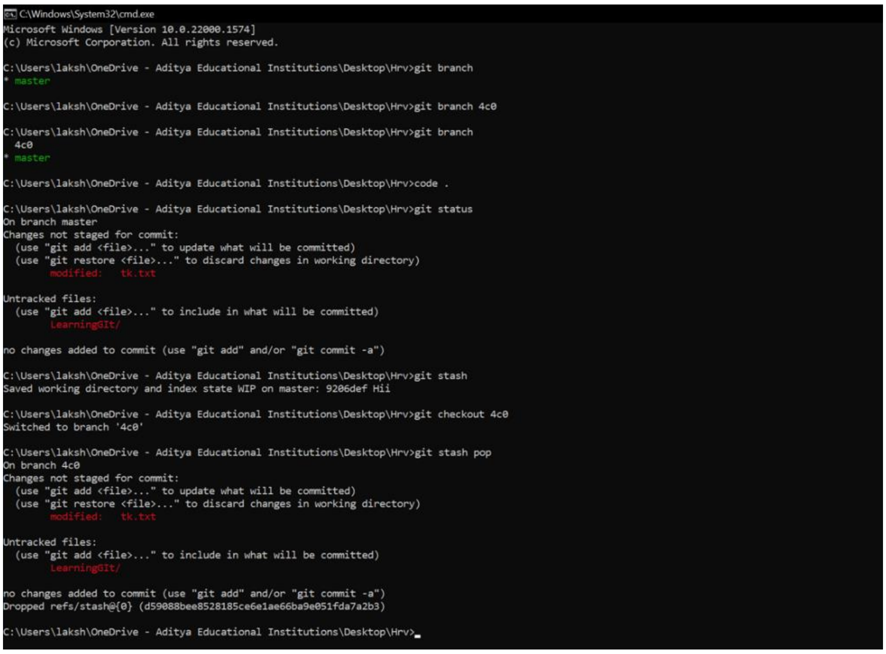
Graded Assignment on DevOps

Q-1: git stash temporarily shelves (or stashes) changes you've made to your working copy so you can work on something else, and then come back and re-apply them later on. Stashing is handy if you need to quickly switch context and work on something else, but you' re mid-way through a code change and aren't quite ready to commit.



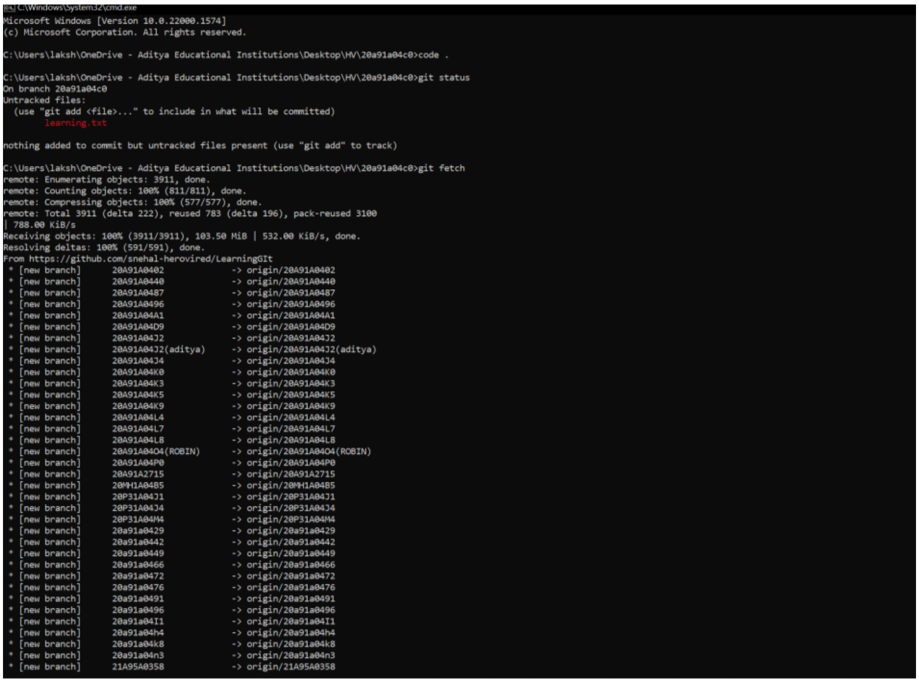
C:\Windows\System32\cmd.exe

```
Microsoft Windows [Version 10.0.22000.1574]
(c) Microsoft Corporation. All rights reserved.

C:\Users\laksh\OneDrive - Aditya Educational Institutions\Desktop\Hrv>git branch
* master

C:\Users\laksh\OneDrive - Aditya Educational Institutions\Desktop\Hrv>git branch 4c0

C:\Users\laksh\OneDrive - Aditya Educational Institutions\Desktop\Hrv>git branch
  4c0
* master

C:\Users\laksh\OneDrive - Aditya Educational Institutions\Desktop\Hrv>code .

C:\Users\laksh\OneDrive - Aditya Educational Institutions\Desktop\Hrv>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:   tk.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        LearningGIt/

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

C:\Users\laksh\OneDrive - Aditya Educational Institutions\Desktop\Hrv>git stash
Saved working directory and index state WIP on master: 9206def Hii

C:\Users\laksh\OneDrive - Aditya Educational Institutions\Desktop\Hrv>git checkout 4c0
Switched to branch '4c0'

C:\Users\laksh\OneDrive - Aditya Educational Institutions\Desktop\Hrv>git stash pop
On branch 4c0
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:   tk.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        LearningGIt/

no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (d59088bee8528185ce6e1ae66ba9e051fda7a2b3)

C:\Users\laksh\OneDrive - Aditya Educational Institutions\Desktop\Hrv>_
```

Steps to work with stash:

1. First, we need go to our master branch.
2. Then we need to create a file in the branch (if file doesn’t exist), if file exists then go to next step.
3. Then modify (add some content) in the text in the file.
4. Now apply **git stash** command.
5. Now create another branch and move to the created branch.
6. Apply **git stash pop** command.

