DevOps Assignment

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.

Git Stash command: The git stash command enables you to switch branches without committing the current branch.

Git stash command helps in saving the previously written code and then goes back to the last commit for a fresh start. Now we can add the new feature without disturbing the old one as it is saved locally saved.

After committing the new feature you can go on working with the old one which was incomplete and not committed.

```
Poojitha@DESKTOP-CFKNIRE MINGW64 ~ (hello-world)
$ git config --global user.name "saipoojitha28"

Poojitha@DESKTOP-CFKNIRE MINGW64 ~ (hello-world)
$ git config --global user.email "20a91a05e2@aec.edu.in"

Poojitha@DESKTOP-CFKNIRE MINGW64 ~ (hello-world)
$ git clone "https://github.com/poojitha2803/HeroViredAssignment.git"
fatal: destination path 'HeroViredAssignment' already exists and is not an empty directory.

Poojitha@DESKTOP-CFKNIRE MINGW64 ~ (hello-world)
$ cd HeroViredAssignment
```

A new directory is created with the name HeroViredAssignment.

The present working directory is changed to the new directory.

```
Poojitha@DESKTOP-CFKNIRE MINGW64 ~/HeroViredAssignment (main)
$ vi newfile.txt
```

New File is created with the name newfile.txt.

Adding the content into the file makes the changes to the file and gets saved.

```
Hello World
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
```

```
Poojitha@DESKTOP-CFKNIRE MINGW64 ~/HeroViredAssignment (main)
$ git status
On branch main
Your branch is based on 'origin/main', but the upstream is gone.
  (use "git branch --unset-upstream" to fixup)

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

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

Poojitha@DESKTOP-CFKNIRE MINGW64 ~/HeroViredAssignment (main)
$ git commit -m "Commit"
On branch main
Your branch is based on 'origin/main', but the upstream is gone.
  (use "git branch --unset-upstream" to fixup)

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

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

The file is now modified,and it is not committed,now if you want to pull the code on the other branch,then you have to remove these uncommitted changes,so use git stash command.

```
Poojitha@DESKTOP-CFKNIRE MINGW64 ~/HeroViredAssignment (main)
$ git status
On branch main
Your branch is based on 'origin/main', but the upstream is gone.
  (use "git branch --unset-upstream" to fixup)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   newfile.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:   newfile.txt
```

```
Poojitha@DESKTOP-CFKNIRE MINGW64 ~/HeroViredAssignment (main)
$ git stash
warning: in the working copy of 'newfile.txt', LF will be replaced by CRLF the next time Git touches it
Saved working directory and index state WIP on main: 9f3ebcd Commited

Poojitha@DESKTOP-CFKNIRE MINGW64 ~/HeroViredAssignment (main)
$ vi newfile.txt
```

```
Hello World
Hero Vired
~
~
~
~
~
~
~
~
~
~
~
~
~
~
```

The file is now stashed and it is under untracked state.

By default,running git stash will stash the changes that have been added to your index(staged changes)and unstages changes.To stash your untracked files,use git stash -u.

```
Poojitha@DESKTOP-CFKNIRE MINGW64 ~/HeroViredAssignment (main)
$ git status
On branch main
Your branch is based on 'origin/main', but the upstream is gone.
  (use "git branch --unset-upstream" to fixup)

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

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

Listing stashes: You can create multiple slashes and view them using git stash list command.

```
Poojitha@DESKTOP-CFKNIRE MINGW64 ~/HeroViredAssignment (main)
$ git stash list
stash@{0}: WIP on main: 9f3ebcd Commited
```

Providing additional message:

To provide more context to the stash we create the stash using the following command.

git stash save “message”

Getting back stashed changes:

You can reapply the previously stashed changes with the ‘git stash pop’ or ‘git stash apply’ command.

