Travaux Pratiques Git

Michel Meynard

7 septembre 2015

1 Généralités

Le site `https://gitlab.info-ufr.univ-montp2.fr` est dédié à l'hébergement de projets informatiques des étudiants et enseignants de la FDS. Un projet nommé `tccp20xy` a été créé (20xy correspond à l'année courante). Vous pouvez aller anonymement visiter ce projet `https://gitlab.info-ufr.univ-montp2.fr/michel.meynard/tccp20xy/`.

Vous deviendrez membres (developer) de ce projet au fur et à mesure que vous aurez envoyés vos adresses email au propriétaires (meynard).

2 Git

Git est un logiciel de gestion de versions décentralisé. C'est un logiciel libre créé par Linus Torvalds, créateur du noyau Linux, et distribué selon les termes de la licence publique générale GNU version 2. L'intérêt de Git par rapport à SVN réside dans le fait qu'il suffit de forker un projet ou de le cloner pour commencer à travailler dessus (avec tout l'historique du projet en local) et ensuite de proposer sa contribution (pull request) au repository principal (mainteneur principal du projet). On peut utiliser Git sur de nombreux serveurs dont :

- Google Code qui permet de créer des projets libres sous SVN, Git ou Mercurial. La version payante permet de créer des projets privés.
- GitHub qui permet de créer des projets libres sous Git. La version payante permet de créer des projets privés.
- GitLab de la FDS qui permet de créer des projets PRIVES gratuitement !

3 Introduction à GitLab du Service Info. de la FDS

GitLab est un serveur de version Git intégré dans un site Web permettant également :

- d'éditer un Wiki ;
- de signaler des problèmes (Issue) et leur résolution ;
- de gérer des groupes et des permissions sur les projets ;

3.1 Premiers pas

1. ouvrir le site `https://gitlab.info-ufr.univ-montp2.fr/` dans un navigateur ;
2. saisir son login et mot de passe de l'ENT ;
3. on arrive sur le tableau de bord (*dashboard*) sur lequel on peut toujours revenir en cliquant sur l'icône de renard ;
4. dans le bandeau du haut, on peut examiner/modifier son profil (bonhomme) mais surtout **créer un nouveau projet** (+) ;
5. après avoir créer un projet de nom "test" en cliquant sur le symbole d'addition, on arrive sur la page du projet qui contient un nouveau bandeau horizontal contenant plusieurs onglets :
   - Activity : recense les dernières actions sur le projet ;
   - Files : les fichiers du projet ;
   - Issues : contient la liste des problèmes liés à ce projet ;
   - Wiki pour structurer de l'information sur le projet.

Attention, le projet existe mais ne contient rien. On va devoir, dans la section suivante, créer un dépôt local, y ajouter des fichiers, valider, puis envoyer cette version sur le serveur GitLab.

3.2 Débuter le projet test

Depuis la page d'accueil du projet test, apparaît une liste des commandes à réaliser sur le poste client (unix ou windows ou ...) dans un terminal pour pouvoir utiliser ce projet :

Git global setup configuration globale à ne faire qu'une seule fois pour tous ses projets et qui définit le nom et l'email (ENT) de l'utilisateur dans un fichier de configuration `~/.gitconfig` ;

- `git config --global user.name "Michel Meynard"` définit le nom ;
- `git config --global user.email "michel.meynard@univ-montp2.fr"` définit l'email ;

— `cat ~/.gitconfig` vérifie vos enregistrements

Create repository Vous souhaitez initialisez le projet à vide : après avoir créé le répertoire `test` et s'y être déplacé, on effectuera `git init` qui permet de créer une arborescence dans le répertoire `~/test/.git`. On pourra ensuite, créer des fichiers avec son éditeur préféré, les ajouter `git add toto.txt tutu.txt`, valider les modifications sur le dépôt local `git commit -m "ajout toto et tutu"`. Enfin, l'association du dépôt local avec le serveur distant désigné par "origin" sera réalisé grâce à la commande : `git remote add origin git@gitlab.info-ufr.univ-montp2.fr:michel.meynard/test.git`. On voit donc qu'il n'existe qu'un seul compte git (de nom git) qui sera accédé via ssh uniquement par clé privée publique (pas par login, mot de passe). Il faut donc avoir créé sur la machine ce couple de clé en utilisant la commande :

```
ssh-keygen -t rsa -C "michel.meynard@univ-montp2.fr"
```

Cette commande, qui est situé dans "tableau de bord / Help / ssh keys" génère un répertoire `~/.ssh` qui contient le fichier `~/.ssh/id_rsa.pub`. Ce fichier contient la clé publique que l'on doit recopier et coller dans GitLab sous "Profile / SSH Keys / Add new". Après avoir sauvé cette clé, il ne reste plus qu'à envoyer votre version actuelle sur le serveur distant `git push -u origin master` où master est la branche principale de votre dépôt local. Attention à visualiser où ssh sauve le répertoire `.ssh` (dans ~ mais parfois dans `/home/meynard`. Ouf ! Bien entendu, durant le cycle de vie du projet, les commandes git à effectuer seront plus simples !

Existing Git Repo ? si vous souhaitez utilisez un projet git existant sur votre machine locale, il suffira de définir le serveur distant (git remote add origin ...), de créer la clé ssh si ce n'est pas déjà fait, et de faire le push initial ;

3.3 Cycle de vie du projet

Par la suite, vous n'aurez qu'à ajouter des collaborateurs à GitLab ("Projet / Accueil / Team"), ajouter des fichiers (add) en supprimer (rm), valider votre version (commit) et envoyer fréquemment votre version sur le serveur (push). En cas de conflit, votre push sera refusé et un merge sera automatique effectué dans les fichiers conflictuels. Il faudra alors les éditer puis faire un commit et un push pour que le serveur distant soit à jour. Pour récupérer le travail des autres, il suffit de faire un pull qui effectue un fetch (récupération) et un merge (fusion).

Si vous êtes un nouveau collaborateur au projet, il vous faudra juste cloner le projet depuis le serveur distant (`git clone git@gitlab.info-ufr.univ-montp2.fr:michel.meynard/test.git`). Pour cela, vous devrez bien sur posséder une clé publique ssh que vous aurez enregistré sur votre profil GitLab. Ensuite, comme le créateur, vous pourrez récupérer (pull), ajouter (add), valider (commit) et envoyer (push) vos modifications.

3.4 Divers

Les versions Git n'étant pas agréables car codées sur 10 chiffres hexadécimaux, il est conseillé d'étiqueter les versions avec des tags : `git tag 1.0.0 1b2e1d63ff`.

4 A faire

Exercice 1 (CV et lettre de motivation (add, commit))
Sur le site de projet, visitez les pages wiki en activant le menu de gauche Wiki. Les deux premières tâches que vous devrez accomplir concerne la mise en ligne de votre présentation (une fois que vous l'aurez effectuée) et de votre CV LaTeX accompagnée d'une lettre de candidature motivée au master informatique.

1. Créez un répertoire `dupont.jean` dans votre répertoire `tccp20xy/Cvs` ;
2. créez dans ce répertoire les fichiers tex et pdf demandés ;
3. ajoutez (git add ...) ces différents fichiers puis validez vos modifications (git commit ...) ;
4. enfin il vous faudra déposez cette nouvelle version (git push ...) ;
5. vérifiez sur la page web que vos fichiers sont maintenant présents dans le projet ;
6. lorsque vous aurez terminé votre présentation, publiez-la également dans le dépôt sous le répertoire Presentations.