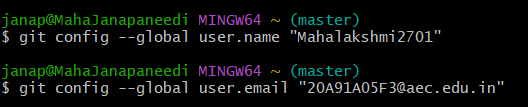
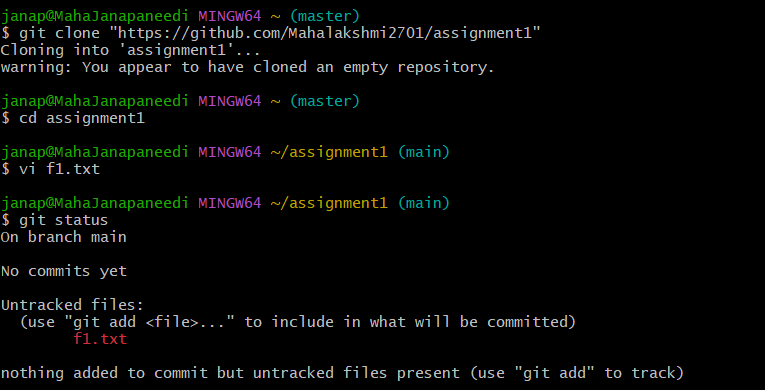
DevOps Assignment

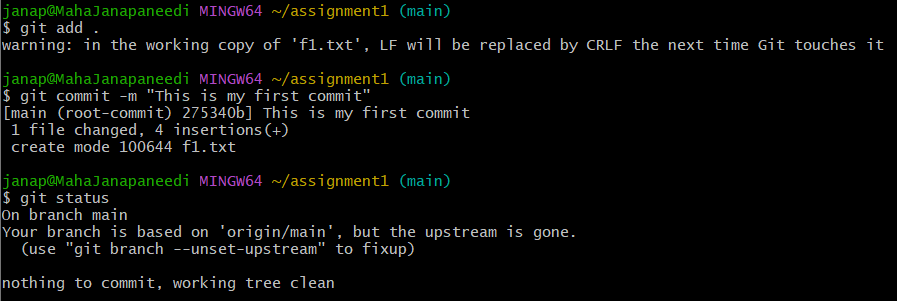
**Q1. Describe the usage of the git stash command by using an example and also state the process by giving the screenshot of all the commands written in git bash.**

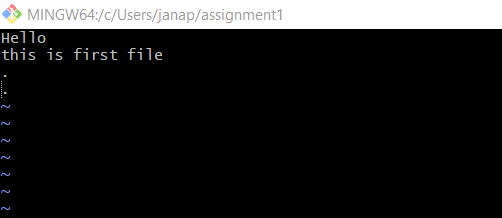
1. **Git Stash Command:**

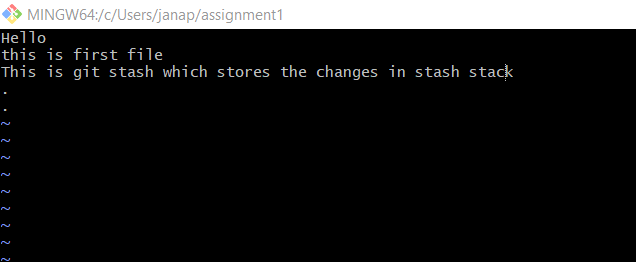
Git stash is a built-in command with the distributed Version control tool in Git that **locally stores all the most recent changes in a workspace and resets the state of the workspace to the prior commit state**.The Git stash command saves the previously written code and then goes back to the last commit for a fresh start. Now you can add the new feature without disturbing the old one as it is saved locally. After committing the new feature you can go on working with the old one which was incomplete and not committed.



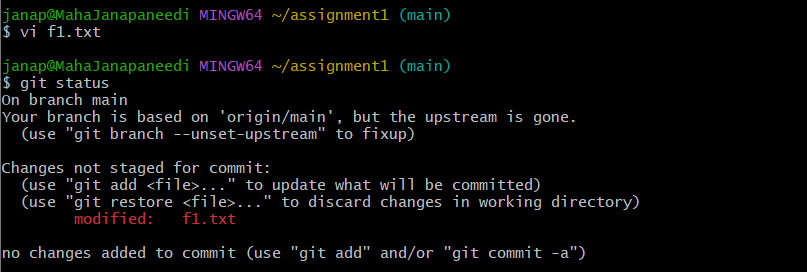




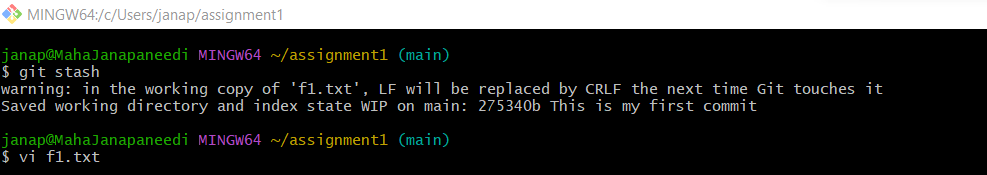


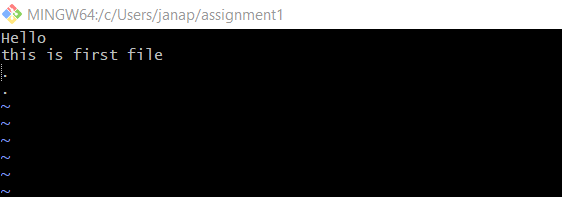


The file is now modified, and it is not committed, now if you want to pull the code on the other branch, then you have to remove these uncommitted changes, so use the git stash command.



