Compte-rendu : Mise en place d'un serveur Apache HTTPS avec Docker

Étudiant : Péniel LAWSON-BODY

Environnement : GitHub Codespaces (Linux / Docker)

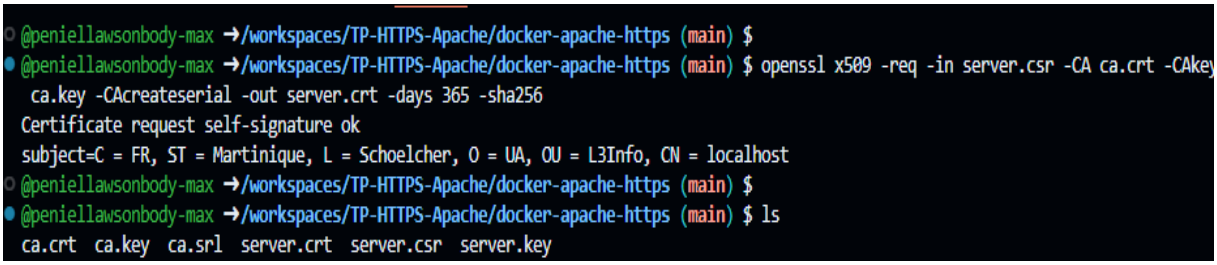
1. Introduction

L'objectif de cet atelier est de déployer un serveur web Apache sécurisé via le protocole HTTPS. Pour surmonter les limitations techniques de l'hôte local (Docker non reconnu sur Windows), le projet a été réalisé sur **GitHub Codespaces**, offrant un environnement conteneurisé natif.

2. Création de l'Autorité de Certification (CA)

Pour signer nos propres certificats, nous avons créé une CA factice.

- **Commande de génération de la clé privée :** `openssl genrsa -out ca.key 2048`.
- **Commande de génération du certificat racine :** `openssl req -x509 -new -nodes -key ca.key -sha256 -days 3650 -out ca.crt -subj "/C=FR/ST=Martinique/L=Schoelcher/O=UA/OU=L3Info/CN=MonAutorite"`.



3. Création et signature du certificat serveur

Le certificat du serveur est configuré pour répondre au nom d'hôte `localhost`.

- **Génération du CSR :** `openssl req -new -key server.key -out server.csr -subj "/C=FR/ST=Martinique/L=Schoelcher/O=UA/OU=L3Info/CN=localhost"`.
- **Signature par la CA :** `openssl x509 -req -in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out server.crt -days 365 -sha256`.

4. Configuration Docker et Apache

Le fichier `Dockerfile` automatise l'activation du module SSL et l'inclusion de la configuration personnalisée.

- **Fichier `httpd-ssl.conf` :** Nous avons activé le moteur avec `SSLEngine on` et spécifié les chemins `SSLCertificateFile` et `SSLCertificateKeyFile`.

```
@peniellawsonbody-max →/workspaces/TP-HTTPS-Apache/docker-apache-https (main) $ # Création du Dockerfile
cat <<EOF > Dockerfile
FROM httpd:2.4.63

COPY ./server.crt /usr/local/apache2/conf/server.crt
COPY ./server.key /usr/local/apache2/conf/server.key
COPY ./httpd-ssl.conf /usr/local/apache2/conf/extra/httpd-ssl.conf

RUN sed -i '/ssl_module/s/^#//g' conf/httpd.conf
RUN echo "Include conf/extra/httpd-ssl.conf" >> /usr/local/apache2/conf/httpd.conf
EOF
```

5. Déploiement et Test

- **Construction de l'image :** `docker build -t my-apache-https .`
- **Lancement du conteneur :** `docker run -dit --name apache-https -p 443:443 my-apache-https`.