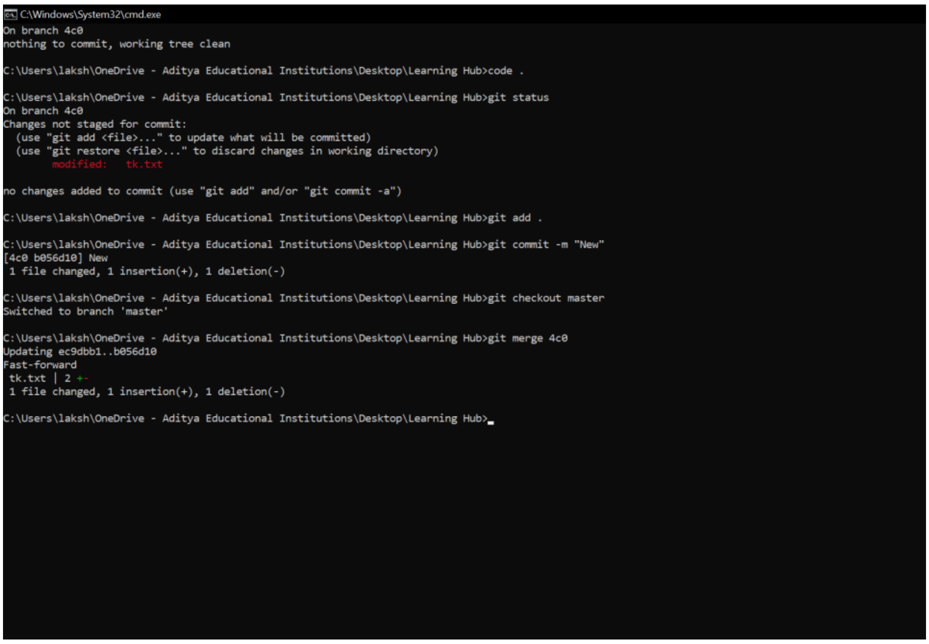
Q-2: The git fetch command downloads all branches, tags, and data from a project to the local machine. Existing local code is not overwritten. Fetch is commonly used with the git reset command to bring a local repository up to date with a remote repository.



C:\Windows\System32\cmd.exe

```
Microsoft Windows [Version 10.0.22000.1574]
(c) Microsoft Corporation. All rights reserved.

C:\Users\laksh\OneDrive - Aditya Educational Institutions\Desktop\HV\20a91a04c0>code .

C:\Users\laksh\OneDrive - Aditya Educational Institutions\Desktop\HV\20a91a04c0>git status
On branch 20a91a04c0
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        learning.txt

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

C:\Users\laksh\OneDrive - Aditya Educational Institutions\Desktop\HV\20a91a04c0>git fetch
remote: Enumerating objects: 3911, done.
remote: Counting objects: 100% (811/811), done.
remote: Compressing objects: 100% (577/577), done.
remote: Total 3911 (delta 222), reused 783 (delta 196), pack-reused 3100
| 788.00 KiB/s
Receiving objects: 100% (3911/3911), 103.50 MiB | 532.00 KiB/s, done.
Resolving deltas: 100% (591/591), done.
From https://github.com/snehal-herovired/LearningGIt
 * [new branch]      20A91A0402          -> origin/20A91A0402
 * [new branch]      20A91A0440          -> origin/20A91A0440
 * [new branch]      20A91A0487          -> origin/20A91A0487
 * [new branch]      20A91A0496          -> origin/20A91A0496
 * [new branch]      20A91A04A1          -> origin/20A91A04A1
 * [new branch]      20A91A04D9          -> origin/20A91A04D9
 * [new branch]      20A91A04J2          -> origin/20A91A04J2
 * [new branch]      20A91A04J2(aditya)  -> origin/20A91A04J2(aditya)
 * [new branch]      20A91A04J4          -> origin/20A91A04J4
 * [new branch]      20A91A04K0          -> origin/20A91A04K0
 * [new branch]      20A91A04K3          -> origin/20A91A04K3
 * [new branch]      20A91A04K5          -> origin/20A91A04K5
 * [new branch]      20A91A04K9          -> origin/20A91A04K9
 * [new branch]      20A91A04L4          -> origin/20A91A04L4
 * [new branch]      20A91A04L7          -> origin/20A91A04L7
 * [new branch]      20A91A04L8          -> origin/20A91A04L8
 * [new branch]      20A91A04O4(ROBIN)   -> origin/20A91A04O4(ROBIN)
 * [new branch]      20A91A04P0          -> origin/20A91A04P0
 * [new branch]      20A91A2715          -> origin/20A91A2715
 * [new branch]      20MH1A04B5          -> origin/20MH1A04B5
 * [new branch]      20P31A04J1          -> origin/20P31A04J1
 * [new branch]      20P31A04J4          -> origin/20P31A04J4
 * [new branch]      20P31A04M4          -> origin/20P31A04M4
 * [new branch]      20a91a0429          -> origin/20a91a0429
 * [new branch]      20a91a0442          -> origin/20a91a0442
 * [new branch]      20a91a0449          -> origin/20a91a0449
 * [new branch]      20a91a0466          -> origin/20a91a0466
 * [new branch]      20a91a0472          -> origin/20a91a0472
 * [new branch]      20a91a0476          -> origin/20a91a0476
 * [new branch]      20a91a0491          -> origin/20a91a0491
 * [new branch]      20a91a0496          -> origin/20a91a0496
 * [new branch]      20a91a04I1          -> origin/20a91a04I1
 * [new branch]      20a91a04h4          -> origin/20a91a04h4
 * [new branch]      20a91a04k8          -> origin/20a91a04k8
 * [new branch]      20a91a04n3          -> origin/20a91a04n3
 * [new branch]      21A95A0358          -> origin/21A95A0358
```

