CONSULTADD

GITHUB ASSIGNMENT

110 Wall Street,

New York, NY 10005

United States

QUESTIONS

1. What are the advantages of using Git?
   Git branches are cheap and easy to merge
   each developer gets their own local repository, complete with a full history of commits.
   faster release cycle
   Excellent support for parallel development, support for hundreds of parallel branches.
   Fully Distributed

2. What language is used in Git?
   C language is used in Git.

3. What is the meaning of "Index" or "Staging Area" in Git?

   When you work on your project making changes you are dealing with your project's working directory. This is the project directory on your computer's filesystem. All the changes you make will remain in the working directory until you add them to the staging area (via git add command). The staging area is best described as a preview of your next commit.

   Meaning, when you do a git commit, git will take the changes that are in the staging area and make the new commit out of those changes

   You can add and remove changes from staging area until you are satisfied with how your next commit will look like, at which point you can do git commit.

   Although it is often useful to think of staging area as some real area (or directory) where git stores changes (like it does in .git/objects) this is not entirely true. Git doesn't have a dedicated staging directory where it puts some objects representing file changes (blobs).

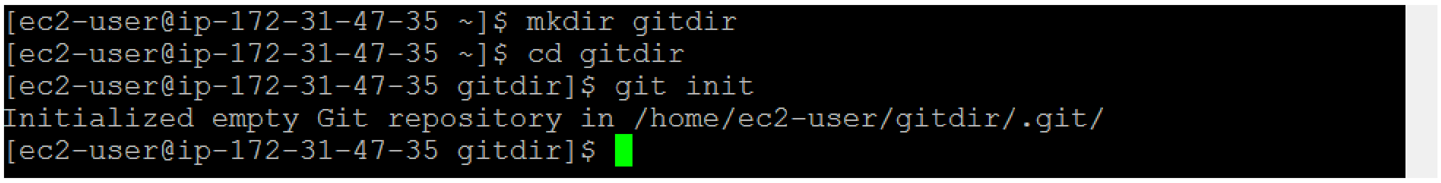
   Instead, git has a file called the index that it uses to keep track of the file changes over the three areas: working directory, staging area, and repository.

   And when you add changes to your staging area, git updates the information in the index about those changes and creates new blob objects, but puts them in the same .git/objects directory with all the other blobs that belong to previous commits.

index is not a directory but a file, so git is not actually storing objects (blobs) into it. Instead, git is storing information about each file in our repository.

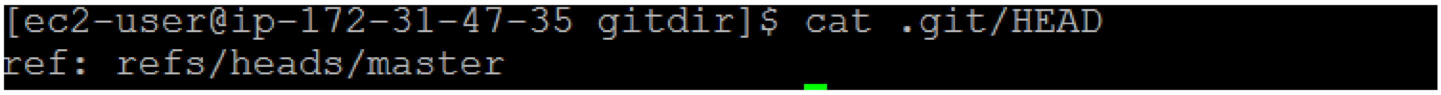
4. What is the process for creating a repository in Git?

   1) mkdir <name of directory>
   2)cd <path of the directory>
   3)git init



```
[ec2-user@ip-172-31-47-35 ~]$ mkdir gitdir
[ec2-user@ip-172-31-47-35 ~]$ cd gitdir
[ec2-user@ip-172-31-47-35 gitdir]$ git init
Initialized empty Git repository in /home/ec2-user/gitdir/.git/
[ec2-user@ip-172-31-47-35 gitdir]$
```

5. What is 'head' in Git and how many heads can be created in a repository?



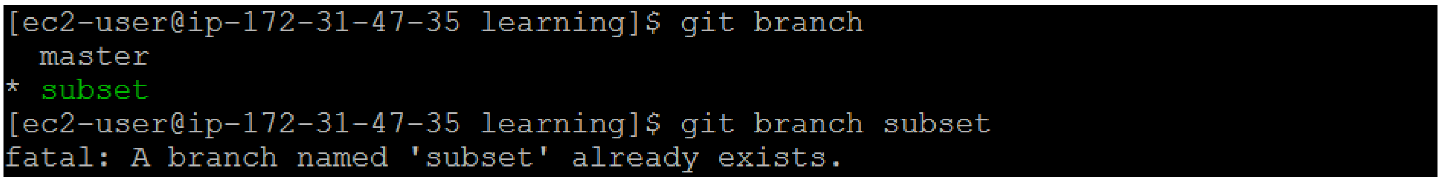
```
[ec2-user@ip-172-31-47-35 gitdir]$ cat .git/HEAD
ref: refs/heads/master
```

HEAD is a reference to the last commit in the currently check-out branch.
You can think of the HEAD as the "current branch". When you switch branches with git checkout , the HEAD revision changes to point to the tip of the new branch.

