Assignment-1

Q1. 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.
Ans: -
<u>Git Stash:</u>

Git stash stores the uncommitted changes and overlooks untracked and ignored files. When we want to record the current state of working directory, but want to go back to clean working directory. The command saves your local modifications away and reverts from the working directory to match the head commit.

In this stash we have some types like: -

- git stash
- git stash push -m “message”
- git stash apply –index <index_no>
- git stash list
- git stash pop
- git stash drop
- git stash clear

1. **git stash** is used to save the changes without any message.
2. **git stash push -m “message”** is used to save the changes with a message.
3. **git stash list** is used to list of stashes in the directory.
4. **git stash apply -index <index_no>** is used to re-apply the changes that we stashed by using git stash command.
5. **git stash pop** removes the changes from stash and applies them to your working file and it returns the message in the file.
6. **git stash drop** is used to delete a stash from the queue.
7. **git stash clear** allows deleting all the available stashes at once.

Firstly, we create three files with name file1.txt, file2.txt and file3.txt and add the files by using the **git add .** and commit the changes by using **git commit -m “commit_message”.** After that modify the file and add the file using **git add <filename>** and used the **git stash push -m “message”** and repeat the process for remaining files.

After creating stashes I display the list of stashes in the working directory using **git stash list** after that I used the **git stash apply -index <index_no>, git stash pop, git stash drop, git stash clear.**

MINGW64:/c/Users/durga/OneDrive/Desktop/Iswaryagitass

```
durga@LAPTOP-4G7C1UBO MINGW64 ~ (master)
$ cd /c/Users/durga/OneDrive/Desktop/Iswaryagitass

durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswaryagitass (master)
$ git init
Initialized empty Git repository in C:/Users/durga/OneDrive/Desktop/Iswaryagitas
s/.git/

durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswaryagitass (master)
$ git config user.name "iswarya"

durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswaryagitass (master)
$ git config list
error: key does not contain a section: list

durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswaryagitass (master)
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager-core
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.email=ishutirumalaraju@gmail
user.name=iswarya
core.repositoryformatversion=0
core.filemode=false
core.bare=false
core.logallrefupdates=true
core.symlinks=false
core.ignorecase=true
user.name=iswarya

durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswaryagitass (master)
$
```

```
durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya (master)
$ vi file1.txt

durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya (master)
$ vi file2.txt

durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya (master)
$ vi file3.txt

durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya (master)
$ git add .
warning: LF will be replaced by CRLF in file1.txt.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in file2.txt.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in file3.txt.
The file will have its original line endings in your working directory

durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya (master)
$ git commit -m "commit1"
[master (root-commit) 2188968] commit1
 3 files changed, 3 insertions(+)
 create mode 100644 file1.txt
 create mode 100644 file2.txt
 create mode 100644 file3.txt

durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya (master)
$ vi file1.txt

durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya (master)
$ git add file1.txt
warning: LF will be replaced by CRLF in file1.txt.
The file will have its original line endings in your working directory

durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya (master)
$ git stash push -m "First stash"
Saved working directory and index state On master: First stash

durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya (master)
$ vi file2.txt
```

```
durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya (master)
$ vi file2.txt

durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya (master)
$ git add file2.txt
warning: LF will be replaced by CRLF in file2.txt.
The file will have its original line endings in your working directory

durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya (master)
$ git stash push -m "Second stash"
Saved working directory and index state On master: Second stash

durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya (master)
$ vi file3.txt

durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya (master)
$ git stash push -m "Third stash"
warning: LF will be replaced by CRLF in file3.txt.
The file will have its original line endings in your working directory
Saved working directory and index state On master: Third stash

durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya (master)
$ git stash list
stash@{0}: On master: Third stash
stash@{1}: On master: Second stash
stash@{2}: On master: First stash

durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya (master)
$ git stash apply --index 1
<stdin>:9: new blank line at EOF.
+
warning: 1 line adds whitespace errors.
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   file2.txt


durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya (master)
$ cat file2.txt
Hello

stash2


durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya (master)
$ git stash pop
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   file2.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:   file3.txt

Dropped refs/stash@{0} (32b76c34f61856796e93f8061662768039d47653)

durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya (master)
$ git stash list
stash@{0}: On master: Second stash
stash@{1}: On master: First stash

durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya (master)
$ git stash drop
Dropped refs/stash@{0} (cda61c4c72d4ae53dfebc6283e42b92cafb6a6ed)

durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya (master)
$ git stash list
stash@{0}: On master: First stash

durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya (master)
$ git stash clear

durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya (master)
$ git stash list

durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya (master)
$
```

Q2. 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.

Ans: -

<u>Git fetch:</u>

Git fetch downloads commits, objects and refers from another repository. It fetches branches and tags from one or more repositories. It holds repositories along with the objects that are necessary to complete their historied to keep updated remote-tracking branches.

