

Git et Gitlab au quotidien

Commandes Principales et Workflows

Récapitulatif

Benoît Bayol

June 9, 2015

Outline

Licence

Création

Changement locaux

Historique des commits

Branches & Tags

Mise à jour & Publication

Merge & Rebase

Annuler

Topic

Licence

Création

Changement locaux

Historique des commits

Branches & Tags

Mise à jour & Publication

Merge & Rebase

Annuler

- ▶ Les commandes sont extraites de : <https://github.com/Kapeli/cheatsheets>
- ▶ Les graphiques présents dans les slides sont extraits du livre "Pro Git 2" qui est aussi présent dans l'archive.
- ▶ La licence de ce document est :
<https://creativecommons.org/licenses/by-nc-sa/3.0/>

Topic

Licence

Création

Changement locaux

Historique des commits

Branches & Tags

Mise à jour & Publication

Merge & Rebase

Annuler

Création I

- ▶ Cloner un répertoire existant

```
1 git clone ssh://user@domain.tld/repo.git
```

- ▶ Cloner un répertoire existant avec ses sous-modules

```
1 git clone --recursive ssh://user@domain.tld/repo.git
```

- ▶ Créer un nouveau dépôt local

```
1 git init
```

Topic

Licence

Création

Changement locaux

Historique des commits

Branches & Tags

Mise à jour & Publication

Merge & Rebase

Annuler

Changement locaux I

- ▶ List changed files in your working directory

```
1 git status
```

- ▶ List changes to tracked files

```
1 git diff
```

- ▶ Add all current changes to the next commit

```
1 git add .
```

- ▶ Add some changes to the next commit

```
1 git add -p <file>
```

- ▶ Commit all local changes in tracked files

```
1 git commit -a
```

- ▶ Commit previously staged changes

```
1 git commit
```

- ▶ Change the last commit

```
1 git commit --amend
```

- ▶ Note: You shouldn't amend published commits!

Topic

Licence

Création

Changement locaux

Historique des commits

Branches & Tags

Mise à jour & Publication

Merge & Rebase

Annuler

Historique des commits I

- ▶ Show all commits

```
1 git log
```

- ▶ Show changes over time for a specific file

```
1 git log -p <file>
```

- ▶ Show changes over time for a specific committer

```
1 git log --author=<committer name>
```

Note: <committer name> is a pattern, so Ed will match Edward Smith. Quotes are optional if the pattern doesn't contain spaces.

- ▶ Who changed what and when in file

```
1 git blame <file>
```

- ▶ Store changes temporarily

```
1 git stash
```

- ▶ Remove and apply stashed changes

```
1 git stash pop
```

Topic

Licence

Création

Changement locaux

Historique des commits

Branches & Tags

Mise à jour & Publication

Merge & Rebase

Annuler

Branches & Tags I

- ▶ List all existing branches

```
1 git branch
```

- ▶ Switch HEAD branch

```
1 git checkout <branch>
```

- ▶ Create a new branch based on your current HEAD

```
1 git branch <new-branch>
```

- ▶ Create a new tracking branch based on a remote branch

```
1 git branch --track <new-branch> <remote-branch>
```

- ▶ Delete a local branch

```
1 git branch -d <branch>
```

- ▶ Delete a remote branch

```
1 git push origin --delete <branch>
```

- ▶ Tag the current commit

```
1 git tag <tag-name>
```

Topic

Licence

Création

Changement locaux

Historique des commits

Branches & Tags

Mise à jour & Publication

Merge & Rebase

Annuler

Mise à jour & Publication I

- ▶ List all currently configured remotes

```
1 git remote -v
```

- ▶ Show information about a remote

```
1 git remote show <remote>
```

- ▶ Add new remote repository

```
1 git remote add <remote> <url>
```

- ▶ Download all changes from remote, but don't merge into HEAD

```
1 git fetch <remote>
```

- ▶ Download all changes from remote, but don't merge into HEAD and clean up deleted branches from origin

```
1 git fetch -p <remote>
```

- ▶ Download changes and directly merge into HEAD

```
1 git pull <remote> <branch>
```

Mise à jour & Publication II

- ▶ Publish local changes on a remote

```
1 git push <remote> <branch>
```

- ▶ Track a remote repository

```
1 git remote add --track <remote-branch> <remote> <url>
```

- ▶ Publish your tags

```
1 git push --tags
```

Topic

Licence

Création

Changement locaux

Historique des commits

Branches & Tags

Mise à jour & Publication

Merge & Rebase

Annuler

Merge & Rebase I

- ▶ Merge branch into your current HEAD

```
1 git merge <branch>
```

- ▶ Rebase your current HEAD onto branch

```
1 git rebase <branch>
```

Note: You shouldn't rebase published commits!

- ▶ Abort a rebase

```
1 git rebase --abort
```

- ▶ Continue a rebase after resolving conflicts

```
1 git rebase --continue
```

- ▶ Resolve conflicts using your configured merge tool

```
1 git mergetool
```

- ▶ Manually resolve conflicts using your editor and mark file as resolved

```
1 git add <resolved-file>
```

```
1 git rm <resolved-file>
```

Topic

Licence

Création

Changement locaux

Historique des commits

Branches & Tags

Mise à jour & Publication

Merge & Rebase

Annuler

Annuler I

- ▶ Discard all local changes in your working directory

```
1 git reset --hard HEAD
```

- ▶ Discard local changes in a specific file

```
1 git checkout HEAD <file>
```

- ▶ Revert a commit by providing a new commit with contrary changes

```
1 git revert <commit>
```

- ▶ Restore a specific file from a previous commit

```
1 git checkout <commit> <file>
```

- ▶ Reset your HEAD pointer to a previous commit

- ▶ Discarding local changes:

```
1 git reset --hard <commit>
```

- ▶ Preserving all changes as unstaged changes:

```
1 git reset <commit>
```

- ▶ Preserving uncommitted local changes

```
1 git reset --keep <commit>
```