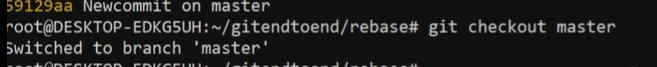
1.what is the difference b/w git merge and rebase?

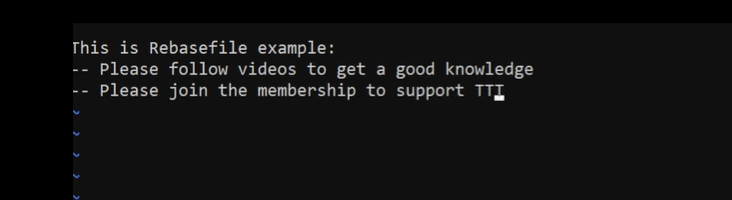
2.

wt is git rebase

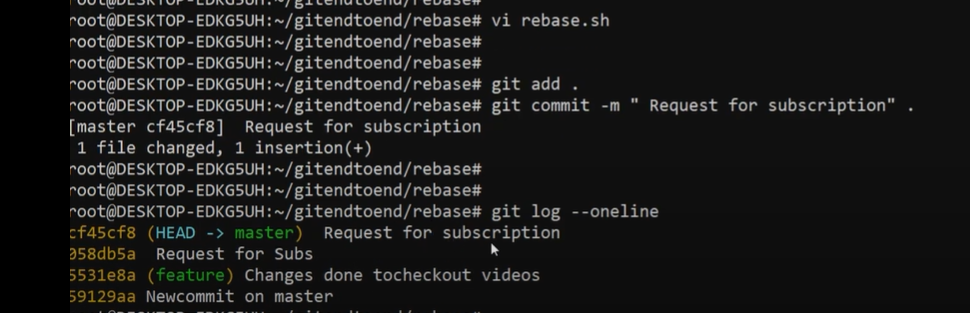
1.step switch master branch



2,step edit file & added that file & commit that file

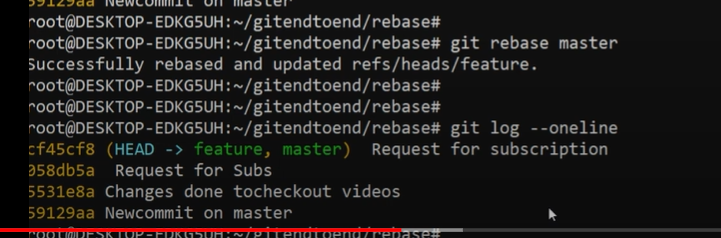


3.again modified in same file&added & save that file



4.step

swith feature &latest commited files added in feature brach &check log in feature branch



Git merge:using for merging all  commits changes to another branch.

<https://www.atlassian.com/git/tutorials/merging-vs-rebasing>

<https://www.baeldung.com/git-merge-vs-rebase>

Git merge:using for merging all  commits changes to another branch.

Wheather branches are created for testing,bug fixes or other reasons,

It takes contents of a souce branch and integrates it with  target barce

Git rebase: using that allows develpers to integrate changes from one brach to another,

Git rebase compress all the changes into a single patch.and then integrate with one the target branch.

Unlike merging,rebase flattens history.it transfer the completed work from one branch to anthoer process,unwanted history eliminated

**Git merge is a command that allows you to merge branches from Git.** **Git rebase is a command that allows developers to integrate changes from one branch to another**.