6. Why do we need branching in Git?
   everyone uses branches created from master to experiment, make edits and additions and changes, before eventually rolling that branch back into the master once they have been approved and are known to work. Master then is updated to contain all the new stuff.

7. Write a way to create a new branch in Git?

   git branch <branch_name>



```
[ec2-user@ip-172-31-47-35 learning]$ git branch
  master
* subset
[ec2-user@ip-172-31-47-35 learning]$ git branch subset
fatal: A branch named 'subset' already exists.
```

8. How do you define a ‘conflict’ in Git?

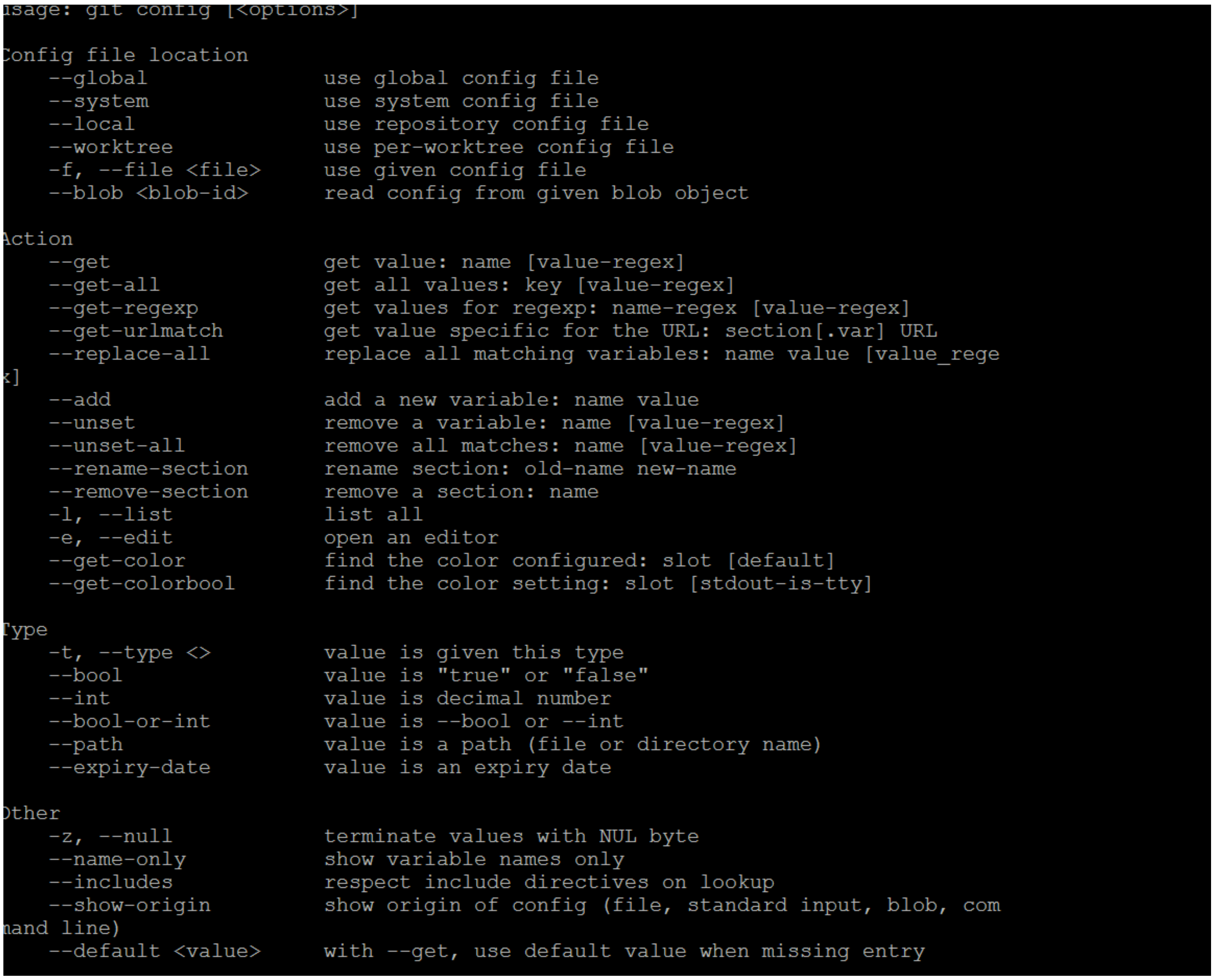
   Conflicts generally arise when two people have changed the same lines in a file, or if one developer deleted a file while another developer was modifying it. In these cases, Git cannot automatically determine what is correct. Conflicts only affect the developer conducting the merge, the rest of the team is unaware of the conflict. Git will mark the file as being conflicted and halt the merging process. It is then the developers' responsibility to resolve the conflict.

