DevOps Assignment-1

[Venkata Krishna Reddy]-[20A91A1239]

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:

The git stash command takes your uncommitted changes (both staged and unstaged), saves them away for later use, and then reverts them from your working copy.

When you run git stash, Git takes all of the changes in your working directory that have not been committed and saves them in a new stash. This stash is stored in a separate location within the Git repository, and can be retrieved later if needed.

The git stash command can be used in a number of different ways, depending on your specific needs. Some common usage scenarios include:

- Stashing changes temporarily: You can run git stash to save any uncommitted changes that you are working on, so that you can switch to a different branch or work on a different feature without losing your progress.
- Retrieving stashed changes: Once you have stashed your changes, you can retrieve them later using the git stash apply or git stash pop command.
- Listing stashes: You can view a list of all the stashes that you have saved using the git stash list command.
- Applying stashes selectively: If you have multiple stashes saved, you can apply a specific stash using the git stash apply stash@{n} command, where n is the index of the stash you want to apply.
- Stashing only specific changes: You can also use the git stash command with specific arguments to stash only specific changes, such as untracked files or ignored files.

Different git commands regarding stashing:

- git stash: Stash changes in the working directory that have not yet been committed.
- git stash list: List all stashes that have been saved.
- git stash show: Show the changes in the most recent stash.
- git stash show stash@{n}: Show the changes in a specific stash.
- git stash apply: Apply the most recent stash.
- git stash apply stash@{n}: Apply a specific stash.
- git stash pop: Apply the most recent stash and remove it from the stash list.
- git stash pop stash@{n}: Apply a specific stash and remove it from the stash list.
- git stash drop: Remove the most recent stash from the stash list.
- git stash branch <branch-name>: Create a new branch and apply the most recent stash to it..
- git stash clear: Remove all stashes from the stash list.

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    CODEWHISPERER REFERENCE LOG

Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (master)
$ git stash
Saved working directory and index state WIP on master: 4449921 after apply

Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (master)
$ git stash list
stash@{0}: WIP on master: 4449921 after apply

Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (master)
$ git stash
Saved working directory and index state WIP on master: 4449921 after apply

Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (master)
$ git stash list
stash@{0}: WIP on master: 4449921 after apply
stash@{1}: WIP on master: 4449921 after apply

Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (master)
$ git stash apply
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   file2

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

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    CODEWHISPERER REFERENCE LOG

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

Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (master)
$ git add .

Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (master)
$ git commit -m "commit for stash"
[master b973a26] commit for stash
 1 file changed, 1 deletion(-)

Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (master)
$ git stash drop
Dropped refs/stash@{0} (0bf78d6c6ad824c04c83dced69e3f0278c39cbd3)

Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (master)
$ git stash list
stash@{0}: WIP on master: 4449921 after apply

Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (master)
$ git stash drop
Dropped refs/stash@{0} (2f8098deb667f99ba16a43254f2860a61e899924)

Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (master)
$ git stash clear

Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (master)
$ 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.

