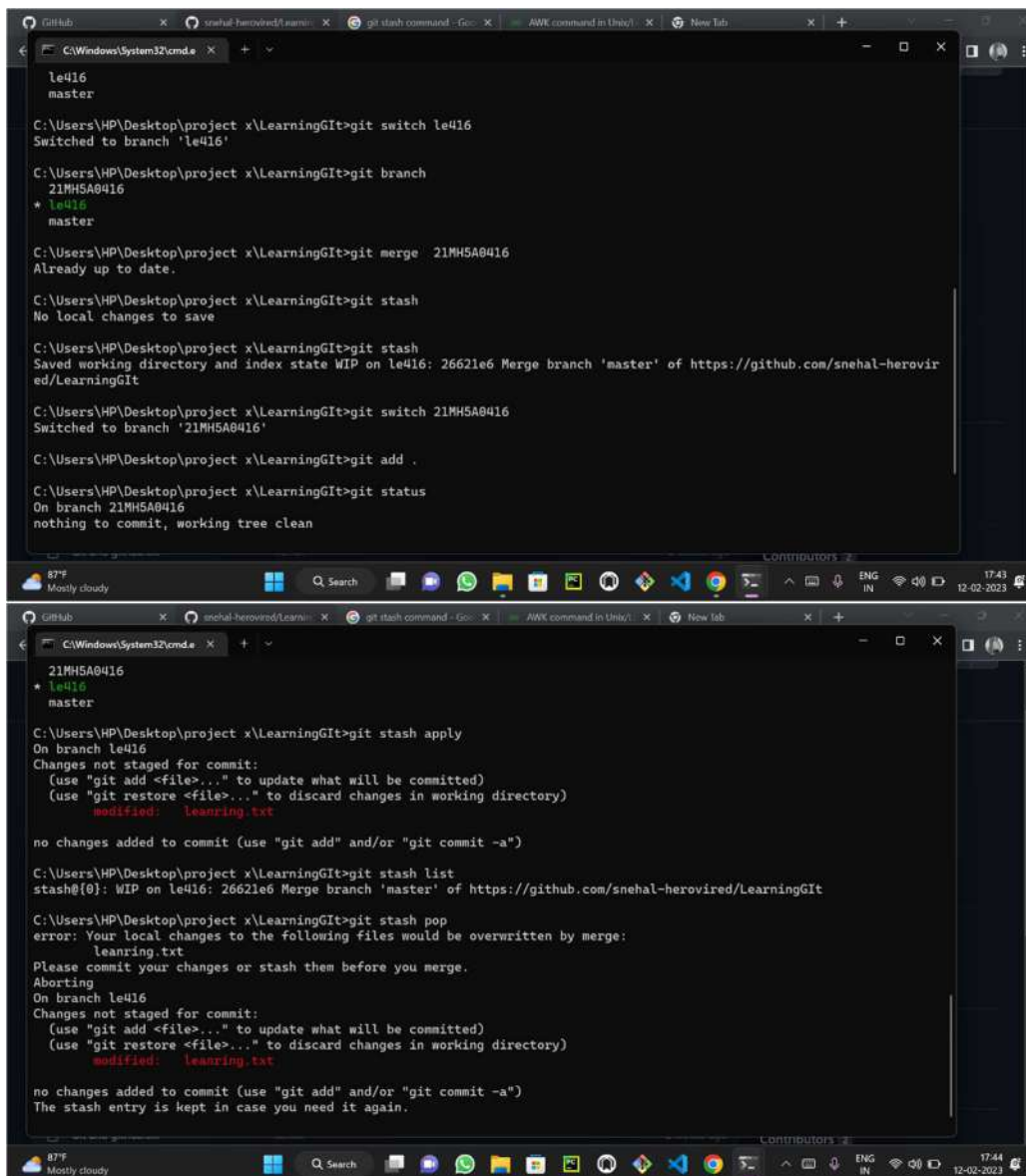


## Q1 - GIT STASH COMMAND-

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.

Use git stash when you want to record the current state of the working directory and the index, but want to go back to a clean working directory. The command saves your local modifications away and reverts the working directory to match the HEAD commit.



The image consists of two screenshots of a Windows command prompt window, showing the execution of various Git commands related to stashing and switching branches.

**Top Screenshot:**

```
le416
master
C:\Users\HP\Desktop\project x\LearningGit>git switch le416
Switched to branch 'le416'
C:\Users\HP\Desktop\project x\LearningGit>git branch
 21MHSA0416
* le416
  master
C:\Users\HP\Desktop\project x\LearningGit>git merge 21MHSA0416
Already up to date.
C:\Users\HP\Desktop\project x\LearningGit>git stash
No local changes to save
C:\Users\HP\Desktop\project x\LearningGit>git stash
Saved working directory and index state WIP on le416: 26621e6 Merge branch 'master' of https://github.com/snehal-herovired/LearningGit
C:\Users\HP\Desktop\project x\LearningGit>git switch 21MHSA0416
Switched to branch '21MHSA0416'
C:\Users\HP\Desktop\project x\LearningGit>git add .
C:\Users\HP\Desktop\project x\LearningGit>git status
On branch 21MHSA0416
nothing to commit, working tree clean
```

**Bottom Screenshot:**

```
21MHSA0416
* le416
  master
C:\Users\HP\Desktop\project x\LearningGit>git stash apply
On branch le416
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:   learning.txt
no changes added to commit (use "git add" and/or "git commit -a")
C:\Users\HP\Desktop\project x\LearningGit>git stash list
stash@{0}: WIP on le416: 26621e6 Merge branch 'master' of https://github.com/snehal-herovired/LearningGit
C:\Users\HP\Desktop\project x\LearningGit>git stash pop
error: Your local changes to the following files would be overwritten by merge:
  learning.txt
Please commit your changes or stash them before you merge.
Aborting
On branch le416
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:   learning.txt
no changes added to commit (use "git add" and/or "git commit -a")
The stash entry is kept in case you need it again.
```

We use git stash command to save the current work without any commits .

