



Nous traiterons ici de la mise en place d'un serveur Cloud collaboratif sécurisé au moyen de [PydioCells](#) et **Docker**. Pour la stack nous avons besoin d'une image de **PydioCells** et d'une image du serveur de base de données **MySQL**. Sur ce TP nous avons déjà configuré APACHE Web Server, ainsi que les certificats SSL pour un nom d'hôte de type : *service.mydom.local*

Pré requis

- Un nom d'hôte sécurisé et accessible : <https://dev.local>
- Un sous domaine du nom d'hôte installé : <https://pydio.dev.local>

Étape 1 : Docker Compose

Nous allons nous servir d'un outil bien pratique pour lancer plusieurs conteneurs empilés et connectés (**une stack**) en évitant les lignes de commandes à rallonge avec l'utilitaire : **docker-compose**.

```
docker-compose -h
```

Lancer la commande :

L'aide doit s'afficher, sinon installer « docker-compose ».

- Création du répertoire de travail et s'y déplacer :

```
cd && mkdir pydio && cd pydio
```

- Nous allons ensuite créer un fichier au format [YAML](#), nommé :
« **docker-compose.yml** »

```
nano docker-compose.yml
```

- dans lequel nous allons définir notre stack PYDIO :

```
version: '3.7'
services:
  cells:
    image: pydio/cells:latest
    restart: unless-stopped
    ports: ["7999:8080"]
    container_name: pydioCells
    environment:
      - CELLS_LOG_LEVEL=production
    volumes:
      - data:/var/pydio/app/data
      - cellsdire:/var/pydio/app
  mysql:
    image: mysql:5.7
    restart: unless-stopped
    container_name: pydioDB
    environment:
      MYSQL_ROOT_PASSWORD: PaSsSW0rd
      MYSQL_DATABASE: cells
      MYSQL_USER: pydio
      MYSQL_PASSWORD: PaSsSW0rd
    command: [mysqld, --character-set-server=utf8mb4, --collation-server=utf8mb4_unicode_ci]
    volumes:
      - mysqldire:/var/pydio/app/mysql

volumes:
  data: {}
  cellsdire: {}
  mysqldire: {}
```

ATTENTION A L'INDENTATION : les tabulations retourneront des erreurs sur les fichiers YAML ou « *yamèl* » (.yml). Le format de fichier doit être LF (Unix) / UTF8

Unix (LF)

UTF-8

- Une fois ce fichier mis en place on peut lancer tous nos conteneurs avec la commande :

```
docker-compose up -d
```

(*up = create, update and start, -d= detach*)

Il faut attendre quelques instants afin que la stack de LAMP soit téléchargée et déployée (*pull*).

Vérification des conteneurs actifs :

```
docker ps -a
```

Notre stack Pydio est en route :

```
root@ubuntu:~/pydio# docker ps -a
CONTAINER ID   IMAGE                COMMAND                  PORTS                NAMES
9e9b1aec3a3e   mysql:5.7            "docker-entrypoint.s..."  3306/tcp, 33060/tcp  pydioDB
7ae4cd9a3039   pydio/cells:latest   "docker-entrypoint.s..."  0.0.0.0:7999->8080/tcp  pydioCells
```

Pour tester le bon fonctionnement du serveur, on test l'accès avec CURL :

```
curl http://localhost:7999
```

➔ **Client sent an HTTP request to an HTTPS server.**

Ce message indique que notre serveur est bien installé, par contre, il faut configurer Apache afin d'installer nos certificats SSL.

Étape 2 : Configuration d'Apache

Ajouter une nouvelle configuration d'apache pour la stack de Pydio :

```
nano /etc/apache2/sites-available/pydio.conf
```

La configuration suivante se base sur le certificat précédemment installé, et va faire le lien entre l'hôte local sur le port **7999** (interne) et sur le port **443** (https externe). Cette méthode s'appelle :

[reverse proxy](#)

```
<VirtualHost *:443>

ServerName pydio.dev.local

# SSL configuration
SSLEngine on
# Container uses a unique non-signed certificate
SSLProxyEngine On
SSLProxyVerify None
SSLProxyCheckPeerCN Off
SSLProxyCheckPeerName Off

SSLCertificateFile /etc/apache2/ssl2/domains.crt
SSLCertificateKeyFile /etc/apache2/ssl2/private.key
SSLCertificateChainFile /etc/apache2/ssl2/domains.crt

AllowEncodedSlashes On
RewriteEngine On

## This rewrite condition is required if using Cells-Sync
RewriteCond %{HTTP:Content-Type} =application/grpc [NC]
RewriteRule /(.*) h2://localhost:7999/$1 [P,L]

ProxyPassMatch "/ws/(.*)" wss://localhost:7999/ws/$1 nocanon

ProxyPass "/" "https://localhost:7999/"
ProxyPassReverse "/" "https://localhost:7999/"

ErrorLog ${APACHE_LOG_DIR}/pydio_error.log
CustomLog ${APACHE_LOG_DIR}/pydio_access.log combined

</VirtualHost>
```

