Installation d'un serveur LAMP

Le serveur sera installé dans une machine virtuelle (invitée) sous VirtualBox, avec la configuration de deux CMS : Wordpress et Dokuwiki.

Pour les “novices” ou peu habitués de la ligne de commande, le serveur fonctionne uniquement dans ce mode, soit dans sa console, soit via ssh qui ouvre un terminal à partir d'un poste distant (sur la machine hôte par exemple). Les copier-coller fonctionne, mais avec la touche “shift” en plus : shift-ctrl-c pour copier, shift-ctrl-v pour coller.

Suggestion : une fois le serveur installé, n'utilisez la console dans la machine invitée que pour vous authentifier, exécuter la commande 'ip a' pour connaître l'IP et arrêter la machine. OpenSSH étant installé, il vous sera beaucoup plus facile de la coontrôler à distance, tout comme vous le feriez pour un serveur dédié dans un datacenter payant.

> Pour les noms d'utilisateurs et mots de passe utilisés sur vos serveurs, évitez l'utilisation de noms d'utilisateur et mots de passe personnels.

Installation de VirtualBox

Installation via ppa

Pour installer la version la plus récente, suivre les instructions de la page https://www.skyminds.net/linux-installer-virtualbox-via-le-ppa-doracle/ :

```
sudo apt-get install build-essential dkms linux-source linux-headers-`uname -r`
```

```
echo "deb http://download.virtualbox.org/virtualbox/debian `lsb_release -sc` contrib" | sudo tee -a /etc/apt/sources.list.d/virtualbox.list && wget -q http://download.virtualbox.org/virtualbox/debian/oracle_vbox.asc -O- | sudo apt-key add - && sudo apt-get update && sudo apt-get install virtualbox-4.3
```

```
sudo usermod -G vboxusers -a $USER
```

(remplacez la dernière variable par votre nom d'utilisateur Linux).

```
version=$(VBoxManage --version|cut -dr -f1|cut -d'_' -f1) && wget -c http://download.virtualbox.org/virtualbox/$version/Oracle_VM_VirtualBox_Extension_Pack-$version.vbox-extpack && VBoxManage extpack install Oracle_VM_VirtualBox_Extension_Pack-$version.vbox-extpack --replace && rm *.vbox-extpack
```

Génération de clés SSH

L'authentification par un système de clés publique/privée vous sera très utile pour gérer vos serveurs. Générez ces clés avec la commande suivante :

```
ssh-keygen -t rsa
```

Les clés publiques et privées sont stockées dans votre répertoire ~/.ssh. Les répertoires cachés (hidden) peuvent être visualisés dans votre gestionnaire de fichier (Nautilus) en tapant ctrl-h. La clés privées ne doit JAMAIS être communiquée. Votre clé publique doit être placée dans le fichier ~/.ssh/authorized_keys des serveurs auxquels vous souhaitez accéder.

ssh-copy-id (ajouter)

Installation du serveur

Installation standard de Linux, Apache, MySQL, Php + outils. Base choisie : Ubuntu server

- téléchargement de Ubuntu server
- Lancement de VirtuaBox et création d'une nouvelle machine virtuelle (VM). Par exemple :
  - Ubuntu-14.04.2_server
  - 1024 Mo
  - Disque dur virtuel (.vdi) 8 Go (dynamiquement alloué)
- Ajustement (configuration) de quelques paramètres de la VM :
  - stockage : renseigner le fichier iso dans le lecteur CD/DVD virtuel
  - réseau : accès par pont
- Démarrer la VM. Le live-CD virtuel de Ubuntu propose le choix de la langue (prendre français) et la touche F3 permet de choisir le clavier (par exemple belge). Sélectionner “installation du serveur”.
- Renseigner des paramètres :
  - nom de la machine (ex : vmix)
  - nom complet de l'utilisateur (ex : Virtual User)
  - nom d'utilisateur (ex : vmuser)
  - mot de passe (ex : vmuser_mdp) (CHOISISSEZ UN MOT DE PASSE SÛR)
  - choisir de chiffrer ou non, renseigner la timezone
  - partitionnement : choisir “assisté, utiliser un disque entier”
  - appliquer les changements
  - renseigner si nécessaire un serveur mandataire (proxy)
  - Sélection de logiciels à installer
    - openSSH server
    - LAMP server (à moins d'installer séparément après)
  - Définir un mot de passe superutilisateur de MySQL (ex : vmmysqladmin)
  - installer GRUB sur le secteur d'amorçage
  - Fin d'installation et redémarrage du serveur (enlever ultérieurement le CD d'installation virtuel). Vérifier votre login (user/mdp) et relevez l'IP du serveur par la commande “ip a”
  - mises à jour :
    - sudo apt-get upgrade
    - sudo apt-get update