Firstly, we clone the repository from github to git using **git clone "path"** then check the commits using **git log or git log –oneline**. Then modify the file in github and save the changes and check the commits another time.

```
durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya (master)
$ git clone https://github.com/iswarya12346/ishu.git
Cloning into 'ishu'...
remote: Enumerating objects: 20, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 20 (delta 1), reused 0 (delta 0), pack-reused 11
Receiving objects: 100% (20/20), done.
Resolving deltas: 100% (5/5), done.

durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya (master)
$ cd ishu

durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya/ishu (student)
$ git log --oneline
38b30f7 (HEAD -> student, origin/student, origin/HEAD) Merge pull request #2 from iswarya12346/PROJ-4-Yamuna-Explain
40e8f64 (origin/PROJ-4-Yamuna-Explain) Create yam.txt
99cd5e2 Merge pull request #1 from iswarya12346/PROJ-3-Testing
3c08f9e (origin/PROJ-3-Testing) Update s.txt
c8cc2b3 add
b453b32 added s file to student branch
20d4d6e (origin/master) rm c.txt
34eae9b second file
ceb467e add two files

durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya/ishu (student)
$ ls
b.txt  filea.txt  s.txt  yam.txt

durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya/ishu (student)
$ git fetch
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), 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), 716 bytes | 65.00 KiB/s, done.
From https://github.com/iswarya12346/ishu
   38b30f7..fcf44a6  student    -> origin/student
```

```
durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya/ishu (student)
$ git log --oneline
38b30f7 (HEAD -> student) Merge pull request #2 from iswarya12346/PROJ-4-Yamuna-Explain
40e8f64 (origin/PROJ-4-Yamuna-Explain) Create yam.txt
99cd5e2 Merge pull request #1 from iswarya12346/PROJ-3-Testing
3c08f9e (origin/PROJ-3-Testing) Update s.txt
c8cc2b3 add
b453b32 added s file to student branch
20d4d6e (origin/master) rm c.txt
34eae9b second file
ceb467e add two files

durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya/ishu (student)
$ git log origin/student
commit fcf44a6787664f794dd86937715c7072da286b98 (origin/student, origin/HEAD)
Author: iswarya12346 <84374289+iswarya12346@users.noreply.github.com>
Date:   Fri Feb 17 15:27:57 2023 +0530

    Update b.txt

commit 38b30f739511d873bef9fbd20dce7c1abb66140a (HEAD -> student)
Merge: 99cd5e2 40e8f64
Author: iswarya12346 <84374289+iswarya12346@users.noreply.github.com>
Date:   Tue Feb 7 10:50:31 2023 +0530

    Merge pull request #2 from iswarya12346/PROJ-4-Yamuna-Explain

    PROJ-4-Yamuna-Explain

commit 40e8f64156cc22294be2379975268ff0d793a999 (origin/PROJ-4-Yamuna-Explain)
Author: iswarya12346 <84374289+iswarya12346@users.noreply.github.com>
Date:   Tue Feb 7 10:49:56 2023 +0530

    Create yam.txt

    git commit -m "PROJ-4 <message>"

commit 99cd5e283b92d7040e557cc33425f59010c37038
Merge: c8cc2b3 3c08f9e
Author: iswarya12346 <84374289+iswarya12346@users.noreply.github.com>
Date:   Tue Feb 7 10:32:11 2023 +0530

    Merge pull request #1 from iswarya12346/PROJ-3-Testing

    PROJ-3-Testing

commit 3c08f9e569c836b1a0ceae120f10cebab5fcf2d1 (origin/PROJ-3-Testing)
Author: iswarya12346 <84374289+iswarya12346@users.noreply.github.com>

    Update s.txt

    git commit -m "PROJ-3 <message>"

commit c8cc2b3ac166e66927ed1099db3eea9c740ff38d
Author: iswarya12346 <84374289+iswarya12346@users.noreply.github.com>
Date:   Tue Nov 23 15:44:41 2021 +0530

    add

commit b453b32a7b2b349f4a207048dd6d364da3e4dba2
Author: iswarya12346 <ishutirumalaraju@gmail.com>
```

Git merge:

A git merge operation is performed by running the command **git merge <name of the branch>** and the command will merge the specified commit to a specified branch by passing in the branch name in commit.

```
durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya/ishu (student)
$ git merge origin/student
Updating 38b30f7..fcf44a6
Fast-forward
 b.txt | 1 +
 1 file changed, 1 insertion(+)

durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya/ishu (student)
$ git log --oneline
fcf44a6 (HEAD -> student, origin/student, origin/HEAD) Update b.txt
38b30f7 Merge pull request #2 from iswarya12346/PROJ-4-Yamuna-Explain
40e8f64 (origin/PROJ-4-Yamuna-Explain) Create yam.txt
99cd5e2 Merge pull request #1 from iswarya12346/PROJ-3-Testing
3c08f9e (origin/PROJ-3-Testing) Update s.txt
c8cc2b3 add
b453b32 added s file to student branch
20d4d6e (origin/master) rm c.txt
34eae9b second file
ceb467e add two files
```

Q3. 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.

