**Git Workflow:**

We can store the code in repository.

**Working Directory:**

All the files are untracked in this stage.

**Staging Area:**

* We can send the files or code to the staging area by using **“git add filename” or “git add . ”** command.
* All the files are tracked in this stage.

**Local Repository:**

We can push the changes to the local repo from staging area by using **“git commit -m “commit message” “** command.

**Remote Repository:**

We can push the changes from local repo to remote repo by using “**git push repositoryurl”** command.

**GIT Commands:**

**Git clone repositoryurl:**

This command will download the entire repository from remote repo into the local repo.

**Git Pull repourl:**

This command will download the only modified files from remote repo into the local repo.

**Git Log:**

This command is used to display the commit history like commit id,commit message,author name,date.

**Git log --oneline:**

This command is used to display the one line only like commit id,commit message.

**Git log --1:**

This command is used to display the latest commit history.

**git Config --global user.name “username”:**

This command is used to configure the git in our local system.

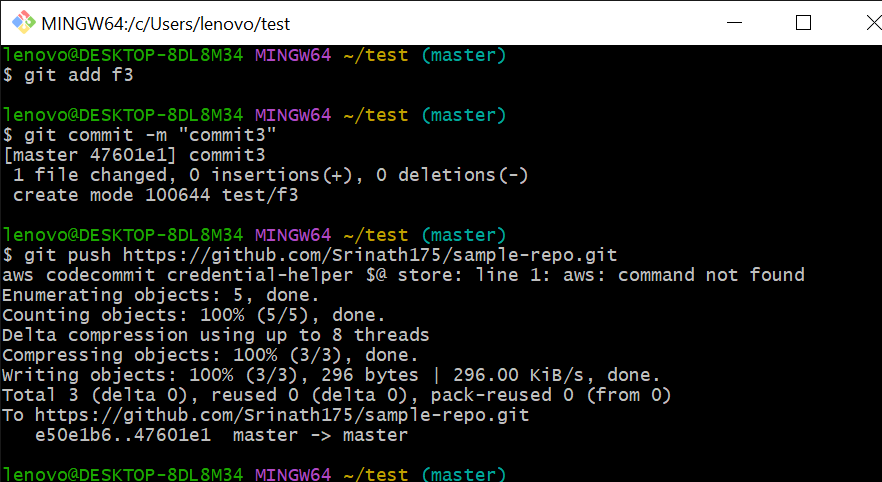
**git remote add origin remoterepositoryurl:**

This command is used to add the remote repo to the local repo.It will establish connection b/w local and remote repo.

(Instead of **origin** we can also give any name in the above command)

**Git remote -v:**

It will list the remote repository url.



**Git Merge:**

This command is used to merge the changes from one branch to another branch.

First we need to switched to the branch( to which branch we want to merge the changes)

We need to execute the **git merge branchname** command( here we need to specify the branch name from which branch we want to merge the changes).

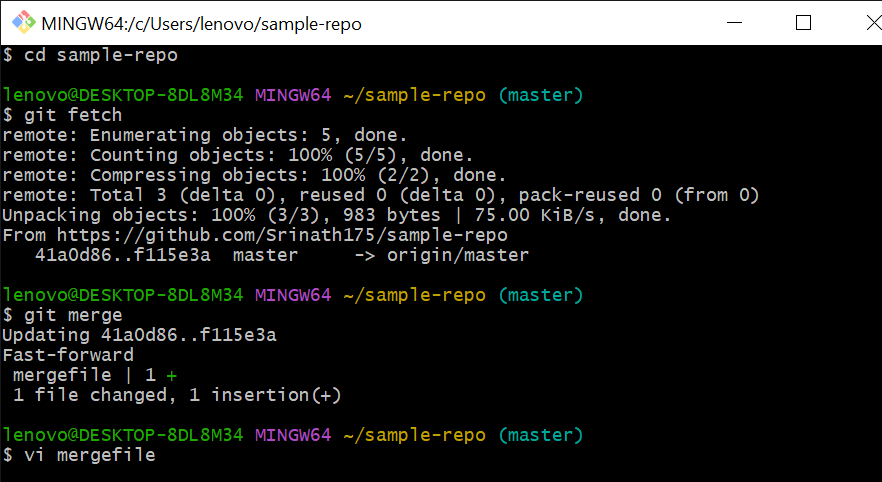
**Git cherry-pick commit\_id:**

This command is used to merge the particular commit from one branch to another branch

**Git Fetch:**

This command is used to download the modified files from remote repo to local repo but not downloaded into working directory.

We need to use **git merge** command after the **git fetch** command to download the modified files into working directory.



**Git branch branchname:**

This command is used to create the new branch.

**Git checkout branchname** :

This command is used to switched to the particular branch.

**Git rebase command:**

It is also used to merge the changes from one branch to another branch.

**Difference b/w Git merge and Git rebase:**

When we use the **git merge** command it will create the one extra commit or new commit(merge commit) which represents the merge has happened.

Whereas when we use **git rebase** command it will not create the extra commit(merge commit).

**Git Revert:**

This command is used to remove the commits or modify the commit message from local repo or remote repo.If we made a mistake in commit message we can modify that message by using this command.

Command**: Git revert commit\_id**

**Git Rebase:**

It is also used to remove the commits or move the commits from one area to another area in git.

**Types:**

**Git reset –soft commitid:**

This command is used to move the files from local repo to staging area.

**Git reset –hard commitid:**

When we use this command it will remove the commits or files from working directory as well.

**Git reset –mixed commitid:**

This command is used to move the commits or files from local repo to working directory.

**Merge Conflicts:**

When two developers are working on the same lines of code in a same file at that time conflicts occurred.

**Git Stash:**

It is like temporary storage developers can save their code at some place he don’t want to commit these changes into local repo by using **git stash** command.

Example:

Developer is working on one branch like writing or modifying the code or file but he needs to work on or switch to the another branch without affecting these current changes because his work is in incomplete in his current branch he can not commit this changes to local repo at that time he can store these changes at some place by using **git stash**.

**Git stash apply:**

It will reapply or move the changes or files from stash to staging area without removing these changes from stash.

**Git stash pop:**

It will reapply the changes or files from stash to staging area but it will remove these changes from stash.By default It will apply on latest stash(stash@{0})

**Git stash pop stash@{1}** which means this command is apply on previous stash.