1.‘git stash pop’ removes the changes from stash and reapplies the changes in working copy,

2.‘git stash apply’ do not remove changes .but reapplies the changes in working copy.

```
Poojitha@DESKTOP-CFKNIRE MINGW64 ~/HeroViredAssignment (main)
$ git stash pop
error: The following untracked working tree files would be overwritten by merge:
        newfile.txt
Please move or remove them before you merge.
Aborting
On branch main
Your branch is based on 'origin/main', but the upstream is gone.
  (use "git branch --unset-upstream" to fixup)

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

nothing added to commit but untracked files present (use "git add" to track)
The stash entry is kept in case you need it again.
```

Now check whether stash is removed or not.

```
Poojitha@DESKTOP-CFKNIRE MINGW64 ~/HeroViredAssignment (main)
$ git stash list
stash@{0}: WIP on main: 9f3ebcd Commited
```

By using “git stash apply” We got the previous uncommitted changes.

```
Poojitha@DESKTOP-CFKNIRE MINGW64 ~/HeroViredAssignment (main)
$ git stash apply
error: The following untracked working tree files would be overwritten by merge:
        newfile.txt
Please move or remove them before you merge.
Aborting
On branch main
Your branch is based on 'origin/main', but the upstream is gone.
  (use "git branch --unset-upstream" to fixup)

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

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

To view the stash summary:

Git stash show is used to view the summary

```
Poojitha@DESKTOP-CFKNIRE MINGW64 ~/HeroViredAssignment (main)
$ git stash show
 newfile.txt | 3 +++
 1 file changed, 3 insertions(+)
```

Deleting stashes:

To delete a particular stash:

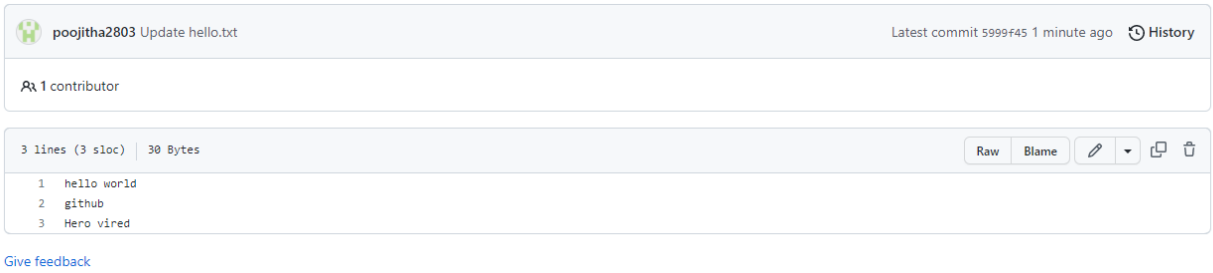
git stash drop stash

To delete all stashes at once,use the below comman

git stash clear

```
Poojitha@DESKTOP-CFKNIRE MINGW64 ~/HeroViredAssignment (main)
$ git stash clear

Poojitha@DESKTOP-CFKNIRE MINGW64 ~/HeroViredAssignment (main)
$ git stash list
```

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.




```
Poojitha@DESKTOP-CFKNIRE MINGW64 ~ (hello-world)
$ cd Herovired

Poojitha@DESKTOP-CFKNIRE MINGW64 ~/Herovired (main)
$ git add .

Poojitha@DESKTOP-CFKNIRE MINGW64 ~/Herovired (main)
$ git commit -m "commit"
[main 6e4b0f2] commit
 1 file changed, 2 insertions(+)
 create mode 100644 hello.txt

Poojitha@DESKTOP-CFKNIRE MINGW64 ~/Herovired (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

By using *git push* command we push the file hello into the github.

```
Poojitha@DESKTOP-CFKNIRE MINGW64 ~/Herovired (main)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 285 bytes | 142.00 KiB/s, done
.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/poojitha2803/Herovired.git
   fea0979..6e4b0f2  main -> main
```

Git log→ gives the changes which are made at particular time.

```
Poojitha@DESKTOP-CFKNIRE MINGW64 ~/Herovired (main)
$ git log
commit 6e4b0f2bd58e60456a3d7936f3a076b6f3cff3f3 (HEAD -> main, origin/m
ain)
Author: saipoojitha28 <20a91a05e2@aec.edu.in>
Date:   Fri Feb 17 10:09:29 2023 +0530

    commit

commit fea09799b36f9ff7950063915c25eb8ec4d06a7a
Author: saipoojitha28 <20a91a05e2@aec.edu.in>
Date:   Fri Feb 17 09:56:07 2023 +0530

    first python file
```

Changes can be made in the file in github directly and then commit.