After the stash it will be stored in stash memory

We can switch to another branch or directory to complete any work and we can return to previous stashed branch so that we can continue our work.

We use git stash pop to retrieve the previous info regarding the stashed work .

We use stash apply to apply the changes and we can see the list of the stashed files using the stash list command.

```
C:\Windows\System32\cmd.exe X + v
C:\Users\HP\Desktop\project x\LearningGit>git branch
21MH5A0416
* le416
master

C:\Users\HP\Desktop\project x\LearningGit>git stash apply
On branch le416
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:   leanring.txt

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

C:\Users\HP\Desktop\project x\LearningGit>git stash list
stash@{0}: WIP on le416: 26621e6 Merge branch 'master' of https://github.com/snehal-herovired/LearningGit

C:\Users\HP\Desktop\project x\LearningGit>git stash pop
error: Your local changes to the following files would be overwritten by merge:
  leanring.txt
Please commit your changes or stash them before you merge.
Aborting
On branch le416
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:   leanring.txt

no changes added to commit (use "git add" and/or "git commit -a")
The stash entry is kept in case you need it again.

C:\Windows\System32\cmd.exe X + v
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   fetchandpull.txt

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:   leanring.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        gitFetch and pull.txt

C:\Users\HP\Desktop\project x\LearningGit>git stash pop
error: Your local changes to the following files would be overwritten by merge:
  leanring.txt
Please commit your changes or stash them before you merge.
Aborting
On branch le416
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   fetchandpull.txt

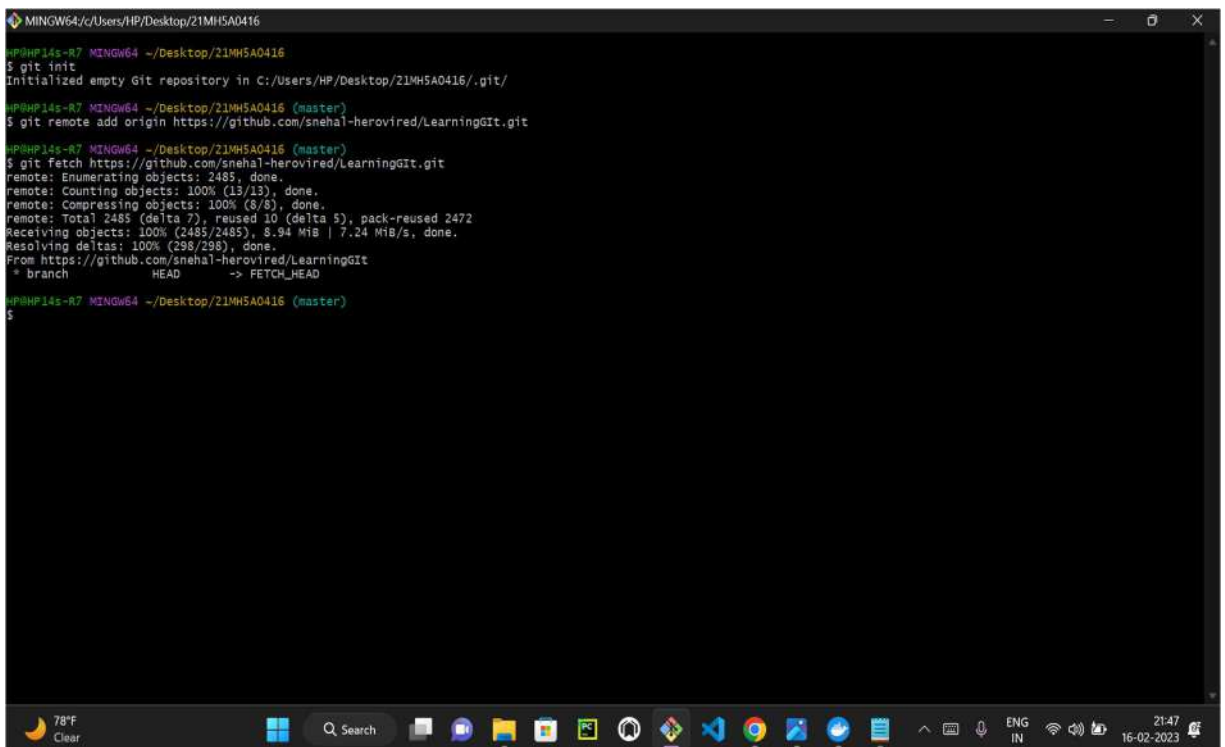
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   leanring.txt

Untracked files:
```

