Accès git Simple Hosting

Prérequis

* L'accès git s'adresse aux personnes maîtrisant l'utilisation de l'outil
* Nous vous conseillons de lire la documentation de git afin de bien utiliser cet accès
* **{vhost}** == un vhost créé sur votre instance
* **{login}** == votre identifiant unique. Un numéro à plusieurs chiffres, fourni dans l'email de confirmation envoyé lors de la création de votre instance, visible aussi sur la page de gestion de votre instance. Différent de l'identifiant Gandi. Le même identifiant que les accès console/administration/sftp…
* **{datacenter\_location}** == l'emplacement de votre instance : dc0 pour Paris, dc1 pour Baltimore, dc2 pour le Luxembourg

Créer un dépôt

Pour le moment nous limitons à un dépôt par vhost. Voici un petit exemple à exécuter localement sur votre machine :

$ mkdir {vhost}

$ cd {vhost}

$ git init

$ git remote add origin ssh+git://{login}@git.{datacenter\_location}.gpaas.net/{vhost}.git

$ mkdir htdocs

$ echo "Hello world" > htdocs/index.html

$ git add htdocs

$ git commit -m "first version of index.html" htdocs

$ git push origin master

Une fois le dépôt créé, vous pouvez avoir une visualisation de celui-ci dans votre administration Simple Hosting via l'outil *gitweb*.

Si c'est votre premier dépôt vous aurez à redémarrer votre instance afin d'avoir cette visualisation disponible dans votre administration.

Le répertoire htdocs

Pour les instances PHP, la racine de votre dépôt n'est pas le dossier servi par le serveur web. Par conséquent afin que vos fichiers soient accessibles, vous devez les mettre dans le dossier nommé *htdocs* placé à la racine de votre dépôt.

Cloner un dépôt

Cette commande va vous permettre de copier le dépôt localement sur votre machine.

$ git clone ssh+git://{login}@git.{datacenter\_location}.gpaas.net/{vhost}.git

Déployer son code

Cette commande va faire un *git checkout* dans votre dossier vhost afin de déployer votre code. Vous pouvez optionnellement préciser un objet git comme un tag par exemple ([plus d'infos](http://git-scm.com/docs/git-checkout)).

$ ssh {login}@git.{datacenter\_location}.gpaas.net 'deploy {vhost}.git [ {git\_object} ]'

L'usage des [submodules Git](http://git-scm.com/book/fr/Utilitaires-Git-Sous-modules) n'est pas supporté pour le moment.

Les répertoires déjà présents dans 'htdocs' de l'hôte virtuels doivent avoir les droits d'écriture sur l'utilisateur à minima afin que les fichiers puissent être remplacé lors du déploiement. Soit l'équivalent d'un 'chmod 644'. Cette action est à effectuer si vous rencontrez une erreur de type :

error: unable to unlink old 'htdocs/sites/default/settings.php' (Permission denied)

Vous pouvez optionnellement nettoyer votre répertoire vhost. Cette commande va exécuter un git clean sur votre répertoire ([plus d'infos](http://git-scm.com/docs/git-clean)).

Cette commande va **supprimer tous les fichiers qui ne sont pas suivis par git** !

$ ssh {login}@git.{datacenter\_location}.gpaas.net 'clean {vhost}.git'

Si vous désirez conserver certains fichiers ou répertoires même s'ils ne sont pas suivis par git, il vous suffira d'ajouter une liste dans un fichier *.gitignore* à la racine de votre dépôt (voir comment [ignorer des fichiers](http://git-scm.com/book/fr/Les-bases-de-Git-Enregistrer-des-modifications-dans-le-d%C3%A9p%C3%B4t#Ignorer-des-fichiers)). Cela est utile par exemple si vous avez des fichiers créés par votre site.

Méthode d'accès

Vous pouvez utiliser votre mot de passe habituel pour accéder à vos dépôts (le même que les accès console/administration/sftp…) ou ajouter votre clef publique ssh sur votre disque ([plus d'infos](https://wiki.gandi.net/fr/simple/ssh_key)).

Supprimer un dépôt

Afin de supprimer un dépôt, vous pouvez supprimer directement le répertoire /lamp0/vcs/git/{vhost}.git (/lamp0/vcs/git/default.git dans le cas d'une instance Node.js) via le sftp.

Les instances Node.js et Python

Pour ce type d'instance, il existe un seul dossier *default* dans vos vhosts, ainsi l'url d'accès sera du type:

ssh+git://{login}@git.{datacenter\_location}.gpaas.net/default.git

Chaque déploiement (voir la commande deploy) va automatiquement redémarrer le process node chargé de recevoir les requêtes. Les dépendances de votre application seront aussi mises à jours. De plus, il n'est pas nécessaire d'avoir un dossier *htdocs* comme pour les instances PHP.

Et sous Windows ?

Nous allons prochainement ajouter une documentation pour l'accès Git sous Windows en utilisant 'msysgit' : <http://msysgit.github.com/>

Il y a un tutoriel pour l'utiliser à l'adresse suivante : <http://guides.beanstalkapp.com/version-control/git-on-windows.html>

Voir aussi

* [site web git](http://git-scm.com/)
* [documentation git](http://git-scm.com/book/fr/)

Dernière modification: le 24/06/2014 à 15:47 par Yann A. (Gandi)