9. How to resolve a conflict in Git?

   The most direct way to resolve a merge conflict is to edit the conflicted file. Git will see that the conflict has been resolved and creates a new merge commit to finalize the merge

10. What is the function of ‘git config’?

    The git config command is a convenience function that is used to set Git configuration values on a global or local project level. These configuration levels correspond to .gitconfig text files. Executing git config will modify a configuration text file


```
usage: git config [<options>]

Config file location
    --global              use global config file
    --system              use system config file
    --local               use repository config file
    --worktree            use per-worktree config file
    -f, --file <file>     use given config file
    --blob <blob-id>      read config from given blob object

Action
    --get                 get value: name [value-regex]
    --get-all             get all values: key [value-regex]
    --get-regexp          get values for regexp: name-regex [value-regex]
    --get-urlmatch        get value specific for the URL: section[.var] URL
    --replace-all         replace all matching variables: name value [value_rege
x]
    --add                 add a new variable: name value
    --unset               remove a variable: name [value-regex]
    --unset-all           remove all matches: name [value-regex]
    --rename-section      rename section: old-name new-name
    --remove-section      remove a section: name
    -l, --list            list all
    -e, --edit            open an editor
    --get-color           find the color configured: slot [default]
    --get-colorbool       find the color setting: slot [stdout-is-tty]

Type
    -t, --type <>         value is given this type
    --bool                value is "true" or "false"
    --int                 value is decimal number
    --bool-or-int         value is --bool or --int
    --path                value is a path (file or directory name)
    --expiry-date         value is an expiry date

Other
    -z, --null            terminate values with NUL byte
    --name-only           show variable names only
    --includes            respect include directives on lookup
    --show-origin         show origin of config (file, standard input, blob, com
mand line)
    --default <value>     with --get, use default value when missing entry
```

11. What is Git fork?

A fork is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. Most commonly, forks are used to either propose changes to someone else's project or to use someone else's project as a starting point for your own idea

12. Difference between fork, branch and clone?

A repository code branch, like a branch of a tree, remains part of the original repository. The code that is branched (main trunk) and the branch know and rely on each other. Like a tree trunk's branch, a code branch knows about the trunk (original code base) it originated from.

unlike a branch, a fork is independent from the original repository. If the original repository is deleted, the fork remains. If you fork a repository, you get that repository and all of its branches.

the cloned repository is your active repo. It is where you do all your work. But other people generally don't have access to your personal cloned repo, because it's on your laptop. So that's why you have the forked repo, so you can push changes to it for others to see and review

13. What's the difference between a "pull request" and a "branch"?

When you file a pull request, all you're doing is requesting that another developer (e.g., the project maintainer) pulls a branch from your repository into their repository. This means that you need to provide 4 pieces of information to file a pull request: the source repository, the source branch, the destination repository, and the destination branch.

14. What is the difference between "git pull" and "git fetch"?

    You can consider git fetch the 'safe' version of the two commands. It will download the remote content but not update your local repo's working state, leaving your current work intact. git pull is the more aggressive alternative,
    it will download the remote content for the active local branch and immediately execute git merge
    to create a merge commit for the new remote content.

15. How to revert previous commit in Git?

    A revert is an operation that takes a specified commit and creates a new commit which inverses the specified commit. git revert can only be run at a commit level scope and has no file level functionality.

16. Explain the advantages of Forking Workflow

    A Git Workflow is a recipe or recommendation for how to use Git to accomplish work in a consistent and productive manner.
    Git workflows encourage users to leverage Git effectively and consistently.

    Git offers a lot of flexibility in how users manage changes.

To ensure the team is on the same page, an agreed upon Git workflow should be developed or selected.

it gives every developer their own local copy of the entire project. This isolated environment lets each developer work independently of all other changes to a project - they can add commits to their local repository and completely forget about upstream developments until it's convenient for them.

