

Rapport Système d'exploitation (SYSG5)

Cedric Thonus et Meihdi El Amouri

Avril 2020

1 Introduction

/! AVANT TOUTE CHOSES /! Changez le SiteDirectory dans le user.json "SiteDirectory" :"/home/lemed/Documents/projet/Site-SYSG5" changez le chemin par votre chemin ou contien le dossier Site-SYSG5.

Notre projet consiste à automatiser la mise en place d'un hébergement mutualisé. C'est à dire, un serveur web subdivisé en plusieurs parties, chaque partie étant occupé par un utilisateur. Chaque partie correspond à un site étant un sous-domaine de localhost. Le répertoire de chaque utilisateur correspond à la racine de son site. Et chaque site est associé à sa propre base de données. L'utilisateur ne peut accepter à son répertoire que via le protocole ftp et il ne peut se connecter en ssh au serveur.

1.1 Mise en place du script

Nous avons créée un script bash (touch script.sh), qui va permettre d'automatiser le projet. Dedans nous nous assurons de l'installation de plusieurs packages :

- Apache2
- MariaDB
- ProFTPD
- PhpMyAdmin
- jq

1.2 Création d'utilisateur

Pour installer ses packages, il faut utiliser cette commande 'sudo zypper install -f -y apache2'. Le "-f" permet de forcer l'installation du package, et le "-y" permet d'automatiser l'installation. Une fois ses paquets installés, il faudra démarrer mysql et apache2 grâce à la commande suivante 'sudo systemctl start [package]'. Le script prendra en paramètre, le nombre d'utilisateurs que nous souhaitons créer. Pour la suite du projet, nous mettrons en place un fichier texte qui reprendra toutes les données souhaitable par l'utilisateur mais pour cette première remise, nous nous contenterons de prendre juste le

nombre d'utilisateurs. Le script fera appelle à la fonction createAccount, dans cette fonction nous bouclerons sur le nombre d'utilisateurs passé en paramètre. Nous allons créer l'utilisateur avec comme nom par défaut "account" suivi de son numéro (sudo useradd "account+i"). Il faut créer un dossier pour l'utilisateur, qui sera donc son répertoire racine (sudo mkdir /home/"account+i") et donner les droits de lecture, d'écriture et d'exécution à l'utilisateur (sudo chmod 755 /home/"account+i") ainsi que le définir propriétaire de celui-ci (chown -R "account+i" /home/"account+i"). On va créer un fichier index.html par défaut pour chaque utilisateur afin de tester le serveur (sudo touch /home/"account+i"/index.html et sudo chmod 755 /home/"account+i"/index.html).

2 lancement de l'application avec ou sans argument

Nous pouvons exécuter le script de deux façons :

- Sans argument donc en utilisant le document JSON
- avec un argument

Si on exécute le script avec un argument, il faut que ce soit un entier qui sera la nombre d'utilisateur sur l'hébergement. Au contraire, il vous faudra remplir le document JSON (voir point suivant).

nous vérifions si il y a un argument de cette grâce à cette condition :

```
if [ $# -eq 0 ]; then
    createAccountJson
else
    create_account
fi
```

3 lancement de l'application avec un document JSON

Comme vu plus haut, si on lance sans argument, il fera appel au JSON, oui mais comment ? Nous allons utiliser un package JQ qui va nous permettre de récupérer les données d'un document JSON. Exemple :

```
name=$(sudo jq '.ListUser['${i}'].user[0].name' user.json)
```

Cela permet de récupérer la variable name stocker dans le tableau ListUser à l'index i qui sera l'utilisateur sur lequel le programme travaille. le user.json est le fichier ou on va aller chercher les informations. Petit bémol, le string retourner comporte des guillemets. Exemple name = "Meihdi" Nous allons donc utiliser le code suivant afin de supprimer le 1er caractère et le dernier :

```
nameUser="${name:1:${#name}-2}"
```

