktop/stash/git-fetch (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```



github.com/Chaitanya58/git-fetch/tree/main

Chaitanya58 / git-fetch Public

main · 3 branches · 0 tags

Your main branch isn't protected
Protect this branch from force pushing or deletion, or require status checks before merging. Learn more

Chaitanya58 third commit · 54fd8cb 1 minute ago · 3 commits

README.md	Initial commit	yesterday
file3	third commit	1 minute ago
main.txt	Create main.txt	yesterday

README.md

git-fetch

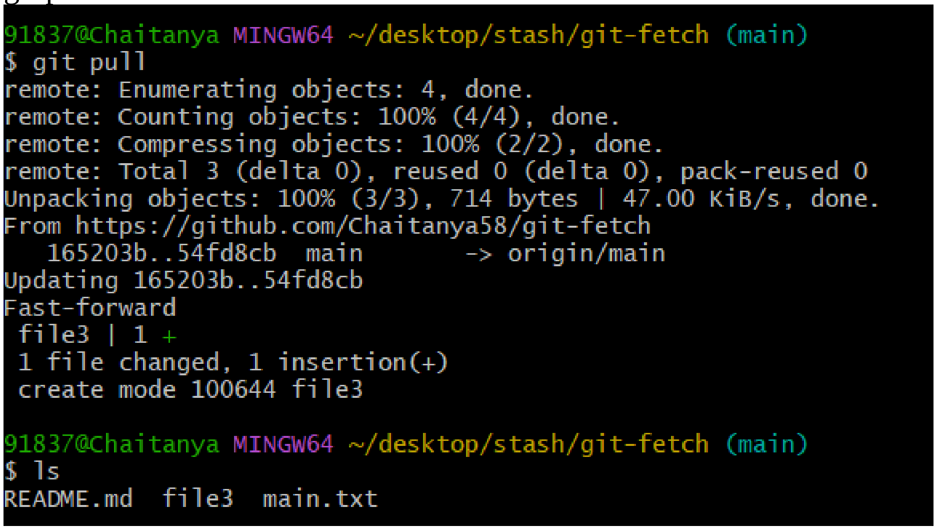
About
No description, website, or topics provided.
Readme
0 stars
1 watching
0 forks

Releases
No releases published
Create a new release

Packages
No packages published
Publish your first package

From the steps the changes we done in a git repository in not visible in local repository

Step4:-
The changes we are done in git repository are visible in local repository when you excute the git pull command



```
91837@Chaitanya MINGW64 ~/desktop/stash/git-fetch (main)
$ git pull
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 714 bytes | 47.00 KiB/s, done.
From https://github.com/Chaitanya58/git-fetch
   165203b..54fd8cb  main       -> origin/main
Updating 165203b..54fd8cb
Fast-forward
 file3 | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 file3

91837@Chaitanya MINGW64 ~/desktop/stash/git-fetch (main)
$ ls
README.md  file3  main.txt
```

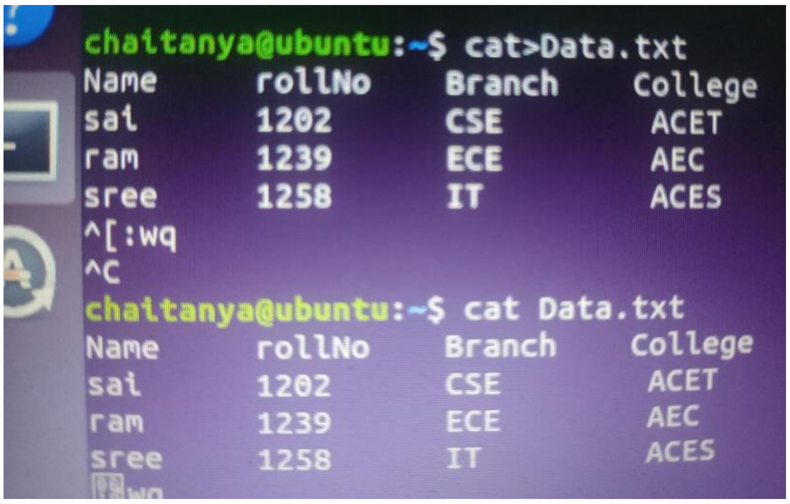
4)Try to find out about the awk command and use it while reading a file created by yourself. Also, make a bash script file and try to find out the prime number from the range 1 to 20.