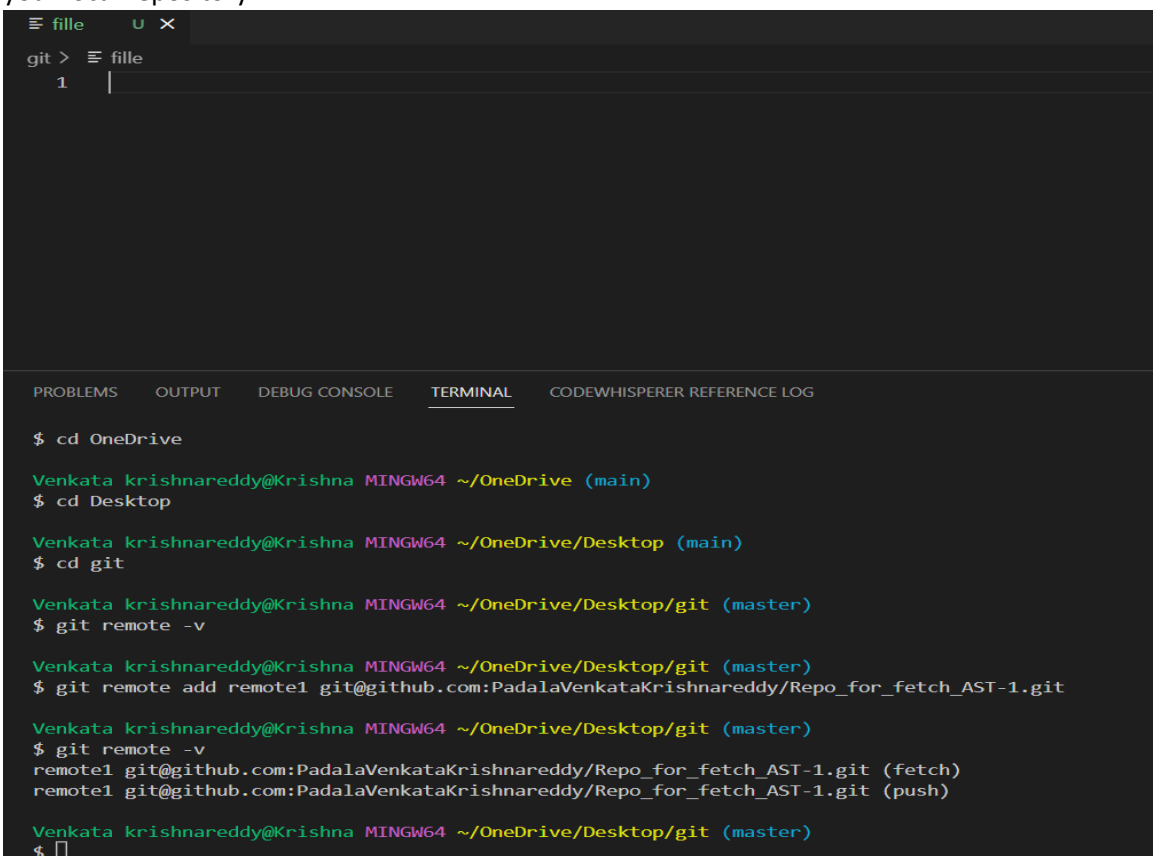
Git Fetch:

The git fetch command is used to retrieve the latest changes from a remote Git repository, without automatically merging them into your local branch. This command is often used to update your local repository with changes that have been made by other contributors to the same project, allowing you to keep your working copy up-to-date with the latest changes.

When you run git fetch, Git fetches the latest changes from the remote repository and stores them in your local repository, but does not automatically merge them into your current branch. This means that you can review the changes before deciding to merge them into your local branch.

Here are some common ways to use git fetch:

- Update the local repository: Use git fetch to download the latest changes from the remote repository and update your local repository with them.
- View changes: After running git fetch, you can use git log or other commands to view the changes that have been made in the remote repository since the last time you fetched changes.
- Merge changes: If you want to merge the changes you fetched from the remote repository into your local branch, you can run git merge after running git fetch.
- Check out remote branches: After running git fetch, you can check out a remote branch using git checkout origin/<branch-name>. This allows you to work on the remote branch in your local repository.



file_r_1 U file_r_2 U file_r_3 U

git > file_r_3

```
file 3 using for fetch command
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL CODEWHISPERER REFERENCE LOG

```
Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (master)
$ git remote add remote4 git@github.com:PadalaVenkataKrishnareddy/Repo_for_fetch_AST-1.git

Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (master)
$ git push remote4 master
Enumerating objects: 29, done.
Counting objects: 100% (29/29), done.
Delta compression using up to 12 threads
Compressing objects: 100% (23/23), done.
Writing objects: 100% (29/29), 71.07 KiB | 103.00 KiB/s, done.
Total 29 (delta 7), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (7/7), done.
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:      https://github.com/PadalaVenkataKrishnareddy/Repo_for_fetch_AST-1/pull/new/master
remote:
To github.com:PadalaVenkataKrishnareddy/Repo_for_fetch_AST-1.git
 * [new branch]      master -> master

Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (master)
$
```

Search or jump to... Pull requests Issues Codespaces Marketplace Explore

PadalaVenkataKrishnareddy / Repo_for_fetch_AST-1 Public

Pin Unwatch 1 Fork 0 Star 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master had recent pushes less than a minute ago Compare & pull request

master 2 branches 0 tags Go to file Add file Code

This branch is 9 commits ahead, 1 commit behind main. Contribute

PadalaVenkataKrishnareddy commit for stash b973a26 1 hour ago 9 commits

Git Cheat Sheet.pdf	file4 committed	17 hours ago
file1	commit for stash 3	15 hours ago
file2	commit for stash	1 hour ago
file3	final commit	2 hours ago
file4.txt	file4 committed	17 hours ago
git	file4 committed	17 hours ago

Help people interested in this repository understand your project by adding a README. Add a README

About

No description, website, or topics provided.

0 stars
1 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Search or jump to... / Pull requests Issues Codespaces Marketplace Explore

PadalaVenkataKrishnareddy / Repo_for_fetch_AST-1 Public

Pin Unwatch 1 Fork 0 Star 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Repo_for_fetch_AST-1 / file1 in master

Cancel changes

Edit file Preview changes

Spaces 2 No wrap

```
changlng in order to apply
stash command for file1
again changing inorder for fetch

```

file_r_1 U file_r_2 U file_r_3 U

git > file_r_3

```
file 3 using for fetch command

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL CODEWHISPERER REFERENCE LOG

```
remote: Resolving deltas: 100% (7/7), done.
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:      https://github.com/PadalaVenkataKrishnareddy/Repo_for_fetch_AST-1/pull/new/master
remote:
To github.com:PadalaVenkataKrishnareddy/Repo_for_fetch_AST-1.git
 * [new branch]      master -> master

Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (master)
$ git fetch remote4 master
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 714 bytes | 39.00 KiB/s, done.
From github.com:PadalaVenkataKrishnareddy/Repo_for_fetch_AST-1
 * branch            master     -> FETCH_HEAD
   b973a26..fe8c409  master     -> remote4/master

Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (master)
$
```

file_r_1 U | file_r_2 U | file_r_3 U | file1

git > file1

```
changlng in order to apply
stash command for file1
again changing inorder for fetch
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL CODEWHISPERER REFERENCE LOG

```
$ git fetch remote4 master
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 714 bytes | 39.00 KiB/s, done.
From github.com:PadalaVenkataKrishnareddy/Repo_for_fetch_AST-1
 * branch            master     -> FETCH_HEAD
   b973a26..fe8c409  master     -> remote4/master

Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (master)
$ git pull remote4 master
From github.com:PadalaVenkataKrishnareddy/Repo_for_fetch_AST-1
 * branch            master     -> FETCH_HEAD
Updating b973a26..fe8c409
Fast-forward
 file1 | 1 +
 1 file changed, 1 insertion(+)

Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (master)
$
```

Git Merge:

The git merge command is used to integrate changes from one branch into another branch. This is typically used when you want to merge the changes that you have made on a feature branch into the main branch of the Git repository.

Here are the steps to follow to use the git merge command:

- Switch to the branch that you want to merge changes into: First, you need to switch to the branch that you want to merge changes into. For example, if you want to merge changes from the feature-branch into the main branch, you should switch to the main branch by running git checkout main.
- Run git merge: Once you are on the target branch, run the git merge command and specify the name of the branch that you want to merge changes from. For example, if you want to merge changes from the feature-branch into the main branch, you should run git merge feature-branch. Git will then attempt to merge the changes from the feature-branch into the main branch.
- Resolve any conflicts: If there are any conflicts between the changes on the two branches, Git will display a message indicating which files have conflicts. You will need to manually resolve these conflicts by editing the affected files, and then running git add to stage the changes. Once you have resolved all the conflicts and staged the changes, run git commit to create a new commit that includes the merged changes.
- Push the changes: Finally, you need to push the changes to the remote repository by running git push. This will update the remote repository with the merge