The primary use of **git merge** is to combine two branches. It is also used to merge multiple commits into one history. In the following illustration, git merge takes two branch tips and finds a common case commit between them. The common base commit creates a new commit that merges the changes of the sequence of each merge commit. Here we have two branches: a master and a stage.
We should merge the stage branch to the master branch.

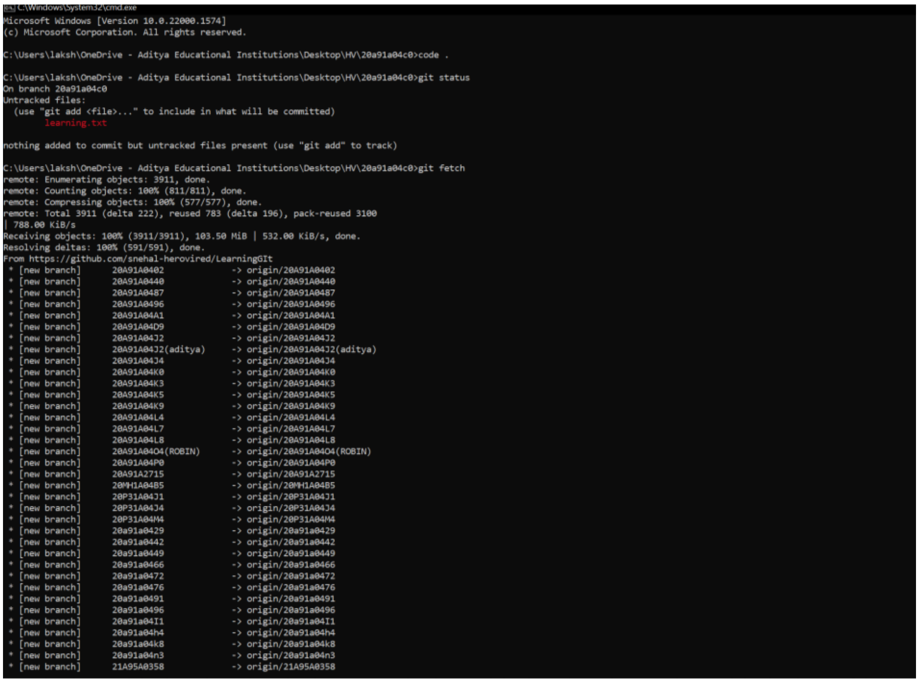


```
C:\Windows\System32\cmd.exe
On branch 4c0
nothing to commit, working tree clean

C:\Users\laksh\OneDrive - Aditya Educational Institutions\Desktop\Learning Hub>code .

C:\Users\laksh\OneDrive - Aditya Educational Institutions\Desktop\Learning Hub>git status
On branch 4c0
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:   tk.txt

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

C:\Users\laksh\OneDrive - Aditya Educational Institutions\Desktop\Learning Hub>git add .

C:\Users\laksh\OneDrive - Aditya Educational Institutions\Desktop\Learning Hub>git commit -m "New"
[4c0 b056d10] New
 1 file changed, 1 insertion(+), 1 deletion(-)

C:\Users\laksh\OneDrive - Aditya Educational Institutions\Desktop\Learning Hub>git checkout master
Switched to branch 'master'

C:\Users\laksh\OneDrive - Aditya Educational Institutions\Desktop\Learning Hub>git merge 4c0
Updating ec9dbb1..b056d10
Fast-forward
 tk.txt | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

C:\Users\laksh\OneDrive - Aditya Educational Institutions\Desktop\Learning Hub>
```

Steps to work with **git merge**:

1. Create a new branch using **git branch branch_name(ex:4c0)**
2. Move to the branch 4c0.
3. Modify the existing file or create a new file and add content.
4. Now check the **git status** and add the file by using **git add .** command.
5. Then commit the file using **git commit**.
6. After commit move to master branch using **git checkout master.**
7. Now merge the master branch and created branch using **git merge 4c0.**

