# Git Cheat-sheet

Git commands and workflow in short

| | | |
|---|---|---|
| git add | Add changes to the staging area | • git add <file><br>• git add <file1> <file2><br>• git add . |
| git commit | Commit staged changes | • git commit –m “message”<br>• git commit –am “message” |
| git status | Compares latest commit with current changes | • git status |
| git log | Shows the commit log | • git log<br>• git log –oneline |
| git branch | Shows all the branches of the repository | • git branch<br>• git branch –a<br>• git branch –M <new_bname><br>• git branch –D <branch><br>• git branch -r |
| git switch | Changes the current working branch | • git switch <branch><br>• git switch –c <branch> |
| git merge | Merges current branch with a specified branch | • git merge <branch><br>• git merge –abort |
| git stash | Save changes that you are not ready to commit in the stashing area | • git stash<br>• git stash pop<br>• git stash apply<br>• git stash list<br>• Git stash clear<br>• Git stash drop ‘stash@{i}’ |

| | | |
|---|---|---|
| git checkout | Checkout something. (branches,commits, …) | • git checkout <chash><br>• git checkout <branch><br>• git checkout HEAD |
| git restore | Undo changes in our local repository since our last commit | • git restore <file><br>• git restore –source HEAD <file><br>• git restore –staged <file> |

When we collaborate with other people we use **revert**!

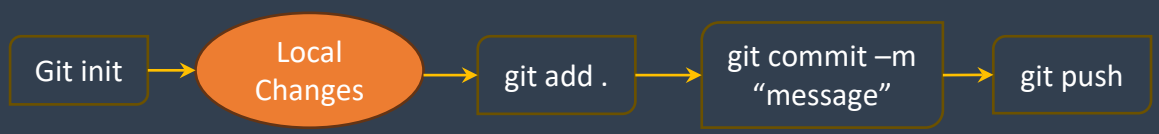
| | | |
|---|---|---|
| git reset | Mover the branch pointer in a past commit. Reseting changes. | • git reset <chash><br>• git reset –hard <chash><br>• git restore –staged <file> |
| git revert | Creates a new commit which reverses the changes until the specified commit. | • git revert <chash> |

| | | |
|---|---|---|
| git restore | Undo changes in our local repository since our last commit | • git restore <file><br>• git restore –source HEAD <file><br>• git restore –staged <file> |
| git clone | Clones a remote repository to your local machine. | • git clone <url> |
| git remote | Add a remote repository reference. | • git remote –v<br>• git remote add <name> <url> |
| git push | Push commits from the local repo to the corresponding remote | • git push <remote> <branch><br>• git push –u <remote> <branch><br>• git push |