    - redémarrage : sudo reboot

Vérification du fonctionnement et configuration complémentaire

- Tentez de vous connecter à votre serveur vituel via ssh à partir de votre machine hôte pas la commande “ssh vmuser@ip” ou ip est l'adresse ip de votre serveur virtuel (machine invitée). Vous pouvez aussi connecter le système de fichier virtuel via votre navigateur de fichier.
- création d'un utilisateur administrateur “de secours” : sudo adduser rescueadmin suivi de sudo usermod -a -G sudo rescueadmin
- à partir d'une autre machine sous Linux, ajout de la clé publique SSH : “ssh-copy-id -i ~/.ssh/id_rsa.pub <username>@<ipaddress>”
- test du serveur web (apache) : tapez “http://ip/” dans un navigateur
- fichier de configuration de php dans /etc/php5/apache2/php.ini

php

Installation complémentaire (utilité incertaine par rapport à l'installation standard) :

```
sudo apt-get install php5 php5-mysql php-pear php5-gd  php5-mcrypt php5-curl
```

- test de php : sudo nano /var/www/html/phpinfo.php et ajouter <?php phpinfo(); ?> comme seule ligne à ce fichier. Sauver et test dans un navigateur : http://ip/phpinfo.php

Groupe www-data

ajouter vmuser au groupe www-data :

```
sudo usermod -a -G www-data vmuser
```

phpmyadmin

Gestion via interface web des base de données :

```
sudo apt-get install phpmyadmin
```

SI ré-installation :

```
sudo dpkg-reconfigure phpmyadmin
```

(choisir le serveur web à reconfigurer : apache2)

- le mot de passe superutilisateur de MySQL sera demandé
- user : phpmyadmin
- mot de passe de connexion : vmuser_phpmyadmin_mdp
- mise à jour du fichier de configuration
    - sudo nano /etc/apache2/apache2.conf

    - ajouter à la fin la ligne “Include /etc/phpmyadmin/apache.conf”
    - sauvegarder, quitter nano
    - sudo service apache2 restart
- accéder à http://ip/phpmyadmin

Installation de wordpress

Création de la DB

- accéder à http://ip/phpmyadmin et s'authentifier avec le code superutilisateur (user=root, pwd=vmmysqladmin)
- activer l'onglet “Utilisateurs”, cliquer sur “Ajouter un utilisateur”
- dans le champ suivant “Entrez une valeur:”, tapez “wordpress” (un bon choix)
- dans le champ client, optez pour “local” et la valeur “localhost”
- pour le mot de passe, choisissez “Entrez une valeur:”, tapez “mdp_wp” (remplacez par un bon mot de passe)
- cochez la case “Créer une base portant son nom et donner à cet utilisateur tous les privilèges sur cette base”
- en bas de la page, cliquez sur “Exécuter”. La DB de wordpress est créée ! vous pouvez quitter phpmyadmin.