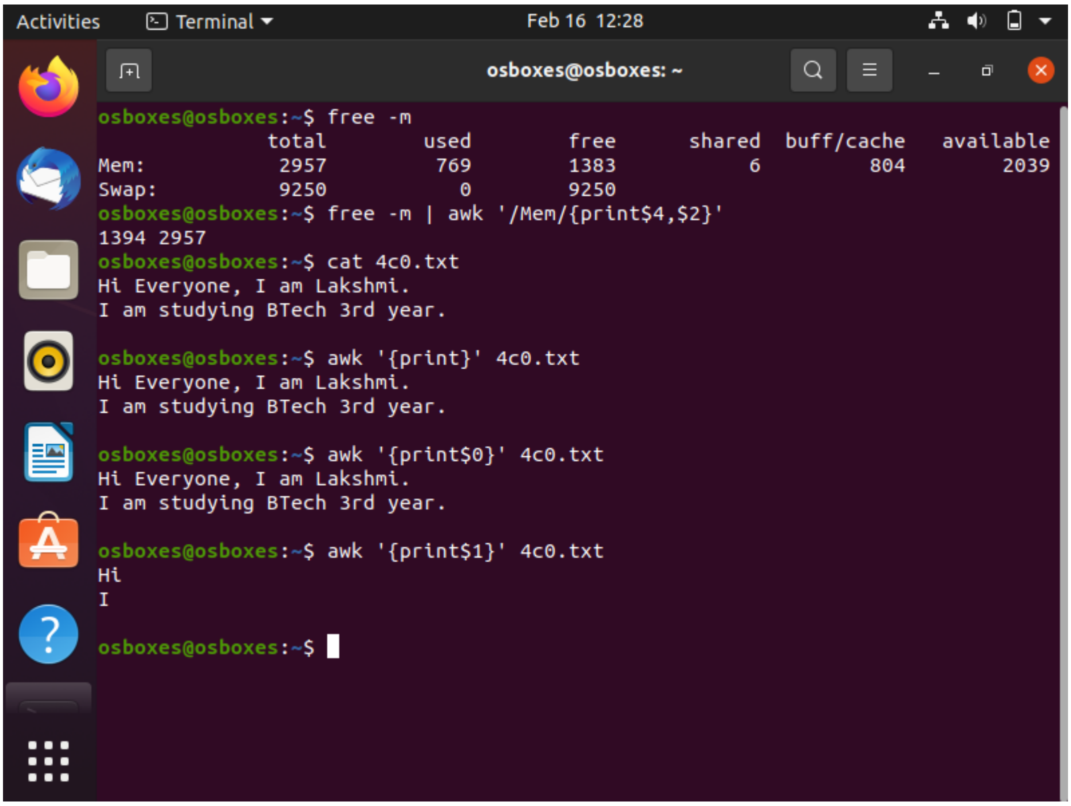
Q-3: The git fetch command downloads all branches, tags, and data from a project to the local machine. Existing local code is not overwritten. Fetch is commonly used with the git reset command to bring a local repository up to date with a remote repository.