it gives you access to Git's robust branching and merging model

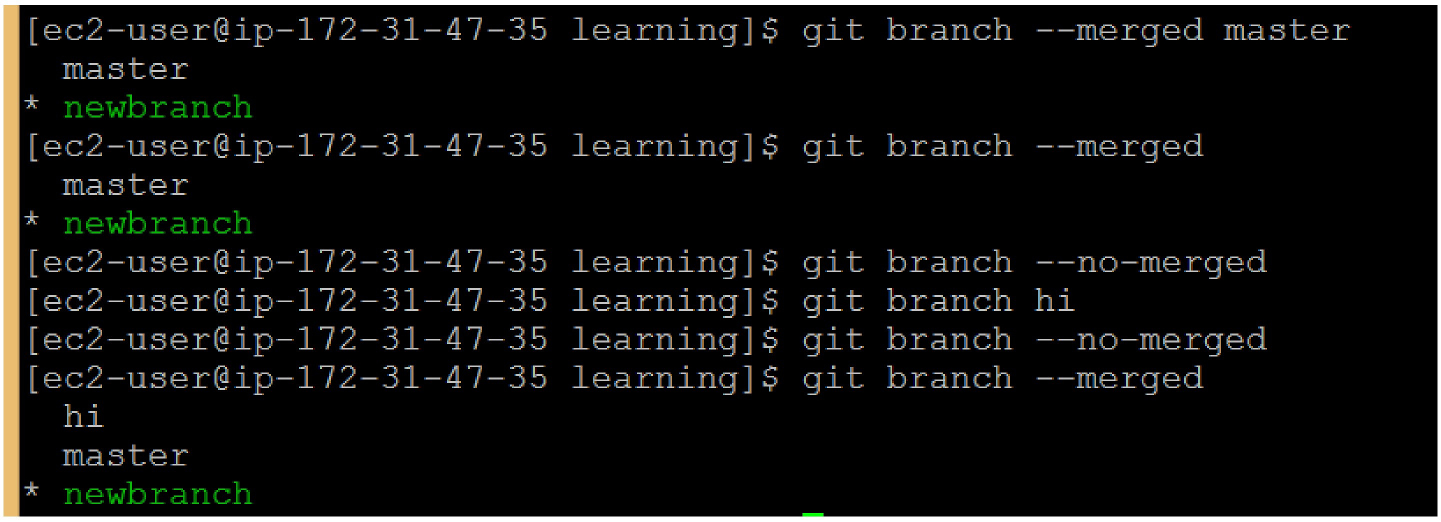
17. Difference between HEAD, working tree and index, in Git?

Working trees: They are nothing but the files that you are currently working on.

HEAD: HEAD is a pointer to the branch or commit that you last checked out, and which will be the parent of a new commit if you make it

Index: The git "index" is where you place files you want commit to the git repository.The index is a staging area where the new commit is prepared.

18. How to identify if a certain branch has been merged into master?



```
[ec2-user@ip-172-31-47-35 learning]$ git branch --merged master
  master
* newbranch
[ec2-user@ip-172-31-47-35 learning]$ git branch --merged
  master
* newbranch
[ec2-user@ip-172-31-47-35 learning]$ git branch --no-merged
[ec2-user@ip-172-31-47-35 learning]$ git branch hi
[ec2-user@ip-172-31-47-35 learning]$ git branch --no-merged
[ec2-user@ip-172-31-47-35 learning]$ git branch --merged
  hi
  master
* newbranch
```

19. What is the use of a Git clone?

Git clone is primarily used to point to an existing repo and make a clone or copy of that repo at in a new directory, at another location. The original repository can be located on the local filesystem or on remote machine accessible supported protocols. The git clone command copies an existing Git repository.

20. What is Git stash?

    git stash temporarily shelves (or stashes) changes you've made to your working copy so you can work on something else, and then come back and re-apply them later on. Stashing is handy if you need to quickly switch context and work on something else, but you're mid-way through a code change and aren't quite ready to commit.

    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.

21. When should I use "git stash"?

    you can use git stash when you want to you keep your current work aside and and work on another high priority task before you merge your current code.

22. What is Git stash drop?

If you decide you no longer need a particular stash, you can delete it with git stash drop

23.What is Git stash save?

This command works same as git stash .
but during stashing if you want to print a function then you can use

git stash save “Your stash message”

You can also stash untracked files
git stash save -u
or
git stash save --include-untracked

