
DÉPLOIEMENT DE NIVEAU ENTREPRISE AVEC GIT, DRUSH ET FABRIC

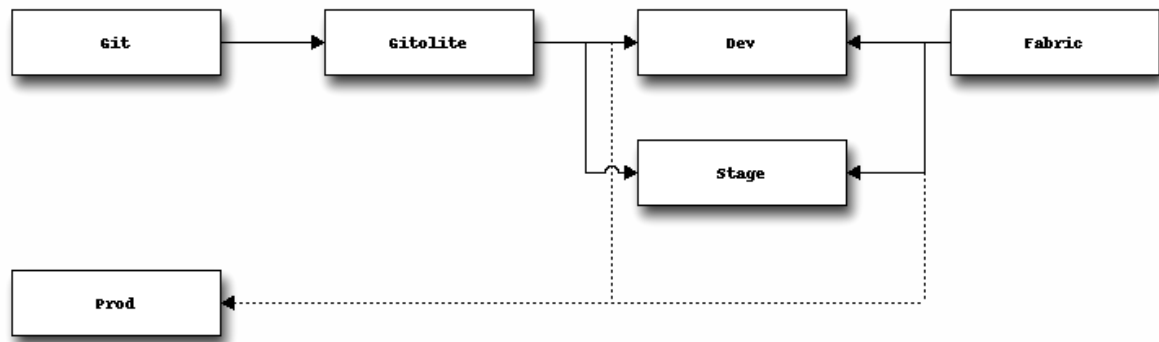
Pierre Paul Lefebvre
Pierre Buyle



INTRO



BUT ULTIME



LE PROCESSUS DOIT ÊTRE AUTOMATISÉ ?

L'être humain est

- lent
- pas très fiable
- approximatifs
- doté de peu de mémoire

Les ordinateurs sont

- rapides
 - fiables
 - précises
 - dotés d'une excellente mémoire
-

NOS OUTILS

- Drush
- Git
- Fabric

DRUSH

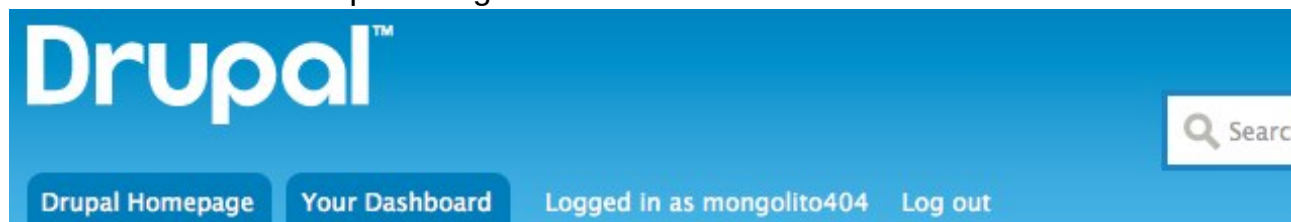


FONCTIONS DE DRUSH

- Télécharger des modules
- Activer/Désactiver des modules
- Déinstaller des modules
- Rouler les tests*
- Vider les cache
- Cron
- Gérer les utilisateurs
- Récupérer et changer les valeurs des variables de configuration
- ...

DRUSH EST EXEXTENSIBLE

- Plus de 100 modules qui s'intègrent avec Drush



Download & Extend

[Download & Extend Home](#) [Drupal Core](#) [Distributions](#) [Modules](#) [Themes](#)

127 Modules match your search

Modules categories:

Filter by compatibility:

Status:

Search Modules:

Sort by: Most installed

Search

- Implementations de hooks pour définir ses propres commandes

FONCTIONS DE DRUSH



POURQUOI DRUSH EST-IL IMPORTANT?

- Rapide
- Simple

- Simple
- Complet
- Scriptable
- Automatisable

COMMENT INSTALLER DRUSH

PEAR

```
pear channel-discover pear.drush.org
pear install drush/drush
```

WINDOWS INSTALLER

http://www.drush.org/drush_windows_installer

GESTIONNAIRE DE PACKAGES

```
sudo aptitude install drush
```

COMMENT INSTALLER DRUSH (PEAR)

Pear est la solution la plus simple pour avoir rapidement accès à la dernière version.