## Q2      \_- GIT FETCH AND GIT MERGE -\_

### ■ GIT FETCH

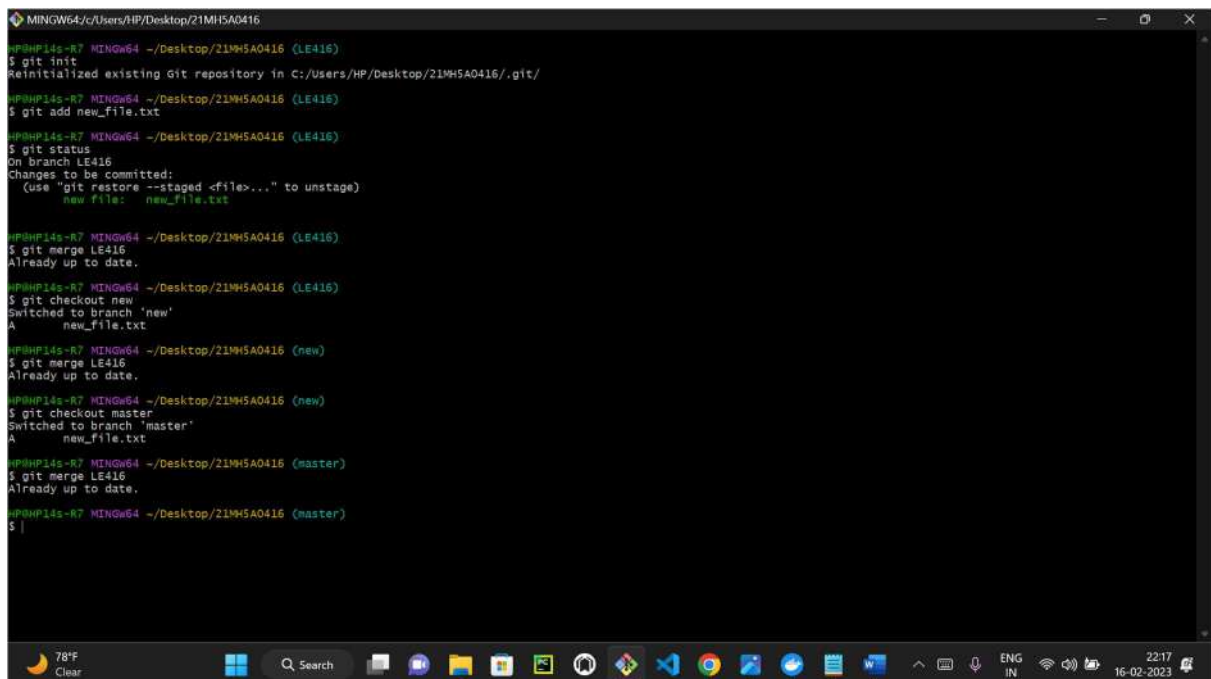
1. git fetch is a primary command used to download contents from a remote repository.
2. git fetch is used in conjunction with git remote , git branch , git checkout
3. git reset to update a local repository to the state of a remote.
4. The git fetch command is a critical piece of collaborative git work flows.
5. The "git fetch" command is used to pull the updates from remote-tracking branches.
6. Additionally, we can get the updates that have been pushed to our remote branches to our local machines.
7. As we know, a branch is a variation of our repositories main code,
8. so the remote-tracking branches are branches that have been set up to pull and push from remote repository



```
MINGW64/c/Users/HP/Desktop/21MHSAD0416
HPHP14s-R7 MINGW64 ~/Desktop/21MHSAD0416
$ git init
Initialized empty Git repository in C:/Users/HP/Desktop/21MHSAD0416/.git/
HPHP14s-R7 MINGW64 ~/Desktop/21MHSAD0416 (master)
$ git remote add origin https://github.com/snehal-herovired/LearningGit.git
HPHP14s-R7 MINGW64 ~/Desktop/21MHSAD0416 (master)
$ 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 | 7.24 MiB/s, done.
Resolving deltas: 100% (298/298), done.
From https://github.com/snehal-herovired/LearningGit
 * branch            HEAD       -> FETCH_HEAD
HPHP14s-R7 MINGW64 ~/Desktop/21MHSAD0416 (master)
$
```

## ■ **\_ - GIT MERGE - \_**

- 1.The "merge" command is used to integrate changes from another branch.
- 2.Creates a merge commit even when a fast-forward would be possible.
- 3.the merging of branches or master will involves the conflicts due to modifications
- 4.we use "git merge <branch\_name>" for merging a branch  
And "git merge <master>" for merging master with a branch.
- 5.it will show uptodate if they were alredy merged.



```
MINGW64/c/Users/HP/Desktop/21MH5A0416
HPHP14s-R7 MINGW64 ~/Desktop/21MH5A0416 (LE416)
$ git init
Reinitialized existing Git repository in C:/Users/HP/Desktop/21MH5A0416/.git/
HPHP14s-R7 MINGW64 ~/Desktop/21MH5A0416 (LE416)
$ git add new_file.txt
HPHP14s-R7 MINGW64 ~/Desktop/21MH5A0416 (LE416)
$ git status
On branch LE416
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   new_file.txt
HPHP14s-R7 MINGW64 ~/Desktop/21MH5A0416 (LE416)
$ git merge LE416
Already up to date.
HPHP14s-R7 MINGW64 ~/Desktop/21MH5A0416 (LE416)
$ git checkout new
Switched to branch 'new'
A       new_file.txt
HPHP14s-R7 MINGW64 ~/Desktop/21MH5A0416 (new)
$ git merge LE416
Already up to date.
HPHP14s-R7 MINGW64 ~/Desktop/21MH5A0416 (new)
$ git checkout master
Switched to branch 'master'
A       new_file.txt
HPHP14s-R7 MINGW64 ~/Desktop/21MH5A0416 (master)
$ git merge LE416
Already up to date.
HPHP14s-R7 MINGW64 ~/Desktop/21MH5A0416 (master)
$
```

### Q3 \_-DIFFERENCE BETWEEN GIT FETCH AND GIT PULL- \_

**GIT FETCH:** Gives the information of a new change from a remote repository without merging into the current branch. and

