SCIENCES SORBONNE UNIVERSITÉ

Travaux Pratiques

UE: MU5SP05

Lab2 : **Git**

Git est un logiciel de gestion de versions décentralisé. C'est un logiciel libre créé par Linus Torvalds, auteur du noyau Linux, et distribué selon les termes de la licence publique générale GNU version 2. En 2016, il s'agit du logiciel de gestion de versions le plus populaire. Vous trouverez dans la page suivante un résumé des commandes git les plus populaires.

1. Installation des git packages
   - First, let's install software packages that we will need throughout the practical labs:

     sudo apt install git gitk git-email

2. Git configuration
   - After installing git on a your machine, the first thing to do is to let git know about your name and e-mail address

     git config --global user.name "[votre_prenom]"

     git config --global user.email [me@mydomain.net]

Such information will be stored in commits. It is important to configure it properly when the time comes to generate and send patches, in particular.

Vérifier que ces informations ont été bien pris en compte en utilisant les commandes suivantes : git config --global user.name et git config --global user.email

1. Créer un répertoire que vous appellerez lab2 (commande mkdir).

2. Dans ce répertoire, créer un fichier 'nom de famille' et écrivez dans ce fichier le texte suivant : hello M2 syscom.

3. Mentionnez que ce répertoire sera un git repository avec la commande git init

4. Utilisez la commande git status pour voir l'état du fichier test.

5. Utiliser la commande git add pour rajouter le fichier et retappez la commande git status pour vérifier si le fichier test à été rajouter à l'index ou pas.

6. Utilisez la commande git commit -m "[Mettre votre message initial]"

7. Créer un compte sur le site git HUB et un nouveau repository. Ne rajoutez pas de fichier d'initialisation de fichier README dans votre repository. Laissez le vide.

8. Connecter votre local repository au remote repository avec la commande git remote (copiez les deux commandes se trouvant dans votre compte git Hub et mettez à jour votre repository distant avec le contenu de votre répertoire se trouvant dans votre machine locale.

   Commandes :

   git remote add origin [https://github.com/votrecompte/toto.git]
   git push -u origin master

9. Créer un fichier README.md sur votre compte git HUB et mettre dans ce fichier un message.

10. Utilisez la commande git pull pour mettre à jour votre répertoire locale.

11. Aller vers votre $HOME et créer un clone du répertoire que vous avez créer et vérifier que vous arrivez à récupérer le répertoire se trouvant dans git HUB.

    Commande : git clone [URL-to-git-HUB].

Dans le reste des TP, vous utiliserez les commandes git pour récupérer les fichiers sources des différentes parties et pour rendre vos comptes rendus.

git cheat sheet

learn more about git the simple way at *rogerdudler.github.com/git-guide/*
cheat sheet created by Nina Jaeschke of *ninagrafik.com*

create & clone

create new repository	*git init*
clone local repository	*git clone /path/to/repository*
clone remote repository	*git clone username@host:/path/to/repository*

add & remove

add changes to INDEX	*git add <filename>*
add all changes to INDEX	*git add **
remove/delete	*git rm <filename>*

commit & synchronize

commit changes	*git commit -m "Commit message"*
push changes to remote repository	*git push origin master*
connect local repository to remote repository	*git remote add origin <server>*
update local repository with remote changes	*git pull*

branches

create new branch	*git checkout -b <branch>* *e.g. git checkout -b feature_x*
switch to master branch	*git checkout master*
delete branch	*git branch -d <branch>*
push branch to remote repository	*git push origin <branch>*

merge

merge changes from another branch	*git merge <branch>*
view changes between two branches	*git diff <source_branch> <target_branch>* *e.g. git diff feature_x feature_y*

tagging

create tag	*git tag <tag> <commit ID>* *e.g. git tag 1.0.0 1b2e1d63ff*
get commit IDs	*git log*

restore

replace working copy with latest from HEAD	*git checkout -- <filename>*