QUESTIONS

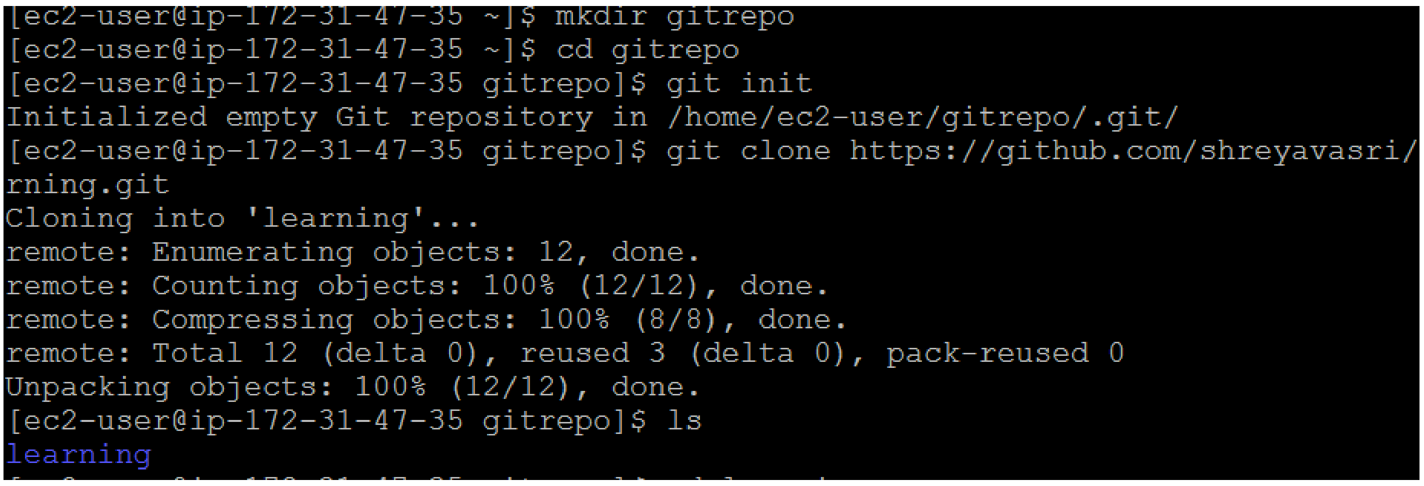
24. What README.MD ? What is its purpose? What does MD stands for?

A readme file, often named "READ ME" to get the user's attention, is a text file containing useful information about a software program. It often accompanies the program's installer or is installed with the program. A typical readme file contains instructions on how to install the program, how to use the basic functions of the program, and what the program does. It may also include a list of recent updates made to the program. Sometimes the readme file will include warnings and other important notices regarding the operation of the program. So when you see a readme file accompanying a new software program, it is best to do what the file says and read it!.

MD : MARKDOWN

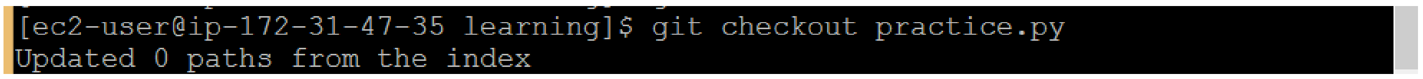
An MD file is a text file created using one of several possible dialects of the Markdown language. It is saved in plain text format but includes inline text symbols that define how to format the text.

25. How to create repository from command prompt?



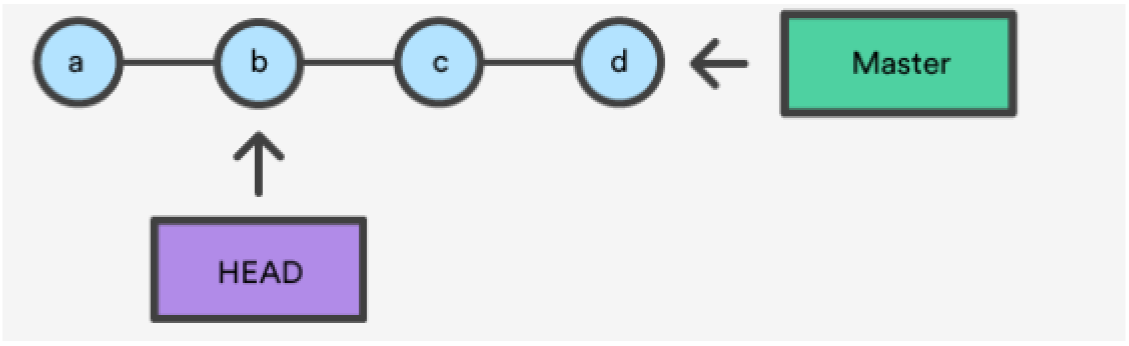
```
[ec2-user@ip-172-31-47-35 ~]$ mkdir gitrepo
[ec2-user@ip-172-31-47-35 ~]$ cd gitrepo
[ec2-user@ip-172-31-47-35 gitrepo]$ git init
Initialized empty Git repository in /home/ec2-user/gitrepo/.git/
[ec2-user@ip-172-31-47-35 gitrepo]$ git clone https://github.com/shreyavasri/
rning.git
Cloning into 'learning'...
remote: Enumerating objects: 12, done.
remote: Counting objects: 100% (12/12), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 12 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (12/12), done.
[ec2-user@ip-172-31-47-35 gitrepo]$ ls
learning
```

26. What is the function of ‘git checkout’ in Git?



```
[ec2-user@ip-172-31-47-35 learning]$ git checkout practice.py
Updated 0 paths from the index
```

A checkout is an operation that moves the HEAD ref pointer to a specified commit. The HEAD ref and master branch ref currently point to commit d. Now let us execute git checkout b



This is an update to the "Commit History" tree. The git checkout command can be used in a commit, or file level scope. A file level checkout will change the file's contents to those of the specific commit.