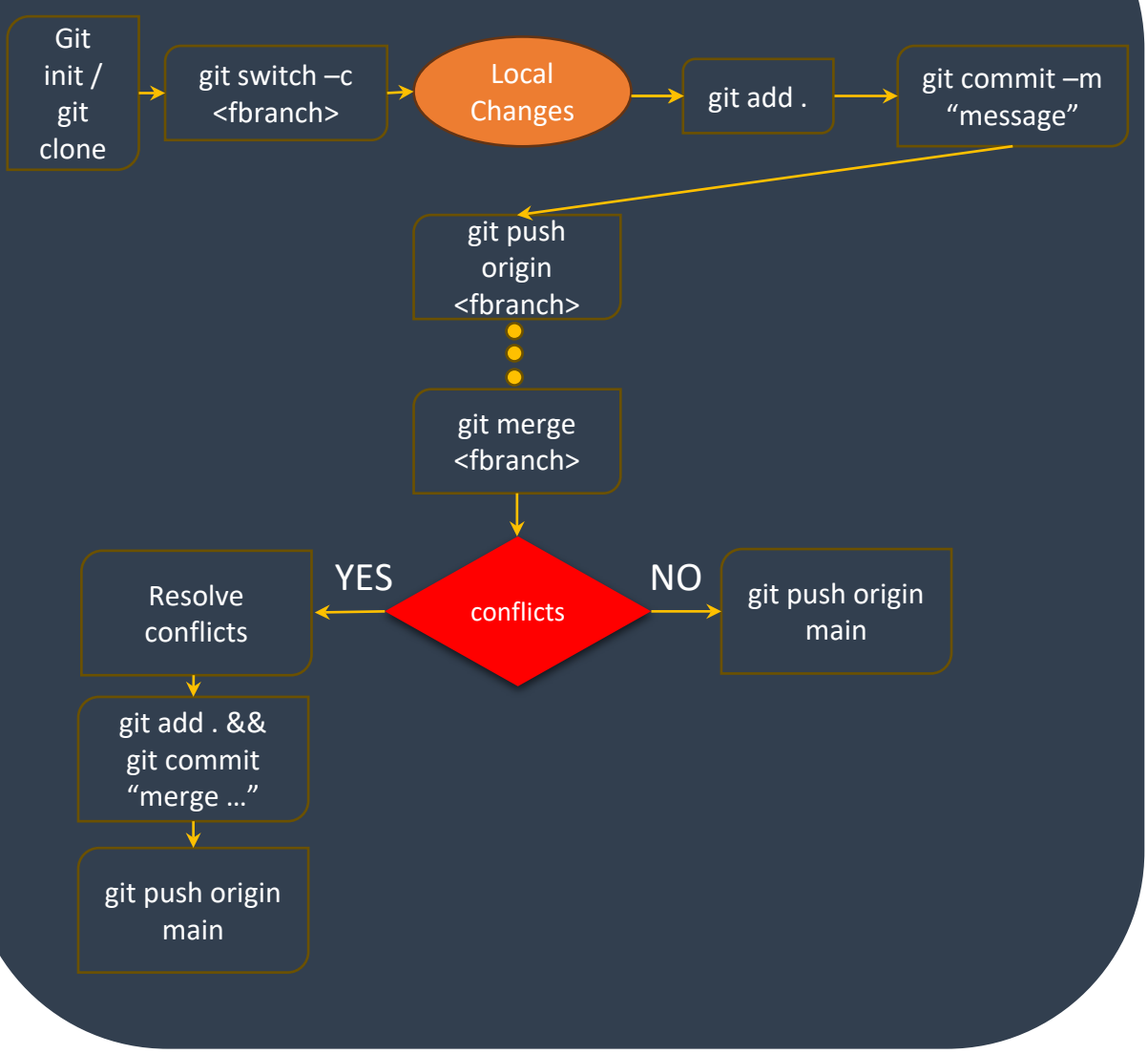
| | | |
|---|---|---|
| git fetch | Updates the remote tracking branch. | • git fetch<br>• git fetch <remote> <branch> |
| git pull | Does fetch + merge. Updates HEAD branch from the remote. Applies the changes if it can automatically be merged. | • git pull<br>• git pull <remote> <branch> |