Téléchargement

- on peut se placer dans un répertoire temporaire : “mkdir tempo”, “cd tempo”
- wget http://wordpress.org/latest.tar.gz (téléchargement de la dernière version)
- tar -xzvf latest.tar.gz (décompression de l'archive)
- mv wordpress blog (on renomme wordpress en blog, par exemple)
- On rentre dans le répertoire : “cd blog”
- on crée le fichier de configuration à partir de l'exemple fourni : “cp wp-config-sample.php wp-config.php”
- on édite le fichier de configuration wordpress configuration : nano wp-config.php
- dans la section suivante, remplacez 'database_name_here' par 'wordpress', 'username_here' par 'wordpress' et 'password_here' par 'mdp_wp' comme définis lors de la création de la DB. Sauvez et quittez l'éditeur nano

```
// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'database_name_here');

/** MySQL database username */
define('DB_USER', 'username_here');

/** MySQL database password */
define('DB_PASSWORD', 'password_here');
```

: il faut aussi générer des clés de salage avec ce générateur et les placer dans le fichier wp-config.php !

Mise en place, propriétaire et droit

- cd ..
- sudo mv blog /var/www/html
- cd /var/www/html
- changement de propriétaire et groupe vers www-data : “sudo chown -R www-data:www-data /var/www/html/blog”
- “cd /var/www/html/blog” puis changement des droits d'accès
    - fichiers : sudo find . ! -type d -exec chmod 660 {} \;
    - répertoires : sudo find . -type d -exec chmod 770 {} \;
- N.B. : les propriétaire et groupe ont tout, le reste du monde rien. D'autres utilisent 644 et 755 : tout au propriétaire et tout sauf l'écriture au groupe et au reste du monde. Dans des cas spécifiques où l'utilisateur n'a pas les droits d'administration, des fichiers peuvent avoir comme propriétaire l'utilisateur, mais www-data comme groupe. il est nécessaire de maintenir le groupe pour les créations en utilisant les guid (set guid actif). Utiliser alors 0660 et 2770 plutôt que 660 et 770.

Configuration initiale

- dans un navigateur : “http://ip/blog/wp-admin/install.php”
- choisissez la langue d'installation
- Donnez un titre à votre blog, un identifiant administrateur (vous en créerez un autre comme simple utilisateur ou éditeur,...) et un identifiant, ainsi qu'une adresse de messagerie
- exécutez l'installation et connectez-vous ensuite. Vous arrivez dans le tableau de bord de votre site WordPress installé !

Remarques

- Si l'IP de la VM change, il est nécessaire de modifier 2 paramètres de Worpress via phpmyadmin : WP_HOME et WP_SITEURL.

Installation de dokuwiki

Installation de Dokuwiki (https://www.dokuwiki.org/install:upgrade) :

- cd /var/www/html puis sudo mkdir wiki et cd wiki pour se retrouver dans /var/www/html/wiki
- sudo wget http://download.dokuwiki.org/src/dokuwiki/dokuwiki-stable.tgz
- sudo tar -xzvf dokuwiki-stable.tgz –strip-components=1
- sudo rm dokuwiki-stable.tgz
- owner & permissions https://www.dokuwiki.org/install:permissions#unix
- sudo chown -R www-data:www-data /var/www/html/wiki
- cd /var/www/html/wiki
    - fichiers : sudo find . ! -type d -exec chmod 660 {} \;
    - répertoires : sudo find . -type d -exec chmod 770 {} \;
- en cas de migration, effacement de fichiers obsolètes : sudo grep -Ev '^($|#)' data/deleted.files | xargs -n 1 rm -vf
- à partir d'un autre PC : installation http://ip/wiki/install.php

