By default, the Docker daemon is only accessible from the machine where it is running. However, we can expose the Docker socket securely, enabling us to interact with Docker remotely. In this lesson, we'll cover how to securely set up mutual client/server certificate authentication so that we can connect to a remote Docker daemon.

Relevant Documentation

- https://docs.docker.com/engine/security/https/

Lesson Reference

Follow along with this lesson using two playground servers:

- Image - Ubuntu 18.04 Bionic Beaver LTS
- Size - Micro

Generate a certificate authority and server certificates for your Docker server. Make sure you replace `<server private IP>` with the actual private IP of your server.

Note: If you get a message that says `Can't load /home/cloud_user/.rnd into RNG`, it is safe to ignore that message. The command will still succeed.

```
openssl genrsa -aes256 -out ca-key.pem 4096
openssl req -new -x509 -days 365 -key ca-key.pem -sha256 -out ca.pem -subj "/C=US/ST=Texas/L=Keller/O=Linux A
openssl genrsa -out server-key.pem 4096
openssl req -subj "/CN=$HOSTNAME" -sha256 -new -key server-key.pem -out server.csr
echo subjectAltName = DNS:$HOSTNAME,IP:<server private IP>,IP:127.0.0.1 >> extfile.cnf
echo extendedKeyUsage = serverAuth >> extfile.cnf
openssl x509 -req -days 365 -sha256 -in server.csr -CA ca.pem -CAkey ca-key.pem \
  -CAcreateserial -out server-cert.pem -extfile extfile.cnf
```

Then generate the client certificates:

```
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 365 -sha256 -in client.csr -CA ca.pem -CAkey ca-key.pem \
  -CAcreateserial -out cert.pem -extfile extfile-client.cnf
```

Set appropriate permissions on the certificate files:

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

Configure your Docker host to use `tlsverify` mode with the certificates that were created earlier:

```
sudo vi /etc/docker/daemon.json
```

```
{
  "tlsverify": true,
  "tlscacert": "/home/cloud_user/ca.pem",
  "tlscert": "/home/cloud_user/server-cert.pem",
  "tlskey": "/home/cloud_user/server-key.pem"
}
```

```
sudo vi /lib/systemd/system/docker.service
```

Look for the line that begins with `ExecStart` and change the `-H` so that it looks like this:

```
ExecStart=/usr/bin/dockerd -H=0.0.0.0:2376 --containerd=/run/containerd/containerd.sock
```

```
sudo systemctl daemon-reload
sudo systemctl restart docker
```

Copy the CA cert and client certificate files to the client machine:

```
scp ca.pem cert.pem key.pem cloud_user@<client private IP>:/home/cloud_user
```

On the client machine, configure the client to securely connect to the remote Docker daemon:

```
mkdir -pv ~/.docker
cp -v {ca,cert,key}.pem ~/.docker
export DOCKER_HOST=tcp://<docker server private IP>:2376 DOCKER_TLS_VERIFY=1
```

Test the connection:

```
docker version
```