

Utilisation de Git sous Linux

Léo BAUDOUIN

7 juin 2013

Pour des ajouts, corrections, suggestions ou pour des questions vous pouvez me contacter à l'adresse : `leo.baudouin@univ-bpclermont.fr` ou au bureau **3116**.

1 Ajout de l'adresse du Gitorious du laboratoire

Modifier le fichier `/etc/hosts` pour ajouter la ligne suivante :

172.27.3.1 git.univ-bpclermont.fr redmine.univ-bpclermont.fr

Avec la commande suivante :

```
user:~$ sudo echo 172.27.3.1 git.univ-bpclermont.fr redmine.univ-bpclermont.fr >> /etc/hosts
```

2 Installation

Depuis Ubuntu 12.04 pour installer Git il suffit de faire :

```
user:~$ sudo apt-get install git
```

Avec une plus ancienne version d'Ubuntu, utilisez :

```
user:~$ sudo apt-get install git-core
```

3 Configuration

3.1 Données générales

Créez ou modifiez le fichier `~/.gitconfig` :

```
user:~$ gedit ~/.gitconfig
```

```
[user]
  name = Léo Baudouin
  email = leo.baudouin@univ-bpclermont.fr
[alias]
  st = status
  ci = commit
[color]
  branch = auto
  diff = auto
  interactive = auto
  status = auto
[url "git@git.univ-bpclermont.fr:"]
  insteadOf = univbp:
[core]
  excludesfile = ~/.gitignore
```

Vous pouvez également utiliser la commande :

```
user:~$ git config --global <nom_de_l'option><valeur>
```

Exemple :

```
user:~$ git config --global user.name "Léo Baudouin"
```

3.2 Fichiers à exclure

Créez ou modifiez le fichier `~/.gitignore` afin d'ajouter des fichiers qui ne seront pas pris en compte par Git :

```
user:~$ gedit ~/.gitignore
```

```
build/
makefile
*~
*.o
```

3.3 Crédation d'une clé SSH

Afin de sécuriser la connexion et les communications entre le serveur et votre ordinateur, un paire de clés SSH va devoir être créée. On va pour cela utiliser le algorithme de cryptographie asymétrique RSA.

```
user:~$ ssh-keygen
```

Suivez ensuite la procédure de création en modifiant le nom du fichier si besoin, puis en entrant deux fois un mot de passe.

3.4 Envoi de votre clé publique sur le serveur Gitorious

Allez sur git.univ-bpclermont.fr. Une fois connecté, créez-vous un compte¹.

Rendez-vous ensuite sur votre *Dashboard*, puis appuyez sur **Manage SSH keys**. A droite, utilisez le bouton **Add SSH key**. Copiez l'intégralité de votre clé publique puis validez (la totalité du fichier `~/.ssh/id_rsa.pub`).

4 Utilisation

4.1 Cloner un dépôt existant

```
user:~$ git clone git@git.univ-bpclermont.fr:empty-project/empty_project.git
```

4.2 Créer un dépôt local vide

Créer un nouveau dossier

```
user:~$ mkdir empty_project
```

Entrer dans le dossier

```
user:~$ cd empty_project
```

Initialiser le dépôt

```
user:~/empty_project$ git init
```

Ajouter l'adresse du serveur

```
user:~/empty_project$ git remote add origin git@<url>
```

Créer un fichier

```
user:~/empty_project$ gedit README
```

Ajouter le fichier au dépôt

```
user:~/empty_project$ git add README
```

Générer un commit

```
user:~/empty_project$ git commit -m "Text décrivant les changements"
```

Envoyer les fichiers

```
user:~/empty_project$ git push [-u origin master]
```

4.3 Mise à jour

```
user:~/empty_project$ git pull
```

1. voir avec Christophe TOURNAYRE, si vous ne parvenez pas à créer de compte

4.4 Liste des modifications

Liste les fichiers modifiés `user:~/empty_project$ git status`

Liste les modifications `user:~/empty_project$ git diff [fichier]`

4.5 Opérations courantes

Ajouter tous les fichiers modifiés `user:~/empty_project$ git add -u`

Voir les différentes branches `user:~/empty_project$ git branch -a`

Créer une branche `user:~/empty_project$ git branch <nom_de_la_branche>`

Changer de branche `user:~/empty_project$ git checkout <nom_de_la_branche>`

Fusionner deux branches `user:~/empty_project$ git merge <branche1> <branche2>`

Supprimer un fichier `user:~/empty_project$ git rm <fichier>`

Voir qui a modifié le fichier `user:~/empty_project$ git blame <fichier>`

Voir les différents commits `user:~/empty_project$ git log`

Annuler les dernières modifications `user:~/empty_project$ git reset --hard HEAD` [DANGER]

Annuler le dernières commit `user:~/empty_project$ git revert commit` [DANGER]

4.6 Aide

Pour obtenir de l'aide sur les fonctions de Git :

`user:~$ man git-<fonction>`

`user:~$ git <fonction> --help`

`user:~$ git help <fonction>`

5 Interfaces graphiques

Pour visualiser, créer, envoyer des *commits* vous pouvez passer par une interface graphique, il en existe plusieurs :

- QGit : `user:~$ sudo apt-get install qgit`
- GitK : `user:~$ sudo apt-get install gitk`
- QtCreator (dans le menu **Outils** puis **Git**)
- gitg, git-gui, Giggle, git-cola, GitForce,...