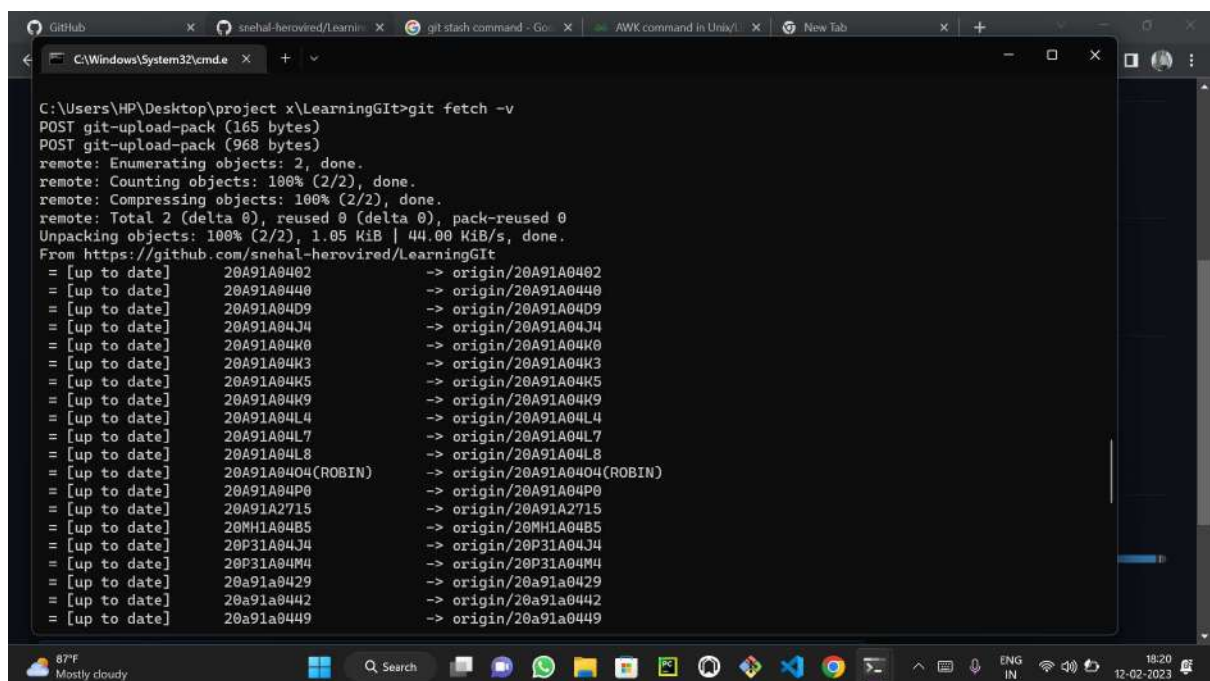
Git fetch basically imports the commits to local branches as to keep data up-to-date that what everybody is working on.

command:- `git fetch <remote>`

**GIT PULL:** Brings the copy of all the changes from a remote repository and merges them into the current branch.

by using git pull we will face merge conflicts.

command:- `git pull origin <branch name>`

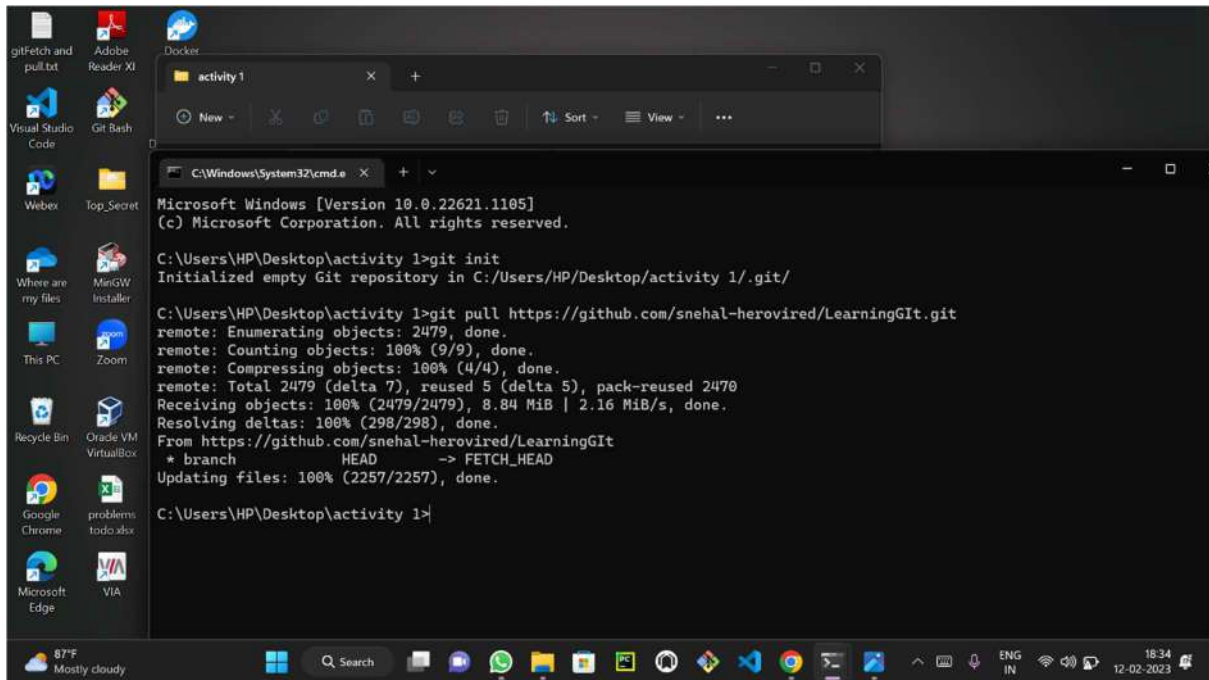


```
C:\Users\HP\Desktop\project x\LearningGit>git fetch -v
POST git-upload-pack (165 bytes)
POST git-upload-pack (968 bytes)
remote: Enumerating objects: 2, done.
remote: Counting objects: 100% (2/2), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 2 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (2/2), 1.05 KiB | 44.00 KiB/s, done.
From https://github.com/snehal-herovired/LearningGit
= [up to date] 20A91A0402 -> origin/20A91A0402
= [up to date] 20A91A0440 -> origin/20A91A0440
= [up to date] 20A91A04D9 -> origin/20A91A04D9
= [up to date] 20A91A04J4 -> origin/20A91A04J4
= [up to date] 20A91A04K0 -> origin/20A91A04K0
= [up to date] 20A91A04K3 -> origin/20A91A04K3
= [up to date] 20A91A04K5 -> origin/20A91A04K5
= [up to date] 20A91A04K9 -> origin/20A91A04K9
= [up to date] 20A91A04L4 -> origin/20A91A04L4
= [up to date] 20A91A04L7 -> origin/20A91A04L7
= [up to date] 20A91A04L8 -> origin/20A91A04L8
= [up to date] 20A91A0404(ROBIN) -> origin/20A91A0404(ROBIN)
= [up to date] 20A91A04P0 -> origin/20A91A04P0
= [up to date] 20A91A2715 -> origin/20A91A2715
= [up to date] 20MH1A04B5 -> origin/20MH1A04B5
= [up to date] 20P31A04J4 -> origin/20P31A04J4
= [up to date] 20P31A04M4 -> origin/20P31A04M4
= [up to date] 20a91a0429 -> origin/20a91a0429
= [up to date] 20a91a0442 -> origin/20a91a0442
= [up to date] 20a91a0449 -> origin/20a91a0449
```

