

ASSIGNMENT-1

Q1.GIT STASH:

- The git stash command takes your uncommitted changes (both staged and unstaged), saves them away for later use, and then reverts them from your working copy.
- Stash means to store (changes) safely in a hidden place (the stash stack).
- Stashing the current working directory's staged or unstaged changes or untracked files and then storing them in the stash stack reverts the current working directory to the last commit.

```
MINGW64/c/Users/chait/OneDrive/Desktop/new
chait@LAPTOP-CM12ONE9 MINGW64 ~ (master)
$ cd OneDrive

chait@LAPTOP-CM12ONE9 MINGW64 ~/OneDrive (master)
$ cd Desktop

chait@LAPTOP-CM12ONE9 MINGW64 ~/OneDrive/Desktop (master)
$ cd new

chait@LAPTOP-CM12ONE9 MINGW64 ~/OneDrive/Desktop/new (21A95A0410)
$ git init
Reinitialized existing Git repository in C:/Users/chait/OneDrive/Desktop/new/.git/

chait@LAPTOP-CM12ONE9 MINGW64 ~/OneDrive/Desktop/new (21A95A0410)
$ code .

chait@LAPTOP-CM12ONE9 MINGW64 ~/OneDrive/Desktop/new (21A95A0410)
$ git status
On branch 21A95A0410
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:   a1.txt

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

chait@LAPTOP-CM12ONE9 MINGW64 ~/OneDrive/Desktop/new (21A95A0410)
$ git stash
Saved working directory and index state WIP on 21A95A0410: 511a346 command

chait@LAPTOP-CM12ONE9 MINGW64 ~/OneDrive/Desktop/new (21A95A0410)
$ git checkout master
Switched to branch 'master'

chait@LAPTOP-CM12ONE9 MINGW64 ~/OneDrive/Desktop/new (master)
$ git status
On branch master
nothing to commit, working tree clean

chait@LAPTOP-CM12ONE9 MINGW64 ~/OneDrive/Desktop/new (master)
$ git checkout 21A95A0410
Switched to branch '21A95A0410'
```

```
MINGW64/c/Users/cha1t/OneDrive/Desktop/new
chait@LAPTOP-CM12ONE9 MINGW64 ~/OneDrive/Desktop/new (master)
$ git checkout 21A95A0410
Switched to branch '21A95A0410'

chait@LAPTOP-CM12ONE9 MINGW64 ~/OneDrive/Desktop/new (21A95A0410)
$ git status
On branch 21A95A0410
nothing to commit, working tree clean

chait@LAPTOP-CM12ONE9 MINGW64 ~/OneDrive/Desktop/new (21A95A0410)
$ git stash pop
On branch 21A95A0410
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:   a1.txt

no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0}: (b9eff9700de9b139b8c4e28d5fca1cd408e4cdf1)

chait@LAPTOP-CM12ONE9 MINGW64 ~/OneDrive/Desktop/new (21A95A0410)
$ git stash apply
error: Your local changes to the following files would be overwritten by merge:
    a1.txt
Please commit your changes or stash them before you merge.
Aborting
On branch 21A95A0410
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:   a1.txt

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

chait@LAPTOP-CM12ONE9 MINGW64 ~/OneDrive/Desktop/new (21A95A0410)
$ git checkout master
Switched to branch 'master'
M    a1.txt

chait@LAPTOP-CM12ONE9 MINGW64 ~/OneDrive/Desktop/new (master)
$ git status
On branch master
Changes not staged for commit:
```