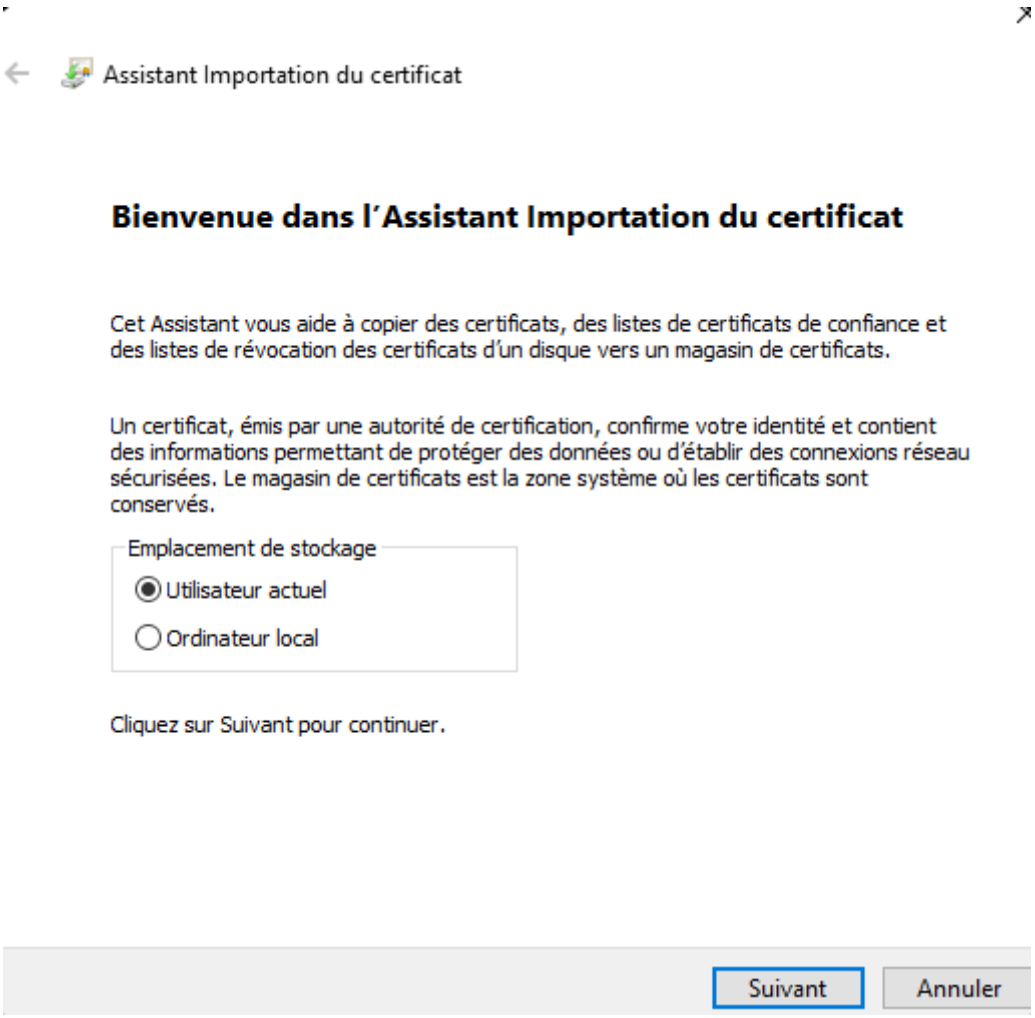
```
# Création du fichier de configuration SSL (Complété)
EOFirtualHost>y>all grantedapache2/htdocs">e2/conf/server.key"
@peniellawsonbody-max →/workspaces/TP-HTTPS-Apache/docker-apache-https (main) $ # Construire l'image
docker build -t my-apache-https .
```