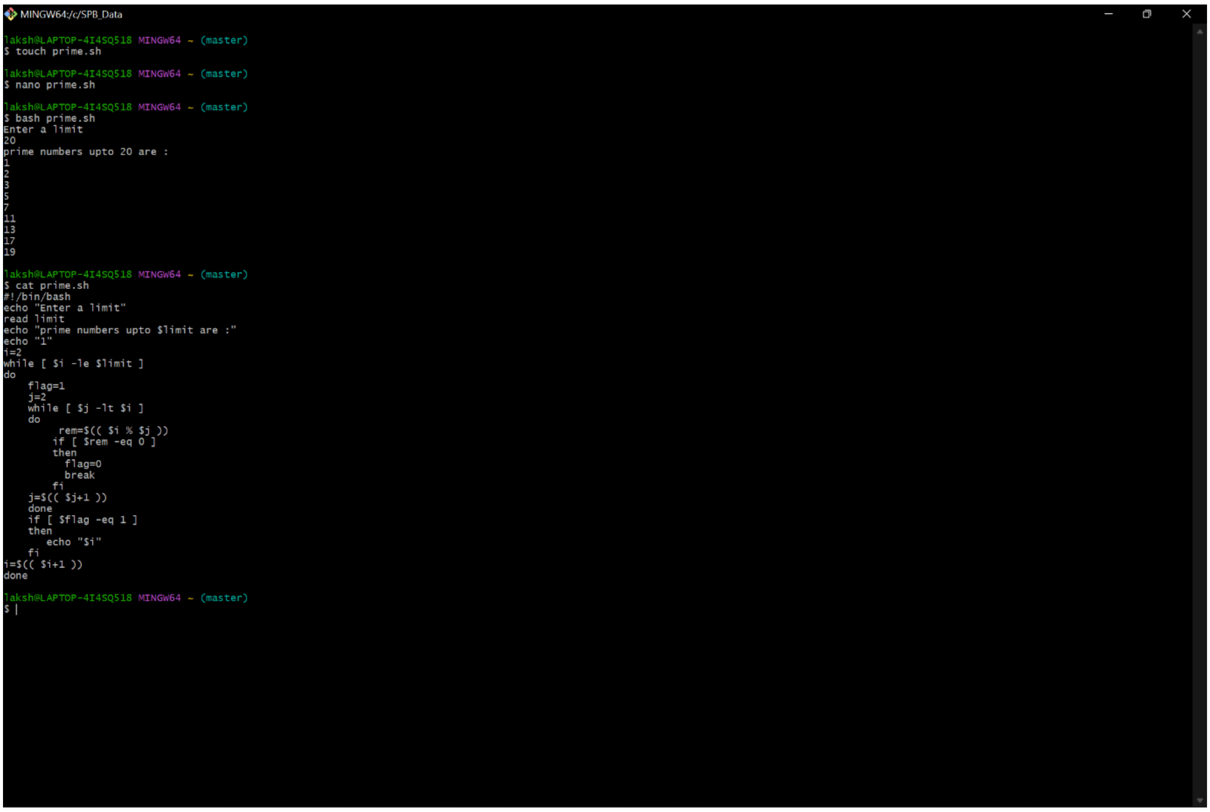
```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22000.1574]
(c) Microsoft Corporation. All rights reserved.

C:\Users\laksh\OneDrive - Aditya Educational Institutions\Desktop\HV\20a91a04c0>code .

C:\Users\laksh\OneDrive - Aditya Educational Institutions\Desktop\HV\20a91a04c0>git status
On branch 20a91a04c0
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        learning.txt

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

C:\Users\laksh\OneDrive - Aditya Educational Institutions\Desktop\HV\20a91a04c0>git fetch
remote: Enumerating objects: 3911, done.
remote: Counting objects: 100% (811/811), done.
remote: Compressing objects: 100% (577/577), done.
remote: Total 3911 (delta 222), reused 783 (delta 196), pack-reused 3100
| 788.00 KiB/s
Receiving objects: 100% (3911/3911), 103.50 MiB | 532.00 KiB/s, done.
Resolving deltas: 100% (591/591), done.
From https://github.com/snehal-herovired/LearningGIt
 * [new branch]      20A91A0402            -> origin/20A91A0402
 * [new branch]      20A91A0440            -> origin/20A91A0440
 * [new branch]      20A91A0487            -> origin/20A91A0487
 * [new branch]      20A91A0496            -> origin/20A91A0496
 * [new branch]      20A91A04A1            -> origin/20A91A04A1
 * [new branch]      20A91A04D9            -> origin/20A91A04D9
 * [new branch]      20A91A04J2            -> origin/20A91A04J2
 * [new branch]      20A91A04J2(aditya)    -> origin/20A91A04J2(aditya)
 * [new branch]      20A91A04J4            -> origin/20A91A04J4
 * [new branch]      20A91A04K0            -> origin/20A91A04K0
 * [new branch]      20A91A04K3            -> origin/20A91A04K3
 * [new branch]      20A91A04K5            -> origin/20A91A04K5
 * [new branch]      20A91A04K9            -> origin/20A91A04K9
 * [new branch]      20A91A04L4            -> origin/20A91A04L4
 * [new branch]      20A91A04L7            -> origin/20A91A04L7
 * [new branch]      20A91A04L8            -> origin/20A91A04L8
 * [new branch]      20A91A04O4(ROBIN)     -> origin/20A91A04O4(ROBIN)
 * [new branch]      20A91A04P0            -> origin/20A91A04P0
 * [new branch]      20A91A2715            -> origin/20A91A2715
 * [new branch]      20MH1A04B5            -> origin/20MH1A04B5
 * [new branch]      20P31A04J1            -> origin/20P31A04J1
 * [new branch]      20P31A04J4            -> origin/20P31A04J4
 * [new branch]      20P31A04M4            -> origin/20P31A04M4
 * [new branch]      20a91a0429            -> origin/20a91a0429
 * [new branch]      20a91a0442            -> origin/20a91a0442
 * [new branch]      20a91a0449            -> origin/20a91a0449
 * [new branch]      20a91a0466            -> origin/20a91a0466
 * [new branch]      20a91a0472            -> origin/20a91a0472
 * [new branch]      20a91a0476            -> origin/20a91a0476
 * [new branch]      20a91a0491            -> origin/20a91a0491
 * [new branch]      20a91a0496            -> origin/20a91a0496
 * [new branch]      20a91a04I1            -> origin/20a91a04I1
 * [new branch]      20a91a04h4            -> origin/20a91a04h4
 * [new branch]      20a91a04k8            -> origin/20a91a04k8
 * [new branch]      20a91a04n3            -> origin/20a91a04n3
 * [new branch]      21A95A0358            -> origin/21A95A0358
```

The git pull command is used to fetch and download content from a remote repository and immediately update the local repository to match that content. Merging remote upstream changes into your local repository is a common task in Git-based collaboration work flows.

Q-4:

