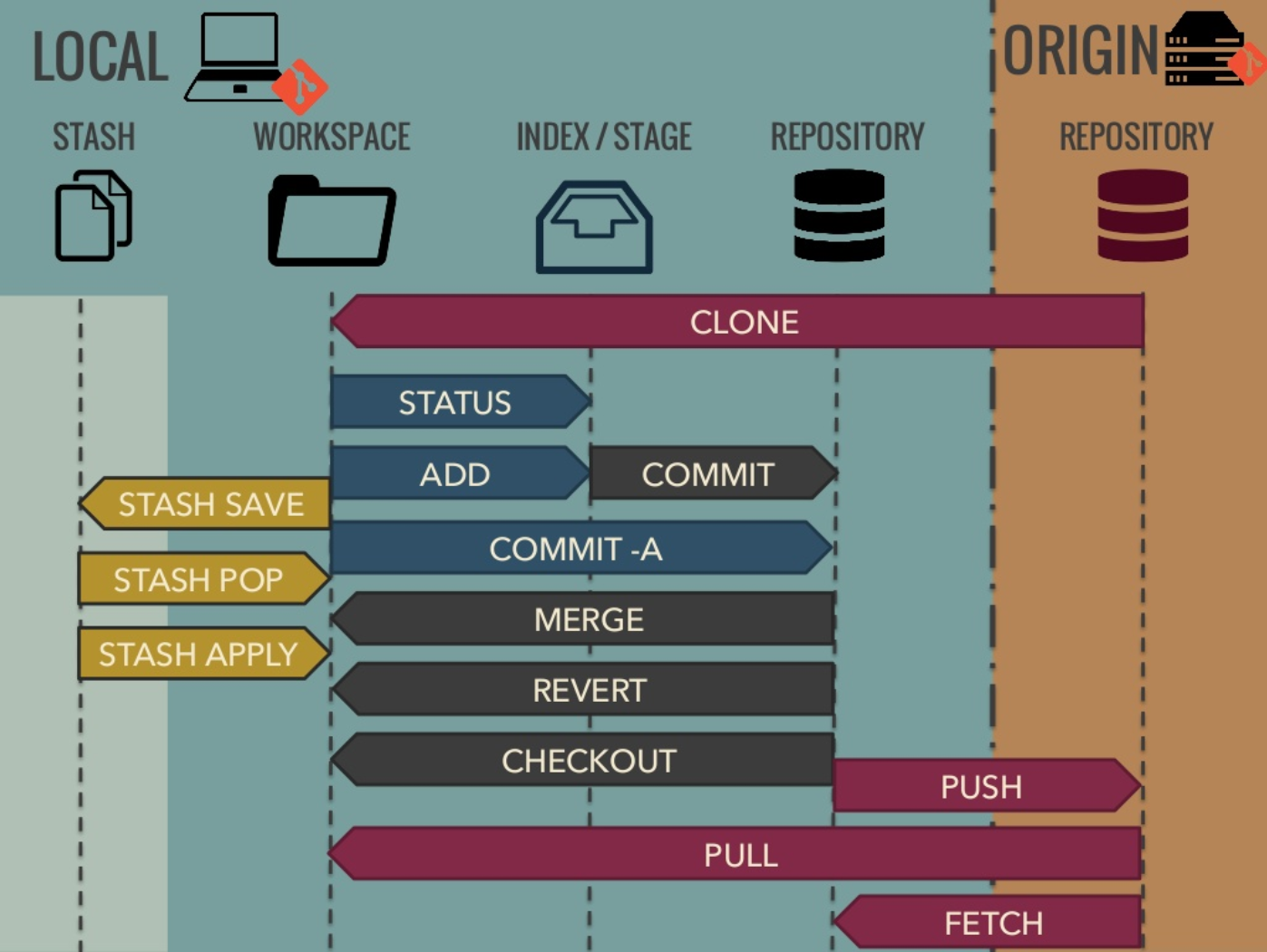
LEARNING GIT

1. Introduction to GIT through diagram and Introduction to basic commands only through diagram.



1. Let’s make the changes to working directory and push these changes using commands.

See the tentative list of commands.

Git add <file name>

Git commit –a –m”<commit message>”

Git push origin branch name.

1. Let’s see dash board or GIT hub webpage.

Write the issues.

Set labels.

Write title 50 words.

Write description 72 words.

1. Making branches and see changes in content.

What is big deal here, can we do complete project and then put in the GIT wrong.

Or

Every time in make separate working folder work there and copy paste in GIT working folder and push the changes?

1. Work feature wise.
2. Set appropriate labels like bug fix etc.
3. Work for only the given branch /task i.e. issue wise. Don’t make any changes unless asked to do so.
4. Things done above are technically ok but professionally not!!
5. Always make branch name as

Feature/branch name (or)

Bug\_fix/branch name (or)

Enhancement/branch\_name etc.

1. In the very first commit of the project.

we must write commit message as “initial commit”

then afterwards ,

git commit –m “[#issue number]<commit\_messgae>”.

1. Prefer to make new branch following the previous branch.

git branch checkout –b <new branch> <branch\_to\_be followed>

1. Always do update the local master or main branch using git fetch –all and rebase to the master/ main branch using

git rebase master.

In case of conflicts resolve it.

One may observe <<<<<<<< HEAD is the start of conflict

And, below this line ====================== is the alternate content

And >>>>>>>>>>>>> here conflicts ends.

Now we need to choose which one to keep and which one to discard i.e. content above the line or below of it or both.

Then,

git rebase --continue

1. Few more git commands.

git status => to know which files have been changed.

git stash => to save the present work with locally.

git stash pop => to get the saved work.