Ans: -

<u>Git fetch:</u>

Git fetch downloads commits, objects and refers from another repository. It fetches branches and tags from one or more repositories. It holds repositories along with the objects that are necessary to complete their historied to keep updated remote-tracking branches.

Firstly, we clone the repository from github to git using **git clone "path"** then check the commits using **git log or git log –oneline**. Then modify the file in github and save the changes and check the commits another time.

```
durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya (master)
$ git clone https://github.com/iswarya12346/ishu.git
Cloning into 'ishu'...
remote: Enumerating objects: 20, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 20 (delta 1), reused 0 (delta 0), pack-reused 11
Receiving objects: 100% (20/20), done.
Resolving deltas: 100% (5/5), done.

durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya (master)
$ cd ishu

durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya/ishu (student)
$ git log --oneline
38b30f7 (HEAD -> student, origin/student, origin/HEAD) Merge pull request #2 from iswarya12346/PROJ-4-Yamuna-Explain
40e8f64 (origin/PROJ-4-Yamuna-Explain) Create yam.txt
99cd5e2 Merge pull request #1 from iswarya12346/PROJ-3-Testing
3c08f9e (origin/PROJ-3-Testing) Update s.txt
c8cc2b3 add
b453b32 added s file to student branch
20d4d6e (origin/master) rm c.txt
34eae9b second file
ceb467e add two files

durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya/ishu (student)
$ ls
b.txt  filea.txt  s.txt  yam.txt

durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya/ishu (student)
$ git fetch
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), 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), 716 bytes | 65.00 KiB/s, done.
From https://github.com/iswarya12346/ishu
   38b30f7..fcf44a6  student    -> origin/student
```

```
durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya/ishu (student)
$ git log --oneline
38b30f7 (HEAD -> student) Merge pull request #2 from iswarya12346/PROJ-4-Yamuna-Explain
40e8f64 (origin/PROJ-4-Yamuna-Explain) Create yam.txt
99cd5e2 Merge pull request #1 from iswarya12346/PROJ-3-Testing
3c08f9e (origin/PROJ-3-Testing) Update s.txt
c8cc2b3 add
b453b32 added s file to student branch
20d4d6e (origin/master) rm c.txt
34eae9b second file
ceb467e add two files

durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya/ishu (student)
$ git log origin/student
commit fcf44a6787664f794dd86937715c7072da286b98 (origin/student, origin/HEAD)
Author: iswarya12346 <84374289+iswarya12346@users.noreply.github.com>
Date:   Fri Feb 17 15:27:57 2023 +0530

    Update b.txt

commit 38b30f739511d873bef9fbd20dce7c1abb66140a (HEAD -> student)
Merge: 99cd5e2 40e8f64
Author: iswarya12346 <84374289+iswarya12346@users.noreply.github.com>
Date:   Tue Feb 7 10:50:31 2023 +0530

    Merge pull request #2 from iswarya12346/PROJ-4-Yamuna-Explain

    PROJ-4-Yamuna-Explain

commit 40e8f64156cc22294be2379975268ff0d793a999 (origin/PROJ-4-Yamuna-Explain)
Author: iswarya12346 <84374289+iswarya12346@users.noreply.github.com>
Date:   Tue Feb 7 10:49:56 2023 +0530

    Create yam.txt

    git commit -m "PROJ-4 <message>"

commit 99cd5e283b92d7040e557cc33425f59010c37038
Merge: c8cc2b3 3c08f9e
Author: iswarya12346 <84374289+iswarya12346@users.noreply.github.com>
Date:   Tue Feb 7 10:32:11 2023 +0530

    Merge pull request #1 from iswarya12346/PROJ-3-Testing

    PROJ-3-Testing

commit 3c08f9e569c836b1a0ceae120f10cebab5fcf2d1 (origin/PROJ-3-Testing)
Author: iswarya12346 <84374289+iswarya12346@users.noreply.github.com>

    Update s.txt

    git commit -m "PROJ-3 <message>"

commit c8cc2b3ac166e66927ed1099db3eea9c740ff38d
Author: iswarya12346 <84374289+iswarya12346@users.noreply.github.com>
Date:   Tue Nov 23 15:44:41 2021 +0530

    add

commit b453b32a7b2b349f4a207048dd6d364da3e4dba2
Author: iswarya12346 <ishutirumalaraju@gmail.com>
```

Git Pull:

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.

```
durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya/ishu (student)
$ git pull
Already up to date.

durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya/ishu (student)
$ git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), 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), 737 bytes | 52.00 KiB/s, done.
From https://github.com/iswarya12346/ishu
   fcf44a6..614a1ea  student    -> origin/student
Updating fcf44a6..614a1ea
Fast-forward
 s.txt | 1 +
 1 file changed, 1 insertion(+)

durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya/ishu (student)
$ git log --oneline
614a1ea (HEAD -> student, origin/student, origin/HEAD) Update s.txt
fcf44a6 Update b.txt
38b30f7 Merge pull request #2 from iswarya12346/PROJ-4-Yamuna-Explain
40e8f64 (origin/PROJ-4-Yamuna-Explain) Create yam.txt
99cd5e2 Merge pull request #1 from iswarya12346/PROJ-3-Testing
3c08f9e (origin/PROJ-3-Testing) Update s.txt
c8cc2b3 add
b453b32 added s file to student branch
20d4d6e (origin/master) rm c.txt
34eae9b second file
ceb467e add two files
```

Difference between git fetch and git pull:

- 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 faster as you're performing multiple actions in one – a better bang for your buck. Using the Git pull command can be seen in one light as a feature of convenience; you're probably less worried about introducing conflicts into your local repo and you just want the most up-to-date changes from the remote branch you're pulling from.
- 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.