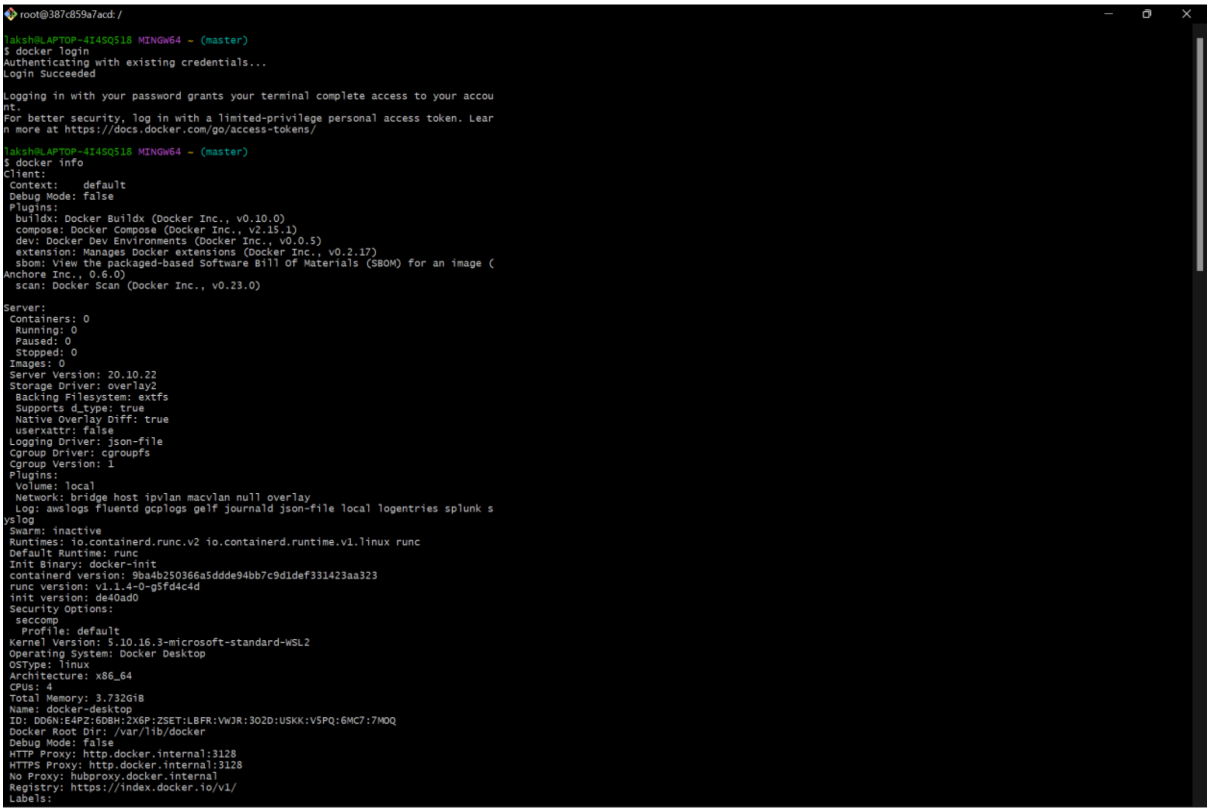


```
osboxes@osboxes:~$ free -m
              total        used        free      shared  buff/cache   available
Mem:           2957         769        1383           6         804        2039
Swap:          9250           0        9250
osboxes@osboxes:~$ free -m | awk '/Mem/{print$4,$2}'
1394 2957
osboxes@osboxes:~$ cat 4c0.txt
Hi Everyone, I am Lakshmi.
I am studying BTech 3rd year.

osboxes@osboxes:~$ awk '{print}' 4c0.txt
Hi Everyone, I am Lakshmi.
I am studying BTech 3rd year.

osboxes@osboxes:~$ awk '{print$0}' 4c0.txt
Hi Everyone, I am Lakshmi.
I am studying BTech 3rd year.

osboxes@osboxes:~$ awk '{print$1}' 4c0.txt
Hi
I

osboxes@osboxes:~$
```

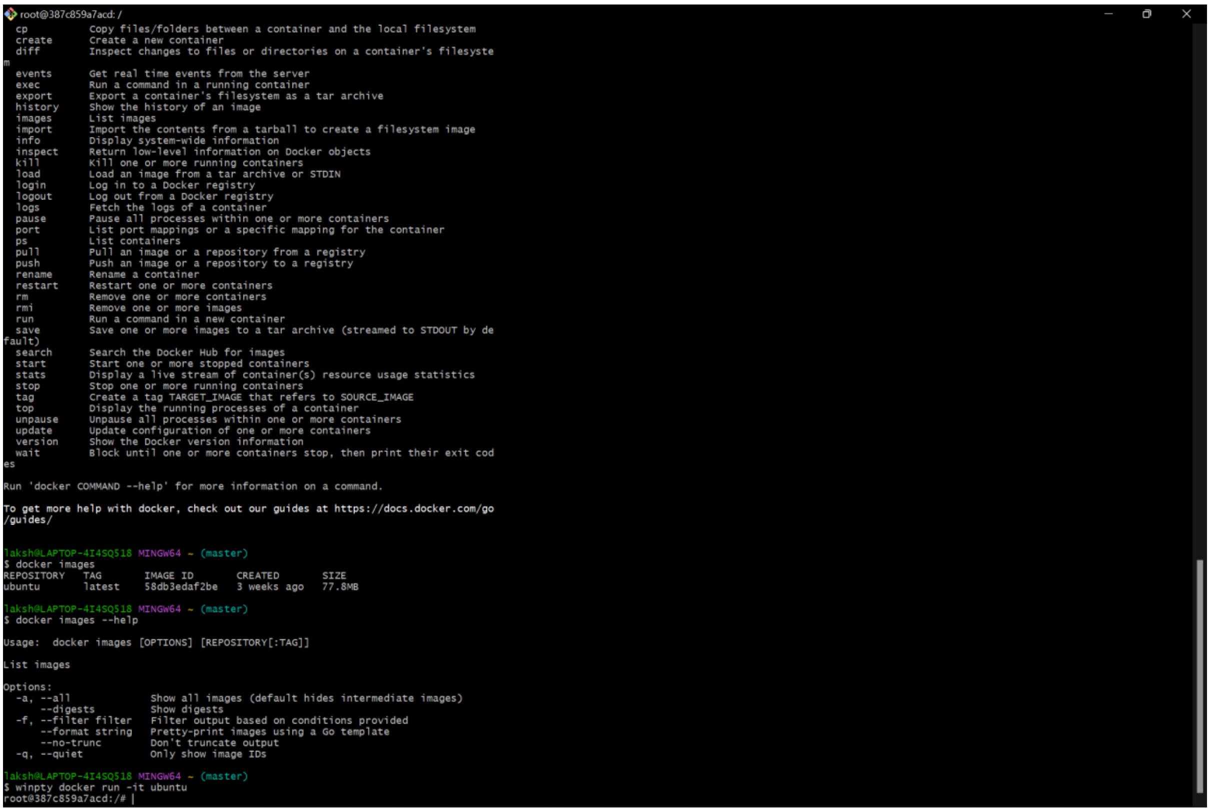


```
laksh@LAPTOP-4I4SQ518 MINGW64 ~ (master)
$ touch prime.sh

laksh@LAPTOP-4I4SQ518 MINGW64 ~ (master)
$ nano prime.sh

laksh@LAPTOP-4I4SQ518 MINGW64 ~ (master)
$ bash prime.sh
Enter a limit
20
prime numbers upto 20 are :
1
2
3
5
7
11
13
17
19

laksh@LAPTOP-4I4SQ518 MINGW64 ~ (master)
$ cat prime.sh
#!/bin/bash
echo "Enter a limit"
read limit
echo "prime numbers upto $limit are :"
echo "1"
i=2
while [ $i -le $limit ]
do
    flag=1
    j=2
    while [ $j -lt $i ]
    do
         rem=$(( $i % $j ))
        if [ $rem -eq 0 ]
        then
          flag=0
          break
        fi
    j=$(( $j+1 ))
    done
    if [ $flag -eq 1 ]
    then
       echo "$i"
    fi
i=$(( $i+1 ))
done

laksh@LAPTOP-4I4SQ518 MINGW64 ~ (master)
$ |
```