4 Création de l'hébergement mutualisé avec JSON

Voyons comment nous allons procéder. Si on lance le programme sans argument, il va appeler la fonction `createAccountJson()`

```
for i in $(seq 0 $nombreUser); do

    name=$(sudo jq '.ListUser['${i}'].user[0].name' user.json)
    nameUser="${name:1:${#name}-2}"
    nameDb=$(sudo jq '.ListUser['${i}'].user[0].DataBaseName' user.json)
    nameDbWithout="${nameDb:1:${#nameDb}-2}"
    passwordDb=$(sudo jq '.ListUser['${i}'].user[0].DataBasePassword'
                      user.json)
    passwordDbWithout="${passwordDb:1:${#passwordDb}-2}"
    src=$(sudo jq '.ListUser['${i}'].user[0].SiteDirectory' user.json)
    srcDirectory="${src:1:${#src}-2}"
    sudo useradd ${nameUser}
    createSiteWithJson ${nameUser} ${srcDirectory}

    create_vhost ${nameUser}

    create_db ${nameUser} ${nameDbWithout} ${passwordDbWithout}
done
```

Nous allons donc boucler sur le nombre d'utilisateur qui est donc la taille du tableau dans le JSON. Ensuite nous allons récupérer toutes les données et appeler le `createSiteWithJson`.

4.1 Création des vhost.

```
create_vhost() {
    local vhost="<VirtualHost *:80>\nDocumentRoot
/home/$1\nServerName $1.localhost\n<Directory /home/$1>
\nOptions All\nAllowOverride All\nRequire all granted
\n</Directory>\n</VirtualHost>"

    sudo touch /etc/apache2/vhosts.d/$1.conf
    sudo echo -e "127.0.0.1 $1.localhost" >>/etc/hosts
    sudo echo -e $vhost >/etc/apache2/vhosts.d/$1.conf
    sudo systemctl reload apache2
}
```

Cette fonction va créer les vhosts, pour clarifier nous allons faire appelle à la méthode `createVhost`, une variable locale reprendra en paramètre le code ci dessous. Ensuite nous allons créer un fichier `.conf` dans le dossier où se trouve les vhosts. La fonction va injecter dans le fichier `/etc/hosts` une règle pour la

résolution des sous domaines . Une fois toutes la configuration du serveur faite , il vous faudra reload apache 2 via cette commande :

4.2 Mise en place de la base de données.

Il vous faudra désormais créer une base de données pour chaque utilisateur. Nous écrivons une fonction createdb qui se chargera de tout ça. Dans cette fonction nous définirons l'utilisateur de la DB ainsi qu'un mot de passe grace au paramètre passé par la fonction createAccountJson(). Nous allons vérifier si la fonction à les 3 paramètres (nom User , nom DB , password). Si elle a pas ces paramètre car par exemple on veut lancer l'application avec un argument alors par défaut 'admin' sera le password et comme nom de db par défaut, le nom de l'utilisateur +DB. Illustrons tout ça avec un bout de code :

```
create_db() {  
  
    local newUser="$1"  
    if [ -z "$2" ] && [ -z "$3" ]; then  
        newDbPassword='admin'  
        newDb="$1Db"  
    else  
        echo {$3}  
        newDbPassword="$3"  
        newDb="$2"  
    fi  
    host="localhost"  
    commands="CREATE DATABASE \'${newDb}\';  
    CREATE USER \'${newUser}\'@\'${host}\'  
    IDENTIFIED BY \'${newDbPassword}\';  
    GRANT USAGE ON *.* TO \'${newUser}\'@\'${host}\'  
    IDENTIFIED BY \'${newDbPassword}\';  
    GRANT ALL privileges ON \'${newDb}\'.*  
    TO \'${newUser}\'@\'${host}\' ; FLUSH PRIVILEGES ;"  
  
    echo "${commands}" | /usr/bin/mysql -u root  
}
```

Grâce à la requête sql mise dans commands, nous allons créer une DB avec le nom , l'utilisateur , le mot de passe que nous allons faire executer.