Q4. 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.

Ans: -

<u>Awk command:</u>

Awk is a scripting language used for manipulating data and generating reports. The awk command programming language requires no compiling and allows the user to use variables, numeric functions, string functions, and logical operators.

Syntax: awk '{ print "message" }'

Awk is a utility that enables a programmer to write tiny but effective programs in the form of statements that define text patterns that are to be searched for in each line of a document and the action that is to be taken when a match is found within a line.

```
durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya/ishu (student)
$ awk '{ print "This is Iswarya"}'

This is Iswarya

This is Iswarya

This is Iswarya
```

```
durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya/ishu (student)
$ awk '{ print "Hello World!!!"}'

Hello World!!!

Hello World!!!

Hello World!!!

Hello World!!!

Hello World!!!

Hello World!!!
```

find out the prime number from the range 1 to 20 using shell script

```
durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya/ishu (student)
$ vi prime.sh
```

MINGW64:/c/Users/durga/OneDrive/Desktop/Iswarya/ishu

```
#!/bin/bash

echo "Prime numbers in the range of 1 to 20 are:"

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

```
prime.sh [unix] (18:13 17/02/2023)                    21,0-1 All
:wq
```

```
durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya/ishu (student)
$ sh prime.sh
Prime numbers in the range of 1 to 20 are:
1
2
3
5
7
11
13
17
19
```

History:

History command used to show the history of the commands which we are executed until now.

MINGW64:/c/Users/durga/OneDrive/Desktop/Iswarya/ishu

```
durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya/ishu (student)
$ history
    1  pwd
    2  touch index.html
    3  git init
    4  touch file.txt
    5  ls
    6  git add .
    7  git status
    8  git config user.email "ishutirumalaraju@gmail.com"
    9  git config user.name "iswarya12346"
   10  git commit -m "add files"
   11  git log
   12  git remote add origin https://github.com/iswarya12346/samplegitproject12.git
   13  git remote
   14  git branch
   15  git push origin master
   16  git pull origin
   17  git pull origin master
   18  pwd
   19  touch filea.txt
   20  touch b.txt
   21  ls -a
   22  git init
   23  ls -a
   24  git add .
   25  git status
   26  git commit -m "added two files"
   27  git config user.name "iswarya12346"
   28  git config user.email "ishutirumalaraju@gmail.com"
   29  git log
   30  git commit -m "add two files"
   31  git log
   32  touch c.txt
   33  git add .
   34  git commit -m "second file"
   35  git log
   36  git log --oneline
   37  git checkout ceb467e
   38  git log
   39  git checkout master
   40  git rm --cached c.txt
   41  git commit -m "rm c.txt"
   42  git log
   43  git push origin master
   44  git branch student
   45  git branch
   46  git checkout student
   47  git branch
   48  touch s.txt
   49  ls -a
```

MINGW64:/c/Users/durga/OneDrive/Desktop/Iswarya/ishu

```
 50  git add .
 51  git status
 52  rm c.txt
 53  ls
 54  git log
 55  git add .
 56  git status
 57  git commit -m "added s file to student branch"
 58  git log --oneline
 59  git remote add origin https://github.com/iswarya12346/ishu.git
 60  git remote
 61  git branch
 62  git push origin student
 63  ls
 64  git checkout master
 65  git push origin master
 66  git merge student master
 67  git pull origin
 68  git pull origin master
 69  git remote
 70  git pull origin
 71  git branch
 72  git pull origin
 73  git pull origin master
 74  git pull origin student
 75  ls -la
 76  git status
 77  cd desktop
 78  cd Desktop
 79  mkdir gitcommands
 80  cd gitcommands
 81  cd ..
 82  cd github
 83  cd gitcommands
 84  ls -al ~/.ssh
 85  ssh-keygen -t ed25519 -C "ishutirumalaraju@gmail.com"
 86  ssh-keygen -t ed25519 -C "ishutirumalaraju@gmail.com"
 87  clear
 88  ssh-keygen -p -f ~/.ssh/id_ed25519
 89  ssh-keygen -t ed25519 -C "ishutirumalaraju@gmail.com"
 90  eval "$(ssh-agent -s)"
 91  ssh-add ~/.ssh/id_ed25519
 92  ls -al ~/.ssh
 93  clip < ~/.ssh/id_ed25519.pub
 94  ls -al ~/.ssh
 95  ls
 96  clear
 97  ssh -T git @github.com
 98  ssh -T git@github.com
 99  git commit -m "PROJ-3 commit1"
100  git config --list
101  git config user.name "ishutirumalaraju@gmail.com"
```

MINGW64:/c/Users/durga/OneDrive/Desktop/Iswarya/ishu