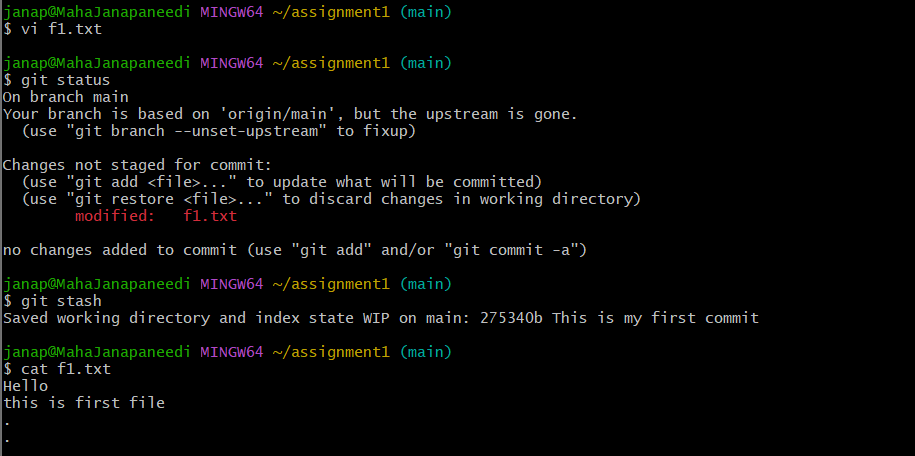
Now changes are removed.



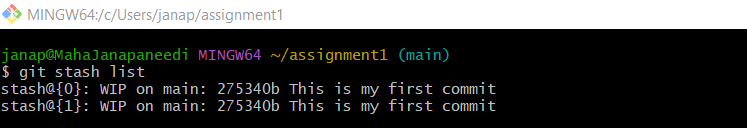


The file is now stashed and it is in an untracked state.

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



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



**3. Providing additional message:**

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

git stash save “message”

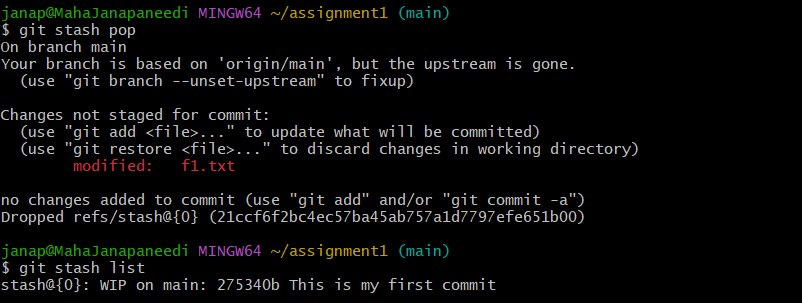
**4. Getting back stashed changes:**

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

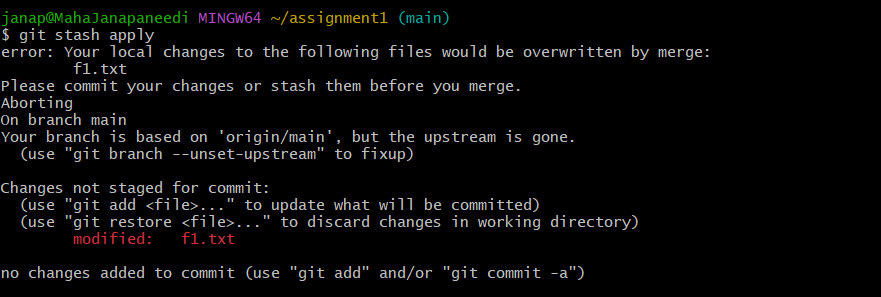
1.‘git stash pop’ removes the changes from stash and reapplies the changes in working copy,

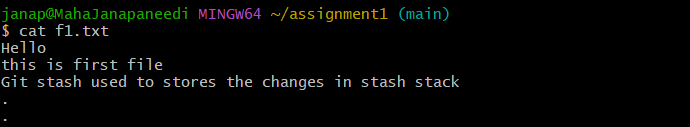
2.‘git stash apply’ do not remove changes .but reapplies the changes in working copy.