- You can reapply previously stashed changes with git stash pop

```
MINGW64/c/Users/cha1t/OneDrive/Desktop/new
$ git status
On branch master
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:   a1.txt

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

chait@LAPTOP-CM12ONE9 MINGW64 ~/OneDrive/Desktop/new (master)
$ git checkout 21A95A0410
Switched to branch '21A95A0410'
M    a1.txt

chait@LAPTOP-CM12ONE9 MINGW64 ~/OneDrive/Desktop/new (21A95A0410)
$ git stash apply
error: Your local changes to the following files would be overwritten by merge:
    a1.txt
Please commit your changes or stash them before you merge.
Aborting
On branch 21A95A0410
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:   a1.txt

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

chait@LAPTOP-CM12ONE9 MINGW64 ~/OneDrive/Desktop/new (21A95A0410)
$ git stash list
stash@{0}: WIP on 21A95A0410: 511a346 command
stash@{1}: WIP on master: 511a346 command
stash@{2}: WIP on 21A95A0410: 511a346 command
stash@{3}: WIP on master: 511a346 command
stash@{4}: WIP on master: 5f4608b git

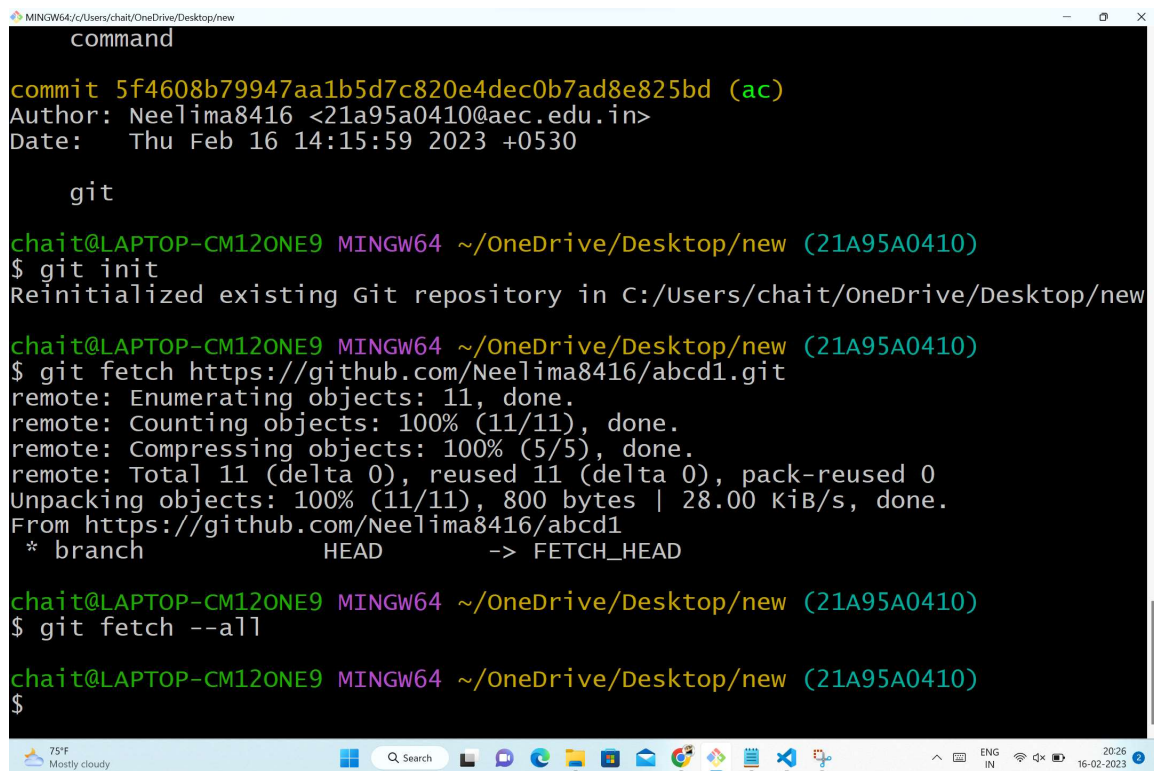
chait@LAPTOP-CM12ONE9 MINGW64 ~/OneDrive/Desktop/new (21A95A0410)
$ git stash clear

chait@LAPTOP-CM12ONE9 MINGW64 ~/OneDrive/Desktop/new (21A95A0410)
$ git stash list

chait@LAPTOP-CM12ONE9 MINGW64 ~/OneDrive/Desktop/new (21A95A0410)
$
```