Now to fetch the changes which are made in the file we use fetch command.

Git fetch

```
Poojitha@DESKTOP-CFKNIRE MINGW64 ~/Herovired (main)
$ 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), 683 bytes | 22.00 KiB/s, done.
From https://github.com/poojitha2803/Herovired
   6e4b0f2..5999f45  main       -> origin/main
```

```
Poojitha@DESKTOP-CFKNIRE MINGW64 ~/Herovired (main)
$ git log
commit 6e4b0f2bd58e60456a3d7936f3a076b6f3cff3f3 (HEAD -> main)
Author: saipoojitha28 <20a91a05e2@aec.edu.in>
Date:   Fri Feb 17 10:09:29 2023 +0530

    commit

commit fea09799b36f9ff7950063915c25eb8ec4d06a7a
Author: saipoojitha28 <20a91a05e2@aec.edu.in>
Date:   Fri Feb 17 09:56:07 2023 +0530

    first python file
```

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.

Git fetch: git fetch is a primary command used to download contents from a remote repository. git fetch is used in conjunction with git remote , git branch , git checkout , and git reset to update a local repository to the state of a remote. The git fetch command is a critical piece of collaborative git work flows.

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.

First, let's use the **git fetch** command. This command will retrieve any changes that have been made in the remote repository since the last time we fetched, but it will not merge those changes with our local repository:

```
$ git fetch
```

If there are any new changes in the remote repository, this command will download them to our local repository, but it will not update our working directory or merge them with our local changes.

Now, let's use the **git pull** command. This command will retrieve any changes that have been made in the remote repository and merge them with our local repository:

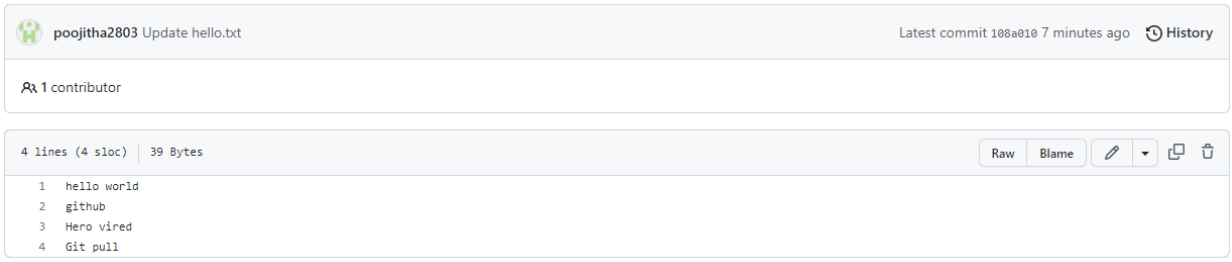
```
$ git pull
```

This command will download any changes from the "master" branch of the remote repository and merge them with our local repository. If there are any conflicts between our local changes and the changes from the remote repository, we will be prompted to resolve them.

So the main difference between **git fetch** and **git pull** is that git fetch downloads the changes from the remote repository but does not merge them with our local repository, while git pull downloads the changes from the remote repository and merges them with our local repository.

```
Poojitha@DESKTOP-CFKNIRE MINGW64 ~ (hello-world)
$ cd Herovired

Poojitha@DESKTOP-CFKNIRE MINGW64 ~/Herovired (main)
$ git add .

Poojitha@DESKTOP-CFKNIRE MINGW64 ~/Herovired (main)
$ git commit -m "commit"
[main 6e4b0f2] commit
 1 file changed, 2 insertions(+)
 create mode 100644 hello.txt

Poojitha@DESKTOP-CFKNIRE MINGW64 ~/Herovired (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

```
Poojitha@DESKTOP-CFKNIRE MINGW64 ~/Herovired (main)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 285 bytes | 142.00 KiB/s, done
.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/poojitha2803/Herovired.git
   fea0979..6e4b0f2  main -> main
```