Now check whether stash is removed or not.

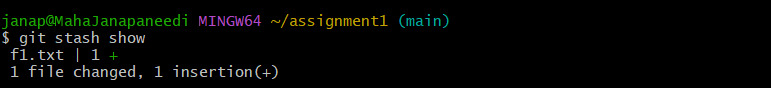


By using “git stash apply” We got the previous uncommitted changes.





**5.To view the stash summary**:

Git stash show is used to view the summary

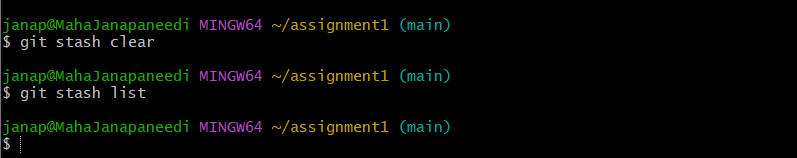
**6. Deleting stashes:**

To delete a particular stash:

git stash drop stash@{1}

To delete all stashes at once, use the below command.

git stash clear



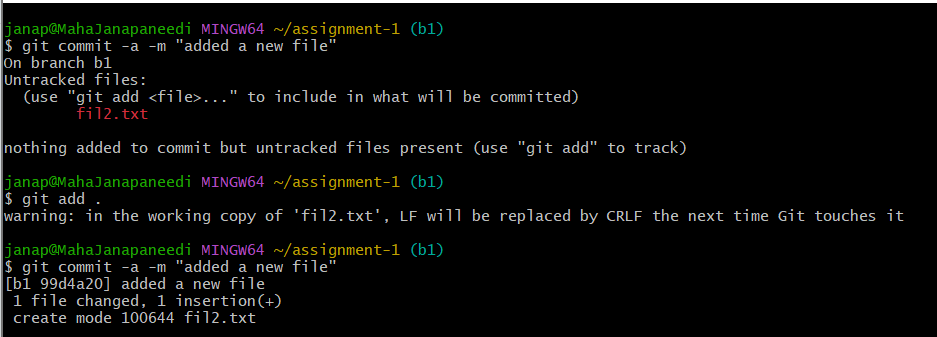
**Q2. By using a sample example of your choice, use the git fetch command and also use the git merge command and describe the whole process through a screenshot with all the commands and their output in git bash.**

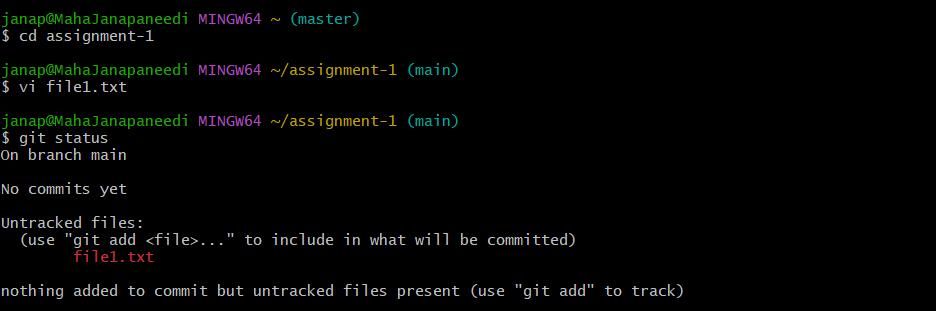
Fetch just downloads the objects and refs from a remote repository and normally updates the remote tracking branches. Pull, however, will not only download the changes, but also merges them - it is the combination of fetch and merge (cf. the section called “Merging”). The configured remote tracking branch is selected automatically.

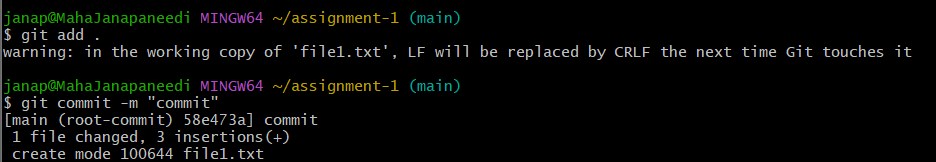
**Git pull=git fetch+git merge**

git fetch is the command that tells your local git to retrieve the latest meta-data info from the original (yet doesn’t do any file transferring. It’s more like just checking to see if there are any changes available).