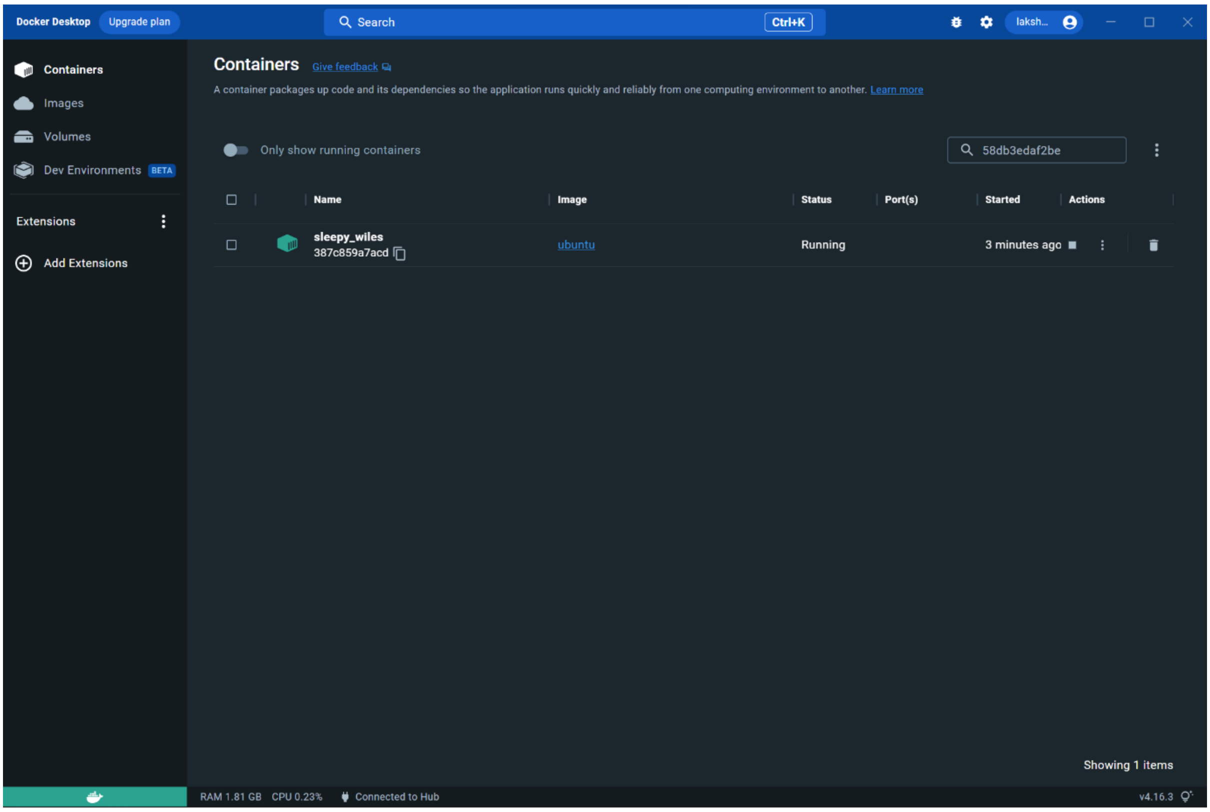
Q-5:



```
root@387c859a7acd: /

laksh@LAPTOP-4I4SQ518 MINGW64 ~ (master)
$ docker login
Authenticating with existing credentials...
Login Succeeded

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

laksh@LAPTOP-4I4SQ518 MINGW64 ~ (master)
$ docker info
Client:
 Context:    default
 Debug Mode: false
 Plugins:
  buildx: Docker Buildx (Docker Inc., v0.10.0)
  compose: Docker Compose (Docker Inc., v2.15.1)
  dev: Docker Dev Environments (Docker Inc., v0.0.5)
  extension: Manages Docker extensions (Docker Inc., v0.2.17)
  sbom: View the packaged-based Software Bill Of Materials (SBOM) for an image (
Anchore Inc., 0.6.0)
  scan: Docker Scan (Docker Inc., v0.23.0)

Server:
 Containers: 0
  Running: 0
  Paused: 0
  Stopped: 0
 Images: 0
 Server Version: 20.10.22
 Storage Driver: overlay2
  Backing Filesystem: extfs
  Supports d_type: true
  Native Overlay Diff: true
  userxattr: false
 Logging Driver: json-file
 Cgroup Driver: cgroupfs
 Cgroup Version: 1
 Plugins:
  Volume: local
  Network: bridge host ipvlan macvlan null overlay
  Log: awslogs fluentd gcplogs gelf journald json-file local logentries splunk s
yslog
 Swarm: inactive
 Runtimes: io.containerd.runc.v2 io.containerd.runtime.v1.linux runc
 Default Runtime: runc
 Init Binary: docker-init
 containerd version: 9ba4b250366a5ddde94bb7c9d1def331423aa323
 runc version: v1.1.4-0-g5fd4c4d
 init version: de40ad0
 Security Options:
  seccomp
   Profile: default
 Kernel Version: 5.10.16.3-microsoft-standard-WSL2
 Operating System: Docker Desktop
 OSType: linux
 Architecture: x86_64
 CPUs: 4
 Total Memory: 3.732GiB
 Name: docker-desktop
 ID: DD6N:E4PZ:6DBH:2X6P:ZSET:LBFR:VWJR:3O2D:USKK:V5PQ:6MC7:7MOQ
 Docker Root Dir: /var/lib/docker
 Debug Mode: false
 HTTP Proxy: http.docker.internal:3128
 HTTPS Proxy: http.docker.internal:3128
 No Proxy: hubproxy.docker.internal
 Registry: https://index.docker.io/v1/
 Labels:
```



```
root@387c859a7acd: /
  cp          Copy files/folders between a container and the local filesystem
  create      Create a new container
  diff        Inspect changes to files or directories on a container's filesyste
m
  events      Get real time events from the server
  exec        Run a command in a running container
  export      Export a container's filesystem as a tar archive
  history     Show the history of an image
  images      List images
  import      Import the contents from a tarball to create a filesystem image
  info        Display system-wide information
  inspect     Return low-level information on Docker objects
  kill        Kill one or more running containers