The whole process should be carried out and by using the history command, give the screenshot of all the processes being carried out.

AWK:-

awk is a powerful command-line tool used for processing and manipulating text files , especially when dealing with large amounts of data.

In the below image, created a file and named as Data.txt and print the same data

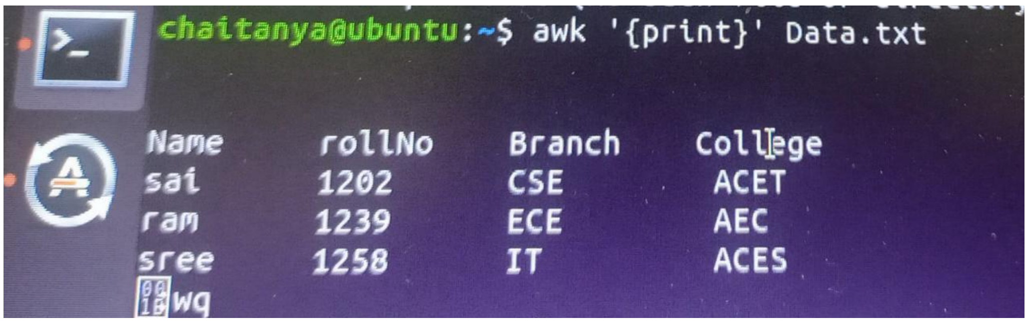


```
chaitanya@ubuntu:~$ cat>Data.txt
Name      rollNo     Branch     College
sai       1202       CSE         ACET
ram       1239       ECE         AEC
sree      1258       IT          ACES
^[:wq
^C
chaitanya@ubuntu:~$ cat Data.txt
Name      rollNo     Branch     College
sai       1202       CSE         ACET
ram       1239       ECE         AEC
sree      1258       IT          ACES
```

Commands on awk:-

Syntax:- **awk '{print}' filename**

This commad prints the data present in the file

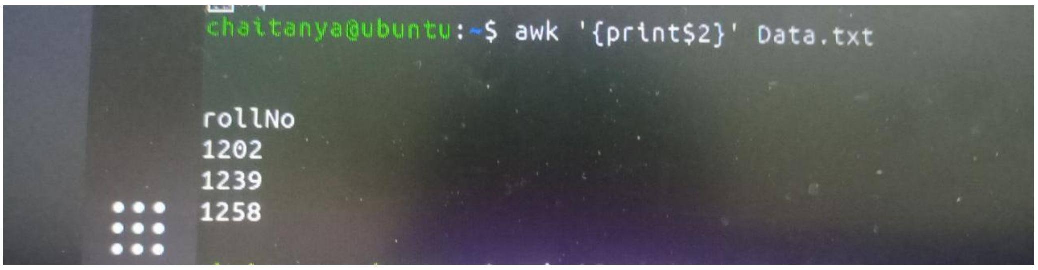


```
chaitanya@ubuntu:~$ awk '{print}' Data.txt

Name      rollNo    Branch    College
sai       1202      CSE        ACET
ram       1239      ECE        AEC
sree      1258      IT         ACES
wq
```

Command:- **awk '{print$column_number}' filename**

Eg:-awk '{print$2}' Data.txt

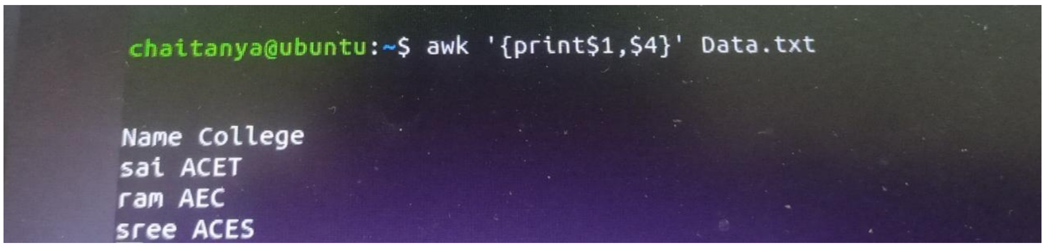
This command prints the second column data in a data.txt file



```
chaitanya@ubuntu:~$ awk '{print$2}' Data.txt

rollNo
1202
1239
1258
```