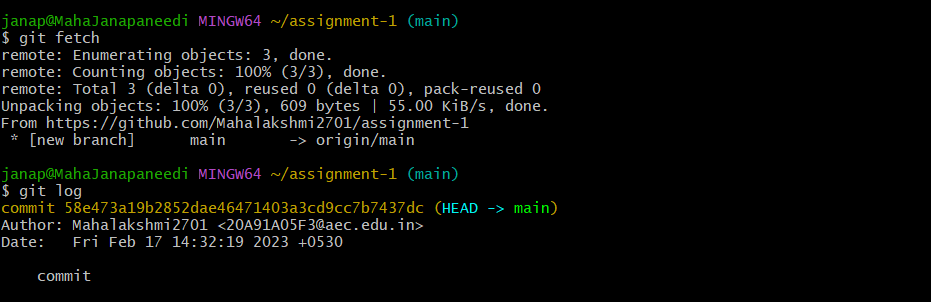
The "git merge" command. The git merge command is used to merge the branches. The syntax for the git merge command is as: $ git merge <query>





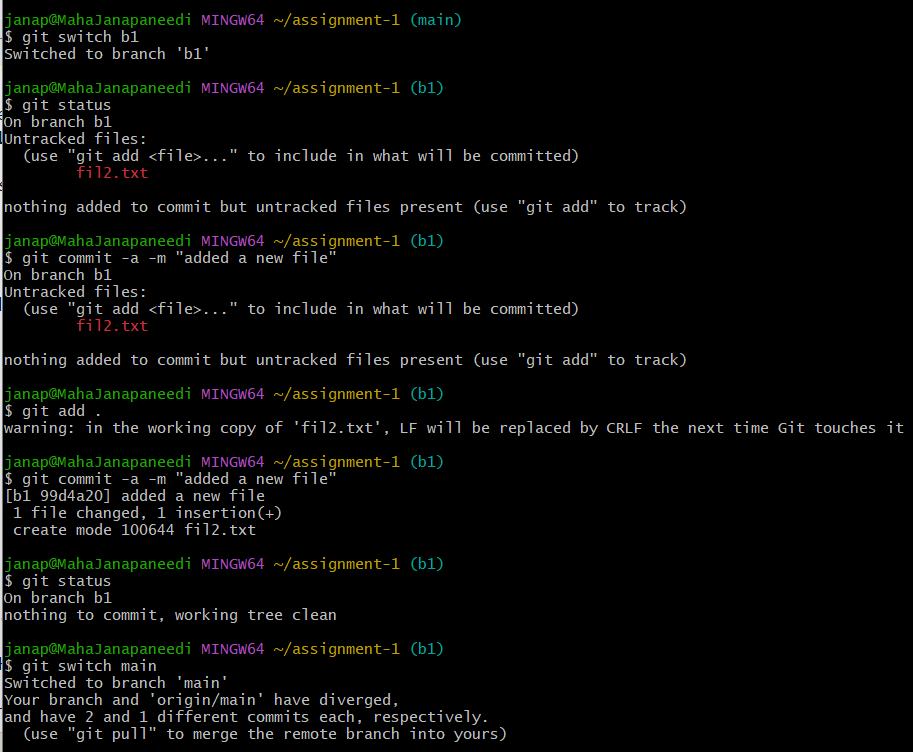
**Git Fetch**

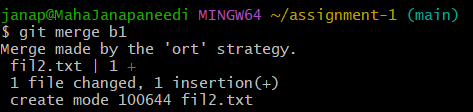
git fetch is the command that tells your local git to retrieve the latest meta-data info from the original (yet doesn’t do any file transferring. It’s more like just checking to see if there are any changes available).



**Git Merge :**

The "git merge" command. The git merge command is used to merge the branches. The syntax for the git merge command is as: $ git merge <query>





**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 command that allows you to download objects from a remote repository but it doesn't integrate any of this new data into your working files.

Fetch just downloads the objects and references from a remote repository and normally updates the remote tracking branches.

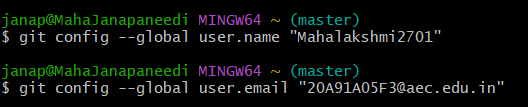
**Git pull:** Git pull is a command that allows you to fetch from and integrate with another repository or local branch. It updates your current HEAD branch with the latest changes from the remote server.

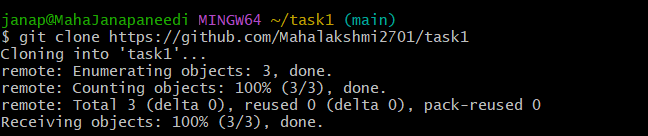
Pull, however, will not only download the changes but also merges them - it is a combination of fetch and merges. The configured remote-tracking branch is selected automatically.

**Git pull=git fetch + git merge**

**Step 1)Git clone an empty repository**

First, give the username and email using the git config. Then create a new repository named task1 and clone using the git clone command.