## Basic workflows

### Centralized workflow (only a main branch)

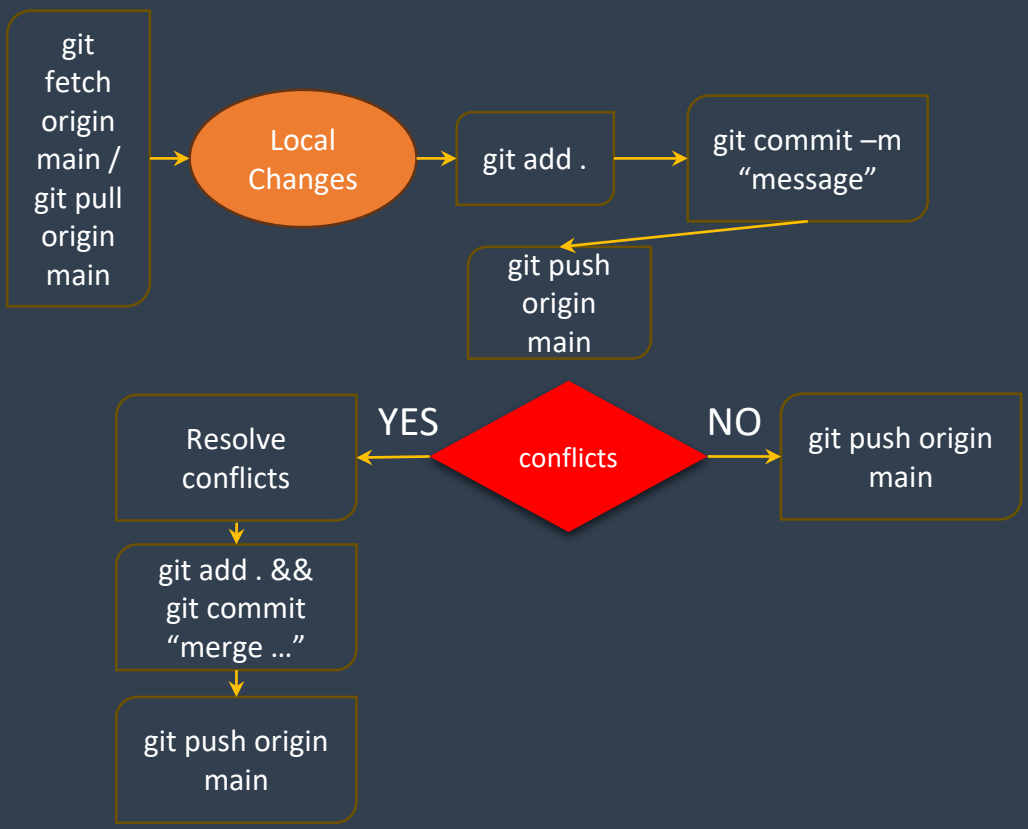



### Feature-branch workflow (new functionality always on a new branch, main branch -> stable)

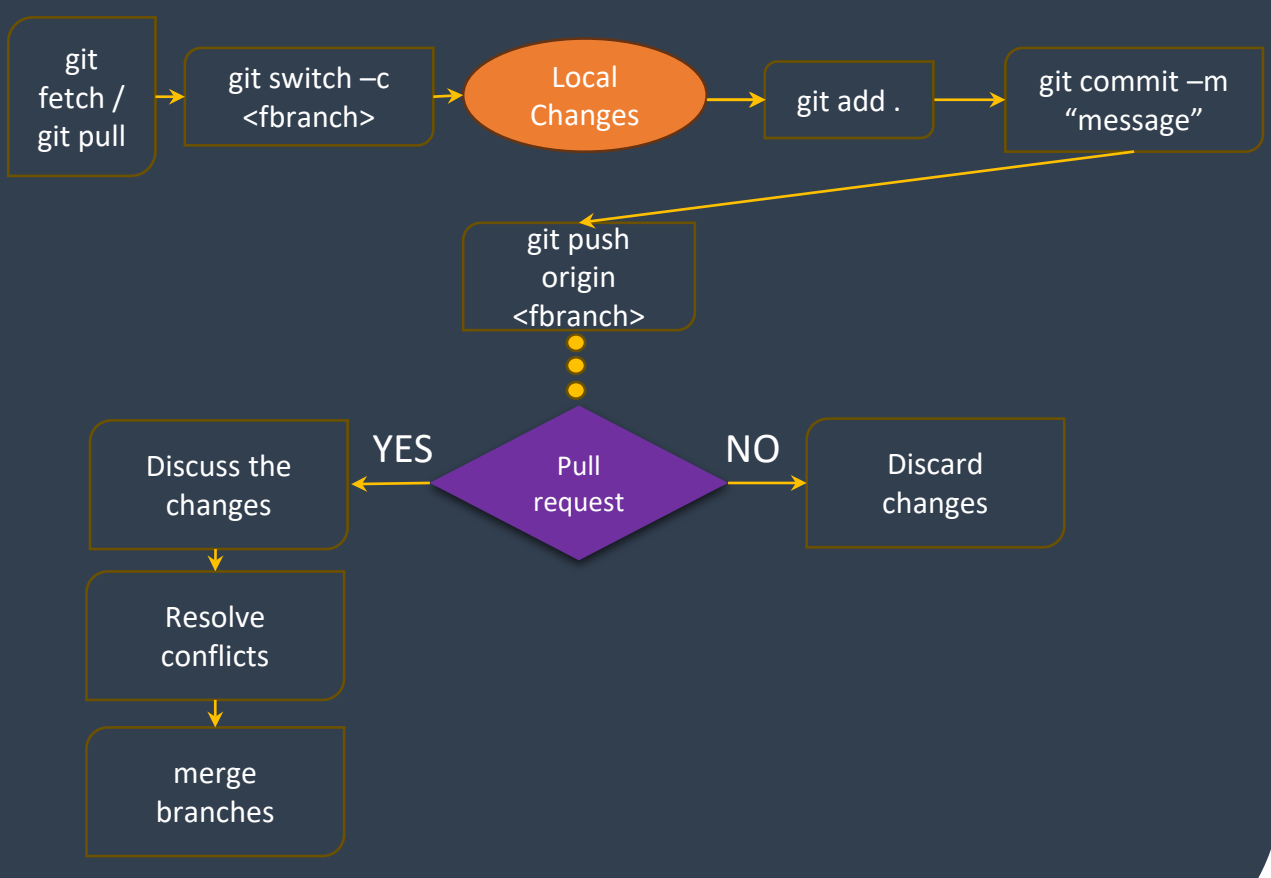


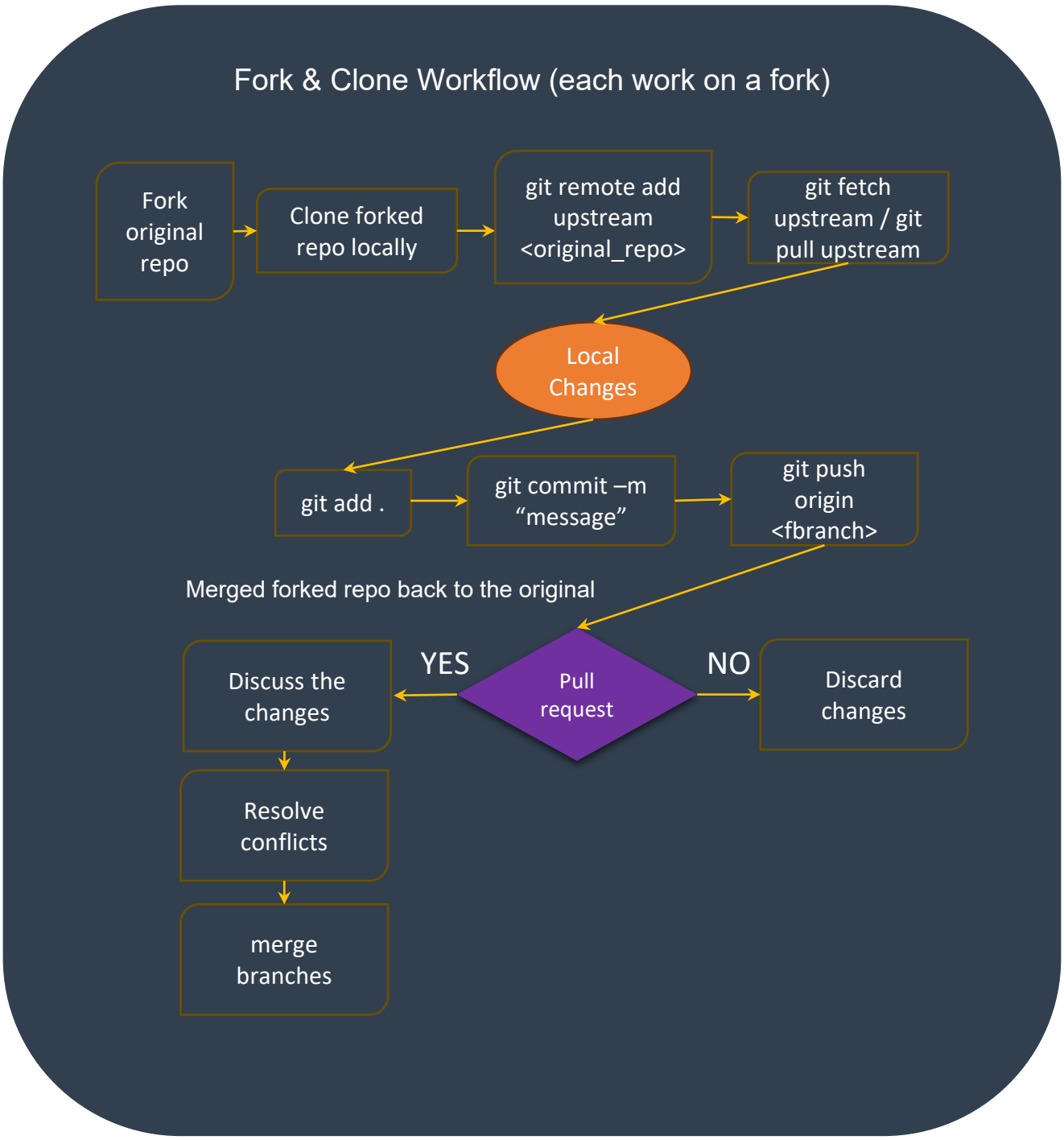
# Colaborative workflows

## Centeralized workflow (only a main branch)




## Feature-branch workflow (new functionality always on a new branch, main branch -> stable)




Fork & Clone Workflow (each work on a fork)
Fork original repo
Clone forked repo locally
git remote add upstream <original_repo>
git fetch upstream / git pull upstream
Local Changes
git add .
git commit –m “message”
git push origin <fbranch>
Merged forked repo back to the original
Pull request
YES
NO
Discuss the changes
Discard changes
Resolve conflicts
merge branches