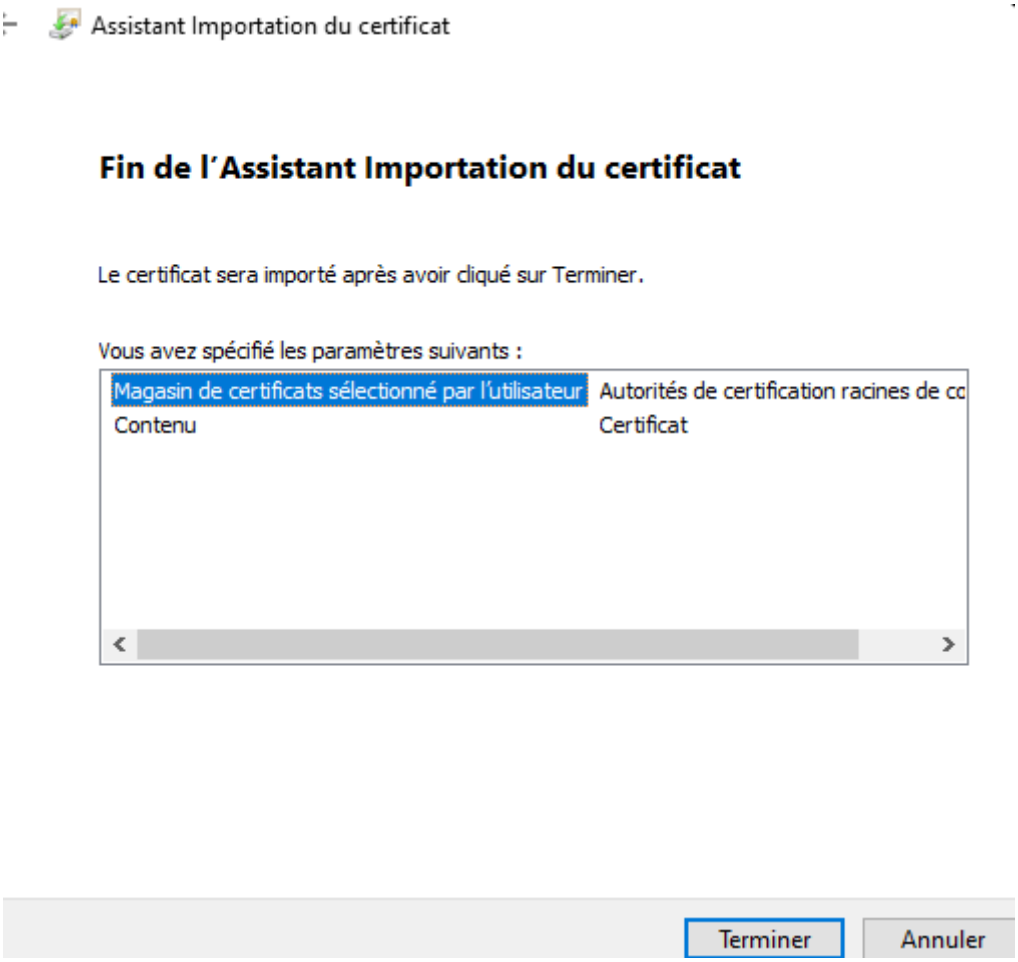
```
# Exécuter le conteneur
docker run -dit --name apache-https -p 443:443 my-apache-https
[+] Building 7.7s (12/12) FINISHED                                    docker:default
 => [internal] load build definition from Dockerfile                          0.0s
 => => transferring dockerfile: 364B                                          0.0s
 => [internal] load metadata for docker.io/library/httpd:2.4.63               0.5s
 => [auth] library/httpd:pull token for registry-1.docker.io                  0.0s
 => [internal] load .dockerignore                                             0.0s
 => => transferring context: 2B                                               0.0s
⊗0⚠0  1   peniellawsonbody-max (maintenant)  Li 17, Col 1 (65 sélectionné)  Espaces : 4  UTF-8  LF  { }
```

```
 => [internal] load build context                                             0.0s
 => => transferring context: 3.44kB                                           0.0s
 => [2/6] COPY ./server.crt /usr/local/apache2/conf/server.crt                0.0s
 => [3/6] COPY ./server.key /usr/local/apache2/conf/server.key                0.0s
 => [4/6] COPY ./httpd-ssl.conf /usr/local/apache2/conf/extra/httpd-ssl.conf  0.0s
 => [5/6] RUN sed -i '/ssl_module/s/^#//g' conf/httpd.conf                    0.3s
 => [6/6] RUN echo "Include conf/extra/httpd-ssl.conf" >> /usr/local/apache2/conf/httpd.conf  0.2s
 => exporting to image                                                        1.0s
⊗0⚠0  1   peniellawsonbody-max (maintenant)  Li 17, Col 1 (65 sélectionné)  Espaces : 4  UTF-8  LF  { }
```

Validation du certificat

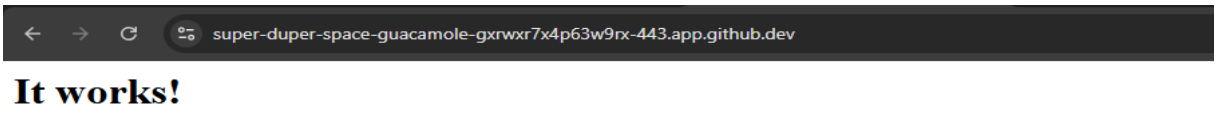
Pour que le navigateur valide la connexion, le fichier `ca.crt` a été importé dans le magasin **Autorités de certification racines de confiance** de Windows.


Assistant Importation du certificat
Bienvenue dans l'Assistant Importation du certificat
Cet Assistant vous aide à copier des certificats, des listes de certificats de confiance et des listes de révocation des certificats d'un disque vers un magasin de certificats.
Un certificat, émis par une autorité de certification, confirme votre identité et contient des informations permettant de protéger des données ou d'établir des connexions réseau sécurisées. Le magasin de certificats est la zone système où les certificats sont conservés.
Emplacement de stockage
Utilisateur actuel
Ordinateur local
Cliquez sur Suivant pour continuer.
Suivant
Annuler



6. Résultat final

Le serveur est accessible et affiche la page par défaut.



7. Réponses aux questions de l'étape 8

- **Signes de sécurité :** Le navigateur affiche "It works!". Le cadenas est validé car l'autorité racine a été ajoutée manuellement au système.

8. Sauvegarde sur GitHub

Le projet est archivé sur GitHub pour garantir la persistance des fichiers de configuration.