

TP - Utilisation basique

Préambule

Deux utilisateurs fictifs, *jojo* et *cap* ont cloné un dépôt git *bac-a-sable* dans le dossier *~/df-git/*. Voici un transcript de ce qu'a fait Cap pour en arriver là (Jojo ayant fait quelque chose de similaire sur son propre pc).

```
cap@bicoque:~$ mkdir ~/df-git/  
cap@bicoque:~$ cd ~/df-git/  
cap@bicoque:~/df-git$ git clone df-git:bac-a-sable  
cap@bicoque:~/df-git$ cd bac-a-sable/  
cap@bicoque:~/df-git/bac-a-sable$
```

Avant de faire quoi que ce soit - la mise à jour

Jojo veut vérifier s'il y a des nouveautés sur le dépôt :

```
jojo@creperie:~$ cd ~/df-git/bac-a-sable/  
jojo@creperie:~/df-git/bac-a-sable$ git pull  
Already up-to-date.  
jojo@creperie:~/df-git/bac-a-sable$
```

`git pull`: *pull* signifie « tirer » en anglais, on tire les mises à jour depuis le serveur, vers notre copie locale)

Visiblement, pas de mise à jour, le dépôt est déjà *up-to-date*.

Ajout de nouveau fichier

Jojo veut alors créer un fichier *crêpes.miam* contenant sa dernière trouvaille.

Il ouvre son éditeur de texte préféré et crée le-dit fichier, de sorte que...

```
jojo@creperie:~/df-git/bac-a-sable$ ls -l  
total 4  
-rw-r--r-- 1 jojo jojo 42 oct. 32 25:61 crêpes.miam  
jojo@creperie:~/df-git/bac-a-sable$
```

Il veut maintenant enregistrer ce changement dans le dépôt.

Comme le fichier *crêpes.miam* n'existait pas, il doit d'abord demander à git de le surveiller.

```
jojo@creperie:~/df-git/bac-a-sable$ git add crêpes.miam
```

```
jojo@creperie:~/df-git/bac-a-sable$
```

Avant de réellement sauvegarder ces modifications :

L'enregistrement des modifications : le commit

```
jojo@creperie:~/df-git/bac-a-sable$ git commit
```

S'ouvre alors son éditeur de texte préféré contenant le texte suivant:

```
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#>new file:   "cr\303\252pes.miam"
```

On voit que le message indique qu'il y a un nouveau fichier "cr\303\252pes.miam" (encodage ascii du nom utf8 du fichier). Il doit indiquer dans cette fenêtre un résumé des opérations effectuées, sur la première ligne. Puis, si nécessaire, une description plus détaillée en dessous (en sautant une ligne). Voici le fichier finalisé de Jojo :

```
Ma nouvelle recette de crêpe :-P

Un peu moins de beurre et un peu plus de bière, que du bon :)
Coquine s'est régalée, trôôôôô !
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#>new file:   "cr\303\252pes.miam"
```

Il enregistre et quitte son éditeur de texte, obtenant dans son terminal :

```
[master 7774c8a] Ma nouvelle recette de crêpe :-P
1 file changed, 1 insertion(+)
create mode 100644 "cr\303\252pes.miam"
jojo@creperie:~/df-git/bac-a-sable$
```

La publication des changements : le push

Les modifications sont donc bien enregistrées sur le dépôt, mais toujours localement sur la machine à Jojo.

Pour envoyer ses modifications vers le serveur DF, il doit lancer la commande suivante:

```
jojo@creperie:~/df-git/bac-a-sable$ git push
Counting objects: 4, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 388 bytes, done.
Total 3 (delta 0), reused 0 (delta 0)
To df-git-jojo:bac-a-sable
d83719f..7774c8a master -> master
jojo@creperie:~/df-git/bac-a-sable$
```

Récupération des mises à jour, encore

Comme sur IRC Jojo vante à Cap le goût inégalable de sa nouvelle recette, ce dernier décide d'aller chercher la recette de Jojo.

```
cap@bicoque:~/df-git/bac-a-sable$ git pull
remote: Counting objects: 4, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
From df-git-cap:bac-a-sable
d83719f..7774c8a master -> origin/master
Updating d83719f..7774c8a
Fast-forward
"cr\303\252pes.miam" | 1 +
1 files changed, 1 insertions(+), 0 deletions(-)
create mode 100644 "cr\303\252pes.miam"
cap@bicoque:~/df-git/bac-a-sable$
```

Et voilà :

```
cap@bicoque:~/df-git/bac-a-sable$ ls -l
total 4
-rw-r--r-- 1 cap cap 42 oct. 32 25:61 crêpes.miam
cap@bicoque:~/df-git/bac-a-sable$
```

Renommage et déplacement

Cap veut réorganiser le dépôt pour mettre la recette à Jojo dans un dossier recettes.