```
102  clear
103  git config user.name "Iswarya"
104  git config user.email "ishutirumalaraju@gmail.com"
105  git config --list
106  git init
107  git clean -n
108  vim file2.txt
109  git clean -n
110  git status
111  git add .
112  git commit -m "commit6"
113  vim file7.txt
114  git status
115  git clean -n
116  git clean -f file7.txt
117  git create -b bran
118  git branch -a
119  git branch
120  git branch -a ish
121  cd Desktop
122  cd /c/Users/durga/OneDrive/Desktop/gitass
123  clear
124  git status
125  cd C:\Users\durga\OneDrive\Desktop\gitasses
126  cd /c/Users/durga/OneDrive/Desktop/gitasses
127  git init
128  clear
129  git config user.name "Iswarya06"
130  git config user.email "ishutirumalaraju@gmail.com"
131  git config --list
132  git status
133  git add .
134  git commit -m "commit1"
135  git status
136  git status
137  git stash
138  git status
139  git stash list
140  git stash push -m "firststash"
141  git stash list
142  git stash push -m "secondstash"
143  git stash list
144  git status
145  git stash show
146  git stash apply --index 1
147  git stash pop
148  git stash list
149  git commit -m "commit2"
150  git add .
151  git commit -m "commit2"
152  git log --oneline
153  git stash apply --index 1
```

MINGW64:/c/Users/durga/OneDrive/Desktop/Iswarya/ishu

```
154  git stash pop
155  git stash list
156  /c/Users/durga/OneDrive/Desktop/gitass
157  cd /c/Users/durga/OneDrive/Desktop/gitass
158  git init
159  git config --list
160  cd /c/Users/durga/OneDrive/Desktop/Iswaryagit ass
161  cd /c/Users/durga/OneDrive/Desktop/Iswaryagit ass
162  cd /c/Users/durga/OneDrive/Desktop/Iswaryagitass
163  git init
164  git config user.name "Iswarya06"
165  git config user.email "ishutirumalaraju@gmail.com"
166  git config --list
167  cat>stashing.txt
168  git add .
169  git commit -m "commit1"
170  git status
171  cat stashing.txt
172  cat>>stashing.txt
173  git status
174  git stash push -m "Stash1"
175  git stash push -m "Stash2"
176  git stash list
177  git stash push -m "Stash3"
178  git stash list
179  git stash apply --index 2
180  git stash pop
181  git stash list
182  git stash
183  git stash list
184  git stash apply --index 2
185  git stash pop
186  git status
187  git add .
188  git status
189  git stash apply --index 2
190  git status pop
191  git stash pop
192  git stash list
193  git stash apply --index 2
194  git stash pop
195  git stash apply --index 2
196  git add .
197  git stash apply --index 2
198  git stash apply --index 2
199  git stash pop
200  git stash list
201  git stash list
202  git stash apply --index 2
203  git add .
204  git stash apply --index 2
205  git commit -m "commit2"
```

MINGW64:/c/Users/durga/OneDrive/Desktop/Iswarya/ishu

```
205  git commit -m "commit2"
206  git stash apply --index 2
207  git stash apply --index 2
208  git stash apply --index 2
209  git stash pop
210  git stash list
211  git stash pop
212  cd /c/Users/durga/OneDrive/Desktop/Iswaryagitass
213  git init
214  git config user.name "Iswarya0610"
215  git stash push -m "stash1"
216  git add .
217  git stash push -m "stash1"
218  git commit -m "commit1"
219  git log --oneline
220  git status
221  git stash push -m "stash1"
222  git stash push -m "stash2"
223  git stash push -m "stash3"
224  git stash list
225  git stash apply --index 2
226  cat sample
227  cat sample.txt
228  git stash apply --index 1
229  cd /c/Users/durga/OneDrive/Desktop/Iswarya
230  git init
231  git config user.name "iswarya0610
232  "
233  git add .
234  git status
235  git commit -m "commit1"
236  git log --oneline
237  git stash push -m "stash1"
238  git stash list
239  cat sample.txt
240  git add sample.txt
241  cat sample.txt
242  git stash push -m "stash2"
243  cat sample.txt
244  git add sample.txt
245  git stash list
246  git stash push -m "stash2"
247  cat sample.txt
248  git stash list
249  git stash apply --index 0
250  cat sample.txt
251  git add sample.txt
252  git stash push -m "stash3"
253  git stash apply --index 0
254  git add sample.txt
255  cat sample.txt
256  git stash pop
```

MINGW64:/c/Users/durga/OneDrive/Desktop/Iswarya/ishu

