Versionner un projet avec GIT

Ce tuto explique comment versionner un projet avec git, sur la forge de l'ECM.

Configuration coté forge

Mise en place d'une clé publique sur la forge

Sur Linux

L'authentification à git se fait via un système de clés exactement comme ssh. L'utilisateur qui va faire les commit doit avoir sa clé publique enregistrée sur la forge.

Pour générer une nouvelle clé publique :

```
$> ssh-keygen
$> cat ~/.ssh/id_rsa.pub
```

Votre clé publique va s'afficher. Il faudra l'ajouter au compte qui va effectuer les mises à jour sur le projet. (A priori votre compte)

La clé privée s'est créée et placée par la commande. Il faut maintenant **ajouter la clé à la forge** en suivant la prochaine étape.

Sur Windows

Allez télécharger puttygen.exe qui vous permettra de générer une belle clé.

Vous ouvrez le logiciel, vous cliquez sur **Generate**, en commentaire de la clé vous mettez username@machine (par exemple rgrondin@rog, où ici *rog* est le petit nom de ma machine). Vous mettez éventuellement une passphrase (mot de passe nécessaire à l'utilisation de la clé). Vous faites **Save public key** et la mettez où vous voulez, où vous pourrez la retrouver.

Pour le **Save private key** mettez là à un endroit un peu caché, connu de vous seul. Cette clé ne doit pas fuiter sinon elle sera compromise et il faudra la supprimer.

Vous copierez la clé publique qui s'affiche (et qui est sauvegardée dans le fichier de clé publique) sur la forge en suivant la prochaine étape.

Ajouter la clé à la forge

La page d'ajout des clés se trouve dans vos options de compte sur la forge. Donnez lui un nom clair pour l'identifier (Exemple : GinfoHeimdall pour le compte ginfo sur heimdall)

Création du projet et du dépôt

Tout d'abord il faut créer un projet sur la forge. Cela se fait à partir de page des projets de celle-ci.



Les projets peuvent avoir un projet parent. Pour les projets propres au GInfo on prendra le projet GInfo comme projet parent.

Une fois le projet créé il faut lui créer un dépôt.

Pour cela il faut aller dans l'onglet configuration, et de là on peut ajouter un dépôt. Un dépôt se caractérise par son identifiant, qui sera contenu dans l'url du dépôt. Pour créer le dépôt laisser les options par défaut.

Une fois le dépôt créé, on a une url du genre `ssh://git@git.centrale-marseille.fr/ginfo/reseau-assos/reseau.git` qu'il faut conserver pour la suite. ici *reseau* est l'identifiant du dépôt, et *reseau-ginfo* le nom du projet. (dont le parent est *ginfo*, d'où le ginfo dans l'url)

Configuration de git côté serveur GInfo

Installation et configuration

Pour l'installation tout simplement :

```
$> sudo apt-get install git
```

La configuration consiste à indiquer votre nom et votre mail.

```
git config --global user.name "GInfo"
git config --global user.email "ginfo@centrale-marseille.fr"
```

Configuration du dépôt

Pour pouvoir envoyer vos fichiers sur votre dépôt il faut avoir un dossier en local qui en sera la copie. Vous y ajouterez les fichiers qui devront aller sur le dépôt git.

Pour l'exemple, imaginons un dossier *depot* qui contiendra le projet. On y initialise git.

```
$> cd depot
$> git init
```

On indique à git où il faudra envoyer les fichiers.

```
$> git remote add origin ssh://git@git.centrale-marseille.fr/ginfo/reseau-assos/reseau.git
```

Ici l'url est celle que vous avez obtenue tout à l'heure en créant le dépôt.

Le premier commit

Pour l'exemple, imaginons que votre dossier *depot* contient ceci :

- depot
  - dossier1/
  - dossier2/
  - fichier.txt

Vous voulez uniquement ajouter *dossier1/ // et tous ses sous-dossiers et fichiers ainsi que //fichier.txt*

```
$> cd depot
$> git add dossier1 fichier.txt
```

Ensuite on effectue le premier commit.

```
$> git commit -m 'Initializing depot repository'
```



La machine doit être connectée au portail captif pour la suite. (A faire attention pour les cron notamment)

Et on balance le tout

```
$> git push origin master
```

Normalement à la fin de l'exécution de cette commande, vous pouvez voir le contenu que vous avez ajouté apparaître dans le dépôt sur la forge.

Utilisation courante

Pour update manuellement le repository Git, naviguez dans le dossier voulu et tapez

```
$> git commit -am 'update manuelle'
```

Puis on push le commit :

```
$> git push origin master
```

Pour récupérer les derniers commits sur votre PC, utilisez :

```
$> git pull
```

Nouveau commit

C'est la même procédure que le premier commit. Après une mise à jour des fichiers (ajout de fichiers ou modifications), il suffit de se placer dans le dossier *depot*, d'ajouter les modifications au projet git, d'effectuer un commit et de push.

Pour aller plus loin

- La documentation git de la forge : https://forge.centrale-marseille.fr/projects/forge/wiki/La_doc#Utilisation-de-Git
- Documentation officielle, en particulier Pro Git : https://git-scm.com/doc
- Embellir ses `git diff` : https://github.com/so-fancy/diff-so-fancy

From:
https://wiki.centrale-med.fr/ginfo/ - **Wiki GInfo**

Permanent link:
https://wiki.centrale-med.fr/ginfo/tutoriels:git

Last update: **06/11/2017 09:06**