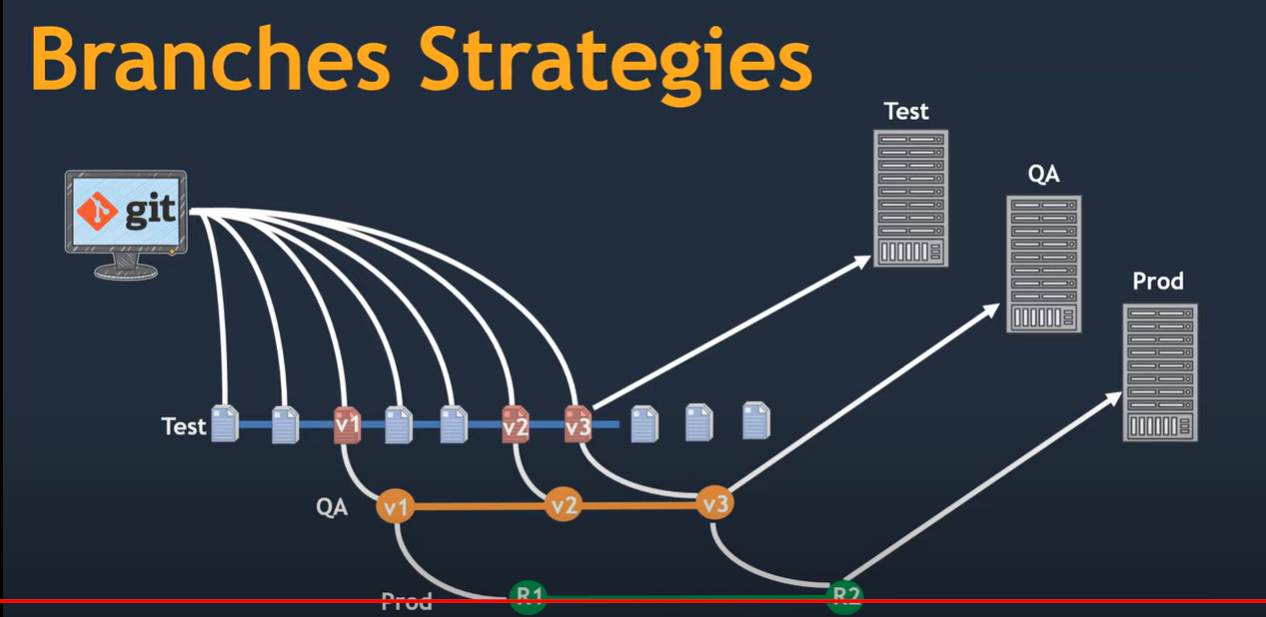
2.what is the difference b/w git pull and fetch?

Git Fetch is the command that tells the local repository that there are changes available in the remote repository without bringing the changes into the local repository. Git Pull on the other hand brings the copy of the remote directory changes into the local repository.

3.what is git cherrypick?

git cherry-pick is **a powerful command that enables arbitrary Git commits to be picked by reference and appended to the current working HEAD**. Cherry picking is the act of picking a commit from a branch and applying it to another. git cherry-pick can be useful for undoing changes.

4.How many branch strategies in git?



1.we have development system & we have test branch and QA & Prod & we have testing env servers & quality servers & prod server

2.our prod system going to used by end user or clients

3.before deploying the code into the production check the code is working for that checking the test code in test env & Quality analysis ENV

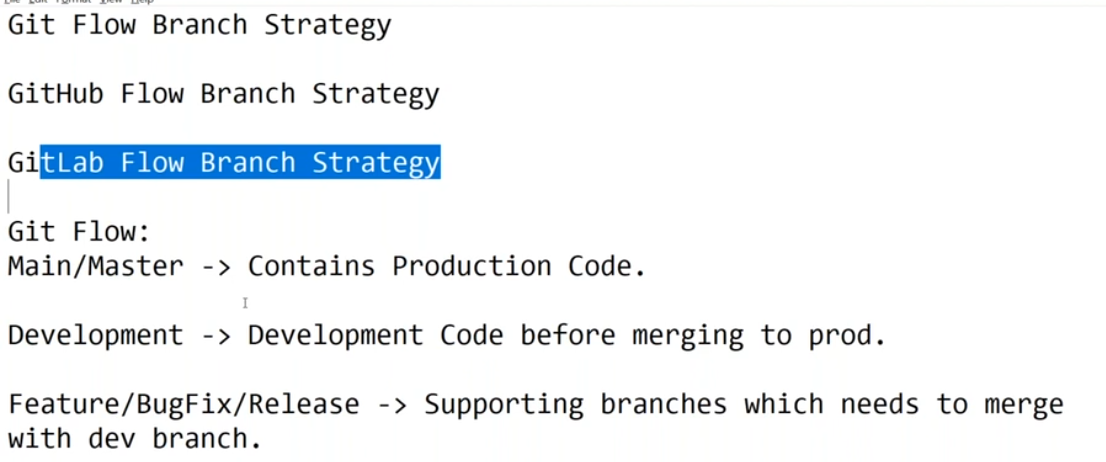
4.we have one peace of code is ready & then check the local system nothing but laptop it is working fine but we dont have confident working well  in production env