Both commands are used to download the data from a remote repository. But both of these commands work differently. Like when you do a git pull, it gets all the changes from the remote or central repository and makes it available to your corresponding branch in your local repository.

When you do a git fetch, it fetches all the changes from the remote repository and stores it in a separate branch in your local repository. You can reflect those changes in your corresponding branches by merging.

git pull = git fetch + git merge



```
Microsoft Windows [Version 10.0.22621.1105]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HP\Desktop\activity 1>git init
Initialized empty Git repository in C:/Users/HP/Desktop/activity 1/.git/

C:\Users\HP\Desktop\activity 1>git pull https://github.com/snehal-herovired/LearningGit.git
remote: Enumerating objects: 2479, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 2479 (delta 7), reused 5 (delta 5), pack-reused 2470
Receiving objects: 100% (2479/2479), 8.84 MiB | 2.16 MiB/s, done.
Resolving deltas: 100% (298/298), done.
From https://github.com/snehal-herovired/LearningGit
* branch      HEAD       -> FETCH_HEAD
Updating files: 100% (2257/2257), done.

C:\Users\HP\Desktop\activity 1>
```



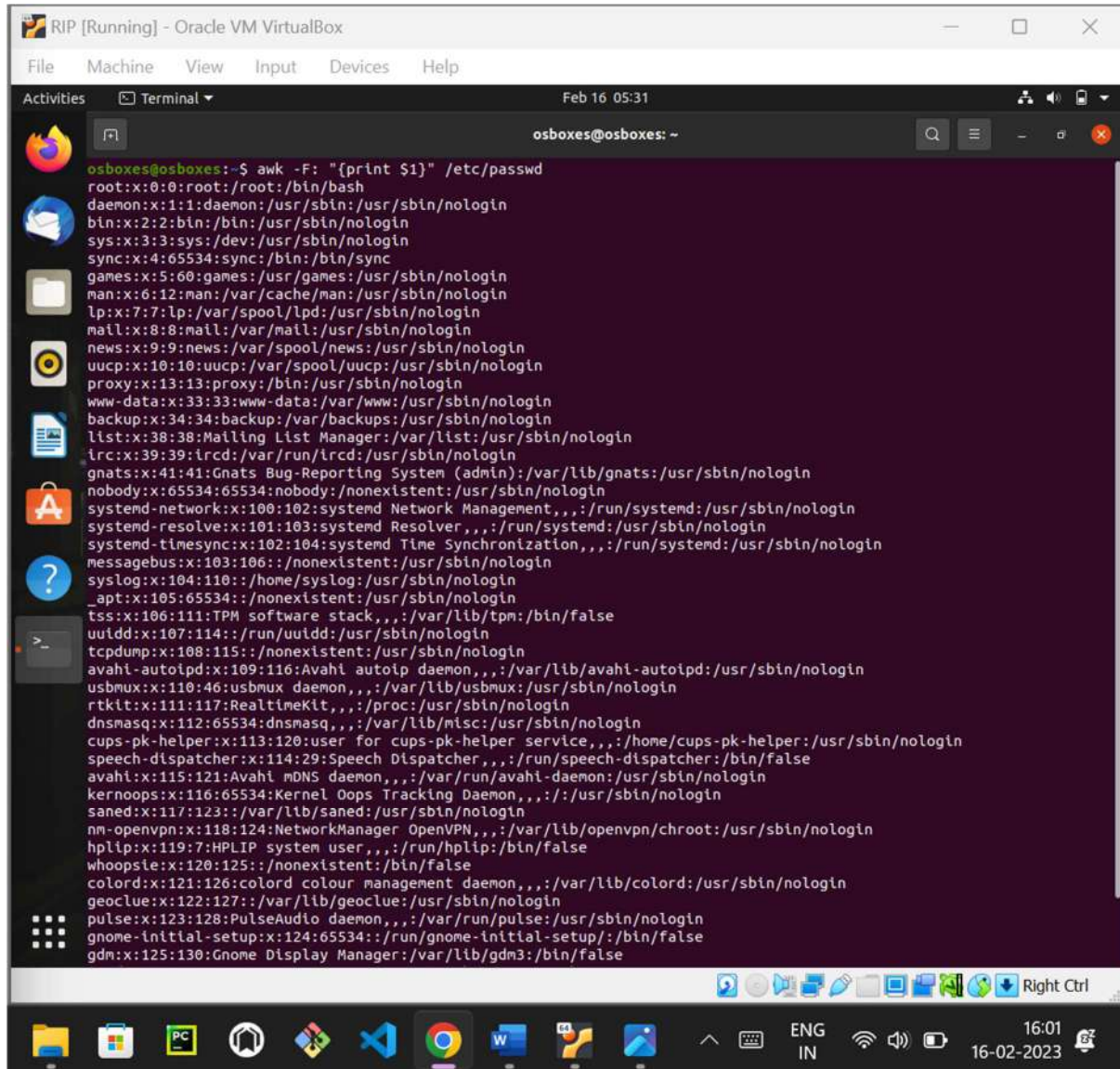
## Q4 \_\_-AWK COMMAND IN LINUX:-\_\_

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

Awk is mostly used for pattern scanning and processing.

It searches one or more files to see if they contain lines

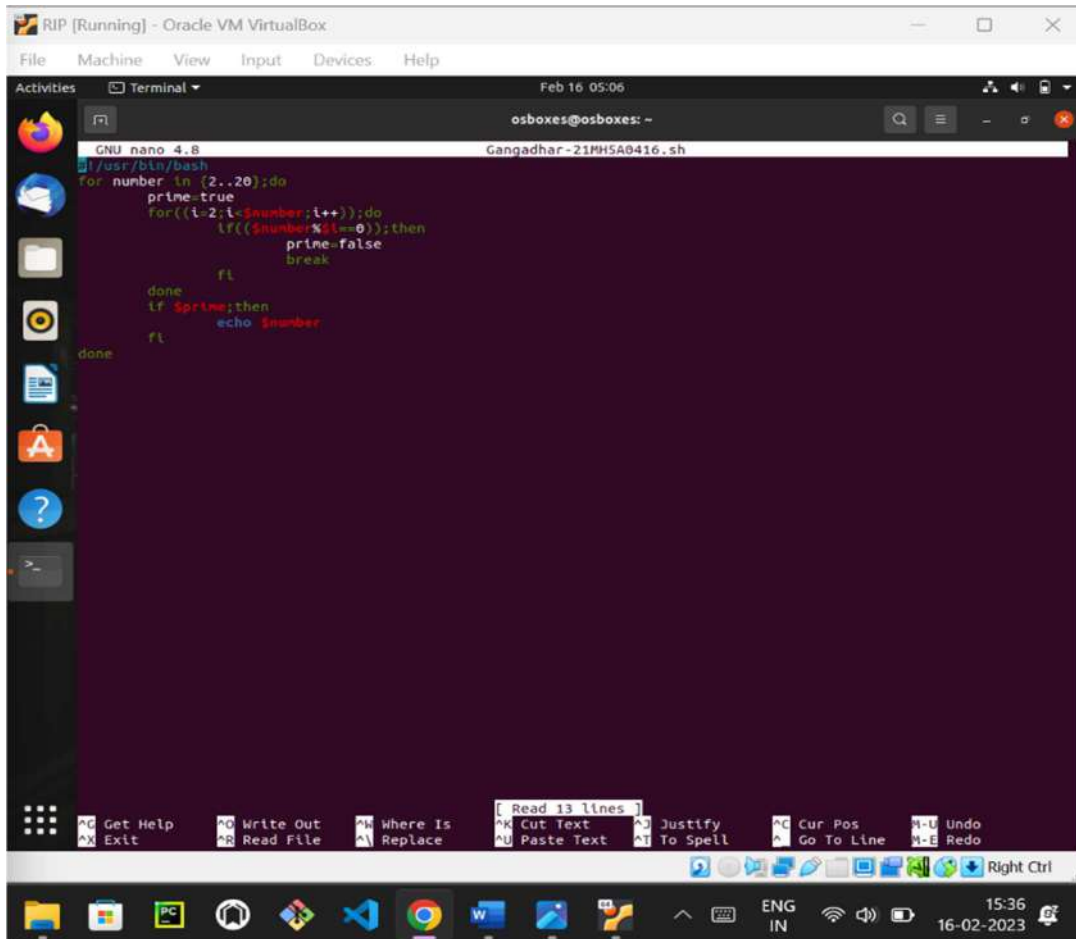
that matches with the specified patterns and then perform the associated actions



```
RIP [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Feb 16 05:31
