

Linux en tant que poste de développeur



Aurélie Vaché
Développeuse Web chez atchikservices

Lorsque j'ai été embauchée par atchikservices il y a 9 ans, J'ai commencé avec une Fedora Core 3 pendant quelques années, puis je suis passée à la distribution Ubuntu. Il a fallu s'habituer à l'environnement de bureau Unity, mais on s'y fait vite :-). Je développe et maintiens des applications en production et je pense qu'il n'y a pas mieux que Linux pour cela. J'ai un double écran et avoir par exemple, mon IDE Eclipse, mon navigateur ou mon logiciel de messagerie Thunderbird sur l'écran de gauche, et, sur celui de droite, l'explorateur de package d'Eclipse ou le terminal, c'est excellent. Vous allez vous rendre compte dans cet article qu'avoir un environnement de Dev sur Linux permet d'améliorer la productivité. Il est plus facile par exemple de gérer différentes versions de vos outils préférés (java, maven ...), de compiler vos projets, et vous pourrez même avoir un environnement de dev semblable à celui de production.

Une distribution tu choisiras

Vous avez choisi de passer votre environnement de développement à Linux, c'est un bon choix ! En premier lieu, il faut que vous choisissiez une distribution, et, parmi la multitude des distributions Linux qui existent, je suis d'accord il y a de quoi être perdu et ne pas savoir quoi choisir. Afin d'aiguiller votre choix, il faut savoir qu'en ce moment il y a 3 distributions qui sont prisées par les développeurs :

- Ubuntu
- Linux Mint
- ArchLinux [Fig.A](#)

Si vous avez déjà entendu parler de Kali Linux, sachez que cette dernière est basée sur Debian et est plus réservée aux tests de sécurité.

On peut choisir une distribution pour son Desktop différente de celles des serveurs de tests et de production : je code sur une Ubuntu et les applications que je déploie tournent sur une CentOS 7. Pourquoi choisir une distribution et pas une autre ? Je dirais que cela dépend de vos besoins, du contexte, de votre affinité avec une distribution, si vos collègues codent déjà sur une distribution particulière, si vous désirez une interface graphique ou non ... Bref, c'est à vous de choisir ! :-)

Des paquets tu installeras :

Vous avez choisi et installé une distribution Linux. Parfait! Maintenant il



faut s'attaquer aux paquets/aux outils que vous devez installer.

La liste des paquets que vous devrez installer dépendra de vos besoins ainsi que des technologies et langages que vous utilisez. Je vais vous en lister quelques-uns mais si vous avez un doute sur le paquet à installer, pas de panique, Internet regorge de tutoriels :-).

Avant toute chose, les exemples que je vous donne ci-dessous sont basés sur Ubuntu donc j'utilise apt-get pour installer les paquets (car il s'agit d'un OS basé sur Debian tout comme Linux Mint). Mais si vous avez installé un système basé sur une Red Hat telle que Fedora, il vous faudra utiliser yum à la place ou encore pacman pour Arch Linux.

Apache

```
$ sudo apt-get install apache2
```

MySQL

```
$ sudo apt-get install mysql-server
```

MongoDB

En premier lieu il faut importer la clé publique dans votre machine :

```
$ sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 7F0CEB10
```

Ensuite il faut ajouter le repository de mongodb :

```
$ echo "deb http://repo.mongodb.org/apt/ubuntu \"$(lsb_release -sc)\"/mongodb-org/3.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb.list
```

Et pour finir, vous pouvez installer le paquet :

```
$ sudo apt-get update
$ sudo apt-get install mongodb
```

PHP

```
$ sudo apt-get install php5 libapache2-mod-php5 php5-mcrypt php5-mysql php5-soap
```

Java

Vous pouvez installer plusieurs versions de la JVM si vous le désirez et surtout si vous en avez besoin. Pour cela il faut que vous alliez sur le site d'Oracle, que vous téléchargiez l'archive, que vous la déplaçiez dans le répertoire /opt/ ou /usr/lib/jvm/, par exemple, et, ensuite vous pouvez l'installer :

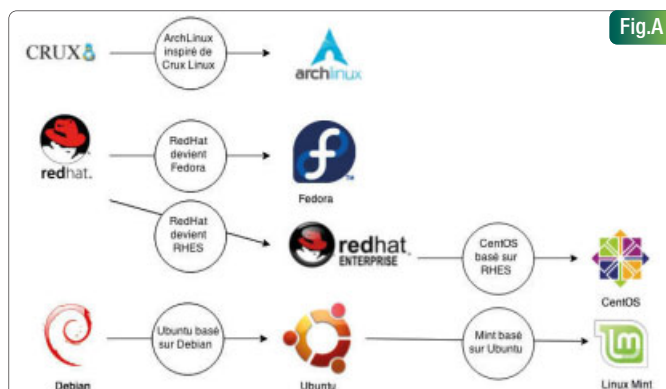


Fig.A

```
$ cd /usr/lib/jvm/
$ tar xzvf jdk-8u45-linux-i586.tar.gz
$ sudo update-alternatives --install "/usr/bin/java" "java" "/usr/lib/jvm/jdk1.8.0_45/bin/java" 500
```

On peut de cette manière installer plusieurs versions de Java sur sa machine et activer à tout moment celle que l'on souhaite :

```
$ sudo update-alternatives --config java
```

Il existe 5 choix pour l'alternative java (qui fournit /usr/bin/java).