Q2.GIT FETCH :

- The git fetch command downloads commits, files, and refs from a remote repository into your local repository.
- Fetching is what you do when you want to see what everybody else has been working on. It's similar to svn update in that it lets you see how the central history has progressed, but it doesn't force you to actually merge the changes into your repository.



```
command

commit 5f4608b79947aa1b5d7c820e4dec0b7ad8e825bd (ac)
Author: Neelima8416 <21a95a0410@aec.edu.in>
Date: Thu Feb 16 14:15:59 2023 +0530

git

chait@LAPTOP-CM12ONE9 MINGW64 ~/OneDrive/Desktop/new (21A95A0410)
$ git init
Reinitialized existing Git repository in C:/Users/chait/OneDrive/Desktop/new

chait@LAPTOP-CM12ONE9 MINGW64 ~/OneDrive/Desktop/new (21A95A0410)
$ git fetch https://github.com/Neelima8416/abcd1.git
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 11 (delta 0), reused 11 (delta 0), pack-reused 0
Unpacking objects: 100% (11/11), 800 bytes | 28.00 KiB/s, done.
From https://github.com/Neelima8416/abcd1
* branch                HEAD      -> FETCH_HEAD

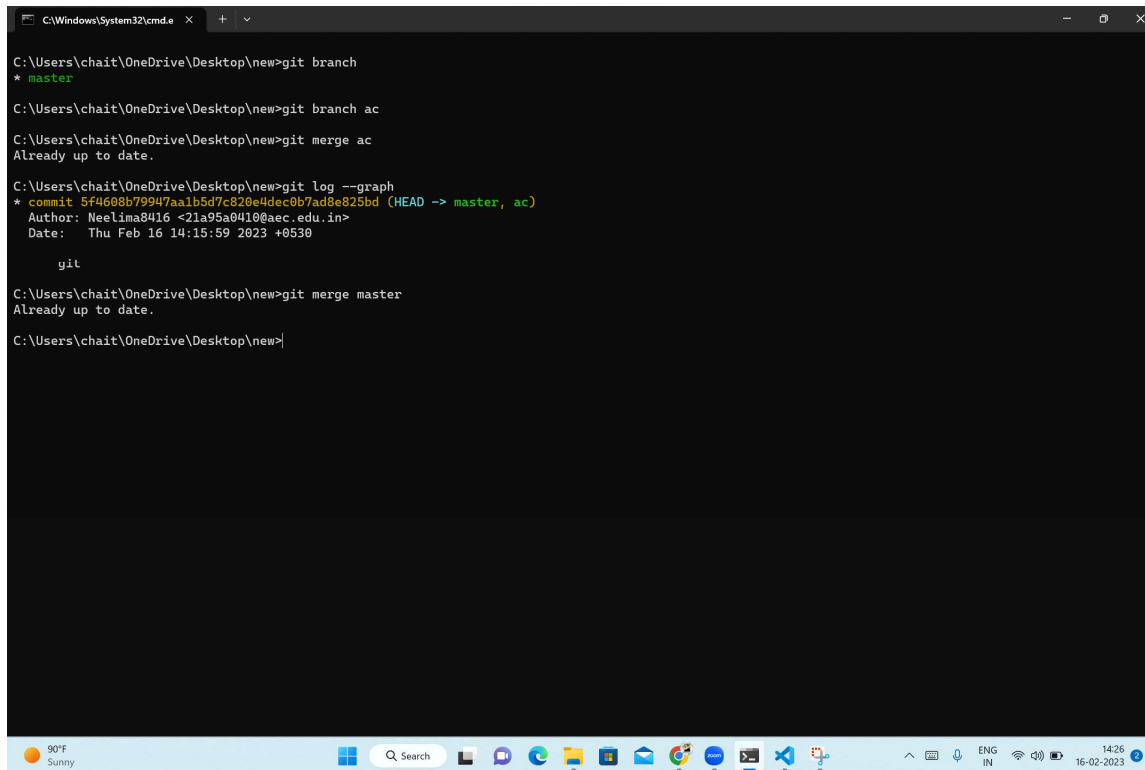
chait@LAPTOP-CM12ONE9 MINGW64 ~/OneDrive/Desktop/new (21A95A0410)
$ git fetch --all

chait@LAPTOP-CM12ONE9 MINGW64 ~/OneDrive/Desktop/new (21A95A0410)
$
```

- Git isolates fetched content from existing local content; it has absolutely no effect on your local development work.
- Fetched content has to be explicitly checked out using the git checkout command.
- This makes fetching a safe way to review commits before integrating them with your local repository.