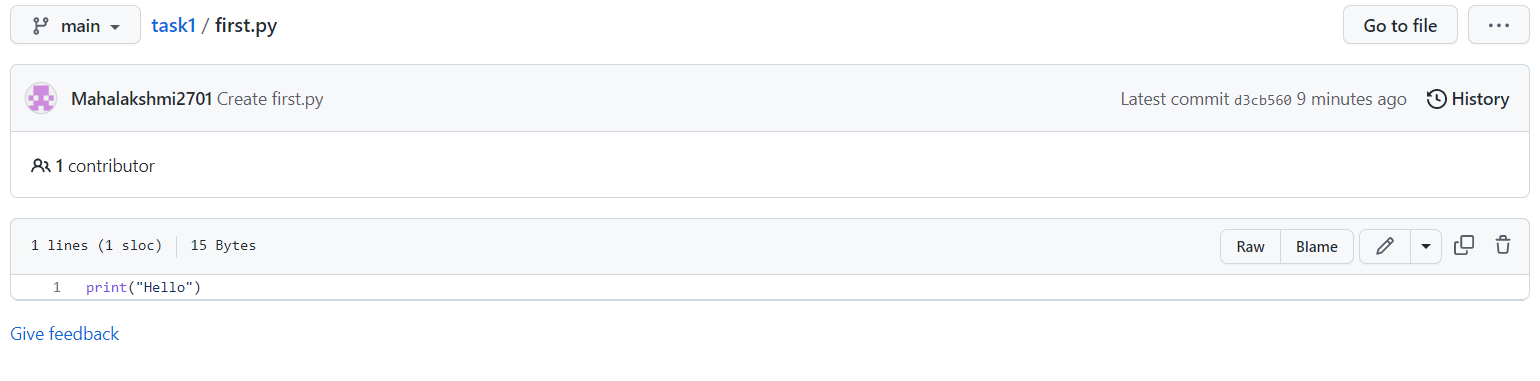
**Step 2) Git log –oneline –all**

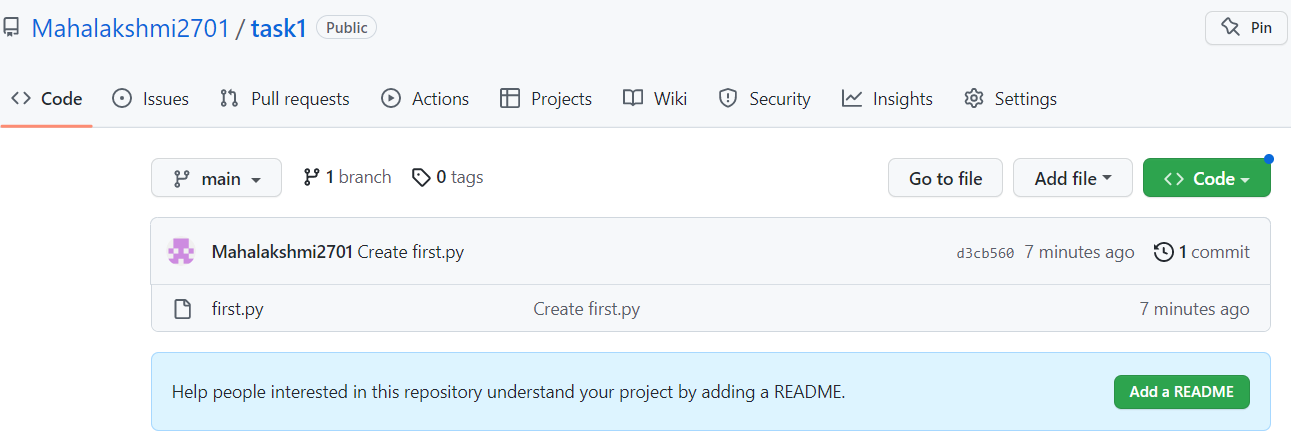
****

**Step 3)Go to the repository and create a file.**

This command gives no output as we have not committed anything yet.

Go to GitHub and go to the repository task1 and create a file and then update it.

****

****

**Step 4)In the git bash, go to the task1 repository.**

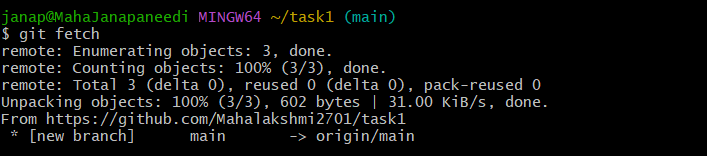
****

****

We have moved to the task1 repository and use the git log command to view the commit history, but here we cannot see any commits, even though we made one commit in the remote repository

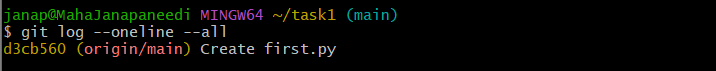
**Step 5)Git fetch**

Git fetch will download the changes in the remote repository, fetch commands just downloads the changes made in the remote repository.