Command:- **awk '{print$1,$4}' Data.txt**

This command prints the first and fourth column data in a Data.txt file



```
chaitanya@ubuntu:~$ awk '{print$1,$4}' Data.txt

Name College
sai ACET
ram AEC
sree ACES
```

BASH SCRIPTING:-

Steps to follow in bash Scripting

Step1:-

Create a file with extension .sh

Step2:-

Give the permissions of read, write and excute

Step3:-

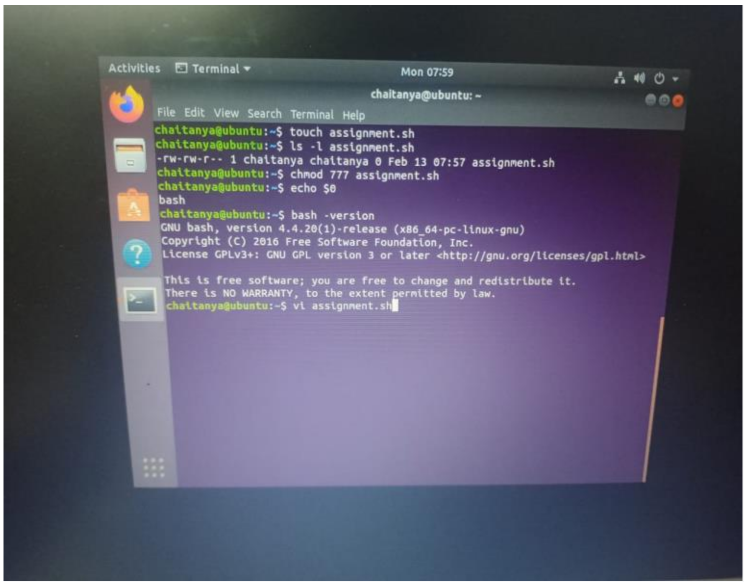
open the shell and write the script

Step4:-

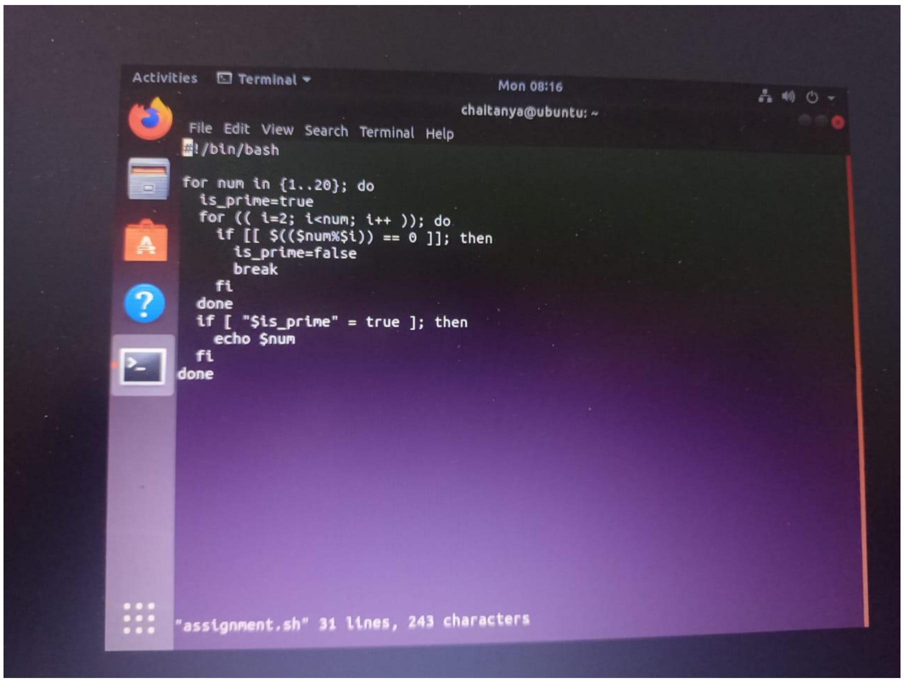
Save the code and run the code

The command to run a code is

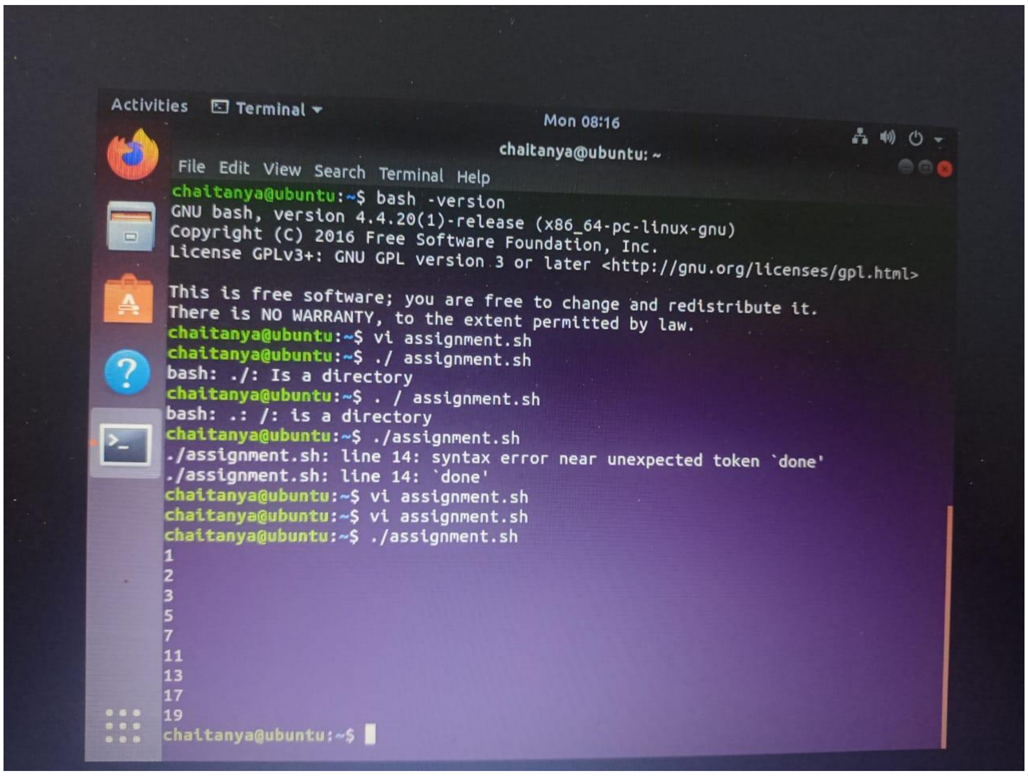
Syntax:- ./filename



```
chaitanya@ubuntu:~$ touch assignment.sh
chaitanya@ubuntu:~$ ls -l assignment.sh
-rw-rw-r-- 1 chaitanya chaitanya 0 Feb 13 07:57 assignment.sh
chaitanya@ubuntu:~$ chmod 777 assignment.sh
chaitanya@ubuntu:~$ echo $0
bash
chaitanya@ubuntu:~$ bash -version
GNU bash, version 4.4.20(1)-release (x86_64-pc-linux-gnu)
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>

This is free software; you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
chaitanya@ubuntu:~$ vi assignment.sh
```



```
#!/bin/bash

for num in {1..20}; do
  is_prime=true
  for (( i=2; i<num; i++ )); do
    if [[ $(($num%$i)) == 0 ]]; then
      is_prime=false
      break
    fi
  done
  if [ "$is_prime" = true ]; then
    echo $num
  fi
done

"assignment.sh" 31 lines, 243 characters
```



```
chaitanya@ubuntu:~$ bash -version
GNU bash, version 4.4.20(1)-release (x86_64-pc-linux-gnu)
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>

This is free software; you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
chaitanya@ubuntu:~$ vi assignment.sh
chaitanya@ubuntu:~$ ./ assignment.sh
bash: ./: Is a directory
chaitanya@ubuntu:~$ . / assignment.sh
bash: .: /: is a directory
chaitanya@ubuntu:~$ ./assignment.sh
./assignment.sh: line 14: syntax error near unexpected token `done'
./assignment.sh: line 14: `done'
chaitanya@ubuntu:~$ vi assignment.sh
chaitanya@ubuntu:~$ vi assignment.sh
chaitanya@ubuntu:~$ ./assignment.sh
1
2
3
5
7
11
13
17
19
chaitanya@ubuntu:~$ 
```

