Transfert de racktables sur un nouveau serveur

Table of Contents

Racktables est un outil conçu pour gérer les actifs des plate-formes. **RackTables** permet de gérer les ressources matérielles des documents, les adresses réseau, l'espace dans les racks, la configuration des réseaux et bien plus encore. **RackTables** nécessite :

- Un serveur Web – Apache ou Nginx
- PHP-FPM pour le front-end
- Serveur de base de données MySQL/MariaDB pour les données back-end.

Afin de transférer l'application sur un nouveau serveur il faut donc récupérer les éléments constitutifs.

Récupération de la Base de données

Sur le serveur d'origine, Repérer le dossier hébergeant les bases de données

```
sudo grep -R --color datadir /etc/mysql

/etc/mysql/mysql.conf.d/mysqld.cnf:datadir      = /var/lib/mysql
```

Dans l'invite de commande **mysql**, récupérer le nom d'utilisateur le l'administrateur de base

```
SELECT user FROM mysql.user;
+------------------+
| user             |
+------------------+
| debian-sys-maint |
| mysql.session    |
| mysql.sys        |
| racktables_user  |
| root             |
+------------------+
5 rows in set (0,04 sec)
```

Faire un backup de la base de données:

```
sudo mysqldump racktables_db > backup-file.sql
```

Transférer la base de données vers le nouveau serveur:

```
sudo rsync -ave ssh backup-file.sql user@xx.xx.xxx.xx:/
```

Sur le nouveau serveur, installer le serveur mysql:

```
dnf install mysql-server
```

Démarrer les service:

```
systemctl start mysqld
systemctl enable mysqld
```

Créer et restaurer la base de données:

```
mysql -e "CREATE DATABASE IF NOT EXISTS racktables_db
mysql racktables_db < /backup-file.sql
```

Dans l'invite de commande **mysql** créer le compte admin de la base de données:

```
CREATE USER 'racktables_user' IDENTIFIED BY 'password';
GRANT ALL privileges ON `racktables_db`.* TO 'racktables_user'@'%';
```

Modifier le mot de passe de l'administrateur de base de données:

```
SELECT * FROM UserAccount;
+---------+-----------+------------------------------------------+----------
----------------+
| user_id | user_name | user_password_hash                       |
user_realname           |
+---------+-----------+------------------------------------------+----------
----------------+
|       1 | admin     | 5baa61e4c9b93f3f0682250b6cf8331b7ee68fd8 |
RackTables Administrator |
+---------+-----------+------------------------------------------+----------
----------------+

UPDATE UserAccount SET user_name = 'admin', user_password_hash =
SHA1('password') where user_id = 1;
```

Récupération de racktables

Installer les prérequis de l'application

```
dnf install php-fpm php-snmp php-pcntl php-pdo php-gd php-mbstring php-json
php-bcmath php-pdo_mysql -y
```

Démarrer et activer le service **php-fpm**:

```
systemctl start php-fpm
systemctl enable php-fpm
```

Depuis l'ancien serveur transférer le dossier `racktables`:

```
sudo rsync -ave ssh /home/jacques.nougat/www/html/racktables
user@xx.xx.xxx.xx:/var/www/
```

Il faut s'assurer que le fichier `secret.php` est bien dans le dossier ou il est attendu:

```
sudo rsync -ave ssh www/html/racktables/inc/secret.php
user@:/var/www/racktables/wwwroot/inc/secret.php
chmod 777 /var/www/racktables/wwwroot/inc/secret.php
```

Préparer la configuration de **NGINX**:

```
server {
        listen 88 default_server;
        server_name _;

        root /var/www/racktables/wwwroot;

        location / {
            index index.php;
        }

        location ~ \.php$ {
            try_files $uri =404;
            fastcgi_pass unix:/var/run/php-fpm/www.sock;
            fastcgi_index index.php;
            fastcgi_param SCRIPT_FILENAME
$document_root$fastcgi_script_name;
            include /etc/nginx/fastcgi_params;
        }
```

```
    }
```

Pour mettre à l'écoute **NGINX** sur le socket **PHP-FPM**, il faut repérer le socket unix utilisé:

```
grep '^listen' /etc/php-fpm.d/www.conf
listen = /run/php-fpm/www.sock
listen.acl_users = apache,nginx
listen.allowed_clients = 127.0.0.1
```

Puis modifier la ligne correspondante: `fastcgi_pass unix:/var/run/php-fpm/www.sock;`

Pour que SELinux accorde à **NGINX** l'accès au socket TCP sur lequel **PHP-FPM** écoute, il faut utiliser **audit2allow** pour créer un module de stratégie personnalisé à partir des sorties **php-fpm** et **nginx** de `SELinux audit.log`:

```
grep php-fpm /var/log/audit/audit.log | audit2allow -M phpfpmlocal
semodule -i phpfpmlocal.pp
grep nginx /var/log/audit/audit.log | audit2allow -M nginx
semodule -i nginx.pp
```

Ouvrir le port dans le pare-feux:

```
firewall-cmd --permanent --zone=public --add-port=88/tcp
```

Redémarrer le service **nginx**:

```
systemctl nginx restart
```