Mais Pear est *brisé* sur MacOSX Mountain Lion

```
sudo cp /private/etc/php.ini.default /private/etc/php.ini
sudo php /usr/lib/php/install_pear_nozlib.php
```

```
sudo php /usr/lib/php/install-pear-nozlib.pear
pear config-set php_ini /private/etc/php.ini
pecl config-set php_ini /private/etc/php.ini
sudo pear upgrade-all
```

USAGE

```
drush --root=/path/to/drupal --uri=dev.mydrupalsite.com
status
Drupal version           : 7.15
Site URI                 : http://dev.mydrupalsite.com/
Database driver          : mysql
Database hostname       : localhost
Database username       : drupal
Database name            : drupal
Default theme            : garland
Administration theme     : seven
PHP configuration        : /etc/php-cli.ini
Drush version            : 5.7
Drush configuration      : /home/fry/.drush/drushrc.php
Drush alias files        : /home/fry/.drush/aliases.drush.rc
Drupal root              : /path/to/drupal
Site path                : sites/default
File directory path      : sites/default/files
```

CONFIGURATION DE DRUSH

DRUSHRC.PHP

ALIASES.DRUSHRC.PHP

```
$aliases['dev'] = array(  
  'root' => '/path/to/drupal',  
  'uri' => 'dev.mydrupalsite.com',  
);  
$aliases['live'] = array(  
  'root' => '/other/path/to/drupal',  
  'uri' => 'mydrupalsite.com',  
  'remote-host' => 'myprodserver.myisp.com',  
  'remote-user' => 'publisher',  
);
```

```
drush @dev status
```

FONCTIONS IMPORTANTES

- drush @dev sql-dump > backup.sql
- drush sql-sync @prod @dev
- drush rsync @dev @stage
- drush rsync ./ @stage:%files/img
- drush @dev clear-cache all
- drush @dev updatedb
- drush make mydrupalsite.make

