DUCHENE Kévin Le 03/04/2016

CORENTHIN Didier

KAID KASBAH Yanis

Projet d’administration réseau

**Sommaire**

Introduction

Etape 1 : L’Installation

Etape 2 : La maquette fonctionnelle

Etape 3 : Le test de l’installation

Etape 4 : L’automatisation de l’installation

Etape 5 : Le changement des utilisateurs (maquette)

Etape 6 : Le changement des utilisateurs (Automatisation)

Etape 7 : L’automatisation à partir d’une image inconnue

Etape 8 : La réalisation d’images de test

Conclusion

**Introduction**

Ce projet a été réalisé de façon collaborative afin de pouvoir accomplir un maximum de tâches dans le temps imparti.

Sur un service déjà existant et fonctionnel, nous devions changer les utilisateurs sans changer de système et conserver ce système fonctionnel après l'opération.

Le changement d'utilisateur devait se faire de façon totalement automatique.

Nous travaillons sous **Debian 8.2** après une NetInstall

Il faut créer l’utilisateur **Carole** avec pour mot de passe **Carole** et lui accorder les droits sudo.

Nous devions installer le service **phpmyadmin** et le configurer manuellement dans un premier temps, puis automatiser cette tâche.

Il fallait ensuite tester le service, changer les utilisateurs **MySQL** et **Apache**, puis effectuer divers tests.

**Etape 1 : L’Installation**

Nous avons installé Debian 8.2 en utilisant VMWare.



Les contraintes imposées sont les suivantes :

* Un serveur **SSH** doit tourner
* Un compte **root** permet de se connecter à la machine avec le mot de passe **daryl**
* Le nom d’hôte est **rick**

Affectation des droits root à « **carole** » (Sudo) :

*Avant :*



Commande : **apt-get install sudo**

*Après :*



Le fichier de configuration : **/etc/sudoers** a été généré

Il faut l’éditer grâce à la commande **visudo**, et ajouter une ligne dans la catégorie **User privilege specification**

**Carole ALL=(ALL:ALL) NOPASSWD:ALL**

visudo -c → contrôle du système de syntaxe et réparation des erreurs

Vérification de l’installation du serveur SSH :

Commande : **ssh -V**

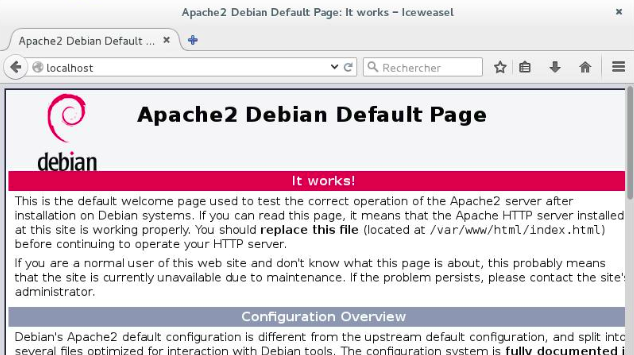


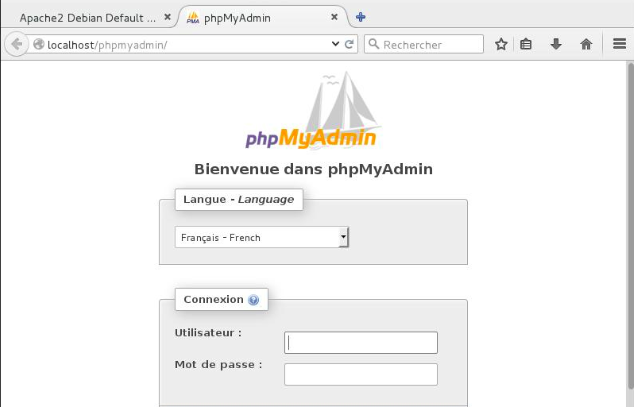
**Etape 2 : La maquette fonctionnelle**

Nous avons installé les logiciels suivants : Apache2, MySQL-Server et phmyadmin.

Les contraintes imposées sont les suivantes :

* MySQL root admin : **glenn**
* Mdp par défaut de phpmyadmin : **maggie**





**Etape 3 : Le test de l’installation**

Nous devions tester l’installation et la bonne configuration de certains logiciels.

Un script devait être créé et retourner les informations de la façon suivante :

hostname : rick

http\_state : OK

mysql\_state : OK

phpmyadmin\_state : OK

http\_user\_conf : www-data

mysql\_user\_conf : mysql

http\_user\_pid : www-data

mysql\_user\_pid : mysql

Nom de la machine : Il faut récupérer le nom d’hôte dans la **variable $hostname**.

hostname=$(hostname)

Serveur **HTTP** fonctionnel : Création d’un fichier **ping\_me** contenant le mot « **michonne** » à placer dans le répertoire **/var/www/html**

Vérification du contenu de ce fichier via la commande suivante :

$(curl -s http://127.0.0.1/ping\_me)" = "michonne"

* **curl** : Outil pour récupérer des données provenant du serveur web
* **-s** : Option pour rendre la commande silencieuse (Pas de messages d’erreur,…)

Vérification du fonctionnement de **MySQL** : Vérification de l’existence de la base de données

Nous utilisons la commande suivante :

echo "show databases;" | mysql -u root --password=glenn

* **-u root** : L’utilisateur MySQL
* **--password** : Le mot de passe lié au compte root
* Cette commande permet d’exécuter la requête « show databases ; » dans le terminal mysql

**Phpmyadmin** accessible : Vérification du code http et utilisation de **RegEx**

curl --head --silent http://127.0.0.1/phpmyadmin/ | head -1 | awk '{print $2}’