Sélection	Chemin	Priorité	État
0	/usr/lib/jvm/java-7-openjdk-i386/jre/bin/java	1071	mode automatique
1	/opt/java/bin/java	1	mode manuel
2	/opt/java/bin/javac	1	mode manuel
3	/usr/lib/jvm/java-7-openjdk-i386/jre/bin/java	1071	mode manuel
* 4	/usr/lib/jvm/jdk1.7.0_25/bin/java	1	mode manuel
5	/usr/lib/jvm/jdk1.8.0_45/bin/java	500	mode manuel

Appuyez sur <Entrée> pour conserver la valeur par défaut[*] ou choisissez le numéro sélectionné :

Vous pouvez également activer la version de Java que vous souhaitez si vous la connaissez :

```
$ sudo update-alternatives --set java /usr/lib/jvm/jdk1.8.0_45/bin/java
```

De cette manière il est super simple d'utiliser java 7 pour un projet, java 8 pour un autre qui utilise les lambda expressions ou java 6 pour un vieux projet par exemple.

Python

```
$ sudo apt-get install libssl-dev openssl
$ wget https://www.python.org/ftp/python/3.4.1/Python-3.4.1.tgz
$ tar -xvf Python-3.4.1.tgz
$ cd Python-3.4.1/
$ ./configure
$ make
$ sudo make install
```

Maven

```
$ sudo apt-get install maven
```

Subversion

```
$ sudo apt-get install subversion
```

Git

Si vous souhaitez installer Git, il vous faudra installer au préalable les bibliothèques suivantes : curl, zlib, openssl, expat et libiconv. Si vous n'avez pas déjà installé la librairie libssl-dev qui est nécessaire pour Python notamment, n'oubliez pas de l'installer pour Git.

```
$ sudo apt-get install libcurl4-gnutls-dev libexpat1-dev gettext libz-dev libssl-dev
```

Puis il ne vous reste plus qu'à installer le package de Git :

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

Android

Si vous souhaitez développer des applications natives sous Android et que vous êtes sur un Ubuntu 64 bits, il vous faudra installer la librairie de compatibilité en 32 bits :

```
$ sudo apt-get install ia32-libs
```



De la configuration tu feras

Pour que votre poste de travail ait toutes les bonnes variables d'environnement configurées, les bons paths, il faut créer un fichier .profile et l'appeler dans votre fichier .bashrc :

```
$ cat ~/.bashrc
```

```
...
source ~/.profile
```

Voici un aperçu de mon fichier .profile :

```
$ cat ~/.profile
```

```
...
# some more ls aliases
alias ll='ls -alF'
alias la='ls -A'
alias l='ls -CF'
```

```
# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
```

```
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi
```

```
JAVA_HOME="/usr/lib/jvm/jdk1.7.0_25/"
JAVACMD="$JAVA_HOME/bin/java"
PATH="/usr/lib/jvm/jdk1.7.0_25/bin:/home/my_user/applis/android-sdk-linux:$PATH"
export JAVA_HOME
export PATH
```

```
export MAVEN_OPTS="-Dmaven.wagon.http.ssl.insecure=true -Dmaven.wagon.http.ssl.allowall=true"
```

```
# The next line updates PATH for the Google Cloud SDK.
source /home/my_user/applis/google-cloud-sdk/path.bash.inc
```

```
# The next line enables bash completion for gcloud.
source /home/my_user/applis/google-cloud-sdk/completion.bash.inc
```

```
export ANDROID_EMULATOR_FORCE_32BIT=true
```

```
export ANDROID_HOME="/home/my_user/applis/android-sdk-linux/"
```

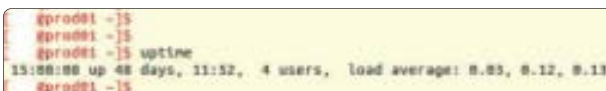
```
# Choose the default editor for crontab edition
export EDITOR=vim
```

Lorsque vous effectuez une modification du `.bashrc`, si vous ne voulez pas sortir de la session, il vous suffit de le sourcer :

```
$ source ~/.bashrc
```

