git –version 🡪git version 2.20.1.windows.1

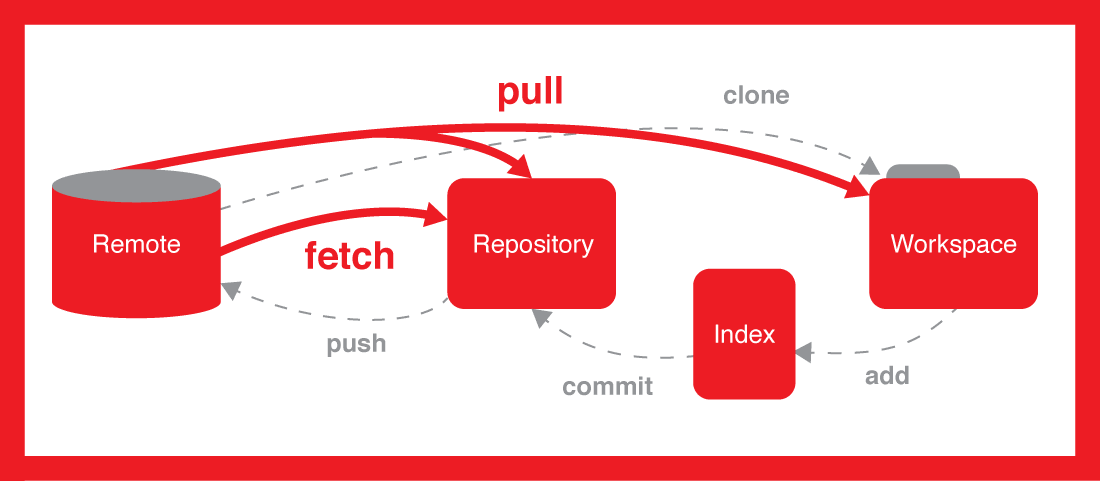
Git allows a team of people to work together, all using the same files. And it helps the team cope with the confusion that tends to happen when multiple people are editing the same files..

Hooks::git related automation script

clone vs pull?

git clone: clone is basically one time activity, it is used to get a local copy of an existing remote repository to work on.

git pull(git fetch + git merge): git pull is how you update the local copy with new commit from the remote repository/pull the conges and add to local repo.

git fetch origin master/URL 🡪to get the files from github to local repo

git pull origin master/URL

git clone URL

git push origin master/URL

**General workflow is as follows:**

You clone the Git repository as a working copy.

You modify the working copy by adding/editing files.

If necessary, you also update the working copy by taking other developer's changes.

You review the changes before commit.

You commit changes. If everything is fine, then you push the changes to the repository.

After committing, if you realize something is wrong, then you correct the last commit and push the changes to the repository.

**Git Conflict** arises when the commit that has to be merged has some alternate in a single place, and the current dedicate additionally has a change on the same place.

git clone -b master URL directory\_name🡪 **To clone a specefic branch:**

find / -name git🡪 **Check Git is installed**

git config --global user.email "dheeraj.gundraserver@infor.com"

git config --global user.name "dheeraj"

git config --global color.ui auto

git config --global --list --> this will list back the user & password

-------

git init🡪 to create an empty repo

touch file.txt

git status🡪to check the status of the file/dir

git add file.txt or git add . 🡪to add the changes to the staging area

**Let us assume, now we want to revert back(after git add) then**

git reset HEAD file.txt🡪revert after adding to staging area

git checkout -- file.txt🡪revert the changes in the file

---------------------------------------

git commit -am "my first commit"🡪add to staging are and commit

git push -u origin master

git log --oneline🡪to view the logs in one line

git show [commitID]🡪to view the metadata and content changes

git ls-files -->

git tag 1.0 [commitID]🡪tag will be created for the commit

**To remove file from git :**

git rm <filename>🡪removes the file from the repo but also deletes its from the local file system.

git rm -cached <file name>🡪 removes the file only from the Git repository and not remove it from the filesystem.

git commit -m "deleted the file"

git push -u origin master

**To view all the files in the master:**

git ls-tree -r master

git ls-tree --name-only -r master

**If you want to ignore some log file in git(which not required for push to master):**

create a .gitignore

touch .gitignore

git add .gitignore

open the .gitignore file & add \*.log

then commit the file

git commit -m "git ignore file is added"

git push -u origin master

**For creation of New branch(here i am creating a branch name called "testing"):**

git branch [Branch name]🡪create branch

git checkout [Branch name]--> Switched to branch 'testing'

git checkout –b [Branch name]🡪create and switch branch

git diff🡪to check the difference a file

git rebase [Branch Name]🡪it replace a series of commits one branch on top of another, directely merge without commit.

git merge [Branch Name]🡪merge data from one branch to another

**Working on GIT Stash :**

git stash/push🡪temporally stashes changes you have made to working copy

git stash show/list - list the modifications stashed away.

git stash apply - To restore the modifications stashed away.

Git stash drop🡪to clean the stash id

Git stash clear🡪to clear all the files in stash

git stash pop🡪apply + drop

For example :

git checkout testing

touch teststash.txt

git add teststash.txt

Now, if you want to move to master without commit the teststash.txt file in branch testing , then it will be big issue...so You need to use stash..

git clone <https://github.com/dgundra/firstproject.git>

git remote –v 🡪it will connect local repo to remote repo

origin  https://github.com/dgundra/firstproject.git (fetch)

origin  https://github.com/dgundra/firstproject.git (push)

**Revert/Reset/checkout?**

Git revert: This command creates a new commit that undoes the changes from a previous commit. This command adds new history to the project (it doesn't modify existing history).

Git reset: Git reset is a simple way to undo changes that haven't been shared with anyone else.

git checkout: The git checkout command is used to update the state of the repository to a specific point in the projects history.

git reset --soft A, will change the commit history and repository; staging and working directory will still be there.

git reset --mixed A, will change the commit history, repository, and staging; working directory will still be there.

git reset --hard A, will change the commit history, repository, staging and working directory; you will go back to the state of A completely.

git revert HEAD🡪 revert the previous commit

git revert –n master~5 master~2

git branch –d [BranchName]🡪delete local repo/branch

git push origin –delete [BranchName]🡪delete the branch in remote repo