```
git > file1
1   changlng in order to apply
2   stash command for file1

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    CODEWHISPERER REFERENCE LOG

Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (master)
$ git branch b1
fatal: a branch named 'b1' already exists

Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (master)
$ bit switch b1
bash: bit: command not found

Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (master)
$ git switch b1
Switched to branch 'b1'

Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (b1)
$ touch fil1_b1

Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (b1)
$ cat > file_b1
file using for merging


[6]+  Stopped                 cat > file_b1

Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (b1)
$

Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (b1)
$ touch fil1_b2

Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (b1)
$ cat > file_b2
another file using for merging
```

file_r_1 U | file_r_2 U | file_r_3 U | file1

git > file1

```
changlng in order to apply
stash command for file1
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL CODEWHISPERER REFERENCE LOG

```
Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (b1)
$ touch fil1_b2

Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (b1)
$ cat > file_b2
another file using for merging


Switched to branch 'master'

Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (master)
$ git merge b1
Already up to date.

Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (master)
$ git log --oneline
fe8c409 (HEAD -> master, remote4/master, b1) Update file1
b973a26 commit for stash
4449921 after apply
5643e34 commit after applying stash
5ba03ea final commit
525feb4  commit for stash 3
3655b29  commit for stash 3
7aafed0 commit for stash 1
05e36db file4 committed
c7970fc first commit

Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (master)
$
```

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.

The main difference between git fetch and git pull is that git fetch only retrieves the latest changes from a remote repository, while git pull both retrieves the latest changes and immediately merges them into the local branch.

Here are some more details about the differences between git fetch and git pull:

- git fetch retrieves the latest changes from a remote repository and stores them in the local repository, but does not merge them into the current branch. This means that you can review the changes before deciding to merge them into your local branch.
- git pull retrieves the latest changes from a remote repository, stores them in the local repository, and immediately merges them into the current branch. This means that the changes from the remote repository will be automatically merged into your local branch.
- git fetch can be useful when you want to update your local repository with changes from a remote repository, but you want to review the changes before merging them into your local branch.
- git pull can be useful when you want to update your local repository with changes from a remote repository and immediately merge them into your local branch. This is often used when you want to quickly incorporate changes made by other contributors to the same project.

```
file_r_1  U    file_r_2  U    file_r_3  U ●
git > file_r_3
1  file 3 using for fetch command
2

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   CODEWHISPERER REFERENCE LOG

remote: Resolving deltas: 100% (7/7), done.
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:      https://github.com/PadalaVenkataKrishnareddy/Repo_for_fetch_AST-1/pull/new/master
remote:
To github.com:PadalaVenkataKrishnareddy/Repo_for_fetch_AST-1.git
 * [new branch]      master -> master

Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (master)
$ git fetch remote4 master
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 714 bytes | 39.00 KiB/s, done.
From github.com:PadalaVenkataKrishnareddy/Repo_for_fetch_AST-1
 * branch            master     -> FETCH_HEAD
   b973a26..fe8c409  master     -> remote4/master

Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (master)
$
```

file_r_1 U file_r_2 U file_r_3 U

git > file_r_3

```
file 3 using for fetch command

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL CODEWHISPERER REFERENCE LOG

```
$ git fetch remote4 master
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 714 bytes | 39.00 KiB/s, done.
From github.com:PadalaVenkataKrishnareddy/Repo_for_fetch_AST-1
 * branch            master     -> FETCH_HEAD
   b973a26..fe8c409  master     -> remote4/master

Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (master)
$ git pull remote4 master
From github.com:PadalaVenkataKrishnareddy/Repo_for_fetch_AST-1
 * branch            master     -> FETCH_HEAD
Updating b973a26..fe8c409
Fast-forward
 file1 | 1 +
 1 file changed, 1 insertion(+)

Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (master)
$
```

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 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.

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. 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.

Awk is abbreviated from the names of the developers – Aho, Weinberger, and Kernighan.

- AWK Operations:
  - Scans a file line by line
  - Splits each input line into fields
  - Compares input line/fields to pattern
  - Performs action(s) on matched lines
- Useful For:
  - Transform data files
  - Produce formatted reports
- Programming Constructs:
  - Format output lines
  - Arithmetic and string operations
  - Conditionals and loops

Syntax:

awk options 'selection _criteria {action }' input-file > output-file

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    CODEWHISPERER REFERENCE LOG

Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (master)
$ touch awk.txt

Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (master)
$ cat >awk.txt
Hi! happy to use awk command

[1]+  Stopped                 cat > awk.txt

Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (master)
$ awk '{print}' awk.txt
Hi! happy to use awk command

Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (master)
$ awk '{"print hi"}'


[2]+  Stopped                 awk '{"print hi"}'

Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (master)
```

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    CODEWHISPERER REFERENCE LOG

Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (master)
$ cat >> awk.txt
Excited for completion of assignment

[3]+  Stopped                 cat >> awk.txt

Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (master)
$ cat awk.txt
Hi! happy to use awk command
Excited for completion of assignment

Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (master)
$ awk '/v/ {print}' awk.txt

Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (master)
$ awk '/h/ {print}' awk.txt
Hi! happy to use awk command

Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (master)
$ 
```

Bash Script for finding prime numbers from 1 to 20:

```
$ prime.sh U
git > $ prime.sh > ...
#!/bin/bash
echo "1 is not a prime number.";
for((k=2;k<=20;k++))
do
  c=0;
  for((i=2; i<=$k/2; i++))
  do
    r=$(( k%i ))
    if [ $r -eq 0 ]
    then
      let c=1;
      break;
    fi
  done
  if [ $c -eq 0 ];
  then
  echo "$k is a prime number.";
  else
  echo "$k is not a prime number.";
fi
done

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    CODEWHISPERER REFERENCE LOG

Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (master)
$ sh prime.sh
1 is not a prime number.
2 is a prime number.
3 is a prime number.
4 is not a prime number.
5 is a prime number.
6 is not a prime number.
7 is a prime number.
8 is not a prime number.
```

$ prime.sh U

git > $ prime.sh > ...

```bash
#!/bin/bash
echo "1 is not a prime number.";
for((k=2;k<=20;k++))
do
  c=0;
  for((i=2; i<=$k/2; i++))
  do
    r=$(( k%i ))
    if [ $r -eq 0 ]
    then
      let c=1;
      break;
    fi
  done
  if [ $c -eq 0 ];
  then
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL CODEWHISPERER REFERENCE LOG

```
Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (master)
$ sh prime.sh
1 is not a prime number.
2 is a prime number.
3 is a prime number.
4 is not a prime number.
5 is a prime number.
6 is not a prime number.
7 is a prime number.
8 is not a prime number.
9 is not a prime number.
10 is not a prime number.
11 is a prime number.
12 is not a prime number.
13 is a prime number.
14 is not a prime number.
15 is not a prime number.
16 is not a prime number.
```

$ prime.sh U

git > $ prime.sh > ...

```bash
#!/bin/bash
echo "1 is not a prime number.";
for((k=2;k<=20;k++))
do
  c=0;
  for((i=2; i<=$k/2; i++))
  do
    r=$(( k%i ))
    if [ $r -eq 0 ]
    then
      let c=1;
      break;
    fi
  done
  if [ $c -eq 0 ];
  then
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL CODEWHISPERER REFERENCE LOG

```
5 is a prime number.
6 is not a prime number.
7 is a prime number.
8 is not a prime number.
9 is not a prime number.
10 is not a prime number.
11 is a prime number.
12 is not a prime number.
13 is a prime number.
14 is not a prime number.
15 is not a prime number.
16 is not a prime number.
17 is a prime number.
18 is not a prime number.
19 is a prime number.
20 is not a prime number.