Il est également possible de personnaliser le prompt de son shell ou celui des serveurs que l'on administre via ssh, c'est pratique et tout se passe dans le fichier `.bashrc` :

```
#prompt changes default color to red in production servers
DEFAULT="\e[00m"
RED="\e[31m"
PS1="$RED[\u@\h \W]\$ $DEFAULT";
```



```
gprod01 -]$
gprod01 -]$
gprod01 -]$ uptime
13:00:00 up 48 days, 11:52, 4 users, load average: 0.05, 0.12, 0.13
gprod01 -]$
```

A noter que sur Ubuntu, au lieu de setter une valeur pour la variable `EDITOR`, comme vu dans le fichier `.profile`, vous pouvez re-sélectionner à tout moment votre éditeur par défaut :

```
$ select-editor
```

Select an editor. To change later, run 'select-editor'.

1. /bin/ed
2. /bin/nano <---- easiest
3. /usr/bin/vim.basic
4. /usr/bin/vim.tiny

Choose 1-4 [2]:

Pour vérifier que les dernières versions par exemple de Java, de maven ou de php que j'ai installées sont bien présentes dans mon PATH, il suffit de le demander en ligne de commande :

```
$ java -version
java version "1.7.0_25"
Java(TM) SE Runtime Environment (build 1.7.0_25-b15)
Java HotSpot(TM) Server VM (build 23.25-b01, mixed mode)
```

```
$ mvn -v
Apache Maven 3.0.5
Maven home: /usr/share/maven
Java version: 1.7.0_25, vendor: Oracle Corporation
Java home: /usr/lib/jvm/jdk1.7.0_25/jre
Default locale: fr_FR, platform encoding: UTF-8
OS name: "linux", version: "3.13.0-49-generic", arch: "i386", family: "unix"
```

```
$ php -v
PHP 5.6.7-1+deb.sury.org~trusty+1 (cli) (built: Mar 24 2015 11:13:27)
Copyright (c) 1997-2015 The PHP Group
Zend Engine v2.6.0, Copyright (c) 1998-2015 Zend Technologies
with Zend OPcache v7.0.4-dev, Copyright (c) 1999-2015, by Zend Technologies
```

```
$ mongo --version
MongoDB shell version: 3.0.0
```

Pour me connecter aux différents serveurs que l'on gère je me connecte en ssh. Et si l'on veut se faciliter la vie et gagner du temps, au lieu de sai-

sir à chaque fois le port du serveur (si différent du port 22) et le login, on peut gérer tout ce petit monde dans le fichier config de SSH :

```
$ cat .ssh/config
Host dev01
  HostName dev01
  Port 5555
  User dev

Host dev02
  HostName dev02
  Port 5555
  User dev

Host rec01
  HostName recette01
  Port 5554
  User dev
```

Et pour éviter d'avoir à chaque fois à saisir le mot de passe, il suffit d'insérer votre clé publique dans le fichier `~/.ssh/authorized_keys` du serveur distant.

Des outils tu te muniras

Pour installer des outils, vous le savez, il existe le gestionnaire d'installation et de désinstallation de paquets en ligne de commande, mais il existe également une version graphique (la Logithèque Ubuntu par ex. ou Synaptic), et il est également possible d'installer un logiciel via son package RPM. Quotidiennement j'utilise une multitude d'outils sur Linux, en voici une liste non exhaustive :

Éditeurs de texte :

- VI(M). J'adore VI personnellement, c'est un éditeur de texte super léger et très pratique. C'est mon partenaire du quotidien. En quelques secondes et en tapant quelques caractères seulement, il est possible d'effectuer des actions sur un fichier texte.

Imaginez que vous avez un fichier contenant des IDs, il est possible de rajouter au début de chaque ligne du fichier une commande SQL SELECT :

```
:%s#^#SELECT name from Client where client_id='#g
```

Et vous pouvez fermer votre instruction SELECT également :

```
:%s#$$#;#g
```

Si vous voulez supprimer 10 lignes d'un coup dans un fichier :

```
10dd
```

Ce n'est pas le sujet de l'article donc je vais arrêter là, mais pour moi il s'agit d'un outil indispensable :-).

