Activer l'API Docker sur une instance

Ici c'est l'exemple de l'instance DB-SRV sur le SINP PACA (silene.eu) qui est donné.

Rendre persistante l'activation

- Afin d'éviter que les modifications effectuées dans le fichier */lib/systemd/system/docker.service* soient écrasées à chaque mise à jour de Docker, vous devez ajouter un fichier qui écrasera les valeurs par défaut.
  - **Source** : Using systemd to control the Docker daemon
- Pour créer automatiquement l'arborescence de dossier et le fichier nécessaire, utiliser la commande suivante : `systemctl edit docker`
  - La commande précédente ouvre l'éditeur par défaut du système, vous pouvez ajouter le contenu suivant et sortir de l'édition du fichier en sauvegardant :

```
[Service]
ExecStart=
ExecStart=/usr/bin/dockerd -H fd:// --
containerd=/run/containerd/containerd.sock -H tcp://10.0.1.20:2376
```

    - **Note** : la première ligne *ExecStart=* vide permet de réinitialiser la commande de lancement de Docker
  - Les modifications devraient être présente dans le fichier suivant : `vi /etc/systemd/system/docker.service.d/override.conf`
- Lancer la prise en compte des modifications qui vérifiera une éventuelle erreur : `systemctl daemon-reload`
- Relancer le service Docker : `systemctl restart docker`
- Vérifier la présence des nouveaux paramètres dans *CGroup* : `systemctl status docker`

Tester temporairement l'activation

- Au préalable, sur le serveur *db-srv*, activer l'API *Docker* sur l'IP de l'hôte du VPN : `vi /lib/systemd/system/docker.service`
  - Modifier la ligne `ExectStart=` en ajoutant l'option `-H tcp://10.0.1.20:2376` juste aprés `-H fd://`
  - Prendre en compte les changements : `systemctl daemon-reload`
  - Redémarrer Docker : `systemctl restart docker`
- Puis accéder à https://manager.silene.eu pour configurer cet instance (voir la doc dédiée).

Utiliser TLS (HTTPS) pour sécuriser l'API (daemon Docker)

- Créer un dossier qui contiendra les certificats : `mkdir -pv /etc/docker/ssl/`
  - Sécuriser le dossier : `chmod 600 /etc/docker/ssl/`
- Générer les différents certificats en suivant la documentation de Docker :
  - Se placer dans le dossier qui hébergera les certificats : `cd /etc/docker/ssl/`

    - `openssl genrsa -aes256 -out ca-key.pem 4096`
        - Créer le mot de passe du certificat et le stocker dans Keepass
    - créer un certificat valable 5 ans (1825 jours) : `openssl req -new -x509 -days 1825 -key ca-key.pem -sha256 -out ca.pem`
        - Répondre aux questions comme suit :

```
Country Name (2 letter code) [AU]: FR
State or Province Name (full name) [Some-State]:Hautes-Alpes
Locality Name (eg, city) []: Gap
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
CBNA
Organizational Unit Name (eg, section) []: SI
Common Name (e.g. server FQDN or YOUR name) []: db-
srv.silene.eu
Email Address []: adminsys@silene.eu
```

    - `openssl genrsa -out server-key.pem 4096`
    - `openssl req -subj "/CN=db-srv.silene.eu" -sha256 -new -key server-key.pem -out server.csr`
    - `echo subjectAltName = DNS:db-srv.silene.eu,IP:10.0.1.20,IP:127.0.0.1 > extfile.cnf`
    - `echo extendedKeyUsage = serverAuth >> extfile.cnf`
    - `openssl x509 -req -days 1825 -sha256 -in server.csr -CA ca.pem -CAkey ca-key.pem -CAcreateserial -out server-cert.pem -extfile extfile.cnf`
    - `openssl genrsa -out key.pem 4096`
    - `openssl req -subj '/CN=client' -new -key key.pem -out client.csr`
    - `echo extendedKeyUsage = clientAuth > extfile-client.cnf`
    - `openssl x509 -req -days 1825 -sha256 -in client.csr -CA ca.pem -CAkey ca-key.pem -CAcreateserial -out cert.pem -extfile extfile-client.cnf`
    - `chmod -v 0400 ca-key.pem key.pem server-key.pem`
    - `chmod -v 0444 ca.pem server-cert.pem cert.pem`
- Renommer les fichiers client :
    - `mv key.pem client-key.pem`
    - `mv cert.pem client-cert.pem`
- Modifier le fichier daemon.json : `vi /etc/docker/daemon.json`

```
{
    "tls": true,
    "tlsverify": true,
    "tlscacert": "/etc/docker/ssl/ca.pem",
    "tlscert": "/etc/docker/ssl/server-cert.pem",
    "tlskey": "/etc/docker/ssl/server-key.pem"
}
```

- Redémarrer le service Docker : `systemctl stop docker.service;systemctl start docker.service`
- Tester la sécurisation : `docker -H 10.0.1.20:2376 --tls --tlscert=/etc/docker/ssl/client-cert.pem --tlskey=/etc/docker/ssl/client-`

`key.pem --tlscacert=/etc/docker/ssl/ca.pem ps -a`

- Créer un dossier sur admin pour récupérer en local les fichiers client : `mkdir /home/admin/ca;cp /etc/docker/ssl/{ca.pem,client-*} /home/admin/ca/;chown admin: -R /home/admin/ca`
- Depuis votre poste local : `scp admin@db-paca-sinp:~/ca/* ~/Documents/Keepass/docker-ca/db-srv/`
- Supprimer sur le serveur le dossier *~/ca/* : `rm -fR /home/admin/ca`
- Via l'interface de Portainer créer un nouvel environnement :
  - Name : db-srv
  - Environment URL : 10.0.1.20:2376
  - TLS : activer
  - TLS with server and client verification : activer
  - TLS CA certificate : uploader le fichier *ca.pem*
  - TLS certificate : uploader le fichier *client-cert.pem*
  - TLS key : uploader le fichier *client-key.pem*
- Stocker les certificats sur le serveur hébergeant le Docker de Portainer (bkp-srv) :
  - Créer un dossier qui hébergera les certificats : `mkdir -p /etc/docker/certs.d/db-srv`
  - Modifier les droits : `chmod 750 -R /etc/docker/certs.d/`
  - Créer un dossier sur admin pour récupérer sur le serveur (bkp-srv) les fichiers client : `mkdir /home/admin/ca;chown admin: -R /home/admin/ca`
  - Depuis le poste local, uploader les fichiers clients : `scp ~/Documents/Keepass/docker-ca/db-srv/* admin@bkp-paca-sinp:~/ca/`
  - Déplacer les fichiers clients sur le serveur bkp-srv : `mv /home/admin/ca/* /etc/docker/certs.d/db-srv/`
  - Donner les bons droits aux fichiers clients : `chown root: /etc/docker/certs.d/db-srv/*`
  - Supprimer le dossier *ca/* sur le serveur bkp-srv : `rm -fR /home/admin/ca`