27. How can you bring a new feature in the main branch?

checkout the branch you want to merge.
git checkout <feature-branch>
merge your updated branch to master
git merge master

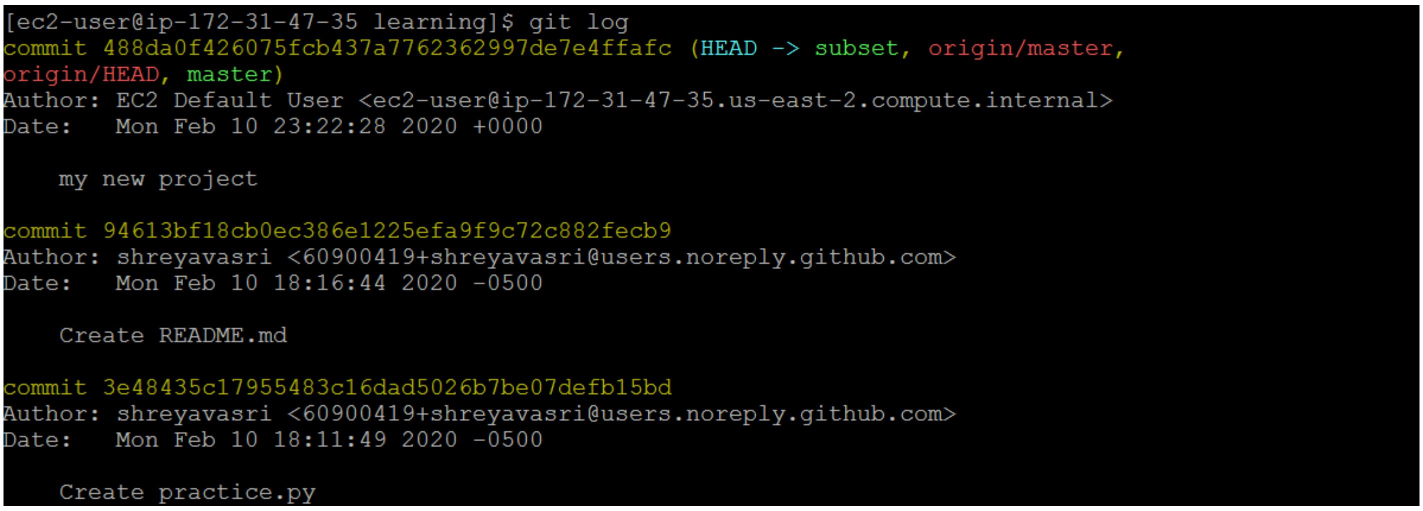
28. What is the function of ‘git rm’?

The git rm command is used to remove files from a Git repository.
git rm can be used to remove files from both the staging index and the working directory. The git rm command operates on the current branch only. The removal event is only applied to the working directory and staging index trees. The file removal is not persisted to the repository history until a new commit is created.

29. What is the function of ‘git stash apply’?

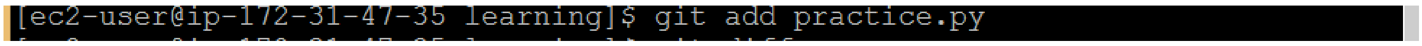
This command takes the top most stash in the stack and applies it to the repository.

30. What is the use of ‘git log’?



```
[ec2-user@ip-172-31-47-35 learning]$ git log
commit 488da0f426075fcb437a7762362997de7e4ffafc (HEAD -> subset, origin/master,
origin/HEAD, master)
Author: EC2 Default User <ec2-user@ip-172-31-47-35.us-east-2.compute.internal>
Date:   Mon Feb 10 23:22:28 2020 +0000

    my new project

commit 94613bf18cb0ec386e1225efa9f9c72c882fecb9
Author: shreyavasri <60900419+shreyavasri@users.noreply.github.com>
Date:   Mon Feb 10 18:16:44 2020 -0500

    Create README.md

commit 3e48435c17955483c16dad5026b7be07defb15bd
Author: shreyavasri <60900419+shreyavasri@users.noreply.github.com>
Date:   Mon Feb 10 18:11:49 2020 -0500

    Create practice.py
```

31.What is ‘git add’ is used for?