5.it is doesnt make sense to push our code to prod  branch directly & git deploy in the production system

<https://www.gitkraken.com/learn/git/best-practices/git-branch-strategy#:~:text=Git%20Flow%20Branch%20Strategy&text=There%20are%20five%20different%20branch,Feature>

* GitFlow.
* GitHub Flow.
* GitLab Flow.
* Trunk-based development.
* but

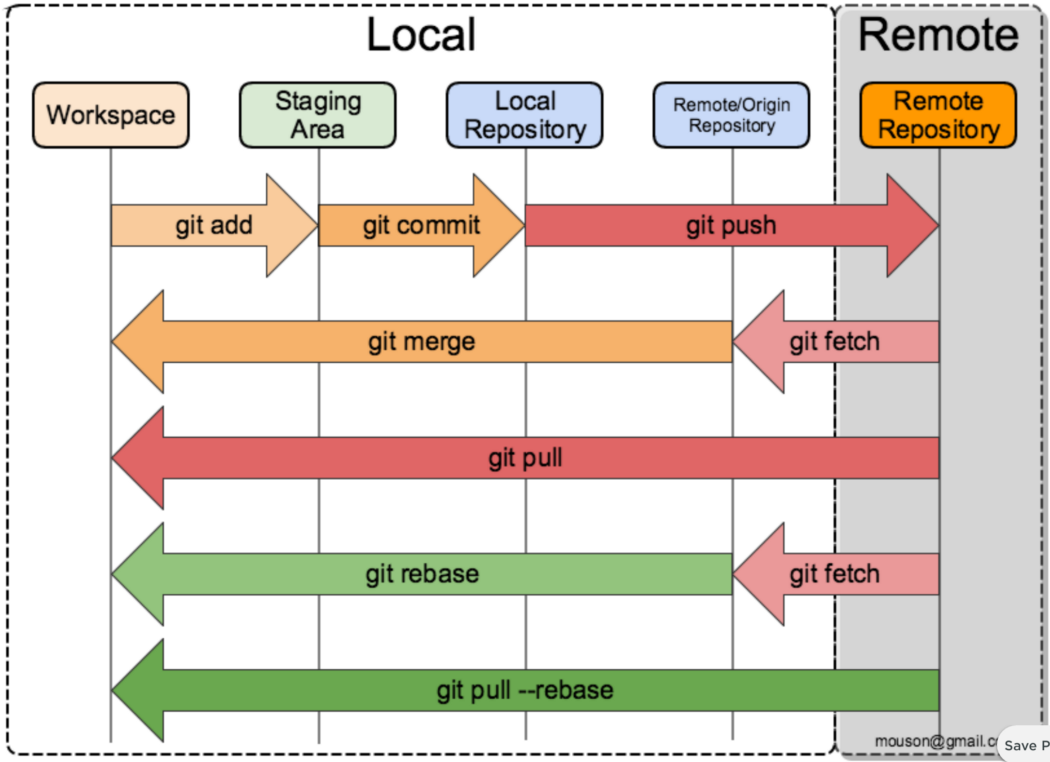
we are folwing git flow strategies



based upon the requirement we are creation supporting branches and the testet & then

we are merge into the devlopment test with cloud env & then code merge from development to master branch

5.what are the stages in git?



6.what is the git fork vs clone?

**A fork creates a completely independent copy of Git repository.** **In contrast to a fork, a Git clone creates a linked copy that will continue to synchronize with the target repository**.

7.what is git remote add origin?

