

dockerd remote tls access

via

- <https://docs.docker.com/engine/security/https/>
- <https://nickjanetakis.com/blog/docker-tip-73-connecting-to-a-remote-docker-daemon>
- <https://success.docker.com/article/how-do-i-enable-the-remote-api-for-dockerd>
- <https://medium.com/@ssmak/how-to-enable-docker-remote-api-on-docker-host-7b73bd3278c6>

setup

- pretendemos «asegurar» la conexión con el **dockerd** de manera que solo los clientes con certificado firmado por la misma CA que el servidor puedan conectarse
- nos permite limitar el acceso al **dockerd** local y además es el paso previo a permitir conexiones remotas para gestionarlo

certificados

1. crear CA:

```
openssl genrsa -aes256 -out ca-key.pem 4096
openssl req -new -x509 -days 365 -key ca-key.pem -sha256 -out ca.pem
```

- el **Common Name** ha de apuntar al FQDN de la máquina

2. crear key y CSR:

```
openssl genrsa -out server-key.pem 4096
openssl req -subj "/CN=$HOST" -sha256 -new -key server-key.pem -out server.csr
```

- \$HOST por el FQDN

3. generar direcciones y atributos:

extfile.cnf

```
echo subjectAltName = DNS:<FQDN>DNS:$HOST,IP:10.10.10.20,IP:127.0.0.1 >>
extfile.cnf
echo extendedKeyUsage = serverAuth >> extfile.cnf
```

- por FQDN y por \$HOST (por si acaso)
- **IP:** solo si es necesario más allá de 127.0.0.1

4. generar el certificado firmado:

```
openssl x509 -req -days 365 -sha256 -in server.csr -CA ca.pem -CAkey ca-
key.pem -CAcreateserial -out server-cert.pem -extfile extfile.cnf
```

5. generar KEY cliente + CSR:

```
openssl genrsa -out key.pem 4096
openssl req -subj '/CN=client' -new -key key.pem -out client.csr
```

6. crear atributos del certificado cliente:

extfile-client.cnf

```
echo extendedKeyUsage = clientAuth > extfile-client.cnf
```

7. generación del certificado firmado:

```
openssl x509 -req -days 365 -sha256 -in client.csr -CA ca.pem -CAkey ca-key.pem -CAcreateserial -out cert.pem -extfile extfile-client.cnf
```

8. ahora ya se pueden eliminar los CSR y atributos:

```
rm -v client.csr server.csr extfile.cnf extfile-client.cnf
```

9. y asegurar el acceso a los ficheros importantes:

```
chmod -v 0400 ca-key.pem key.pem server-key.pem  
chmod -v 0444 ca.pem server-cert.pem cert.pem
```

dockerd

1. decirle al dockerd que solo acepte conexiones «seguras»:

```
sudo systemctl stop docker.service  
sudo dockerd --tlsverify --tlscacert=ca.pem --tlscert=server-cert.pem --  
tlskey=server-key.pem -H=0.0.0.0:2376
```

2. ejecución «siempre»:

1. crear el fichero:

[/etc/systemd/system/docker.service.d/override.conf](#)

```
[Service]  
ExecStart=  
ExecStart=/usr/bin/dockerd -H fd:// -H tcp://0.0.0.0:2376
```

2. recargar el demonio de **systemctl** para que coja la nueva configuración:

```
systemctl daemon-reload
```

3. rearrancar el servicio de dockerd:

```
systemctl restart docker.service
```

docker

1. las conexiones de cliente se tendrán que hacer así:

```
docker --tlsverify --tlscacert=ca.pem --tlscert=cert.pem --tlskey=key.pem -  
H=$HOST:2376 version
```

2. para hacerlo de manera segura «por defecto»:

```
mkdir -pv ~/.docker  
cp -v {ca,cert,key}.pem ~/.docker  
export DOCKER_HOST=tcp://$HOST:2376 DOCKER_TLS_VERIFY=1
```

From:

<https://matebcn.eu/> - miguel angel torres egea

Permanent link:

<https://matebcn.eu/tech:docker:dockerd:remote-tls>

Last update: **11/04/2020 09:55**

