

Fiorella Boucher : git tutorial

The model is now available on `git`. Visit SIM-git repository and go see `git-HELP` file. The `git-HELP` will start you and will give you most basic git commands. The purpose of this git tutorial is to provide information about **safe commands** and **not that safe commands**, a more complete file, and solutions for future errors.

1 Commands : Safe and not that safe

- `git clone dumbo:/storage/common/SIM-git/`
While working on this project you need to clone master version, this command clones master. Note that master is at dumbo machine and master version is fortran 4.4.7, if your local machine supports later fortran you will have some issues with compilation. In order to avoid that, clone master version in a local folder at dumbo.
- `git fetch -all`
It searches information about branches. If another branch was created you will be able to see it after using this command. Totally **safe command**.
- `git branch-l`
Lists branches. With optional `<pattern>...`, e.g. `git branch -list 'maint-*'`, list only the branches that match the pattern(s). Totally **safe command**.
- `git branch-a`
Use this to see all the branches. Your current branch has a '*' preceding it. If you are at master you will see '* master'. Totally **safe command**.
- `git push -u origin branchY`
Use this only for the first time, by default pushes the current branch to a remote branch with the same name.
- `git checkout branchY`
Use this to go branchY. Common error when switching from branchY to master : **local changes would be overwritten by checkout**. Use forcing `git checkout -f branch`. This will destroy local changes in order to keep them use `git stash`.
- `git checkout -b branchY`
- `git add .gitignore`
You are working on a branch then you want to switch to another branch, you get an error. **Error : You have local changes to '.gitignore'; cannot switch branches**. Consider adding '.gitignore', if it exists already make sure it includes the files that you don't need to track. Use `nano .gitignore` and add files you don't need to track or commit.
- `git remote-v`
This will manage set of tracked repositories
- `git commit -am 'message'`
Don't do this with master, place yourself in your branch before committing any changes. If you did commit to master code make sure you don't push them, i.e. don't use `git push`.
- `git push branchY`
Not a super safe command. Make sure you will not push your changes to master code. In order to do that add the name of your branch on which you want to update your changes. Or place yourself in your branch before pushing any changes.
- `git pull`
This does the opposite of `git push`, it will update your local repository from remotely branch (or master code)
- `git status`
It will show you all files you modified. If you want to update your branch you add modified files and then you can commit them with `git commit`. Totally **safe command**.
- `git add /path/fichier`
Add a file that will be later committed. You can see all files that you want to add and commit by using `git status`.
- `git merge master branchY`
Not a super safe command. Use this when you need to update branchY and master code is **ahead** of your branch by some commits. Make sure that when you don't have any conflicts between branchY and new commits of master branch.