**To verify that the remote repo was added to your configuration**, use the git remote –v command. This command will show that GitHub is the fetch and push targets of the local repository.

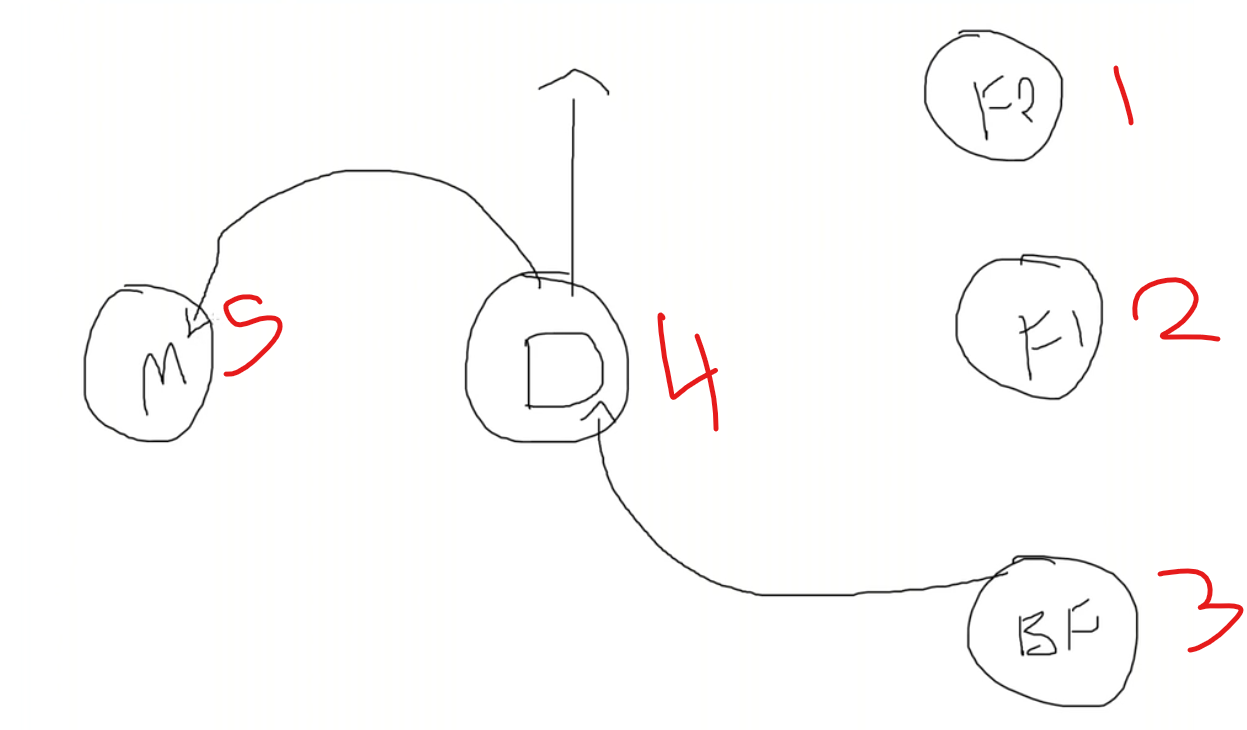
8. what is the diff between git pull and clone?

git pull (or git fetch + git merge ) is how you update that local copy with new commits from the remote repository. git clone is used for just downloading exactly what is currently working on the remote server repository and saving it in your machine's folder where that project is placed.

9.how to revert master branch code if it is failure job  in git hub?

it is dangerous

it is not directly come to prod env



code copy feature brach check code

do bug fixing in feature branch& merge to development & test azure or aws account env if is running properly and then merge to master branch