  load        Load an image from a tar archive or STDIN
  login       Log in to a Docker registry
  logout      Log out from a Docker registry
  logs        Fetch the logs of a container
  pause       Pause all processes within one or more containers
  port        List port mappings or a specific mapping for the container
  ps          List containers
  pull        Pull an image or a repository from a registry
  push        Push an image or a repository to a registry
  rename      Rename a container
  restart     Restart one or more containers
  rm          Remove one or more containers
  rmi         Remove one or more images
  run         Run a command in a new container
  save        Save one or more images to a tar archive (streamed to STDOUT by de
fault)
  search      Search the Docker Hub for images
  start       Start one or more stopped containers
  stats       Display a live stream of container(s) resource usage statistics
  stop        Stop one or more running containers
  tag         Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE
  top         Display the running processes of a container
  unpause     Unpause all processes within one or more containers
  update      Update configuration of one or more containers
  version     Show the Docker version information
  wait        Block until one or more containers stop, then print their exit cod
es

Run 'docker COMMAND --help' for more information on a command.

To get more help with docker, check out our guides at https://docs.docker.com/go
/guides/


laksh@LAPTOP-4I4SQ518 MINGW64 ~ (master)
$ docker images
REPOSITORY   TAG       IMAGE ID       CREATED       SIZE
ubuntu       latest    58db3edaf2be   3 weeks ago   77.8MB

laksh@LAPTOP-4I4SQ518 MINGW64 ~ (master)
$ docker images --help

Usage:  docker images [OPTIONS] [REPOSITORY[:TAG]]

List images

Options:
  -a, --all             Show all images (default hides intermediate images)
      --digests         Show digests
  -f, --filter filter   Filter output based on conditions provided
      --format string   Pretty-print images using a Go template
      --no-trunc        Don't truncate output
  -q, --quiet           Only show image IDs

laksh@LAPTOP-4I4SQ518 MINGW64 ~ (master)
$ winpty docker run -it ubuntu
root@387c859a7acd:/# |
```



Containers

A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. Learn more

Only show running containers

58db3edaf2be

□	Name	Image	Status	Port(s)	Started	Actions
□	sleepy_wiles 387c859a7acd	ubuntu	Running		3 minutes ago	

Showing 1 items

RAM 1.81 GB CPU 0.23% Connected to Hub v4.16.3

By

U. Lakshmi

20A91A04C0