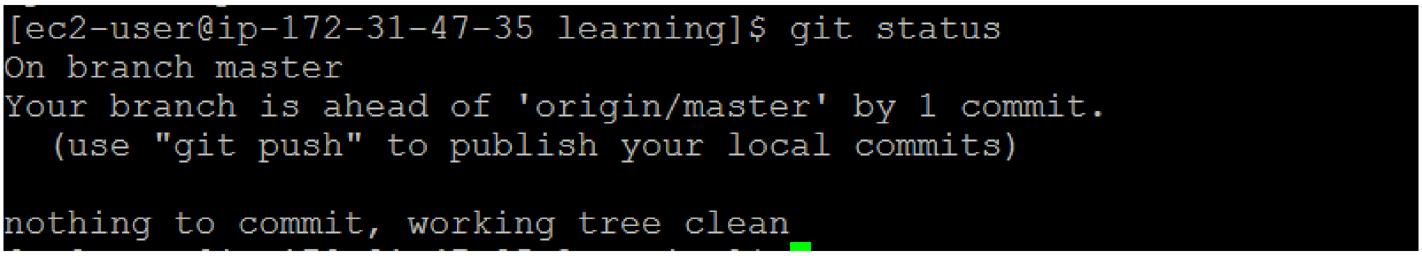


```
[ec2-user@ip-172-31-47-35 learning]$ git add practice.py
```

32. What is 'git diff' is used for?

Diffing is a function that takes two input data sets and outputs the changes between them. git diff is a multi-use Git command that when executed runs a diff function on Git data sources. These data sources can be commits, branches, files and more. This document will discuss common invocations of git diff and diffing work flow patterns.

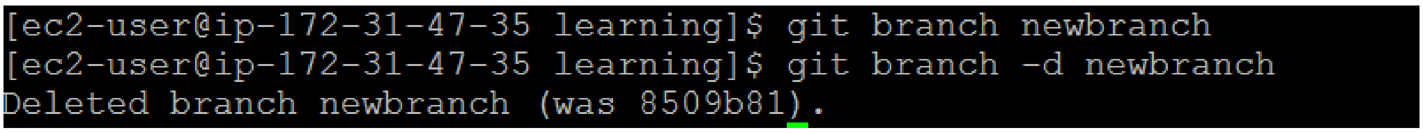
33. What is ‘git status’ is used for?
List the files you've changed and those you still need to add or commit:



```
[ec2-user@ip-172-31-47-35 learning]$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

34. Can we create multiple branch with one command?

35. what is the command that is used to delete a branch?
Delete the specified branch. This is a “safe” operation in that Git prevents you from deleting the branch if it has unmerged changes.
git branch -d <branch>