we are not moving running production env

there are different typo of deployment

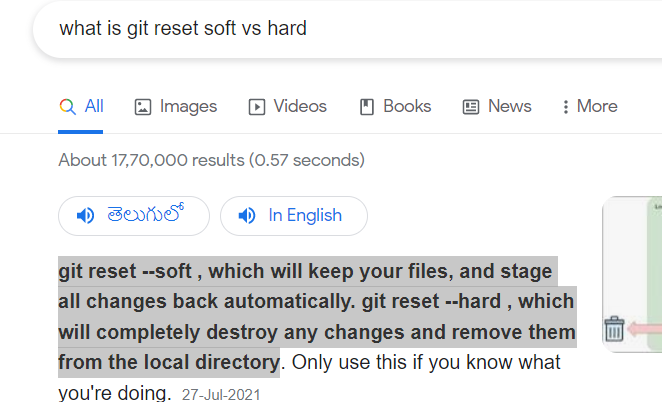
rolling update

blue green deployment

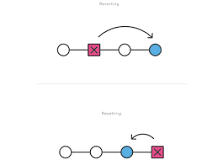
canary deployment

in blue green deployment create separate environment & deploy the master code & then we move the traffic via  load balancer

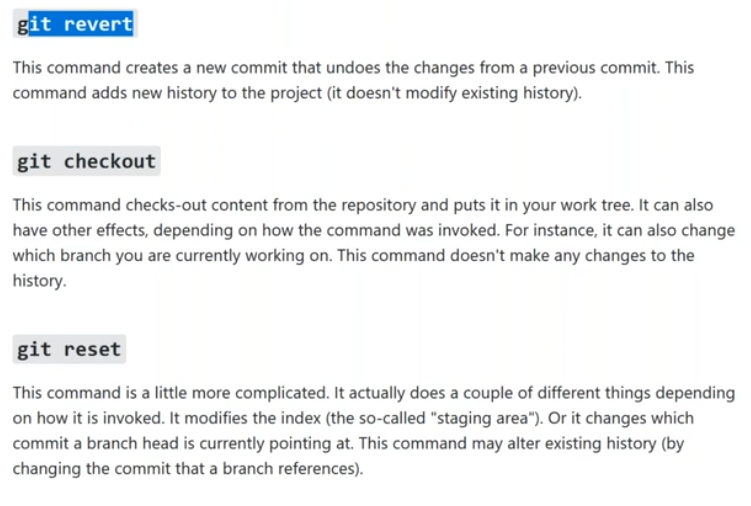
10.



11.wt is difference between the git revert vs git reset?



The git revert command is **a forward-moving undo operation that offers a safe method of undoing changes**. Instead of deleting or orphaning commits in the commit history, a revert will create a new commit that inverses the changes specified. Git revert is a safer alternative to git reset in regards to losing work.



what is git fork…?

ANS:- Git fork command is used like you can create a entire copy of Original One

          Git fork praveen

1. git init - Initializes a new Git repository.
2. git clone [repository] - Creates a copy of a remote repository on your local machine.
3. git add [file] - Adds a file to the staging area, preparing it for a commit.
4. git commit -m "[message]" - Creates a new commit with the changes in the staging area and adds a message describing the commit.
5. git status - Shows the status of the repository, including modified files, new files, and files in the staging area.
6. git log - Shows a log of all commits in the repository.
7. git diff - Shows the differences between the current version of the files and the last commit.
8. git branch - Shows all branches in the repository.
9. git branch [branch name] - Creates a new branch with the given name.
10. git checkout [branch name] - Switches to the specified branch.
11. git merge [branch name] - Merges the specified branch into the current branch.
12. git pull - Fetches the latest changes from the remote repository and merges them into the current branch.
13. git push - Pushes the local changes to the remote repository.

47.Git Stash…?

Ans:-  Git stash command is used like to hidden something Secretly if your in the Middle of the Work

           Not ready to push and commit those commits are you can hold it before the commit.. so that

           the purpose we are using the Git stash command

          Note:- if your Retrive back the commits those are Stashed commits Using the command like

          Git Stash Apply

examples

How do I use 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**. For example: $ git status On branch main Changes to be committed: new file: style. css Changes not staged for commit: modified: index.

**Here's the sequence to follow when using git stash:**

1. Save changes to branch A.
2. Run git stash .
3. Check out branch B.
4. Fix the bug in branch B.
5. Commit and (optionally) push to remote.
6. Check out branch A.
7. Run git stash pop to get your stashed changes back.