- Gedit

IDE :

- Eclipse
- Netbeans
- IntelliJ

Outils en ligne de commande :

- file : permet de connaître les informations d'un fichier,
- wc : permet d'obtenir des statistiques sur un fichier comme par

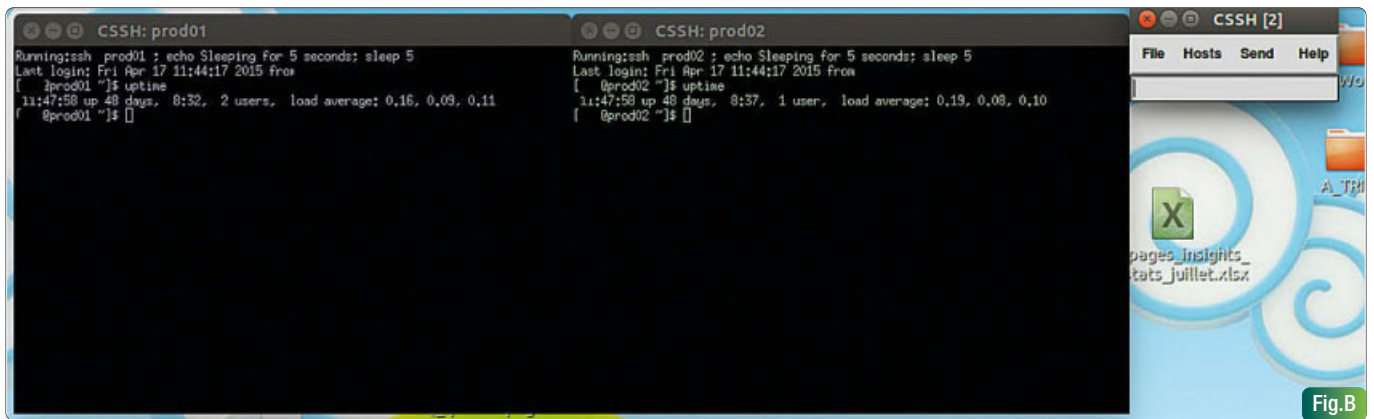


Fig.B

- exemple le nombre de lignes, le nombre de mots ...,
- find/grep : pour rechercher,
- split : pour découper un fichier en plusieurs,
- tar/zip/bzip2/gzip : permet de compresser et décompresser un ensemble de fichiers
- sort : permet de trier,
- head/tail/less : permet de lire un fichier, respectivement les premières, les dernières lignes et son contenu entier,
- cat : n'a aucun rapport avec les chats ;) et permet de concaténer plusieurs fichiers dans un seul fichier par exemple,
- cut : permet d'extraire des zones d'un fichier,
- diff : permet de comparer deux fichiers,
- top : permet d'obtenir la liste dynamique des processus classés par occupation mémoire, CPU ...,
- ps : permet d'obtenir la liste statique des processus qui tournent au moment où vous lancez la commande,
- sed : imaginons que l'on veut remplacer une URL par une autre dans plusieurs fichiers d'un site Web par exemple, on peut faire un petit script shell utilisant sed, c'est super simple :

```
$ cat updater.sh
#!/bin/sh
#####
##
## Ce script remplace un mot par un autre mot sur plusieurs fichiers ##
##
#####
FILES=`grep -rl programmez_185 /home/<your_user>/conf /home/<your_user>/bin
/home/<your_user>/local/install/*.crontab`
for i in $FILES
do
    echo "replacing programmez_185 by programmez_186 in $i"
    sed -r "s:programmez_185:programmez_186:g" $i > $i.new && mv -f $i.new $i
done
```

- awk : très utile pour les logs Apache par exemple...
- wget : permet de télécharger un fichier d'après son url
- ...

Utilitaires :

- Pour rapidement effectuer des captures écran qui sont stockées dans le presse papier, il faut appuyer en même temps sur les touches CTRL+ALT+Impr écran, puis sélectionner ce que vous voulez capturer. C'est super pratique !
- Libre Office

- Cluster SSH : cet outil permet de lancer des commandes sur plusieurs serveurs en même temps.

```
$ cssh prod01 prod02
```

Fig.B

- Remmina : il s'agit d'un visionneur de bureau distant. C'est pratique pour se connecter à distance sur des serveurs/PC qui sont sous Windows par exemple.
- Wine : permet d'exécuter *certaines* programmes Windows sous Ubuntu. C'est un outil pratique pour installer Photoshop par exemple. La liste des logiciels compatible est disponible sur Internet.

Navigateurs :

- Firefox
- Chrome
- Opera
- ...

Graphisme :

- Gimp
- Inkscape

FTP :

- FileZilla
- Il existe également le plugin FireFTP pour Firefox et on peut se connecter à un serveur en FTP en ligne de commande aussi.

Virtualisation :

Il peut arriver que l'on ait besoin d'utiliser des logiciels tournant uniquement sur Windows, pas de problème, les outils de virtualisation sont faits pour ça, ils permettent d'utiliser Windows dans Linux. Les outils de virtualisation permettent de créer une machine virtuelle et d'y installer n'importe quel système d'exploitation.

- VirtualBox
- VMWare

Conclusion

Comme le dit le dicton : "Il y a une application pour tout", il s'agit de la même chose sur un environnement Linux pour le Développement. Et contrairement au mythe, non il ne faut pas recompiler le noyau Linux tous les jours ;-).