```
[ec2-user@ip-172-31-47-35 learning]$ git branch newbranch
[ec2-user@ip-172-31-47-35 learning]$ git branch -d newbranch
Deleted branch newbranch (was 8509b81).
```

Delete the specified branch. This is a “safe” operation in that Git prevents you from deleting the branch if it has unmerged changes
git branch -D <branch>

36. What is another option for merging in git?
As an alternative to merging, you can rebase the feature branch onto master branch using the following commands:

git checkout feature
git rebase master

git rebase

37. How to remove a file from git without removing it from your file system?
# git rm -rf --cached $FILES

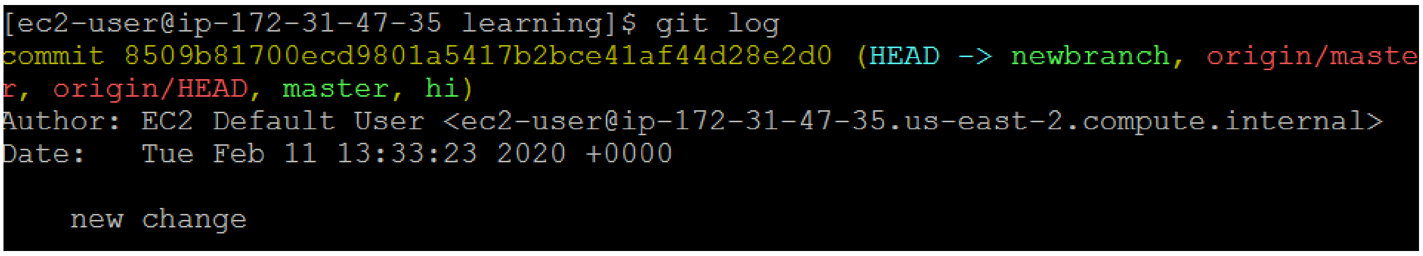
38. Use of "git rebase" instead of "git merge"?
Both of these commands are designed to integrate changes from one branch into another branch—they just do it in very different ways.
rebasing re-writes the project history by creating brand new commits for each commit in the original branch.
The major benefit of rebasing is that you get a much cleaner project history. First, it eliminates the unnecessary merge commits required by git merge
rebasing also results in a perfectly linear project history

39.What is a repository in Git?
Git is a program that tracks changes made to files. Once installed, Git can be initialized on a project to create a Git repository.

A Git repository is the `.git/` folder inside a project. This repository tracks all changes made to files in your project, building a history over time. Meaning, if you delete the `.git/` folder, then you delete your project’s history.

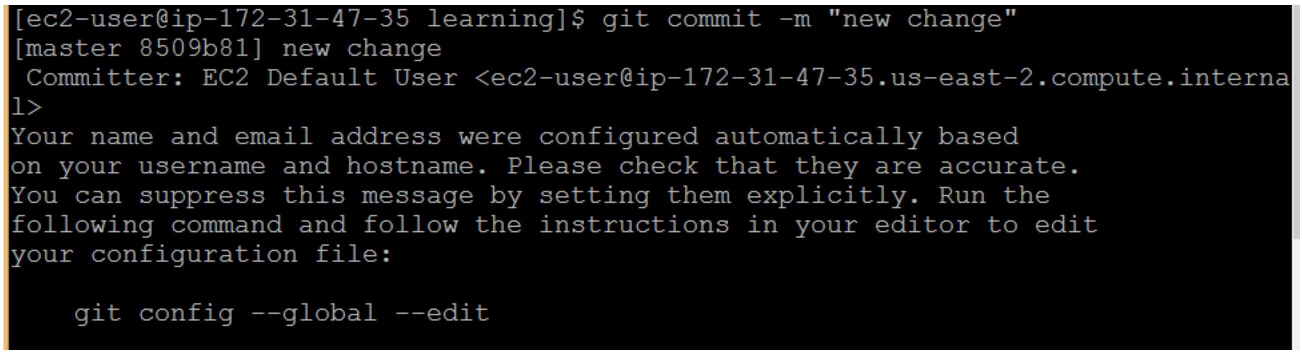
40.What does commit object contain?

The commit object contains the directory tree object hash, parent commit hash, author, committer, date and message

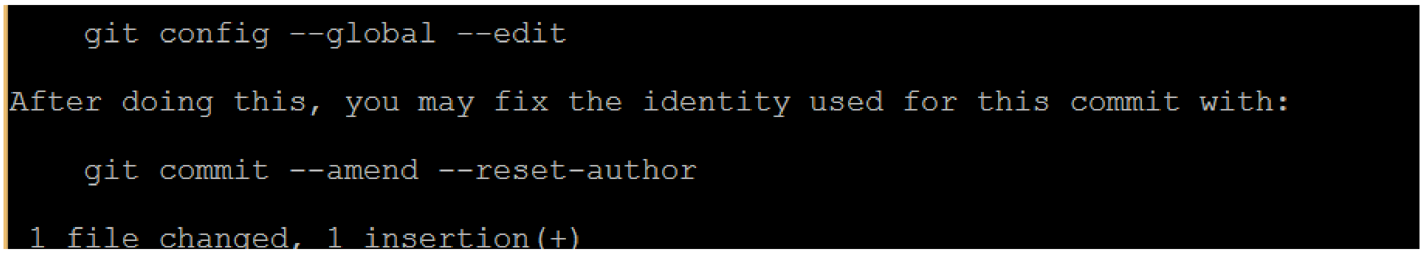


```
[ec2-user@ip-172-31-47-35 learning]$ git log
commit 8509b81700ecd9801a5417b2bce41af44d28e2d0 (HEAD -> newbranch, origin/maste
r, origin/HEAD, master, hi)
Author: EC2 Default User <ec2-user@ip-172-31-47-35.us-east-2.compute.internal>
Date:   Tue Feb 11 13:33:23 2020 +0000

    new change
```

41.Command used to write a commit message?



```
[ec2-user@ip-172-31-47-35 learning]$ git commit -m "new change"
[master 8509b81] new change
 Committer: EC2 Default User <ec2-user@ip-172-31-47-35.us-east-2.compute.interna
l>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit
```



```
    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 1 file changed, 1 insertion(+)
```

42.Write one use-case of Github?

A team of 4 developers working on a same web applcations .
2 people are working on designing front and two are working on bakend.
All of them need to commit there code at one place and generate new release every day for QA.

43.Name some alternative of Git?
GitKarken
AWS CodeCommit
Cloud Source by google
Apache Allura

44.What is a gist in Git?

Every gist is a Git repository, which means that it can be forked and cloned. If you are signed in to GitHub when you create a gist, the gist will be associated with your account .

45.What is a gist programming?

46.Name any two Git repository hosting services which are common?
GitHub
Bitbucket

06

NOTE

The last date for submission of assignments is TOMORROW.

APPRECIATED