****

**Once the git fetch command is used, it downloads the changes made in the remote repository.**

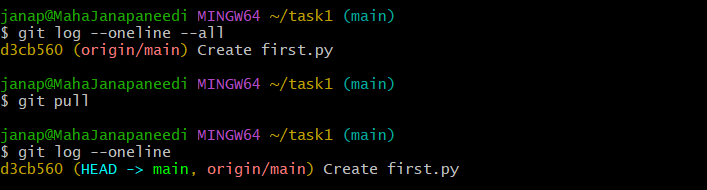
**Git log –oneline –all**

****

Here when the git log command is used then, it shows one commit made in the remote repository.

Here one commit that is made in the remote repository is being shown, but it is not applied. To apply and download we use the git pull command.

**Git pull------downloads and merges**

****

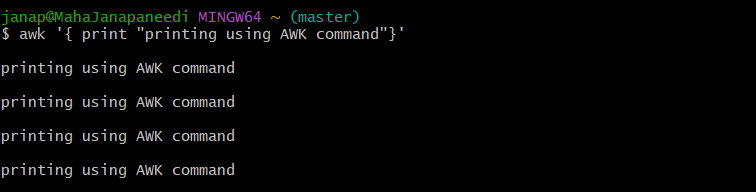
**Now the commits are applied to the main branch by using the git pull command.**

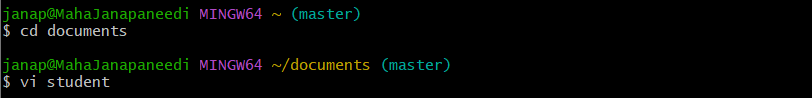
**Q4. Try to find out about the awk command and use it while reading a file created by yourself. Also, make a bash script file and try to find out the prime number from the range 1 to 20.**

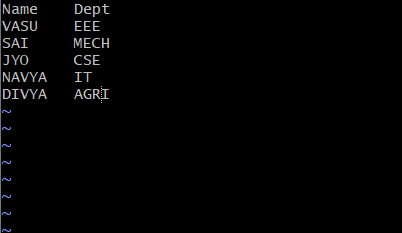
**The whole process should be carried out and by using the history command, give the screenshot**

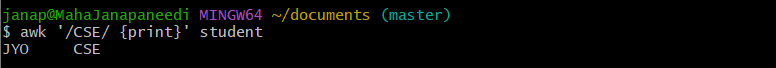
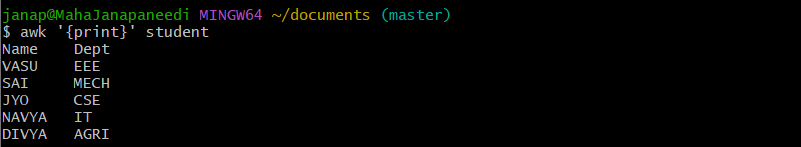
**of all the processes being carried out.**

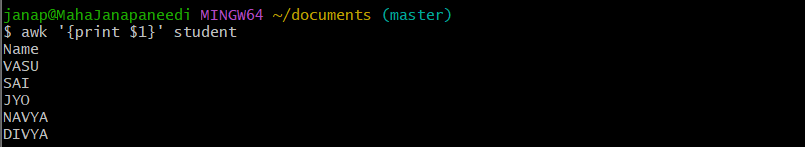
**AWK:** The Awk is a powerful scripting language used for **text scripting**. It searches and replaces the texts and sorts, validates, and indexes the database. It performs various actions on a file like searching a specified text and more.

****

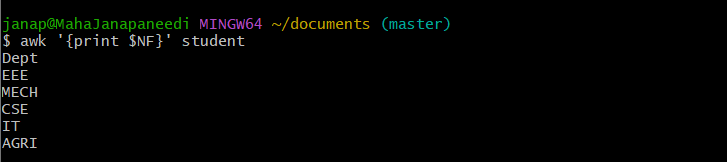
****

****

****

****

* above command will print column1
* \***$NF**: Used to display the last field of the file.

****

Steps to follow bash scripting:

Step 1) create the file with the extension .sh.

Step 2)open the shell and write the script.

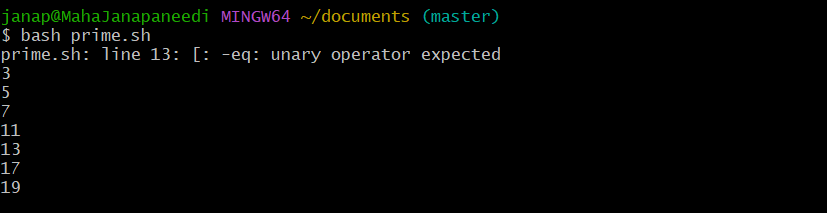
Step 3)save the code and run the code.