osboxes@osboxes: ~
osboxes@osboxes:~$ awk -F: '{print $1}' /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailng List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
systemd-timesync:x:102:104:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:106:nonexistent:/usr/sbin/nologin
syslog:x:104:110:/home/syslog:/usr/sbin/nologin
_apt:x:105:65534:nonexistent:/usr/sbin/nologin
tss:x:106:111:TPM software stack,,,:/var/lib/tpm:/bin/false
uuidd:x:107:114:/run/uuidd:/usr/sbin/nologin
tcpdump:x:108:115:nonexistent:/usr/sbin/nologin
avahi-autoipd:x:109:116:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
usbmux:x:110:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
rtkit:x:111:117:RealtimeKit,,,:/proc:/usr/sbin/nologin
dnsmasq:x:112:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
cups-pk-helper:x:113:120:user for cups-pk-helper service,,,:/home/cups-pk-helper:/usr/sbin/nologin
speech-dispatcher:x:114:29:Speech Dispatcher,,,:/run/speech-dispatcher:/bin/false
avahi:x:115:121:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin
kernoops:x:116:65534:Kernel Oops Tracking Daemon,,,:/usr/sbin/nologin
saned:x:117:123:/var/lib/saned:/usr/sbin/nologin
nm-openvpn:x:118:124:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin/nologin
hplip:x:119:7:HPLIP system user,,,:/run/hplip:/bin/false
whoopsie:x:120:125:nonexistent:/bin/false
colord:x:121:126:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
geoclue:x:122:127:/var/lib/geoclue:/usr/sbin/nologin
pulse:x:123:128:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin
gnome-initial-setup:x:124:65534:/run/gnome-initial-setup:/bin/false
gdm:x:125:130:Gnome Display Manager:/var/lib/gdm3:/bin/false
```