Il commence par créer un dossier et il indique à git la nouvelle position / le nouveau nom du fichier crêpes.miam.

Il enregistre ensuite ses modifications en faisant un *commit* :

```
cap@bicoque:~/df-git/bac-a-sable$ mkdir recettes
```

```
cap@bicoque:~/df-git/bac-a-sable$ git mv crêpes.miam recettes/
```

`git commit -a` permet ensuite d'enregistrer les modifications faites pour tous les fichiers qui étaient déjà suivis par `git`. ie. pour lesquels quelqu'un avait déjà fait `git add` avant.

`git push` envoie les modifications sur le serveur.

```
cap@bicoque:~/df-git/bac-a-sable$ git commit -a
[master 7e63d41] Rangement.
 1 file changed, 0 insertions(+), 0 deletions(-)
 rename "cr\303\252pes.miam" => "recettes/cr\303\252pes.miam" (100%)
```

```
cap@bicoque:~/df-git/bac-a-sable$ git push
Counting objects: 4, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 365 bytes, done.
Total 3 (delta 0), reused 0 (delta 0)
To df-git-cap:bac-a-sable
 7774c8a..7e63d41 master -> master
cap@bicoque:~/df-git/bac-a-sable$
```

On notera que `git cp` existe de manière similaire.

Pourquoi un `git cp` ou un `git mv` plutôt qu'un `cp` ou un `mv` ?
Parce qu'ainsi, `git` garde l'historique des modifications !

Historique

Si vous voulez savoir ce qu'il s'est réellement passé, clonez le dépôt `bac-a-sable`, qui existe vraiment, et tapez la commande suivante :

```
$ git log
commit 7e63d41c89fff6a6d0cd82bdedabfec55125dcd1
Author: Fabien Givors <spam@chezlefab.net>
Date: Sat Oct 27 18:41:17 2012 +0200

    Rangement.
    La cuisine est tout le temps en bazar avec ce jojo !

commit 7774c8a303454bfe8c56b8e3e76552e028ee7ac6
Author: Fabien Givors <spam@chezlefab.net>
Date: Sat Oct 27 18:20:58 2012 +0200

    Ma nouvelle recette de crêpe :-P
    Un peu moins de beurre et un peu plus de bière, que du bon :)
    Coquine s'est régalée, trôôôôô !

commit d83719f23c851d54c8735cb4754669be4732b29d
```

Author: Fabien Givors <spam@chezlefab.net>

Date: Sat Oct 27 17:27:23 2012 +0200

Hello world.

Différences

Vous voyez que chaque commit porte un numéro. En particulier, pour voir les modifications effectuées entre un commit et un autre, la commande suivante est très utile:

```
git diff
7774c8a303454bfe8c56b8e3e76552e028ee7ac6..7e63d41c89fff6a6d0cd82bdedabfec551
25dcd1
diff --git "a/cr\303\252pes.miam" "b/cr\303\252pes.miam"
deleted file mode 100644
index 534d81f..0000000
--- "a/cr\303\252pes.miam"
+++ /dev/null
@@ -1,0,0 @@
-Miam !
diff --git "a/recettes/cr\303\252pes.miam" "b/recettes/cr\303\252pes.miam"
new file mode 100644
index 0000000..534d81f
--- /dev/null
+++ "b/recettes/cr\303\252pes.miam"
@@ -0,0 +1 @@
+Miam !
```

Un fichier a été supprimé, /crêpes.miam, il contenait « Miam ».
Un fichier a été créé, /recettes/crêpes.miam, il contient « Miam »

Retour au tuto git-df

1. [Par ici !](#)

From:
<http://debian-facile.org/> - **Documentation - Wiki**

Permanent link:
<http://debian-facile.org/atelier:git-df:tp-basique>

Last update: **03/07/2014 00:06**