Venkata krishnareddy@Krishna MINGW64 ~/OneDrive/Desktop/git (master)
$
```

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.

- Install Docker on your system if you haven't already. You can download Docker for your specific operating system from the official Docker website.
- Open a terminal or command prompt and run the following command to download the Ubuntu image from Docker Hub:

Execute the following command:

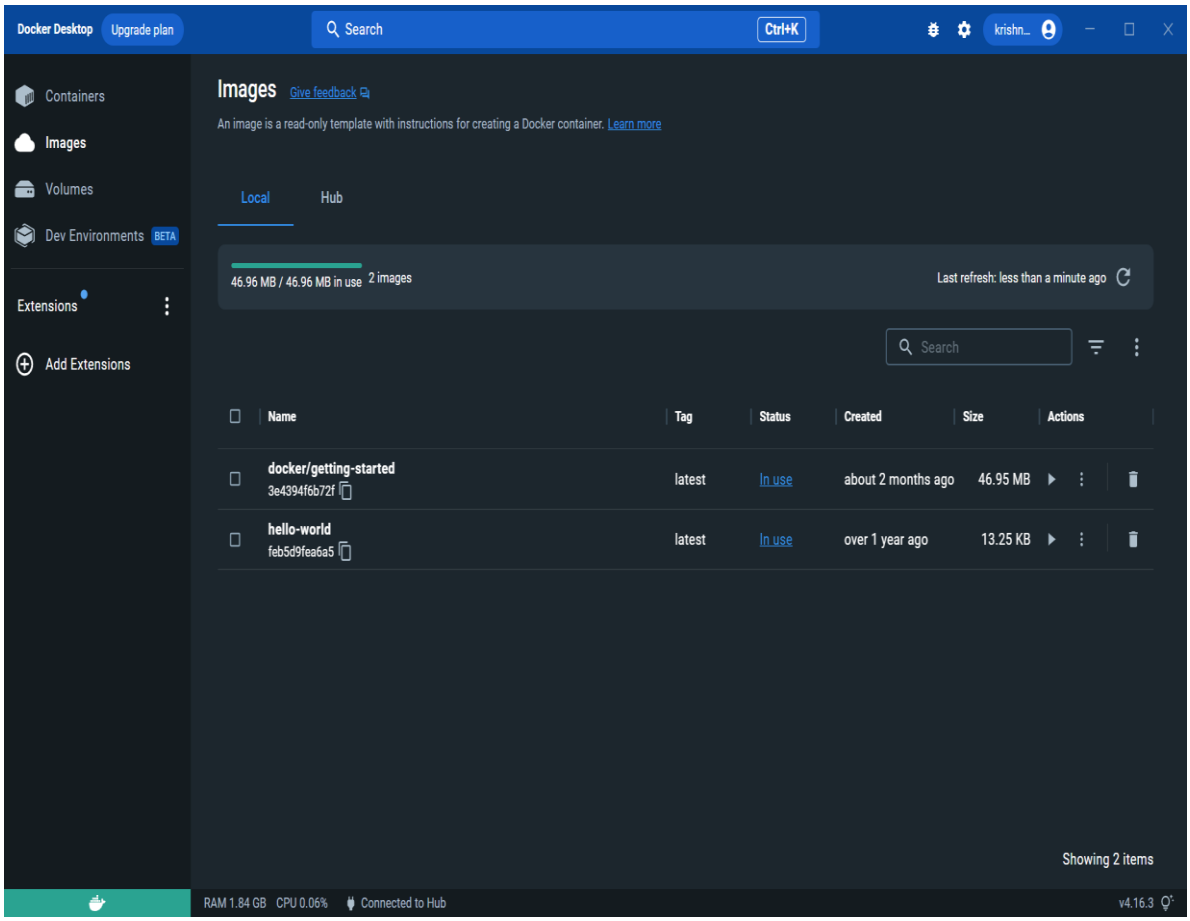
```
docker pull ubuntu
```

- This command will download the latest version of the Ubuntu image from Docker Hub.
- Once the image has finished downloading, run the following command to start a new container with Ubuntu:

Execute the following command:

```
docker run -it ubuntu
```

- This command will start a new container with the Ubuntu image in interactive mode.
- You should now be logged into the container with a command prompt. You can run any command or install any software just as you would on a regular Ubuntu system.



Command Prompt

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

C:\Users\Venkata krishnareddy>docker version
Client:
 Cloud integration: v1.0.29
 Version:           20.10.22
 API version:       1.41
 Go version:        go1.18.9
 Git commit:        3a2c30b
 Built:             Thu Dec 15 22:36:18 2022
 OS/Arch:           windows/amd64
 Context:           default
 Experimental:      true

Server: Docker Desktop 4.16.3 (96739)
 Engine:
  Version:          20.10.22
  API version:      1.41 (minimum version 1.12)
  Go version:       go1.18.9
  Git commit:       42c8b31
  Built:            Thu Dec 15 22:26:14 2022
  OS/Arch:          linux/amd64
  Experimental:     false
 containerd:
  Version:          1.6.14
  GitCommit:        9ba4b250366a5ddde94bb7c9d1def331423aa323
 runc:
  Version:          1.1.4
  GitCommit:        v1.1.4-0-g5fd4c4d
 docker-init:
  Version:          0.19.0
  GitCommit:        de40ad0

C:\Users\Venkata krishnareddy>docker run ubuntu
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
677076032cca: Pull complete
Digest: sha256:9a0bdde4188b896a372804be2384015e90e3f84906b750c1a53539b585fbbe7f
Status: Downloaded newer image for ubuntu:latest

C:\Users\Venkata krishnareddy>
```

Command Prompt

```
 Version:           20.10.22
 API version:       1.41
 Go version:        go1.18.9
 Git commit:        3a2c30b
 Built:             Thu Dec 15 22:36:18 2022
 OS/Arch:           windows/amd64
 Context:           default
 Experimental:      true

Server: Docker Desktop 4.16.3 (96739)
 Engine:
  Version:          20.10.22
  API version:      1.41 (minimum version 1.12)
  Go version:       go1.18.9
  Git commit:       42c8b31
  Built:            Thu Dec 15 22:26:14 2022
  OS/Arch:          linux/amd64
  Experimental:     false
 containerd:
  Version:          1.6.14
  GitCommit:        9ba4b250366a5ddde94bb7c9d1def331423aa323
 runc:
  Version:          1.1.4
  GitCommit:        v1.1.4-0-g5fd4c4d
 docker-init:
  Version:          0.19.0
  GitCommit:        de40ad0

C:\Users\Venkata krishnareddy>docker run ubuntu
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
677076032cca: Pull complete
Digest: sha256:9a0bdde4188b896a372804be2384015e90e3f84906b750c1a53539b585fbbe7f
Status: Downloaded newer image for ubuntu:latest

C:\Users\Venkata krishnareddy>docker run -it ubuntu
root@c1408919a4f5:/# ls
bin  boot  dev  etc  home  lib  lib32  lib64  libx32  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@c1408919a4f5:/# exit
exit

C:\Users\Venkata krishnareddy>
```

Docker Desktop | Upgrade plan | Search Ctrl+K | krishn...

Containers
Images
Volumes
Dev Environments BETA
Extensions
Add Extensions

Images

Give feedback

An image is a read-only template with instructions for creating a Docker container. Learn more

Local | Hub

124.77 MB / 46.96 MB in use 3 images

Last refresh: 3 minutes ago

Search

Name	Tag	Status	Created	Size	Actions
ubuntu 58db3edaf2be	latest	In use	22 days ago	77.8 MB	
docker/getting-started 3e4394f6b72f	latest	In use	about 2 months ago	46.95 MB	
hello-world feb5d9fea6a5	latest	In use	over 1 year ago	13.25 KB	

Showing 3 items

RAM 2.05 GB CPU 0.06% Connected to Hub v4.16.3