4.3 Mise en place du site à déployer.

Pour finir, il va falloir créer le dossier pour stocker le site. Nous allons donc appeler la fonction createSiteWithJson().

```
createSiteWithJson() {  
    sudo mkdir /home/"$1"  
    sudo chmod 755 /home/"$1"  
    sudo cp -r $2/* /home/"$1"/  
}
```

```

sudo chmod 755 /home/"$1"/index.html
sudo chown -R "$1" /home/"$1"
}

```

Le deuxième paramètre de cette fonction permettra d'aller chercher le site directory qui a été notifié dans le JSON. il va copier le contenu de ce dossier et le mettre dans le répertoire racine ou est stocké le site recherché par le vhost.

5 Mise en place du document JSON

Pour pouvoir utilisé l'application à votre guise , il vous faudra remplir le document JSON avec vos données. ce dernier comprend une liste d'utilisateur. L'utilisateur aura son nom, le nom de la base de données, le mot de passe de la DB, le nombre de blocs alloué ainsi que le chemin où se trouve le site à intégré dans le vhost. Le document JSON doit se trouver dans le répertoire source du projet. Exemple de document JSON :

```

{
  "ListUser": [{
    "user": [{
      "name": "Meihdi",
      "DataBaseName": "MeihdiDB",
      "DataBasePassword": "meihdi",
      "bloc": "2000",
      "SiteDirectory": "/home/lemed/Documents/projet/Site-SYSG5"
    } ]
  } ], {
    "user": [{
      "name": "Cedric",
      "DataBaseName": "CedricDB",
      "DataBasePassword": "cedric",
      "bloc": "2000",
      "SiteDirectory": "/home/lemed/Documents/projet/Site-SYSG5"
    } ]
  } ]
}

```

6 lancement de l'application avec un argument

Si on exécute le script avec un argument, l'application se chargera de créer des utilisateurs par défaut ainsi qu'une base de données complète par défaut (voir Mise en place de la base de données). exemple

```
sudo sh install.sh 3
```

Le programme ira donc dans la fonction createaccount()

```
local res="^[0-9]{1,2}$"

if ! [[ $number_account =~ $res ]]; then
    echo "Not a integer"
else
    for i in $(seq 1 $number_account); do
        sudo useradd "account"$i
        sudo mkdir /home/"account"$i
        sudo chmod 755 /home/"account"$i
        sudo touch /home/"account"$i/index.html
        sudo chmod 755 /home/"account"$i/index.html
        sudo echo $htmlpage >/home/"account"$i/index.html
        sudo chown -R "account"$i /home/"account"$i
        create_vhost "account"$i
        create_db "$i"
    done
fi
```

Cette fonction se chargera de créer l'utilisateur ainsi que ses dossier ou sera stocké son site. /home/"accounti"/index.html qui aura comme permission 755. Et comme pour le document json il ira créer les vhost et la db. Mais cette fois ci par défaut le nom de la db sera accounti+DB et le mot de passe sera admin.

7 Accéder aux sites de l'hébergement Mutualisé

Il va falloir voir si tout est en place, ouvrez donc firefox et mettez dans la barre de recherche "account.localhost" remplacer le account par le nom donnée de l'utilisateur si vous avez lancé le programme sans argument, il correspondra sûrement à account1.localhost sinon ce sera le nom énoncé dans le JSON meihdi.localhost. Le site affichera alors le contenu du dossier et chargera le index.html. Maintenant, on va tester la base de données, toujours sur firefox, recherchez localhost et cliquez sur phpmyadmin. et vous pouvez rentrer votre nom d'utilisateur ainsi que le mot de passe. Nous avons eu du mal à implémenter les quotas ainsi que le ProFTPD. Pour le ProFTPD après plusieurs recherches nous nous sommes aperçu que notre machine virtuel n'avait pas d'IP et donc impossible d'utiliser ProFTPD.