* **curl** : Outil pour récupérer des données provenant du serveur web
* **--head** : Option pour récupérer uniquement l’entête de la page (header)
* **-s** : Option pour rendre la commande silencieuse (Pas de messages d’erreur,…)
* **Head -1** : Permet de n’afficher que la première ligne du résultat
* **awk '{print $2}’** : Affiche la 2ème valeur de la chaine de caractère (le code HTTP)

Filtre du code http :

regex\_php='2[0-9][0-9]' (Récupère uniquement les valeurs de 200 à 299)

Nom d’utilisateur théorique **d’Apache** :

sed -n 's/export APACHE\_RUN\_USER=\(.\*\)/\1/gp' /etc/apache2/envvars

* **sed** : Outil permettant de manipuler des chaines de caractères
* **-n** : option permettant de ne pas afficher (ou uniquement sur demande)
* Recherche de la ligne contenant « **export APACHE\_RUN\_USER= »** dans le fichier **/etc/apache2/envvars** et récupération de la valeur se trouvant après le =
* **p** : permet d’afficher seulement la valeur (demande explicite d’affichage)

Nom d’utilisateur théorique de **MySQL** :

sed -n 's/^user\t\t= \(.\*\)/\1/gp' /etc/mysql/my.cnf

* **sed** : Outil permettant de manipuler des chaines de caractères
* **-n**: option permettant de ne pas afficher (ou uniquement sur demande)
* Recherche de la ligne commençant par « **user\t\t=  »** dans le fichier **/etc/mysql/my.cnf** et récupération de la valeur se trouvant après le = (\t symbolisant la tabulation)

**UID** du service **Apache** :

ps -C apache2 -o user= | uniq | grep -v root

* **ps** : Affiche l’état des processus en cours
* **-C** : option pour filtrer l’affichage en fonction du nom du service
* **-o user=** : option permettant de n’afficher que la colonne correspondant aux utilisateurs qui lancent les processus.
* **Uniq**: permet d’éviter que les résultats soient affichés en double.
* **grep** : Afficher les lignes correspondant à un motif donné
* **-v** : option pour sélectionner les lignes ne correspondant pas au motif (on exclut le résultat root)

**UID** du service **MySQL** :

ps -C mysqld -o user=

* **ps** : Affiche l’état des processus en cours
* **-C** : option pour filtrer l’affichage en fonction du nom du service
* **-o user=** : option permettant de n’afficher que la colonne correspondant aux utilisateurs qui lancent les processus.

Il ne reste plus qu’à afficher les résultats en utilisant la commande echo.

Etape 4 : L’automatisation de l’installation

En reprenant l'image de base, il faut automatiser l'installation et les tests.

Un fichier de configuration devait être créé et contenir les mots de passe et préférences d’installation (install.conf)

MYSQL\_PASSWD=glenn

PMA\_WEBSERVER=apache2

PMA\_APP\_USER=maggie

PMA\_MYSQL\_METHOD=unix socket

PMA\_MYSQL\_ADMIN\_USER=root

Nous avons utilisé le test suivant au début du script d’installation :

if [ $# -eq 0 ]

then

source install.conf

else

source $1

fi

Si un fichier de configuration est passé en paramètre lors du lancement du script, on affiche le contenu du fichier grâce à la commande **source.**

Sinon, on utilise le fichier se trouvant dans le même répertoire (install.conf)

Ensuite, nous mettons à jour la liste des dépôts et des paquets :

apt-get -y update && apt-get -y upgrade

* **apt-get** : Utilitaire pour manipuler des paquets
* **-y** : option pour répondre automatiquement “Oui”

Installation automatique **d’Apache** :

apt-get install -y apache2

Démarrer le serveur Apache : /etc/init.d/apache2 start

Installation automatique **de MySQL** :

On enregistre le mot de passe et la confirmation du mot de passe :

debconf-set-selections <<< "mysql-server mysql-server/root\_password password $MYSQL\_PASSWD"

debconf-set-selections <<< "mysql-server mysql-server/root\_password\_again password $MYSQL\_PASSWD"

debconf-set-selections

* Insérer de nouvelles valeurs par défaut dans la BDD de debconf

Installation automatique du paquet MySQL :

apt-get install -y mysql-server

Démarrer le serveur MySQL : /etc/init.d/mysql start

Installation automatique de **phpmyadmin** :

On enregistre des données :

debconf-set-selections <<< "phpmyadmin phpmyadmin/reconfigure-webserver multiselect $PMA\_WEBSERVER"

debconf-set-selections <<< "phpmyadmin phpmyadmin/dbconfig-install boolean true"

debconf-set-selections <<< "phpmyadmin phpmyadmin/app-password-confirm password $PMA\_APP\_USER"

debconf-set-selections <<< "phpmyadmin phpmyadmin/mysql/admin-pass password $MYSQL\_PASSWD"

debconf-set-selections <<< "phpmyadmin phpmyadmin/mysql/app-pass password $PMA\_MYSQL\_ADMIN\_USER"

debconf-set-selections

* Insérer de nouvelles valeurs par défaut dans la BDD de debconf

Installation automatique du paquet phpmyadmin :

apt-get install -y phpmyadmin

Pour finir, on fait appel au script des tests:

source verifE3.sh

**Etape 5 : Le changement des utilisateurs (maquette)**

Pour cette étape, il faut simplement changer les utilisateurs des services **Apache** et **MySQL**.

Coté **Apache** :

Par défaut : www-data

Devient : **carl** (avec U/GID 500)

Coté **MySQL** :

Par défaut : mysql

Devient : **beth** (avec U/GID 501)