Quelques dépendances supplémentaires afin de terminer l'installation :

```
sudo a2enmod proxy
sudo a2enmod proxy_http
sudo a2enmod proxy_wstunnel
sudo a2ensite pydio.conf
sudo service apache2 restart
```

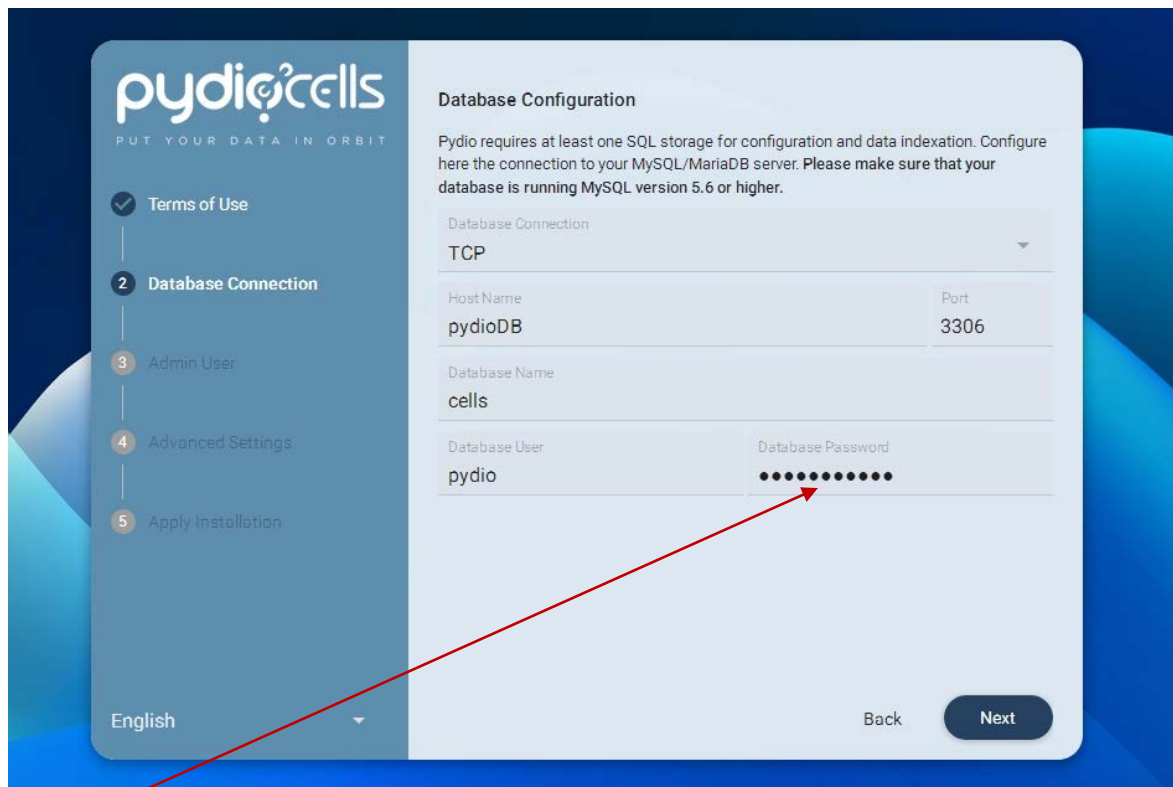
- On test sur un navigateur : <https://pydio.dev.local>

ATTENTION : SUR WINDOWS, vérifiez que les noms d'hôtes soient bien redirigés vers l'IP du serveur dans : **C:\WINDOWS\system32\drivers\etc\hosts** SUR LINUX : **/etc/hosts**

- Obtenir l'IP V4 du serveur, commande : `ip a`
- Ne pas oublier d'ajouter le nouveau nom d'hôte : **ip_v4_du_serveur pydio.dev.local**

Étape 3 : Configuration de Pydio Cells

L'étape suivante consiste à connecter l'image de MySQL gérée par Docker à celle de Pydio :



Le mot de passe MySQL étant celui du fichier de configuration YAML (en page3) pour *docker-compose*.

Découvrez l'interface de **Pydio Cells**, et commencez à le configurer pour votre infrastructure !

