Configuration d'un Client Docker et des Certificats

Lorsque vous configurez un **Docker Registry privé sécurisé avec HTTPS**, il est crucial que le client Docker puisse se connecter au registry de manière sécurisée en vérifiant le certificat SSL du serveur. Voici les étapes pour configurer le client Docker afin qu'il utilise des certificats SSL pour se connecter à un **Docker Registry privé sécurisé**.

1. Ajouter des Certificats SSL

Pour garantir des connexions sécurisées, Docker nécessite que le client fasse confiance au certificat SSL du registry privé. Cela est particulièrement important si vous utilisez un certificat auto-signé, mais même pour des certificats valides d'une autorité de certification (CA), il peut être nécessaire de configurer le client Docker pour qu'il reconnaisse le certificat.

Étapes pour ajouter un certificat SSL au client Docker :

1. **Obtenez le certificat SSL du Docker Registry** :
   - Si vous avez généré un certificat SSL auto-signé (ou utilisé un certificat Let's Encrypt pour un environnement de production), vous devrez l'ajouter à votre client Docker.
   - Si vous avez déjà configuré votre Docker Registry pour utiliser HTTPS avec un certificat (comme montré dans les étapes précédentes), vous devez vous assurer que le client Docker fasse confiance à ce certificat.

2. **Stocker le certificat sur le client Docker** :
   Vous devez stocker le certificat SSL (soit `domain.crt` pour un certificat auto-signé, soit `ca.crt` pour un certificat d'une autorité de certification) dans un répertoire spécifique pour que Docker puisse le reconnaître.

   Emplacement des certificats Docker :
   Sur la machine du client Docker, le certificat SSL du registry privé doit être placé dans le répertoire suivant :

   ```
   /etc/docker/certs.d/<registry-url>:<port>/
   ```

   Exemple :
   Si votre Docker Registry sécurisé utilise `localhost:5000`, copiez le certificat dans le répertoire approprié :

   ```
   sudo mkdir -p /etc/docker/certs.d/localhost:5000
   sudo cp /etc/docker/certs/domain.crt
   /etc/docker/certs.d/localhost:5000/ca.crt
   ```

   - `domain.crt` : Le certificat SSL du serveur.

- `ca.crt` : Le certificat de l'autorité de certification si nécessaire.

Cette configuration permet à Docker de reconnaître et d'accepter les connexions sécurisées au Docker Registry privé.

2. Configurer Docker pour se connecter à un Registry Privé Sécurisé

Une fois le certificat ajouté, vous devez configurer Docker pour qu'il se connecte à votre registry privé sécurisé via HTTPS. Voici les étapes à suivre :

Étape 1 : Activer Docker pour accepter les connexions sécurisées

Docker utilise des certificats SSL pour établir des connexions sécurisées avec un registry privé. Si votre registry utilise HTTPS, Docker doit être configuré pour accepter les connexions sécurisées en vérifiant le certificat SSL. Une fois le certificat installé sur le client, Docker l'utilisera pour valider les connexions HTTPS.

Étape 2 : Configurer le client Docker pour se connecter au registry privé

Lorsque vous utilisez un Docker Registry privé avec HTTPS, vous devez vous connecter à ce registry en utilisant la commande `docker login`, en fournissant le nom du registry sécurisé.

1. **Login au Docker Registry privé sécurisé** :
   Vous pouvez maintenant utiliser `docker login` pour vous connecter à votre Docker Registry privé sécurisé. Docker vous demandera un nom d'utilisateur et un mot de passe si une authentification est configurée (comme expliqué précédemment avec `htpasswd`).

   Exemple de commande :

   ```
   docker login localhost:5000
   ```

   Docker vous demandera d'entrer votre nom d'utilisateur et votre mot de passe définis dans le fichier `htpasswd`. Si l'authentification est réussie, Docker stockera un jeton d'authentification dans `~/.docker/config.json`.

2. **Pousser une image vers le Docker Registry sécurisé** :
   Après vous être connecté, vous pouvez pousser des images vers votre registry sécurisé.

   Exemple de commande pour pousser une image :

   ```
   docker tag my-image localhost:5000/my-image
   docker push localhost:5000/my-image
   ```

3. **Tirer une image depuis le Docker Registry sécurisé** :
   Vous pouvez également tirer des images depuis votre registry privé sécurisé, comme suit :

```
docker pull localhost:5000/my-image
```

Étape 3 : Configuration des utilisateurs et de l'authentification

Si votre registry privé utilise une authentification basique, vous devez d'abord vous authentifier via la commande `docker login`, en utilisant les identifiants d'un utilisateur dans le fichier `htpasswd`. Si vous utilisez un Docker Registry avec authentification via certificat TLS, le processus sera légèrement différent, mais le client Docker doit toujours être configuré avec le certificat approprié pour établir la connexion sécurisée.

Étape 4 : Configurer Docker pour un accès au registry privé en environnement de production

1. **Configurer Docker pour accéder à un registry privé sur plusieurs hôtes** :
   Si vous devez configurer Docker pour accéder à un registry privé depuis plusieurs machines dans votre réseau, assurez-vous que le certificat est installé sur chaque machine qui doit interagir avec le registry. Les étapes pour installer le certificat et configurer Docker seront similaires sur chaque machine cliente.

2. **Configurer Docker pour accéder à un registry privé dans des environnements Docker Swarm ou Kubernetes** :
   Si vous utilisez Docker Swarm ou Kubernetes et que vous devez configurer l'accès à un registry privé sécurisé, vous devrez vous assurer que les nœuds du cluster ont également le certificat SSL installé, et vous devrez vous assurer que les secrets ou volumes utilisés par Swarm ou Kubernetes pour stocker les informations d'authentification sont bien sécurisés.

Résumé

1. **Ajouter des certificats SSL au client Docker** : Pour garantir une connexion sécurisée au Docker Registry privé, vous devez ajouter le certificat SSL du registry dans le répertoire approprié sur le client Docker (`/etc/docker/certs.d/<registry-url>:<port>/`).

2. **Configurer Docker pour se connecter au registry privé sécurisé** : Après avoir installé le certificat, utilisez la commande `docker login` pour vous connecter au registry privé sécurisé. Vous devrez fournir un nom d'utilisateur et un mot de passe si l'authentification est activée.

3. **Pousser et tirer des images** : Une fois la connexion sécurisée établie, vous pouvez utiliser les commandes `docker push` et `docker pull` pour gérer les images Docker dans votre registry privé sécurisé.

Cette configuration assure que les communications entre le client Docker et le Docker Registry privé sont sécurisées, chiffrées et authentifiées, protégeant ainsi vos images Docker sensibles.