GIT MERGE:

- The git merge command is used to merge the branches.



```
C:\Windows\System32\cmd.exe x + v
C:\Users\chait\OneDrive\Desktop\new>git branch
* master
C:\Users\chait\OneDrive\Desktop\new>git branch ac
C:\Users\chait\OneDrive\Desktop\new>git merge ac
Already up to date.
C:\Users\chait\OneDrive\Desktop\new>git log --graph
* commit 5f4688b79947aa1b5d7c820e4dec0b7ad8e825bd (HEAD -> master, ac)
  Author: Neelima8416 <21a95a0410@aec.edu.in>
  Date: Thu Feb 16 14:15:59 2023 +0530

    git
C:\Users\chait\OneDrive\Desktop\new>git merge master
Already up to date.
C:\Users\chait\OneDrive\Desktop\new>
```

- The git merge command lets you take the independent lines of development created by git branch and integrate them into a single branch.

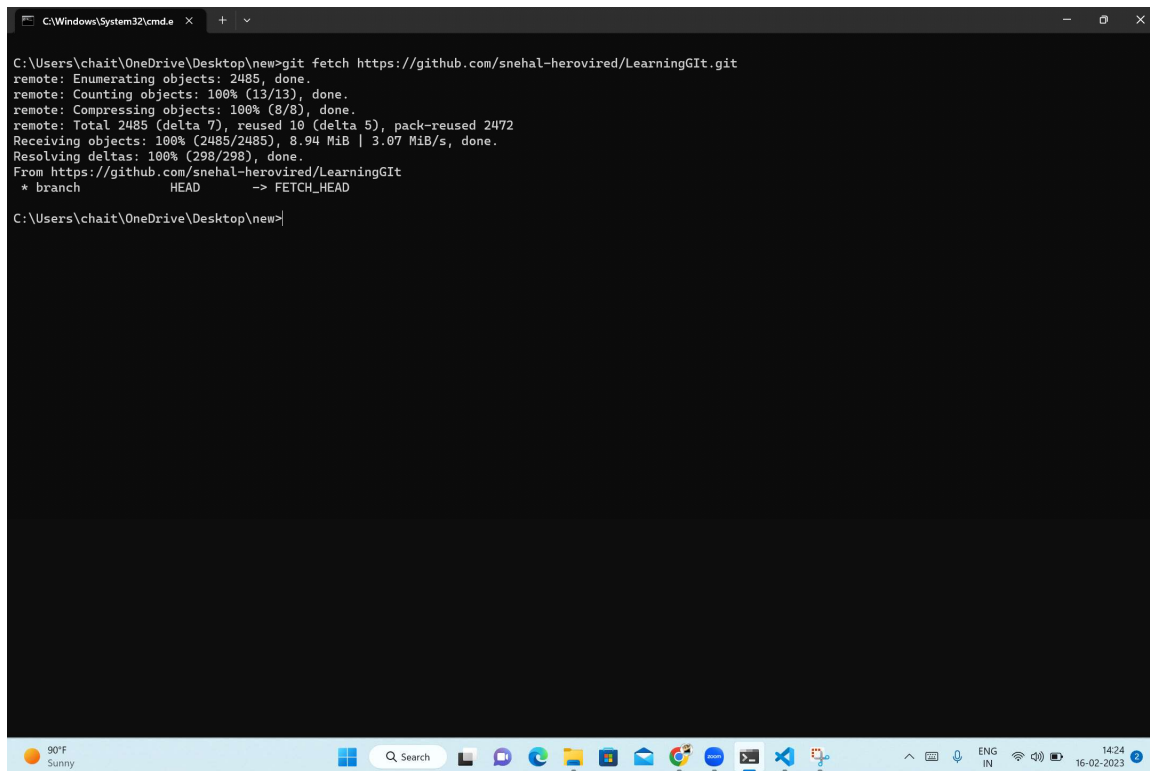
Q3. DIFFERENCE BETWEEN GIT FETCH AND GIT PULL:

- Git Fetch is the command that tells the local repository that there are changes available in the remote repository without bringing the changes into the local repository.
- Git Pull on the other hand brings the copy of the remote directory changes into the local repository.
- When comparing Git pull vs fetch, Git fetch is a safer alternative because it pulls in all the commits from your remote but doesn't make

any changes to your local files.

- Git pull is a more advanced action and it's important to understand that you will be introducing changes and immediately applying them to your currently checked out branch.

Git Fetch Command



```
C:\Windows\System32\cmd.exe
C:\Users\chait\OneDrive\Desktop\new>git fetch https://github.com/snehal-herovired/LearningGit.git
remote: Enumerating objects: 2485, done.
remote: Counting objects: 100% (13/13), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 2485 (delta 7), reused 10 (delta 5), pack-reused 2472
Receiving objects: 100% (2485/2485), 8.94 MiB | 3.07 MiB/s, done.
Resolving deltas: 100% (298/298), done.
From https://github.com/snehal-herovired/LearningGit
* branch            HEAD       -> FETCH_HEAD
C:\Users\chait\OneDrive\Desktop\new>
```

Git Pull Command

```
C:\Windows\System32\cmd.exe
C:\Users\chait\OneDrive\Desktop\folder\LearningGit>git pull origin 21A95A0410
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 5 (delta 0), reused 4 (delta 0), pack-reused 0
Unpacking objects: 100% (5/5), 503.08 KiB | 459.00 KiB/s, done.
From https://github.com/snehal-herovired/LearningGit
* branch                21A95A0410 -> FETCH_HEAD
   60f6313..5049df9      21A95A0410 -> origin/21A95A0410
Merge made by the 'ort' strategy.
 Assignment2/docker1.png | Bin 0 -> 367634 bytes
 Assignment2/docker2.png | Bin 0 -> 193793 bytes
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 Assignment2/docker1.png
 create mode 100644 Assignment2/docker2.png

C:\Users\chait\OneDrive\Desktop\folder\LearningGit>
```

- Git pull and fetch are two commands that are regularly used by Git users.

Q4.AWK COMMAND IN LINUX:

- The awk command is a Linux tool and programming language that allows users to process and manipulate data and produce formatted reports.
- The tool supports various operations for advanced text processing and facilitates expressing complex data selections.
- It searches one or more files to see if they contain lines that matches with the specified patterns and then perform the associated actions.

```
MINGW64/c/Users/chaith/documents
chaith@LAPTOP-CH120NE9 MINGW64 ~ (master)
$ ls
'3D Objects'/'Downloads/' 'My Documents'@
AppData/'Application Data'@ IntelGraphicsProfiles/ NTUSER.DAT
Contacts/'Links/' NTUSER.DAT{a2332f18-cdbf-11ec-8680-002248483d79}.TM.b1f
Cookies@ 'Local Settings'@ NTUSER.DAT{a2332f18-cdbf-11ec-8680-002248483d79}.TM.Container00000000000000000001.regtrans-ms
Documents/ Music/ NetHood@ NTUSER.DAT{a2332f18-cdbf-11ec-8680-002248483d79}.TM.Container00000000000000000002.regtrans-ms
OneDrive/ Pictures/ PrintHood@ Recent@ 'Saved Games'/'Videos/' 'VirtualBox VMs'/'
SendTo@ 'Start Menu'@ Templates@ VIA/

chaith@LAPTOP-CH120NE9 MINGW64 ~ (master)
$ cd documents
chaith@LAPTOP-CH120NE9 MINGW64 ~/documents (master)
$ touch ac
chaith@LAPTOP-CH120NE9 MINGW64 ~/documents (master)
$ nano ac
chaith@LAPTOP-CH120NE9 MINGW64 ~/documents (master)
$ cat ac
hihi my name is chaityanya
hello this is priya
chaith@LAPTOP-CH120NE9 MINGW64 ~/documents (master)
$ awk '{print $1 $2}' ac
hihi my
hellothis
chaith@LAPTOP-CH120NE9 MINGW64 ~/documents (master)
$ awk '{print $1$3}' ac
hihi
hello
chaith@LAPTOP-CH120NE9 MINGW64 ~/documents (master)
$ awk '{print $2 $3}' ac
myname
thisis
chaith@LAPTOP-CH120NE9 MINGW64 ~/documents (master)
$ awk '{print $1 $4}' ac
hihiis
hellopriya
chaith@LAPTOP-CH120NE9 MINGW64 ~/documents (master)
$ awk '{print $1 $2 $4}' ac
hihi myis
hellothispriya
chaith@LAPTOP-CH120NE9 MINGW64 ~/documents (master)
$ awk '{print}' ac
hihi my name is chaityanya
hello this is priya
chaith@LAPTOP-CH120NE9 MINGW64 ~/documents (master)
$
```

BASH PROGRAM TO PRINT PRIME NUMBERS FROM 1 TO 20:

```
GNU nano 7.1 ac1.sh
#!/usr/bin/bash
count=0
for ((i=1;i<=20;i++))
do
    for ((j=1;j<=20;j++))
    do
        if(( $i % $j == 0 ))
        then
            ((count++))
        fi
    done
    if (( $count == 2 ))
    then
        echo $i
        count=0
    else
        count=0
    fi
done

[ Read 20 lines ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line

70°F Clear
```

OUTPUT:


```
MINGW64~/c/Users/chaith/documents
chaith@LAPTOP-CM12ONE9 MINGW64 ~/documents (master)
$ touch ac1.sh

chaith@LAPTOP-CM12ONE9 MINGW64 ~/documents (master)
$ nano ac1.sh

chaith@LAPTOP-CM12ONE9 MINGW64 ~/documents (master)
$ pwd
/c/Users/chaith/documents

chaith@LAPTOP-CM12ONE9 MINGW64 ~/documents (master)
$ nano ac1.sh

chaith@LAPTOP-CM12ONE9 MINGW64 ~/documents (master)
$ bash ac1.sh
2
3
5
7
11
13
17
19

chaith@LAPTOP-CM12ONE9 MINGW64 ~/documents (master)
$
```

HISTORY COMMAND:

- The History command is used to display all the commands used previously by the user of linux.


```
MINGW64/c/Users/chaith/documents
chaith@LAPTOP-CH12ONE9 MINGW64 ~/documents (master)
$ history
1 git config --global --list
2 yum install git
3 sudo yum install git
4 pwd
5 mkdir ac
6 cd ac
7 touch ac1
8 ls
9 touch a.txt
10 vi a.txt
11 git init
12 git add a.txt
13 git status
14 git commit -m 'first line'
15 git config --global user.Neelima8416
16 git remote add origin https://github.com/Neelima8416/demo.git
17 git push -u origin master
18 git config --global user.name 'Neelima8416'
19 git config --global user.email "21a95a0410@aec.edu.in"
20 git config --global --list
21 git push -u origin master
22 ls
23 git status
24 git commit -m "a"
25 git push -u origin master
26 docker login
27 docker info
28 docker images
29 docker pull ubuntu
30 docker images
31 docker images --help
32 docker rmi ubuntu
33 docker images
34 docker stats
35 docker system df
36 docker system --help
37 docker info
38 docker login
39 docker images
40 docker pull ubuntu
41 docker images
42 docker pull ubuntu~
43 docker pull ubuntu
44 docker images
45 docker login
46 docker info
47 docker build ac8
48 docker build -t ac84 .
49 git init
50 git stash
51 ls
52 cd documents
```