- http://ip/wiki puis mettre à jour les extensions via la page d'administration
- Fix Me! procédure de backup de dokuwiki via cron
- réécriture (URL (url joli) : https://www.dokuwiki.org/fr:rewrite#option_1serveur_web
  - sudo a2enmod rewrite déjà fait via wordpress
  - cd /var/www/wiki
  - sudo mv .htaccess .htaccess.old
  - sudo cp .htaccess.dist .htaccess
  - sudo nano .htaccess (édition suivant les recommandations avec décommentaires & RewriteBase /wiki)
  - fichier de configuration d'Apache : pas de changement nécessaire (fait via sites-available ??)
- administration du wiki : notification activée avec subscribe_time = 24*60*60 (1 jour)
- plugin LaTeX https://www.dokuwiki.org/plugin:latex
  - connection ssh + mise à jour & reboot
  - sudo apt-get install texlive-latex-base imagemagick ghostscript
  - dokuwi administration “gestion des extensions” : copie de l'URL http://github.com/marklundeberg/dokuwiki-plugin-latex/tarball/master
  - configuration → config#pluginlatexplugin_settings_name
  - plugin»latex»convert_options initialement à '-density 120 -trim -transparent “#FFFFFF”', test de '-density 180 -trim -transparent “#FFFFFF”'
- plugin csv : https://www.dokuwiki.org/plugin:csv
- plugin blockquote : https://www.dokuwiki.org/plugin:blockquote
- plugin upgrade https://www.dokuwiki.org/plugin:upgrade
- modification des types mime (autoriser csv) : sudo nano /var/www/wiki/conf/mime.conf

Backups automatiques

Pré-requis

- serveur LAMP installé
- accès ssh (y compris clés)
- noms et localisations connus des fichiers de configurations, répertoires et fichiers des sites, bases de données (avec noms d'utilisateurs et mots de passe)
- un autre serveur accessible pour le backup distant (remote server)

Constitution du backup local

À partir du répertoire home de l'utilisateur administrateur, créer un répertoire *backups*, un script (vide) *backup.sh* et une répertoire pour les fichiers *files* :

```
mkdir backups
cd backups
touch -a backup.sh
chmod u+x backup.sh
mkdir files
```

```
#!/bin/sh
```

```
THESITE=""
THEDB="wordpress"
THEDBUSER="wordpress"
THEDBPW="mdp_wp"
THEDATE=`date +%F-%T`

# backup de la base de données
mysqldump -u $THEDBUSER -p${THEDBPW} $THEDB | gzip >
~/${THESITE}backups/files/dbbackup_${THEDB}_${THEDATE}.bak.gz

# backup des fichiers du site
tar -czf ~/${THESITE}backups/files/sitebackup_${THESITE}_${THEDATE}.tar.gz
/var/www/${THESITE}

# éliminer les fichier vieux de plus de 5 jours
find /backups/files/site* -mtime +5 -exec rm {} \;
find /backups/files/db* -mtime +5 -exec rm {} \;
```

- Il s'agit de mettre le script backup.sh rendu exécutable dans le cron : utiliser la commande d'édition du cron (tâches automatiques) "crontab -e", vous pouvez par exemple démarrer la sauvegarde chaque jour à 04h25 par la ligne :
  - `25 4 * * * /home/username/backups/backup.sh`

Sauvegarde distante

En mode “push”, voici que qu'il faut ajouter au script :

```
# synchronisation par rsync (push)
rsync -a ~/backups/files
user@machine_distante:/Path/To/Backup/offsite_backup_name
```

Mode “pull” : ...

Références

- http://doc.ubuntu-fr.org/virtualbox
- http://doc.ubuntu-fr.org/lamp
- http://ubuntuhandbook.org/index.php/2014/04/install-set-up-wordpress-ubuntu-14-04/
- http://smarterware.org/9572/automatically-back-up-your-web-site-every-night : script automatique et transfert par rsync (avec des erreurs/anomalies...). Adapté de cette ref.
- http://www.cyberciti.biz/faq/linux-unix-formatting-dates-for-display/ : formatage de la date et de l'heure

Difficultés

- https://forums.virtualbox.org/viewtopic.php?f=1&t=62339 : virtualbox ne permettant pas le 64 bits

From:
https://dvillers.umons.ac.be/wiki/ - **Didier Villers, UMONS - wiki**

Permanent link:
https://dvillers.umons.ac.be/wiki/floss:server_lamp_install?rev=1434811473

Last update: **2015/06/20 16:44**