To run the run a code

Syntax: bash filename. sh



****

****

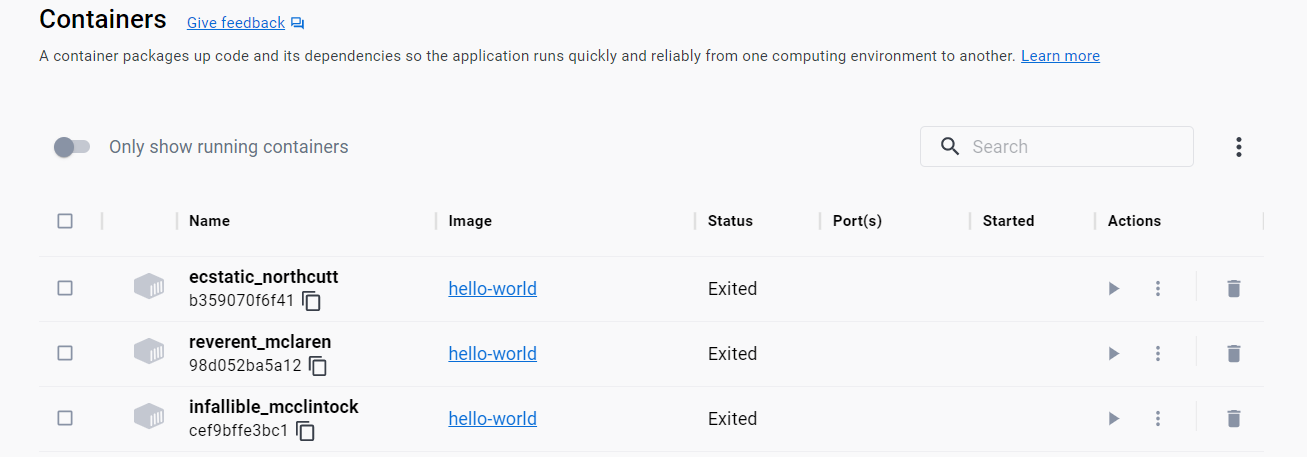
**Q5. Set up a container and run a Ubuntu operating system. For this purpose, you can make use of the docker hub and run the container in interactive mode.**

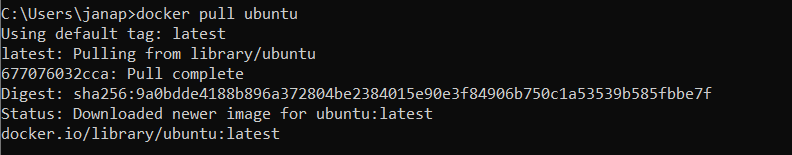
**All the processes pertaining to this should be provided in a screenshot for grading.**

**Image:** Images are used to create containers. It uses a private container registry to share container images within the enterprise and also uses a public container registry to share container images with the whole world.

**Container:** Containers are used to hold the entire package that is needed to run the application. We can say that the image is a template and the container is a copy of the template.

\*These are the containers present in the docker desktop.



****

\*For setting up a container and run the ubuntu os,

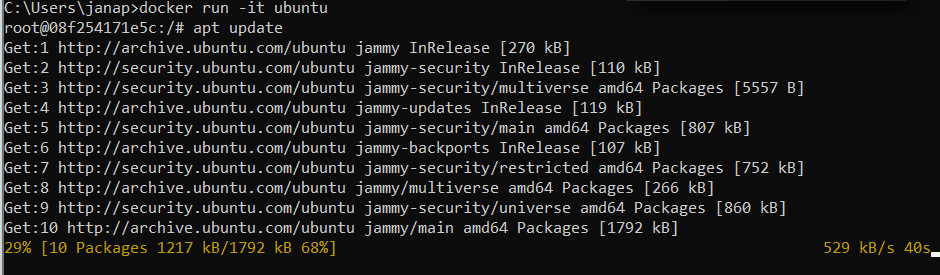
🡪First we need to download the image of Ubuntu from the docker hub using the command docker pull **ubuntu**.