Q5.PROCESS TO SETUP A CONTAINER AND RUN UBUNTU OPERATING SYSTEM:

- Open the folder which is having docker file and open gitbash through that folder.
- Now login with your dockerhub credentials.
- Now type docker images and it will lists you the images in your account.
- Now go to dockerhub.com and select ubuntu and copy the pull command which is present over there and paste it in the gitbash.
- Thus, the ubuntu is pulled and inorder to run the container, enter the command "docker run -it image_name|image_id
- Then, we can find that the a container is setup with ubuntu os is

running in docker desktop.

```
root@407b2886732a: /

chait@LAPTOP-CM12ONE9 MINGW64 ~ (master)
$ docker login
Authenticating with existing credentials...
Login Succeeded

Logging in with your password grants your terminal complete access to your account.
For better security, log in with a limited-privilege personal access token. Learn more at https://docs.docker.com/go/access-tokens/

chait@LAPTOP-CM12ONE9 MINGW64 ~ (master)
$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE

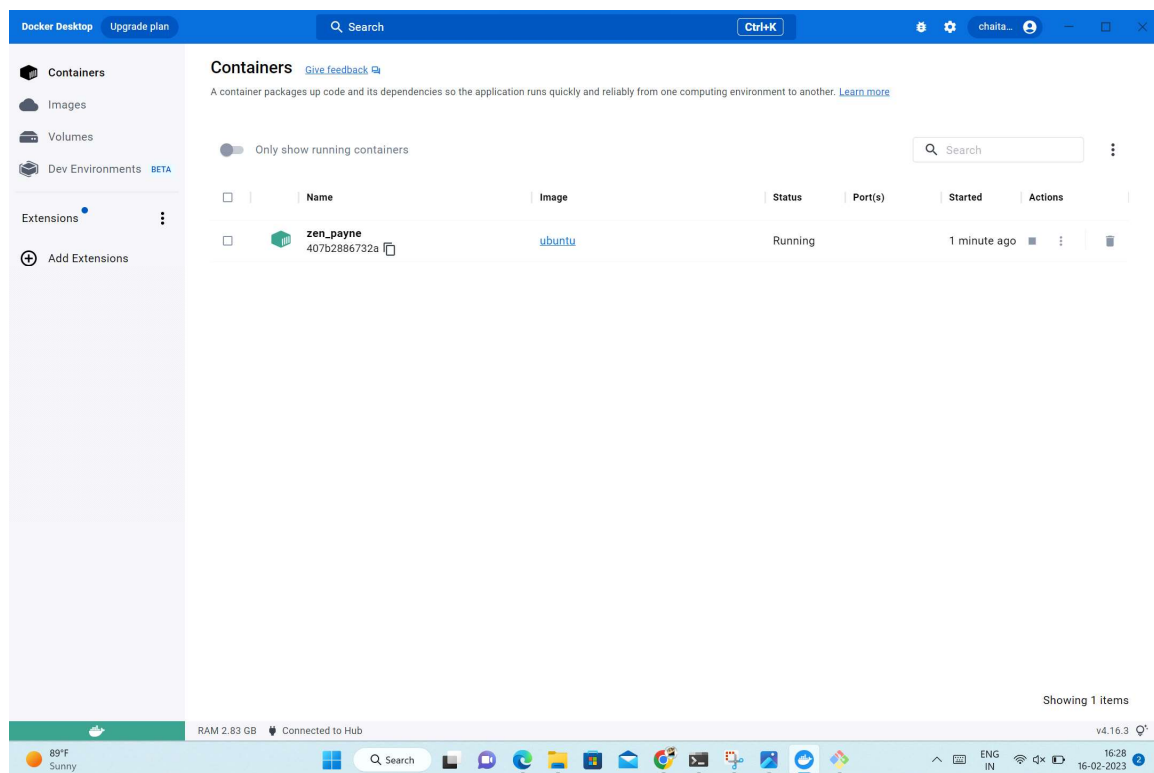
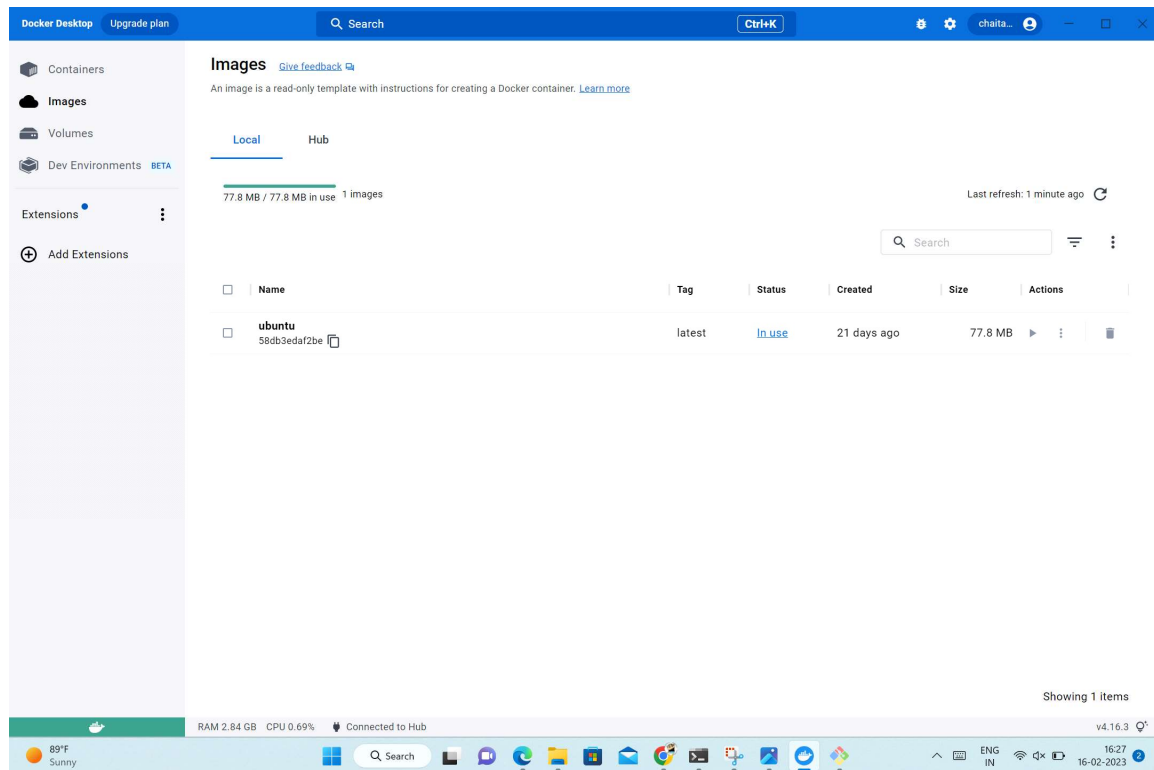
chait@LAPTOP-CM12ONE9 MINGW64 ~ (master)
$ docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
677076032cca: Pulling fs layer
677076032cca: Verifying Checksum
677076032cca: Download complete
677076032cca: Pull complete
Digest: sha256:9a0bdde4188b896a372804be2384015e90e3f84906b750c1a53539b585fbbe7f
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest

chait@LAPTOP-CM12ONE9 MINGW64 ~ (master)
$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
ubuntu        latest    58db3edaf2be   3 weeks ago    77.8MB

chait@LAPTOP-CM12ONE9 MINGW64 ~ (master)
$ docker run -it ubuntu
the input device is not a TTY. If you are using mintty, try prefixing the command with 'winpty'

chait@LAPTOP-CM12ONE9 MINGW64 ~ (master)
$ winpty docker run -it ubuntu
root@407b2886732a:/#
```

89°F Sunny 16-02-2023 16:27



- In this way, we have to setup a container and run ubuntu operating

system.