```
Poojitha@DESKTOP-CFKNIRE MINGW64 ~/Herovired (main)
$ git log
commit 6e4b0f2bd58e60456a3d7936f3a076b6f3cff3f3 (HEAD -> main, origin/m
ain)
Author: saipoojitha28 <20a91a05e2@aec.edu.in>
Date:   Fri Feb 17 10:09:29 2023 +0530

    commit

commit fea09799b36f9ff7950063915c25eb8ec4d06a7a
Author: saipoojitha28 <20a91a05e2@aec.edu.in>
Date:   Fri Feb 17 09:56:07 2023 +0530

    first python file
```

```
Poojitha@DESKTOP-CFKNIRE MINGW64 ~/Herovired (main)
$ 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), 683 bytes | 22.00 KiB/s, done.
From https://github.com/poojitha2803/Herovired
   6e4b0f2..5999f45  main       -> origin/main
```

```
Poojitha@DESKTOP-CFKNIRE MINGW64 ~/Herovired (main)
$ cat hello.txt
hello world
github
Hero vired
```

```
Poojitha@DESKTOP-CFKNIRE MINGW64 ~/Herovired (main)
$ git add .

Poojitha@DESKTOP-CFKNIRE MINGW64 ~/Herovired (main)
$ git commit -m "commit"
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

Poojitha@DESKTOP-CFKNIRE MINGW64 ~/Herovired (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

```
Poojitha@DESKTOP-CFKNIRE MINGW64 ~/Herovired (main)
$ 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), 692 bytes | 23.00 KiB/s, done.
From https://github.com/poojitha2803/Herovired
   5999f45..108a010  main       -> origin/main

Poojitha@DESKTOP-CFKNIRE MINGW64 ~/Herovired (main)
$ cat hello.txt
hello world
github
Hero vired
```

```
Poojitha@DESKTOP-CFKNIRE MINGW64 ~/Herovired (main)
$ git pull
Updating 5999f45..108a010
Fast-forward
 hello.txt | 1 +
 1 file changed, 1 insertion(+)

Poojitha@DESKTOP-CFKNIRE MINGW64 ~/Herovired (main)
$ cat hello.txt
hello world
github
Hero vired
Git pull
```

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.

Awk command → Awk is a powerful tool used in Unix/Linux environments to manipulate and analyze text files. It allows you to easily extract and manipulate data from a file, and also perform more complex operations such as pattern matching and conditional statements.

```
Poojitha@DESKTOP-CFKNIRE MINGW64 ~ (hello-world)
$ vi dem

Poojitha@DESKTOP-CFKNIRE MINGW64 ~ (hello-world)
$ cat dem
Hello world
Gtihub
Awk command
Git fetch
Git commit
```

To print the content in the file we use the following commands.

```
Poojitha@DESKTOP-CFKNIRE MINGW64 ~ (hello-world)
$ awk '/world/ {print}' dem
Hello world

Poojitha@DESKTOP-CFKNIRE MINGW64 ~ (hello-world)
$ awk '{print $1,$2}' dem
Hello world
Gtihub
Awk command
Git fetch
Git commit
```

```
Poojitha@DESKTOP-CFKNIRE MINGW64 ~ (hello-world)
$ awk '{print $1}' dem
Hello
Gtihub
Awk
Git
Git


Poojitha@DESKTOP-CFKNIRE MINGW64 ~ (hello-world)
$ awk '{print $1,$NF}' dem
Hello world
Gtihub Gtihub
Awk command
Git fetch
Git commit
```

Creation of the bash script file.

Create a file with the extension .sh

Now type the code and save the file by giving the permissions like read, write and execute.

Now run the file by using the following command

Bash filename.sh

```
Poojitha@DESKTOP-CFKNIRE MINGW64 ~ (hello-world)
$ vi primenum.sh
```

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

  if $is_prime; then
    echo $i
  fi
done
```

```
Poojitha@DESKTOP-CFKNIRE MINGW64 ~ (hello-world)
$ bash primenum.sh
1
2
3
5
7
11
13
17
19
```

History command→ it gives what all the commands are being executed till now.