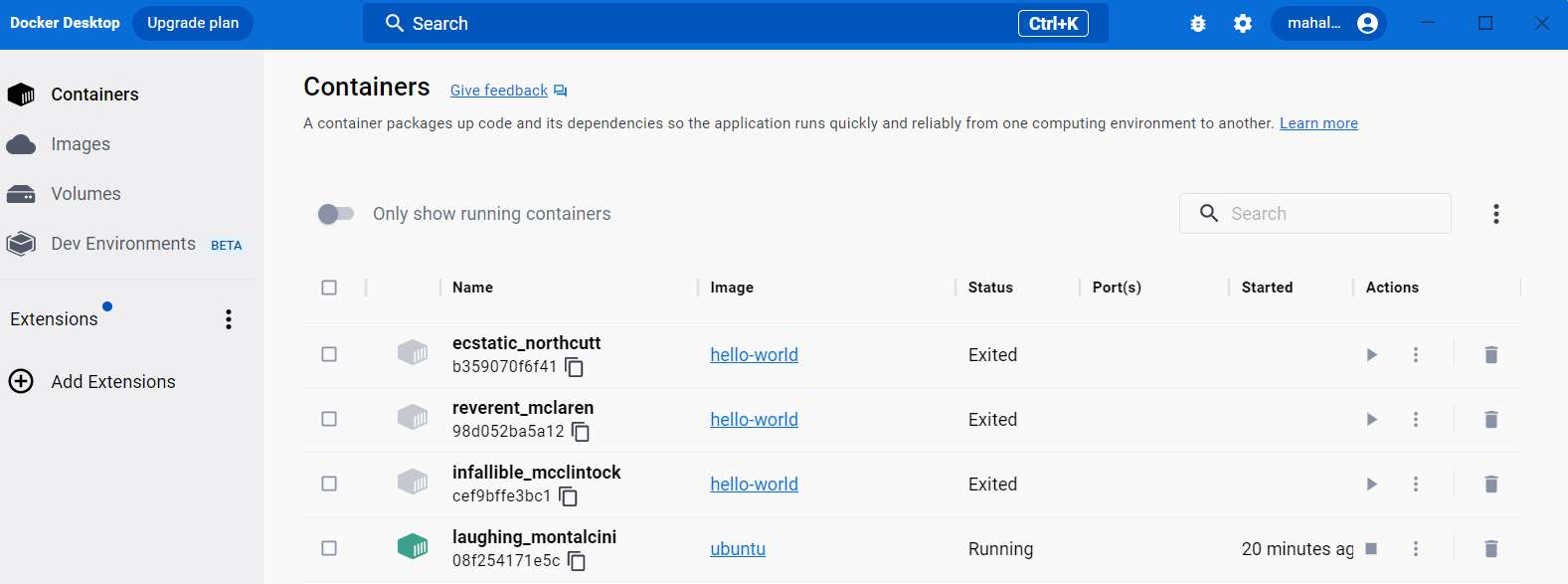
🡪To create a container and execute the image use the command docker run -it ubuntu .

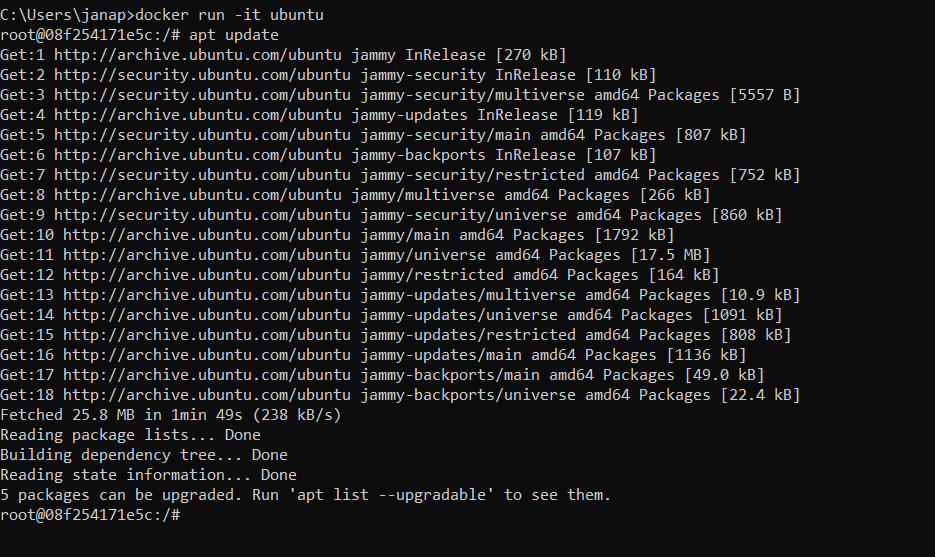
🡪To get an idea about the available update use apt update command.

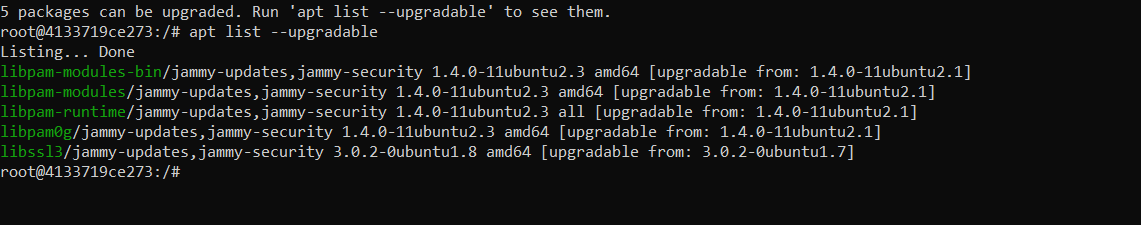
\*\*Download the ubuntu OS image from the docker hub.

\*An image ubuntu got downloaded but a container is not created.

****



****

****

\*\*Now a container got created for the ubuntu image.