

INSTALLATION INFRASTRUCTURE DE DEVELOPEMENT

Outils

Installer les outils suivant notepad++, vagrant, virtualbox, putty, eclipse

NotePad++ : <http://www.notepad-plus-plus.org>
VirtualBox : <http://www.virtualbox.org>
Vagrant : <http://www.vagrantup.com>
PuTTY : <http://www.putty.org>
Eclipse : <http://eclipse.org>

Serveur Git :

- Interface web : <http://gogs.sd.lan:10080>
- Accès SSH : `ssh://gogs.sd.lan:10022`

Documentation

<http://www.croes.org/gerald/blog/introduction-a-git-12/607/>
<https://git-scm.com/book>
<http://www.croes.org/gerald/blog/git-modele-de-branche-efficace/649/>
<http://nvie.com/posts/a-successful-git-branching-model/>
https://wiki.eclipse.org/EGit/User_Guide

Objectif

Le but est d'avoir un environnement reflétant un serveur de production. Cette infrastructure doit fonctionner sur tous systèmes d'exploitation (Windows, Linux, Mac OS X)

Nous utiliserons des machines virtuels (VirtualBox) contrôlées par Vagrant.

La machine virtuelle devra partager ces fichiers avec la machine host.

Installation de l'environnement de développement

- Installer NotePad++, VirtualBox, Vagrant et PuTTY
- Installer java (32-bit ou 64-bit)
- Installer eclipse (32-bit ou 64-bit suivant la version de java)
 - Décompresser le fichier zip, puis copier le répertoire eclipse dans `C:\Users\<<your user>`
 - Créer un raccourci de eclipse.exe sur votre bureau ou autre
- Démarrer eclipse
- Récupérer le fichier contenant la liste des plugins (fichier p2f)
- Importer le File > Import > Install > Install Software Items from File
- Après le redémarrage d'eclipse
- Help > Check for update
- Et faire les mise à jours.

Configurer Eclipse pour Git (<http://git-scm.com>)

Pour utiliser git, il faut créer une clé privée-public dans eclipse qui sera utilisé avec le SSH d'eclipse

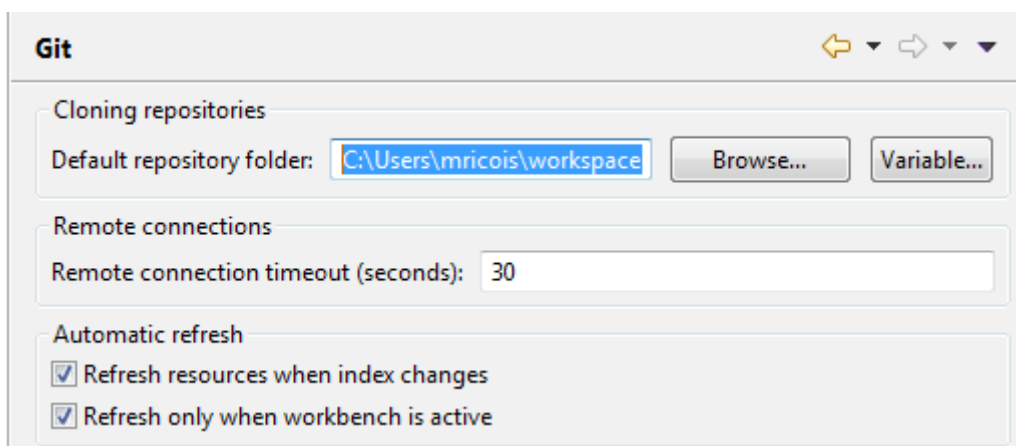
Eclipse > Window > Preferences => General > Network Connections > SSH2
Dans Key Management > Generate RSA Key

Save Private Key (seulement pour vous à ne pas distribuer)
Récupérer le fichier C:\Users\<<your name>\.ssh\key.pub et l'intégrer sur l'interface web de Git.

L'administrateur de Git devra créer les autorisations si nécessaires.