```
Poojitha@DESKTOP-CFKNIRE MINGW64 ~ (hello-world)
$ history
```

```
vi primenum.sh
bash primenum.sh
vi primenum.sh
history
```

Q5. Set up a container and run a Ubuntu operating system. For this purpose, you can make use of the docker hub and run the container in interactive mode.
All the processes pertaining to this should be provided in a screenshot for grading.

Once Docker is installed, open a terminal or command prompt and enter the following command to download the latest Ubuntu image from Docker Hub:

Docker pull ubuntu:latest

This command will download the latest version of the Ubuntu image and save it to your local machine.

Once the download is complete, you can run the container using the following command:

Docker run -it ubuntu:latest

This command will start the container in interactive mode and provide you with a command prompt where you can run Ubuntu commands.

Note that the **-it** flag specifies that the container should run in interactive mode with a TTY attached.

You should now see the command prompt for the Ubuntu container. You can run Ubuntu commands as you would on a regular Ubuntu system.

For example, you can enter **ls** to list the contents of the current directory or **apt-get update** to update the package list.

```
Poojitha@DESKTOP-CFKNIRE MINGW64 /
$ docker search ubuntu
NAME                                DESCRIPTION                                     STARS     OFFICIAL   AUT
OMATED
ubuntu                              Ubuntu is a Debian-based Linux operating sys…   15610     [OK]
websphere-liberty                   WebSphere Liberty multi-architecture images …   291       [OK]
ubuntu-upstart                      DEPRECATED, as is Upstart (find other proces…   112       [OK]
neurodebian                         NeuroDebian provides neuroscience research s…   98        [OK]
ubuntu/nginx                        Nginx, a high-performance reverse proxy & we…   77
open-liberty                        Open Liberty multi-architecture images based…   57        [OK]
ubuntu/apache2                      Apache, a secure & extensible open-source HT…   53
ubuntu-debootstrap                  DEPRECATED; use "ubuntu" instead                50        [OK]
ubuntu/squid                        Squid is a caching proxy for the Web. Long-t…   50
ubuntu/bind9                        BIND 9 is a very flexible, full-featured DNS…   46
ubuntu/mysql                        MySQL open source fast, stable, multi-thread…   41
ubuntu/prometheus                   Prometheus is a systems and service monitori…   36
ubuntu/postgres                     PostgreSQL is an open source object-relation…   23
ubuntu/kafka                        Apache Kafka, a distributed event streaming …   23
ubuntu/redis                        Redis, an open source key-value store. Long-…   16
ubuntu/prometheus-alertmanager      Alertmanager handles client alerts from Prom…   8
ubuntu/grafana                      Grafana, a feature rich metrics dashboard & …   7
ubuntu/dotnet-deps                  Chiselled Ubuntu for self-contained .NET & A…   6
ubuntu/zookeeper                    ZooKeeper maintains configuration informatio…   5
ubuntu/memcached                    Memcached, in-memory keyvalue store for smal…   5
ubuntu/dotnet-runtime               Chiselled Ubuntu runtime image for .NET apps…   5
ubuntu/telegraf                     Telegraf collects, processes, aggregates & w…   4
ubuntu/cortex                       Cortex provides storage for Prometheus. Long…   3
ubuntu/dotnet-aspnet                Chiselled Ubuntu runtime image for ASP.NET a…   3
ubuntu/cassandra                    Cassandra, an open source NoSQL distributed …   2
```

```
Poojitha@DESKTOP-CFKNIRE MINGW64 /
$ docker pull ubuntu: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
```

```
Poojitha@DESKTOP-CFKNIRE MINGW64 /
$ winpty docker run -it ubuntu:latest
root@afbae8515404:/# apt update
Get:1 http://archive.ubuntu.com/ubuntu jammy InRelease [270 kB]
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:4 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [107 kB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [5557 B]
Get:6 http://archive.ubuntu.com/ubuntu jammy/restricted amd64 Packages [164 kB]
Get:7 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [860 kB]
Get:8 http://archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [17.5 MB]
Get:9 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [807 kB]
Get:10 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [752 kB]
Get:11 http://archive.ubuntu.com/ubuntu jammy/main amd64 Packages [1792 kB]
Get:12 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [266 kB]
Get:13 http://archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [10.9 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/restricted amd64 Packages [808 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 1min 24s (307 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@afbae8515404:/#
```