## BASH PROGRAM TO PRINT 1-20 PRIME NUMBERS:

- 1.create a file using the Touch command.
- 2.edit the file using Nano file name command and write down the program
- 3.Run the file using Bash filename command.

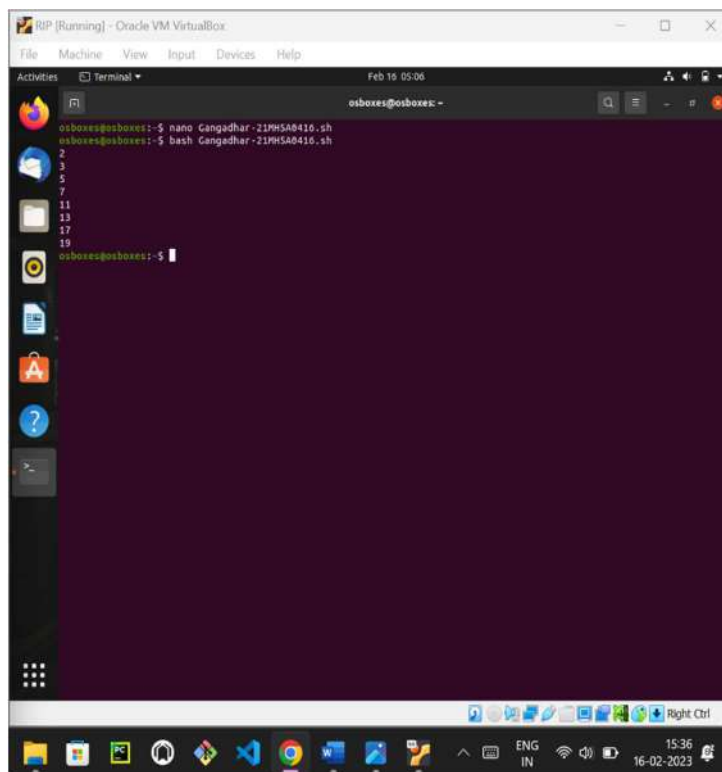


The screenshot shows a VirtualBox window titled "RIP [Running] - Oracle VM VirtualBox". Inside, a terminal window is open with the prompt "osboxes@osboxes: ~". The terminal displays the output of the "cat" command for a file named "Gangadhar-21MH5A0416.sh". The script is a Bash program that iterates through numbers from 2 to 20 and prints the prime ones. The script content is as follows:

```
GNU nano 4.8
#!/usr/bin/bash
for number in {2..20};do
    prime=true
    for((i=2;i<=$number;i++));do
        if((($number%i==0));then
            prime=false
            break
        fi
    done
    if $prime;then
        echo $number
    fi
done
```

At the bottom of the terminal window, there is a status bar with various keyboard shortcuts and a "Right Ctrl" button. The system tray at the bottom of the screen shows the date and time as "15:36 16-02-2023".

