GIT: Quick start Guide

(created by Jean-Jacques PONCIANO)

Setting up a git server

Informations

git init : initialization of empty depot

--bare Create a bare repository.

--shared[=(false|true|umask|group|all|world|everybody|0xxx)]

Specify that the Git repository is to be shared amongst several users.

Add a deposit to the server

To initialize a new deposit

```
git init --bare --shared=group
```

From an existing deposit

```
git clone --bare project project.git
```

To give rights at the group *developer*

```
chgrp -R developer /opt/git/projet.git
```

To send the project to a server

```
scp -r projet.git utilisateur@git.exemple.com:/opt/git
```

Work with a deposit of the server

Get the project

```
git clone user@89.13.202.227:/opt/git/project.git
```

Create a new local branch

```
git checkout -b newBranch
```

Create a new branch tracking a remote branch

```
git  checkout -b [branch] [origin]/[branch]
```

Push your branch at the server

If your branch does not exist at the server

```
git push origin yourBranch
```

else

```
git push yourBranch
```

Remove the branch at the server

```
git push origin :yourBranch
```

Remove local branch

```
git branch -d yourBranch
```

Display log

```
git shortlog
```

Display just n last log

```
git shortlog -n
```

Track a existing remote branch

```
git branch -u origin/correctionserveur
```

Show all difference:

```
git diff --name-status origin/master
```

Ignore file:

(create .gitignore file at the project root directory)

nano .gitignore (à faire à la racine)

Specify the file which you don’t want to.

Manage Tags

Listes all tags

```
git  tag
```

To create a tag

```
git tag -a v1.5 -m 'my signed 1.5 tag'
```

Show the tag

```
git show v1.5
```

Push the tag to the server

```
git push origin v1.5
```

SubModules

Add a new submodule

```
git submodule add user@server:/git/submodule
```

Update all submodule:

```
git submodule init && git submodule update
```

Be Careful when to use submodule:

When you works in a submodule, don’t forget to push all modification in the submodule:

```
cd  yourSubmodule
git checkout master
git merge yourBranch (if you have work in a branch)
git commit -am “your works”
git push
```

When you pull, clone or merge a project containing any submodules:

```
git submodule init
git submodule update
cd  yourSubmodule
git checkout master
git checkout -b yourbranch (if you want work with a branch)
```

It is very important to do not forget the checkout. If you forget it, you will erase the previous modifications when you will push.

Gitolite

Create a user named git

```
su
adduser git
```

Log in this new user

```
su git
cd
```

Get the project:

```
git clone git://github.com/sitaramc/gitolite
```

Install gitolite

```
mkdir bin
gitolite/install -to /home/git/bin
```

Configuration:

```
 /home/git/bin/gitolite setup -pk yourKey.pub
```

Add user or repo

```
git clone git@server:gitolite-admin
```

Now you could edit conf/gitolite.conf to add a repo and you could add a user adding her ssh key in keydir.
Finally, it remains only to send changes to the server.