

Création d'une nouvelle instance(famille) sur PIAFS

****** Voir la section "Connexion à un conteneur Docker avec Rancher" pour savoir comment vous connecter

Vous avez deux options possibles lorsque vous voulez créer des familles. La première option permet de créer qu'une famille à la fois et la deuxième permet de créer plusieurs familles en même temps.

1.1. Exécuter le script de création de famille (createFamily.sh) en vous rendant dans le fichier où il se trouve (par défaut : /var/www/html/) et inscrire la commande `"./createFamily.sh"` pour démarrer le script. Assurez-vous de lui avoir donné les droits requis(`chmod 777 -R createFamily.sh`) pour que le script ait la possibilité d'écrire dans les fichiers de configuration.

1.2. Exécuter le script de création de plusieurs familles (createFamilies.sh). Voir 1.1 pour les étapes suivantes.

2. Rendez-vous sur votre instance, dirigez vous dans la section plugin et ajouter dans le plugin "LocalFiles Editor" les lignes suivantes :

```
if (preg_match('#^/NOM_DE_LA_FAMILLE#', $_SERVER['SCRIPT_NAME']))
#if (preg_match('/^NOM_DE_LA_FAMILLE.yourdomain.com$', $_SERVER['SERVER_NAME']))
{
    $site_dir = 'NOM_DE_LA_FAMILLE';
}
```

S'il y a déjà un "if" dans le plugin, rajouter un "elseif"

3. Rendez vous sur la nouvelle famille (par exemple IP:PORT/NOM_DE_LA_FAMILLE). Vous devriez voir apparaître une page de configuration de la nouvelle famille. Il ne vous reste qu'à inscrire les variables d'environnements :

Host : (Nom de la BD) mariadb

User : piwigo-db-user

Password : piwigo-db-pass

DB Name : piwigo-db-name

DB tables prefix : NOM_DE_LA_FAMILLE_ (Ne pas oublier de mettre "_" après le nom de la famille)

Username : Nom d'utilisateur de l'admin

Password : Mot de passe de l'admin

Email : Adresse courriel de l'admin

Supprimer une famille sur PIAFS

Vous avez deux options possibles lorsque vous voulez supprimer des familles. La première option permet de supprimer qu'une famille à la fois et la deuxième permet de supprimer plusieurs familles en même temps.

1.1. Exécuter le script de suppression de famille (deleteFamily.sh) en vous rendant dans le fichier où il se trouve (par défaut : /var/www/html/) et inscrire la commande `"./deleteFamily.sh"` pour démarrer le script. Assurez-vous de lui avoir donné les droits requis(`chmod 777 -R deleteFamily.sh`) pour que le script ait la possibilité d'écrire dans les fichiers de configuration.

1.2. Exécuter le script de suppression de plusieurs familles (deleteFamilies.sh). Voir 1.1 pour les étapes suivantes.

Installation sur Rancher

Il faut tout d'abord créer une image sur Docker Hub et ensuite Rancher va nous créer une instance avec l'image qu'on lui donne.

Pour pouvoir créer une image sur Docker Hub, il vous faut créer un répertoire de code avec deux fichiers, un fichier Dockerfile et un fichier apache.sh.

Voici celui créé par notre équipe pour PIAFS : <https://github.com/vkoaala/PIAFS-1.0>

Dans le fichier Dockerfile, nous retrouvons la configuration qui permet d'installer les fichiers nécessaires ainsi que de télécharger le code Github (ligne #19), ensuite le code est placé dans le bon emplacement pour que apache puisse le faire afficher.

Le fichier apache.sh reste toujours pareil, il configure apache.

Lorsque le code est créé et accessible sur Github, nous allons nous rendre sur le site de Docker Hub (<https://hub.docker.com/>) pour construire l'image nécessaire à l'installation sur Rancher. Ensuite nous allons créer un "build" automatisé auquel nous allons lui dire de prendre notre code sur Github pour qu'il crée l'image.

Ensuite, sur Rancher il faut créer une nouvelle instance. Il faut bien configurer, la base de données, les variables d'environnements... Pour être sûr qu'il n'y ait pas de problème.

Voici un exemple du docker-compose créé pour PIAFS :

PIAFS-1-0:

ports:

- 20420:80/tcp

labels:

io.rancher.sidekicks: PIAFS-1-0-data

io.rancher.container.pull_image: always

tty: true

image: vkoaala/piafs-1.0

links:

- DB:mariadb

stdin_open: true

PIAFS-1-0-data:

labels:

io.rancher.container.pull_image: always

io.rancher.container.start_once: 'true'

command:

- /bin/true

tty: true

image: vkoaala/piafs-1.0

links:

- DB:mariadb

stdin_open: true

DB:

environment:

MYSQL_DATABASE: piwigo-db-name

MYSQL_PASSWORD: piwigo-db-pass

MYSQL_ROOT_PASSWORD: piwigo-root-pass

MYSQL_USER: piwigo-db-user

domainname: piwigo-mariadb

labels:

io.rancher.sidekicks: DB-data

io.rancher.container.pull_image: always

tty: true

image: mariadb

stdin_open: true

DB-data:

labels:

io.rancher.container.pull_image: always

io.rancher.container.start_once: 'true'

command:

- /bin/true

```
tty: true
image: mariadb
volumes:
- /home/linuq/PIAFS/mariadb:/var/lib/mariadb
volumes_from:
- DB
stdin_open: true
```

Lorsque vous créez l'instance, vous pouvez copier cette configuration et changer l'image ou bien le port par exemple et vous êtes capable d'avoir une configuration fonctionnelle et efficace.

Ensuite, Rancher créera votre instance et vous allez tomber sur la page d'installation.

Il ne vous reste qu'à inscrire les variables d'environnements :

```
Host : mariadb (nom de la base de donnée)
User : piwigo-db-user
Password : piwigo-db-pass
DB Name : piwigo-db-name
DB tables prefix : piwigo_
```

```
Username : Nom d'utilisateur de l'admin
Password : Mot de passe de l'admin
Email : Adresse courriel de l'admin
```

Connection à un conteneur Docker avec Rancher

Pour pouvoir aller directement se connecter au conteneur Docker à l'aide de Rancher il vous faut vous connecter en SSH à la machine hôte sur laquelle il y a le conteneur que vous désirez vous connecter dessus.

Ouvrez un terminal et exécuter la commande : `ssh Nom_Utilisateur_Rancher@IP_de_la_machine`
**Assurez-vous que votre administrateur Rancher vous ait autorisé à vous connecter.

Ensuite, vous aller chercher l'identifiant unique du conteneur que vous désirez rejoindre avec la commande : `docker ps -a`

Dans la liste qui vous affichera, trouvez l'identifiant désiré.

La commande suivante vous permettra d'aller vous connecter directement :
`docker exec -itu root ID_conteneur bash`