Git commands –

1. To remove the untracked file from git – git clean -f (forced fully clean)
2. Different git status

Working directory

Staging/Indexing area

Local Repository

Remote Repository – Bitbucket

1. 1 file is modified in working directory and is added to staging area, now but then wanted to remove file from staging area and again placed it in working directory with changes – cmd -

Git reset

This is mixed mode reset command.

1. Now changes are committed to local repository using cmd – git commit -m “msg”

But want to revoke this commit and placed this changes to staging area – use cmd –

Git reset –soft HEAD~1

1 indicate that number of commit from HEAD pointer need to revoke.

Hence commit is removed, and changes are in staging area.

1. In git reset and git soft mode changes are not lost.
2. To completely remove those committed changes use reset in hard mode – use cmd –

Git reset –hard HEAD~1

It will remove the commit and changes also from git.

1. Reset cmd can also used with commit ID,

From that commit ID above commits gets revoked

|  |
| --- |
| Commit id 4 – HEAD |
| Commit id 3 |
| Commit id 2 |
| Commit id 1 |

Lets use cmd – git reset –soft commit id 2

This cmd will revoke the commit id 3 and commit id 4 changes and HEAD will point to commit id 2 changes. And commit id 3 and id 4 changes placed in staging area.

Example - <https://kodekloud.com/blog/git-uncommit-last-commit/#:~:text=The%20answer%20is%20simple%3A%20use,branch)%20to%20the%20previous%20commit>.

1. To get the history of commands / actions performed on git – use cmd –

Git reflog

1. Merge the feature branch to main branch -using manual merge.

Steps – standing in feature branch git bash console.

->changes did in feature branch

->git commit changes and git push to feature branch

Now suppose master branch is updated with new commits, so to merge feature branch in master branch we need to first add extra commit from master to feature branch manually.

->Git checkout master

->Git pull – latest extra commits are now present in local repository’s master branch.

->Git log

->Git checkout feature branch

->Git merge master – master branch’s extra commits are added to feature branch now.

->may get conflicts if any

->solve the conflicts and save changes. Do git add, git commit changes, git merge master

->Git log

->Git push origin feature branch.

->Raise pull request. – src [feature branch] and destination [master branch]

Do fast forward merge.

That is manually merging.

1. To forcefully /intentionally push the code.

Cmd- git push origin master -f

1. Normal merge and rebase merge-

Both is having same purpose, to add the commits from multiple branch to main branch.

Bcoz of rebase merging, it maintains sequence of linear commits history.

Steps to do rebase merge –

->git checkout master

->git pull

->git checkout feature branch

->git rebase master

->git push origin feature branch -f

Bcoz the tip of the branch is going to change using command rebase.

1. Squash – squash means combining multiple commits into one commit after rebase.

Cmd – git rebase -i HEAD~2

-i = interactive mode

~2 = combine latest topmost 2 commits into 1

After enter

Pick commit1

Squash commit 2

Means add changes of commit 2 into commit 1 changes

Add commit message

Git push origin feature branch – f

1. How to edit a last commit?

User have 2 files in working directory. Committed and pushed 1 file and missed to commit 2nd file.

To add missed file in topmost commit, use command.

->git add missedFile

->git commit --amend

Will ask message

File is added to topmost commit.

Git push origin branchname -f

To view the changes of topmost commit –

->git show

OR  
to add missed file and change the commit msg at a time use below command –

->git commit –amend -m “new commit msg”

1. How to move commit from one branch to another?

Get the commit id of required commit.

Go to branch where need to add above commit .

Cmd – git cherry-pick commitId

Git push origin feature branch

Done

1. Git bisect command

Used if multiple commits are performed and application is not working, in that case to find which commit is causing an issue , bisect command is used

Bisect command works in binary search algorithm.

Need to provide bad commit and good commit.

Git will randomly choose 1 commit in between bad and good commit.

And now branch’s HEAD is pointing to chosen to commit.

Users need to check whether application is working till chosen commit.

If yes, then acknowledge by saying that “git bisect good”.

Then fault commit again need to bisect. And it continues to.

Cmd –

->git bisect start

->git bisect bad commitId

->git bisect good commitId

Gave 1 commit with message – current branch pointing to this commit

Check application is working as expected or not.

->git bisect bad – if given commit is also bad

It will again gave another commit. If given commit is good and application is working say

->git commit good

And git will find out fault commit.

To come out from bisect mode use cmd – git bisect reset

Git will point to latest commit now.

1. To checkout branch on a particular commit use cmd – git checkout commitId

The head is changed, if reset it use cmd – git checkout master

1. Create new branch from particular commit use cmd –

Git checkout commitId - detached branch created on local repository.

git checkout -b branchname – created new branch on local repository.

1. Git stash cmd-

To save working directory changes use cmd – git stash

To retrieve the stashed changes into working directory use cmd – git stash pop

1. Check the changes made in file use cmd – git diff
2. To identify who contributes to work on file1 use cmd – git blame file1
3. To revert commit –

We can use git reset cmd also – reset cmd will remove the commit and history will not show that commit the reverted, its gone

But sometimes it is recommended that removing any commit should also maintain in the history of commits. For that use cmd –

git revert commitId

Git push origin branchName

Git will create 1 commit which contains reverted changes of mentioned commit id.

1. Git fetch – use to download all remote repository changes to local repository.

To check all downloaded changes use cmd – git checkout origin/branchName

That gives all downloaded changes, if changes are looking good then use cmd – git merge origin/branchName

OR

Same is done using cmd – git pull

NOTE : git pull = git fetch + git merge origin branchName

1. Tag

To create a tag – git tag tagName

To verify tags – git tag

To push created tag to remote repo. – git push –tags

To create tag with its metadata – git tag -a tagName -m “message”

To create tag for any commit – git tag tagname commitId

To checkout particular tag – git checkout tagName

1. Source tree tool is similar to Git GUI – we can perform multiple operations from UI instead of git commands.
2. Integrating JIRA tool with bitbucket.