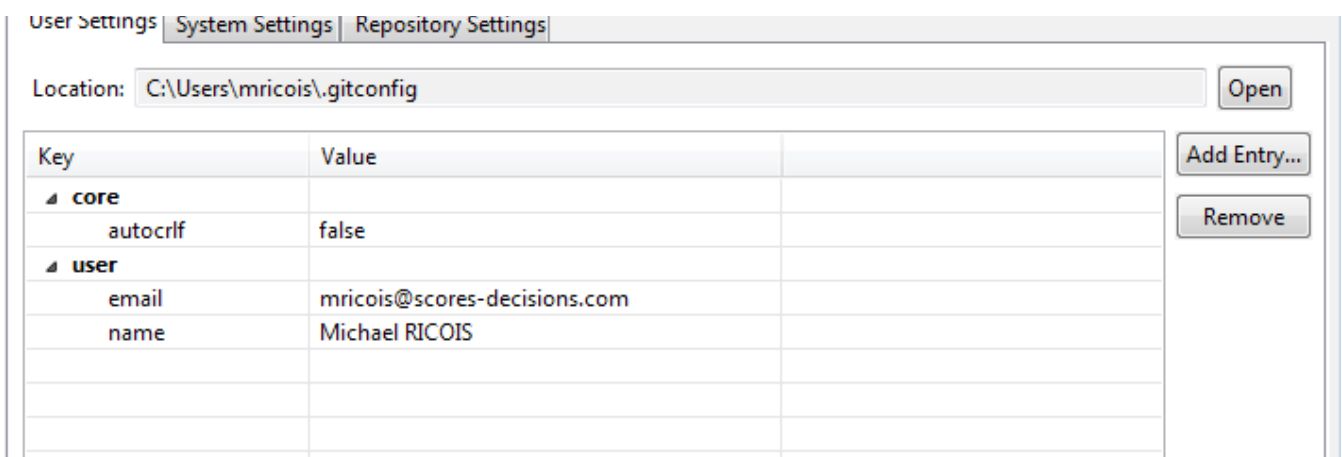
Configurer Git dans eclipse

Eclipse > Window > Preferences > Team > Git



Eclipse > Window > Preferences > Team > Git > Configuration

Définir "email" et "name", utilisé lors des commit



Toutes les préférences sont disponible dans le menu

Window > Preferences > Team > Git

Configurer Subversion dans Eclipse

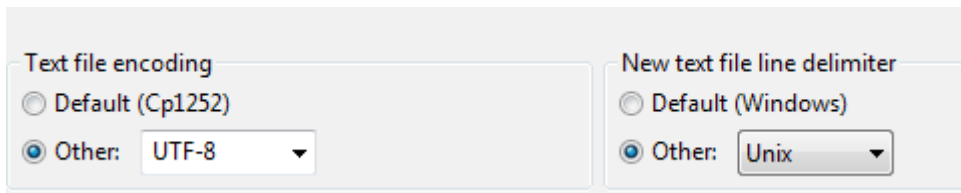
Toutes les préférences sont disponibles dans le menu

Window > Preferences > Team > SVN

Console => On output

Configurer le Workspace

Window > Preferences > General > Workspace



Fichiers à ignorer dans SVN et Git

Window > Preferences > Team > Ignored Ressources

Ajouter les patterns :

```
.project  
.buildpath  
.settings
```

Installation de l'infrastructure

Créer un répertoire pour le stockage du workspace

```
C:\Users\\workspace
```

Ce répertoire doit servir pour le stockage de tout les projets et ainsi réaliser l'execution et le débuggage à partir des machines virtuelles.

Récupérer les fichiers de définition depuis Git (Vagrantfile et autres), projet envdev (ces fichiers sont compatible windows, mac et linux)

```
File > Import > Git : Projects from Git  
ssh://git@gogs.sd.lan:10022/envdev.git
```

Depuis, Existing local repository ou Clone URI
Use the New Projet wizard

Charger une machine virtuelle

Executer **cmd.exe**

```
cd <def name>
```

Ce placer dans un répertoire d'une machine (ex: C:\Users\\workspace\envdev\lamp-trusty64)

Executer la machine **vagrant up**

Au premier démarrage la machine est créer et configurer automatiquement avec tous

les outils nécessaires (phpmyadmin et vhost de test)

Installer les additions invités
vagrant plugin install vagrant-vbguest
vagrant halt
vagrant up

Pour se connecter à la machine virtuelle, Utiliser PuTTY avec les paramètres suivants :

Host Name : 127.0.0.1:2222
Connection > Data > Auto-login username : *vagrant*
Mot de passe : *vagrant*

Dans le répertoire C:\Users*<your user>*\workspace*<def name>*\vhosts, il est possible d'ajouter des vhosts.

Paramètres de la machine virtuelle
IP : 192.168.33.10
Vhost Path : /home/vhosts

Pour provisionner à nouveau, exécuter dans cmd, la commande **vagrant provision**.
(Mise à jour, redémarrage)

Afin de démarrer plus rapidement la machine virtuelle.
Utiliser pour éteindre : **vagrant suspend** et pour redémarrer **vagrant resume**.