```
257  git stash list
258  git stash pop
259  git status
260  git add sample.txt
261  git stash pop
262  git stash pop
263  cd /c/Users/durga/OneDrive/Desktop/Iswaryagitass
264  git init
265  git config user.name "iswarya"
266  git config list
267  git config --list
268  git status
269  git add .
270  git commit -m "commit1"
271  git add sample.txt
272  git stash push -m "stash1"
273  ls
274  vi file1.txt
275  vi file2.txt
276  vi file3.txt
277  git add .
278  git commit -m "commit1"
279  vi file1.txt
280  git add file1.txt
281  git stash push -m "First stash"
282  vi file2.txt
283  git add file2.txt
284  git stash push -m "Second stash"
285  vi file3.txt
286  git stash push -m "Third stash"
287  git stash list
288  git stash apply --index 1
289  git init
290  vi file1.txt
291  vi file2.txt
292  vi file3.txt
293  git add .
294  git commit -m "commit1"
295  vi file1.txt
296  git add file1.txt
297  git stash push -m "First stash"
298  vi file2.txt
299  git add file2.txt
300  git stash push -m "Second stash"
301  vi file3.txt
302  git stash push -m "Third stash"
303  git stash list
304  git stash apply --index 1
305  cat file2.txt
306  git stash pop
307  git stash list
308  git stash drop
```