GIT



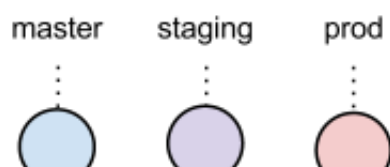


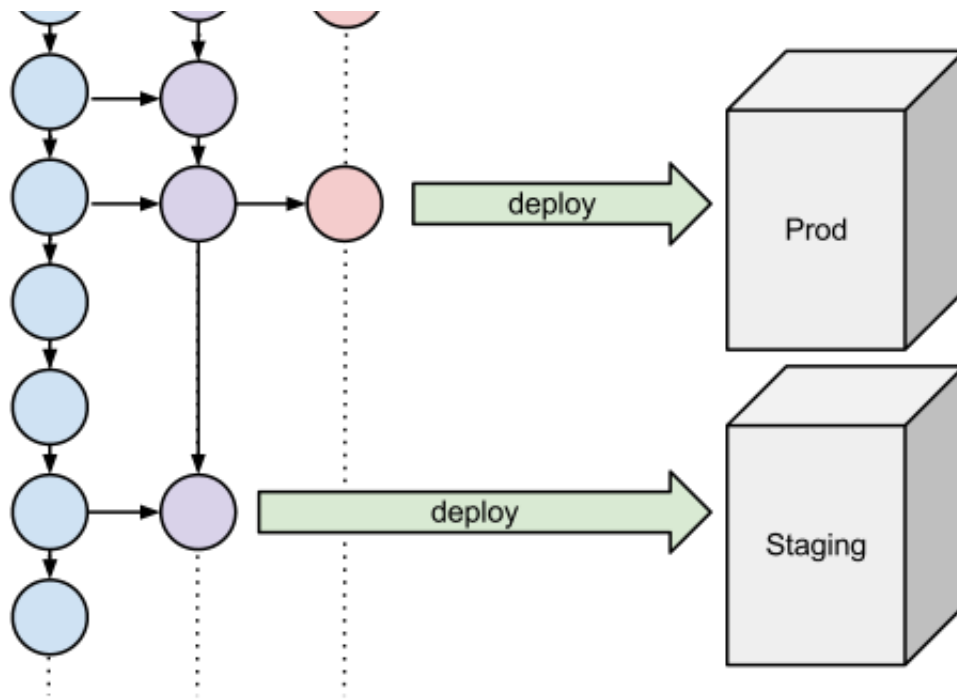
GIT



- Git est un logiciel de gestion de versions décentralisé.
- Permet de maintenir l'historique du versions de code.
- Support de nombreux *workflow* d'édition et de partage de code.
- Utilisé sur Drupal.org, requis pour contribué
- <http://git-scm.com/>
- <http://try.github.com/>
- <http://drupal.org/documentation/git>

BRANCHES





DEPLOYER LE CODE AVEC GIT

Le déploiement se fait par mise à jour d'un dépôt Git *local* au serveur sur lequel on déploie.

PULL

Récupérer les derniers changements dans un dépôt de code central depuis le serveur sur lequel on déploie.

PUSH

Envoyer les derniers changements dans un dépôt local vers le serveur sur lequel on déploie.

GITOLITE/GITOSIS

Permet de facilement gérer un serveur de dépôts de code *centralisés* avec un contrôle d'accès par utilisateurs et groupes d'utilisateurs.

Gitosis est mort, longue vie à Gitosis!

Dernier commit est en 2009. Gitolite est complet et fonctionne bien.

FABRIC



Librairie Python permettant d'automatiser des tâches sur un ou des serveurs distants.

```
from fabric.api import run

def host_type():
    run('uname -s')

$ fab -H localhost,linuxbox host_type
[localhost] run: uname -s
[localhost] out: Darwin
[linuxbox] run: uname -s
[linuxbox] out: Linux

Done.
Disconnecting from localhost... done.
Disconnecting from linuxbox... done.
```

EXEMPLE FABRIC

Exemple de script pour pousser la DB sur un serveur distant

```
from fabric.api import *
env.use_ssh_config = True

@task(default=True)
def pushDB(env='local'):
    local('drush @via sql-dump --result-file=/tmp/via-dump.sql')
    db_connect = local('drush sql-connect --uri=http://supersite.' + env + '.drupalcampmontreal.org', True)
    local(db_connect + ' < /tmp/via-dump.sql')
```

CONFIG SSH

Les dernières versions peuvent lire dans la config SSH du client initiant les commandes.

```
Host      supersite.stage
HostName  10.111.222.333
User      deployuser
```

Avec ce snippet dans votre configuration, vous pouvez vous connecter avec `ssh supersite.stage` sans spécifier le port, le user ou le mot de passe (si la clef SSH est autorisée).

AUTRES OUTILS À CONSIDÉRER

C'est une belle technique qui reste quand même simple et flexible, mais difficile à maintenir quand le nombre de serveurs dépassent une trentaine.

JENKINS

Pour ceux qui veulent se rendre au prochain niveau, Jenkins est là.

Jenkins vous permet de rouler les tests et peut être inséré dans le processus avant de faire un déploiement. C'est long rouler les tests.

Il peut aussi être utiliser pour vérifier que les Coding Standards ont bien été suivis entre chacune des versions du site web.

ANSIBLE - CHEF

ANSIBLE

Belle solution python, probablement la plus simple.

<http://ansible.cc/>

Le nouveau site web est beaucoup plus accessible et convivial.

CHEF

Belle solution ruby un peu plus complexe. Stable avec beaucoup de développement.

<http://wiki.opscode.com/display/chef/Home>

Beaucoup de "cookbooks" sont développés par la communauté

<http://community.opscode.com/cookbooks>

Beaucoup d'autres solutions existent (cdist, puppet, bcfg2, etc) avec des niveaux de complexité variés.

VAGRANT

Vagrant permet de créer des environnements virtualisés pour reproduire les serveurs de productions, mais avec le développeur en tête.

Le développeur peut lancer Vagrant pour avoir un environnement de développement identique à celui de la production. Plus jamais vous allez entendre "Mais ça marche sur ma machine!"

QUESTIONS?