Pour arrêter la machine proprement : **vagrant halt**.

Gestion des vhosts

Dans windows, exécuter notepad++ en tant qu'administrateur et ouvrir le fichier
C:\Windows\System32\drivers\etc\hosts

Définir les vhosts suivants :

```
192.168.33.10    test.sd.dev
192.168.33.10    phpmyadmin.sd.dev
```

- Importer des projets depuis Git

http://wiki.eclipse.org/EGit/User_Guide

Pour le Master, utiliser File > Import
Pour les branches, ouvrir la perspective Git, Branches, Remote Tracking > Clic droit puis Checkout

Pour accéder à git : `ssh://git@gogs.sd.lan:10022/<projet>.git`

- Importer des projets depuis SVN

<http://www.eclipse.org/subversive/documentation/index.php>

Ouvrir la perspective SVN

File > New > Other > Project from SVN

Choisir le projet, Browse puis choisir la branche et ensuite utiliser "Check out as a project configured using the New Project Wizard"

Pour accéder à SVN : <https://xxx.xxx.xxx.xxx/svn/<projet>>

Ancienne Méthode

N.B : Install a MySQL server or MariaDB

Software version

This guide is applicable to these versions, especially on Windows 7

Apache 2.4.x
PHP 5.5.9

Download

<http://www.apachelounge.com/>
httpd-2.4.7-win64-VC11.zip

<http://windows.php.net/>
vcredist_x64.exe
php-5.5.9-Win64-VC11.zip

[http://windows.php.net/downloads/pecl/releases/
php_apcu-4.0.4-5.5-ts-vc11-x64.zip](http://windows.php.net/downloads/pecl/releases/php_apcu-4.0.4-5.5-ts-vc11-x64.zip)

Directory tree to store all files

1. Create, for example : C:\Server

2. Tree example

```
Apache247
PHP
  ini
    5.3.27
    5.5.9
  PEAR
    php-5.3.27
    php-5.5.9
vhosts
```

3. Extract "httpd" in a directory name Apache{version}

4. Extract "php" in a directory name php-{version}

5. Configure your php

Add a directory, to store configuration file, in PHP\ini\{your_php_version}

Add in it all your conf files as php.ini, php_browscap.ini

Edit php.ini and configure "extension_dir"

6. Configure Apache

Create 3 files in C:\Server\Apache247\conf

- httpd.head.conf
- httpd.php{version}.conf
- httpd.foot.conf

Copy the content of httpd.conf from the beginning through "Supplemental" in httpd.head.conf

Copy the content of httpd.conf from "Supplemental" through the end in httpd.conf

Now in file httpd.php559.conf, add

```
ServerRoot "C:/Server/Apache247"  
Include "C:/Server/Apache247/conf/httpd.head.conf"  
ServerName localhost  
LoadModule php5_module "C:/Server/PHP/php-  
5.5.9/php5apache2_4.dll"  
PHPIniDir "C:/Server/PHP/ini/5.5.9"  
AddHandler application/x-httpd-php .php  
Include "C:/Server/Apache247/conf/httpd.conf"
```

Go back in head and foot file

- Comment ServerRoot
- Change path c:\Apache24 par c:\Server\Apache247
- Activate all needed modules
- Uncomment Include extra/httpd.vhosts.conf
- Change extra/httpd.vhosts.conf
- Add all your vhosts conf file in C:\Server\vhosts

7. Install the HTTPD service in windows

```
http.exe -k install -n "Apache247-PHP559" -f  
C:\Server\Apache247\conf\httpd.php559.conf
```

To remove it :

```
http.exe -k uninstall -n "Apache247-PHP559"
```

Debug with xdebug

<http://xdebug.org>

1. Download xdebug module
2. Add it to php

```
zend_extension=php_xdebug-2.2.5-5.5-vc11-x86_64.dll
```

```
[xdebug]  
xdebug.remote_enable = 1  
xdebug.remote_port = 9000  
xdebug.remote_connect_back = 1
```

See <http://xdebug.org/docs/remote>

3. Restart apache

4. Check in phpinfo()

5. Install a browser extension

Firefox : The easiest Xdebug - <https://addons.mozilla.org/en-US/firefox/addon/the-easiest-xdebug/>

Chrome : Xdebug Helper for Chrome -

<https://chrome.google.com/extensions/detail/eadndfjplgielddbigjakmdgkmoaaaoc>

6. Configure eclipse

Preferences > PHP > Debug > Installed debugger