```
309  git stash list
310  git stash clear
311  git stash list
312  git clone
313  $ git stash list
314  durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya (master)
315  $
316  git clone
317  $ git stash list
318  durga@LAPTOP-4G7C1UBO MINGW64 ~/OneDrive/Desktop/Iswarya (master)
319  $
320  git clone https://github.com/iswarya12346/Iswaryagitass.git
321  cd Iswaryagitass
322  git clone https://github.com/iswarya12346/ishu.git
323  git clone https://github.com/iswarya12346/ishu.git
324  cd ishu
325  git log --oneline
326  ls
327  git fetch
328  git log --oneline
329  git log origin/student
330  git merge origin/student
331  git log --oneline
332  git pull
333  git pull
334  git log --oneline
335  awk '{ print "This is Iswarya"}'
336  awk '{ print "Hello World!!!"}'
337  vi prime.sh
338  @noonex chmod prime.sh
339  noonex chmod prime.sh
340  sh prime.sh
341  vi prime.sh
342  vi prime.sh
343  sh prime.sh
344  vi prime.sh
345  sh prime.sh
346  vi prime.sh
347  sh prime.sh
348  git history
349  history
```

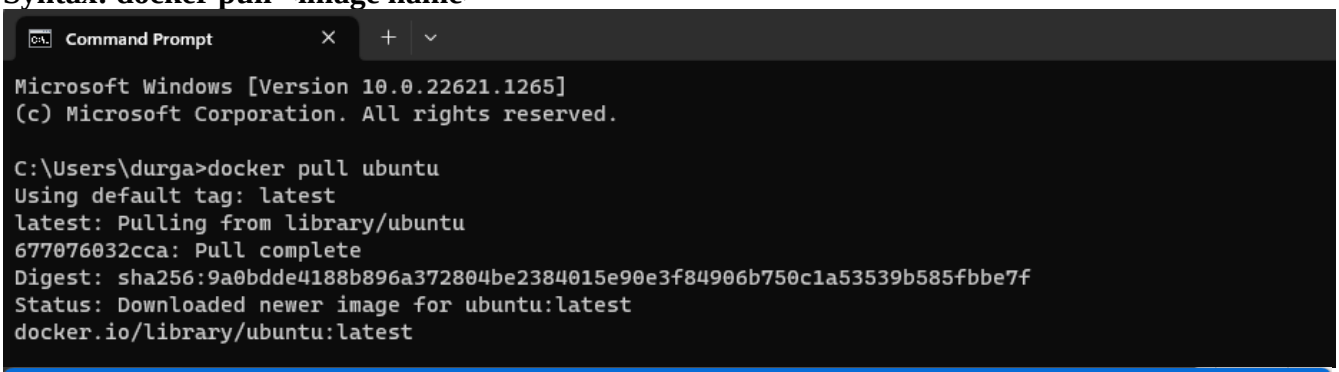
Q5. Set up a container and run an Ubuntu operating system. For this purpose, you can make use of the docker hub and run the container in interactive mode.

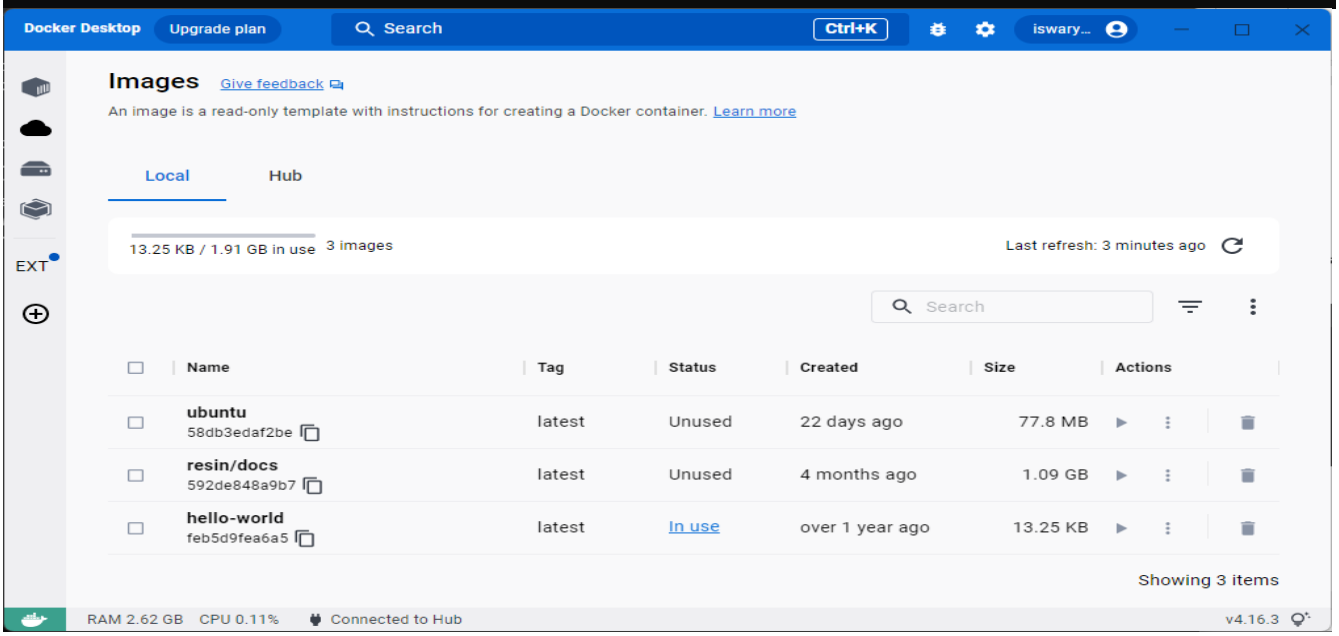
DOCKER

- It is an application.
- It is a tool of DevOps for deployment
- It is used containerization process.
- Containerization is a software deployment process that bundles an application's code with all the files and libraries it needs to run on any infrastructure. Traditionally, to run any application on your computer, you had to install the version that matched your machine's operating system.
- Android application builds with java and xml
- Packing all different interfaces of our application into one container or folder(containerization).
- Dock means attaching something. It is used in shipyards.
- Docker is an open source platform that enables developers to build, deploy, run, update and manage containerized applications.
- It is used to uploaded the container into server.
- Docker is used to transfer our container into server.
- It allows applications to use the same linux kernel as a system on the host computer,rather than creating a whole virtual os
- It uses container on the host's os to run application.