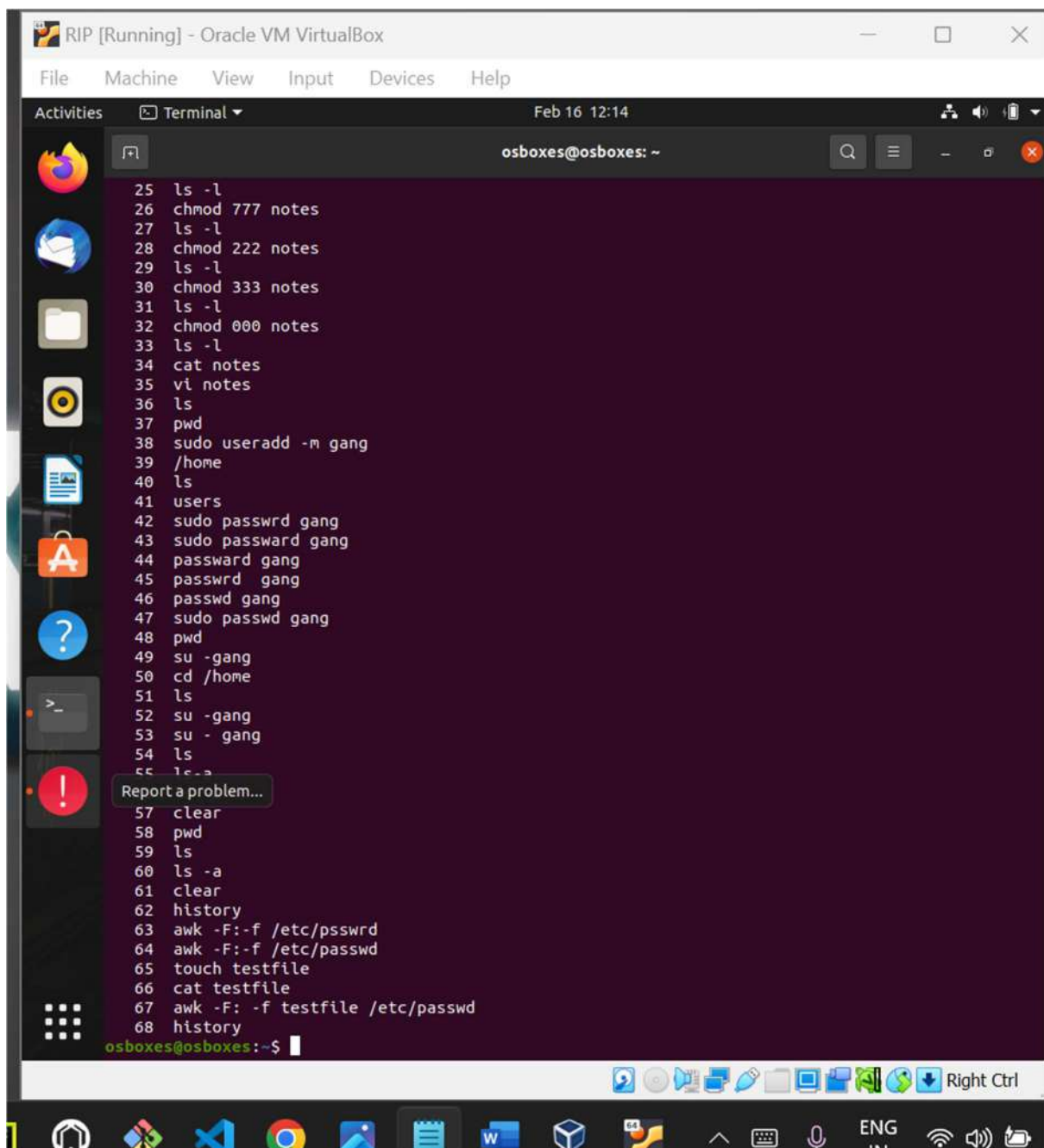




## **HISTORY COMMAND :**

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

It will show all commands used right from the beginning of the usage of the linux or respected os.



## Q5 \_-PROCESS TO SET UP A CONTAINER AND RUN A UBUNTU OPERATING SYSTEM. -\_

- 1.open the folder which is having the docker file .
- 2.open the git bash through the folder and it will directly open sthe git bash.
- 3.login with your docker hub login credentials.
- 4.now use the "docker pull image\_name" to pull the ubuntu image.
- 5.by using the "docker run -it image\_name" command run the ubuntu operating system by creating a container using dockerhub.
6. to exit from the interactive mode use "exit" command.
- 7.it will show that the ubuntu OS is running in the container in dockerhub.
8. after the use terminate or stop the container.

```
MINGW64/C:/Users/HP/Desktop/ClonedData/pythonflask
$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES

$ docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
Digest: sha256:9a0dde4180b896a372804be2384015e90e3f84906b750c1a53539b585fbbe7f
Status: Image is up to date for ubuntu:latest
docker.io/library/ubuntu:latest

$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
gangadhar416/flask_image   latest    0bc9e99f8fed   2 days ago     156MB
gangadhar416/new_image     latest    4b6d0e9f9d20   3 days ago     919MB
gangadhar416/new_image     latest    4b6d0e9f9d20   3 days ago     919MB
gangadhar416/new_file      latest    9dfdadf042fa   3 days ago     919MB
gangadhar416/new_file      latest    9dfdadf042fa   3 days ago     919MB
ubuntu          latest    58db3edaf2be   3 weeks ago    77.6MB

$ docker run -it ubuntu
the input device is not a TTY.  If you are using mntty, try prefixing the command with 'wintty'

$ wintty docker run -it ubuntu
root@46103e4dae0b:/# pwd
/
root@46103e4dae0b:/# ls
bin  boot  dev  etc  home  lib  lib32  lib64  libx32  media  mnt  opt  proc  root  run  sbin  srv  sys  usr  var
root@46103e4dae0b:/# cd home
root@46103e4dae0b:/home# ls
root@46103e4dae0b:/home# pwd
/home
root@46103e4dae0b:/home# ls -la
total 8
drwxr-xr-x 1 root root 4096 Jan 16 21:07 .
drwxr-xr-x 1 root root 4096 Jan 16 21:07 ..
root@46103e4dae0b:/home# exit
exit
$
```

<input type="checkbox"/>	Name	Image	Status	Port(s)	Started	Actions
<input type="checkbox"/>	<b>stupefied_lumiere</b> 248de964114c	<a href="#">flask_image</a>	Exited	5000:5001		
<input type="checkbox"/>	<b>brave_mcnulty</b> f6ab0a853c71	<a href="#">new_image</a>	Exited (255)	5000:5000		
<input type="checkbox"/>	<b>flamboyant_kare</b> 16196f3ff597	<a href="#">new_file</a>	Exited (137)	5000:3000		
<input type="checkbox"/>	<b>youthful_mcnulty</b> 04772f0983bc	<a href="#">ubuntu</a>	Exited (137)			
<input type="checkbox"/>	<b>sad_rosalind</b> 46103e4dae0b	<a href="#">ubuntu</a>	Running		2 seconds ago	