5)Set up a container and run a Ubuntu operating system. For this purpose, you can make use of the docker hub and run the container in interactive mode.All the processes pertaining to this should be provided in a screenshot for grading.

Steps to set up a container and run a ubuntu operating system.

Step1:- Install docker image from a google and set the docker image according to your machine

Step2:-To check the weather the docker installed correctly in your machine excute the below command in the command prompt.

Command:- **docker version**

If you get the description and version about the docker then it installed correctly. Otherwise again install the docker in your machine

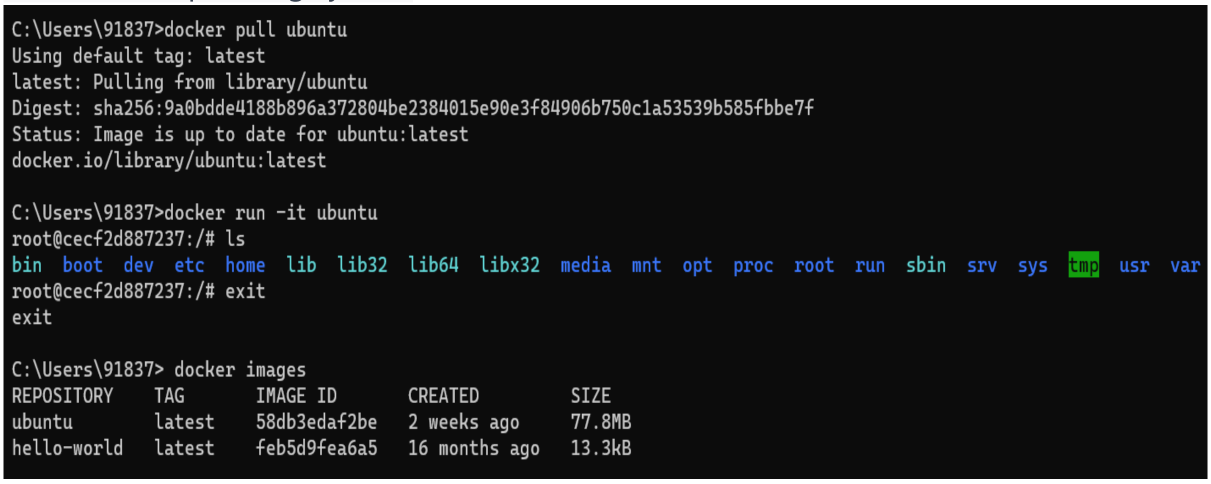
Step3:-Pull the ubuntu image from the docker image by running the below commad:

Command:- **docker pull ubuntu**

Step4:-After the image was downloaded,run the container using the following command

Command:-**docker run -it ubuntu**

"-it" option runs the container in the interactive mode and opens up a shell within the ubuntu operating system



```
C:\Users\91837>docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
Digest: sha256:9a0bdde4188b896a372804be2384015e90e3f84906b750c1a53539b585fbbe7f
Status: Image is up to date for ubuntu:latest
docker.io/library/ubuntu:latest

C:\Users\91837>docker run -it ubuntu
root@cecf2d887237:/# ls
bin  boot  dev  etc  home  lib  lib32  lib64  libx32  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@cecf2d887237:/# exit
exit

C:\Users\91837> docker images
REPOSITORY    TAG       IMAGE ID       CREATED         SIZE
ubuntu        latest    58db3edaf2be   2 weeks ago     77.8MB
hello-world   latest    feb5d9fea6a5   16 months ago   13.3kB
```

To see the downloaded docker images list we use the following command.

Command:- docker image