- Containers ensure that our application works in any environment like developmeny,test, or production.
- Docker includes components the Docker client,server,machine,hub.
- It will have virtual run environment.

By using docker pull command we create or download the image and it gives the status of the image.

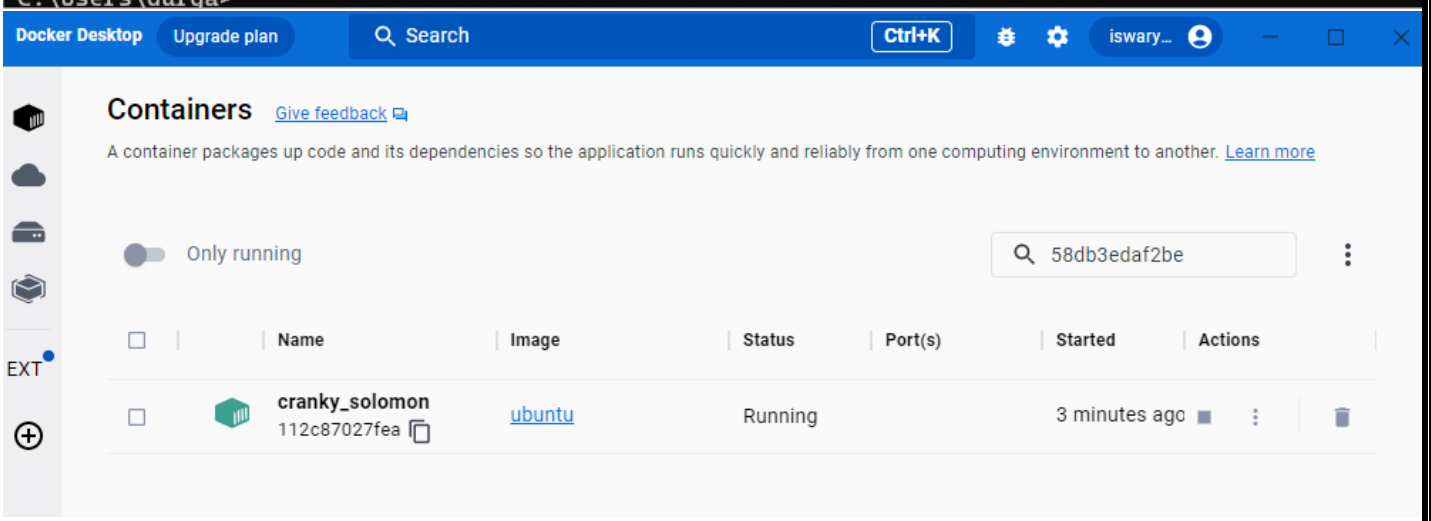
Syntax: docker pull <image name>

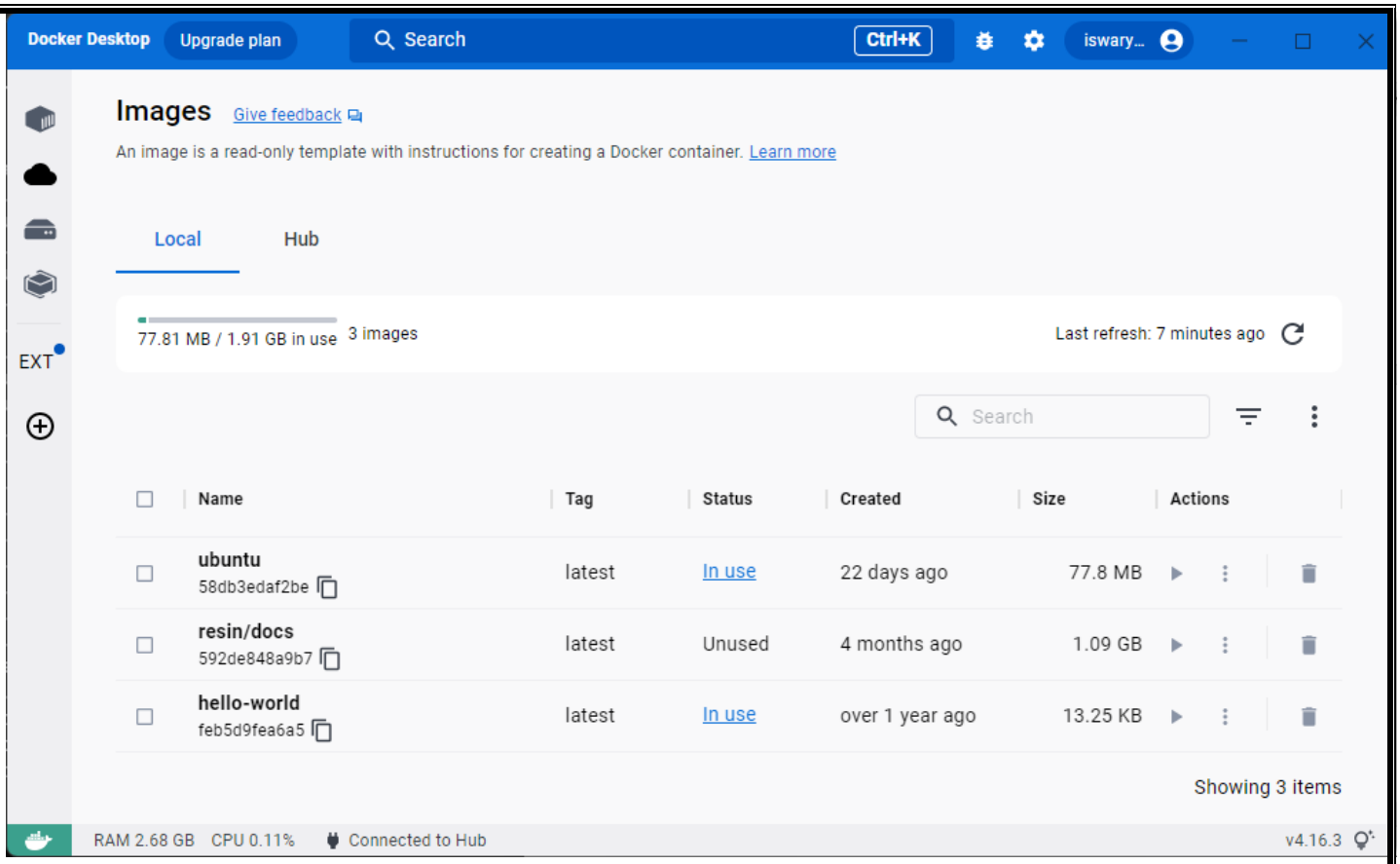




After downloading the image we create container and run the container in interactive mode by using the command **docker run -it ubuntu.**

```
C:\Users\durga>docker run -it ubuntu
root@112c87027fea:/# apt update
Get:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy InRelease [270 kB]
Get:3 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [807 kB]
Get:4 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [5557 B]
Get:6 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [752 kB]
Get:7 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [860 kB]
Get:8 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [107 kB]
Get:9 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [266 kB]
Get:10 http://archive.ubuntu.com/ubuntu jammy/restricted amd64 Packages [164 kB]
Get:11 http://archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [17.5 MB]
Get:12 http://archive.ubuntu.com/ubuntu jammy/main amd64 Packages [1792 kB]
Get:13 http://archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [808 kB]
Get:14 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1091 kB]
Get:15 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1136 kB]
Get:16 http://archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [10.9 kB]
Get:17 http://archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [49.0 kB]
Get:18 http://archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [22.4 kB]
Fetched 25.8 MB in 46s (557 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
6 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@112c87027fea:/#
root@112c87027fea:/#
C:\Users\durga>
```






```
C:\Users\durga>docker ps -all
CONTAINER ID   IMAGE     COMMAND       CREATED          STATUS          PORTS     NAMES
112c87027fea   ubuntu    "/bin/bash"   3 minutes ago    Up 3 minutes              cranky_solomon
```