docker

Connect to Container as root

```
$ docker exec -u 0 -it FW_pgAdmin4 /bin/ash
```

PostgreSQL DEV refresh

```
docker exec -it dev-container_devcontainer_db_1 /bin/bash
```

View filesystem

```
docker image save fueltracker_app >image.tgz
```

SSL/TLS connection

1. Create a working directory

```
mkdir -p ~/.docker/tls && cd ~/.docker/tls
```

2. Generate the CA private key

```
openssl genrsa -aes256 -out ca-key.pem 4096
```

3. Generate the CA certificate

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

You'll be prompted for a passphrase and subject info (Country, CN, etc.).

4. Generate the server private key

```
openssl genrsa -out server-key.pem 4096
```

5. Generate the server CSR (Certificate Signing Request)

```
openssl req -subj "/CN=<your-server-hostname-or-IP>" -sha256 -new -key
server-key.pem -out server.csr
```

Replace <your-server-hostname-or-IP> with your Docker host's hostname or IP (e.g., mydockerhost or 192.168.1.10).

6. Create a SANs (Subject Alternative Names) extension file

```
echo subjectAltName = DNS:<hostname>,IP:<IP>,IP:127.0.0.1 > extfile.cnf
```

```
echo extendedKeyUsage = serverAuth >> extfile.cnf
```

Add all hostnames/IPs clients will use to reach the server.

7. Sign the server certificate with the CA

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

8. Generate the client private key

```
openssl genrsa -out key.pem 4096
```

9. Generate the client CSR

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

10. Create a client extension file

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

11. Sign the client certificate with the CA

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

12. Clean up CSRs and extension files, lock down permissions

```
rm -f client.csr server.csr extfile.cnf extfile-client.cnf
chmod 0400 ca-key.pem key.pem server-key.pem
chmod 0444 ca.pem server-cert.pem cert.pem
```

After this you'll have:

```
┌──────────────────────────────────┬──────────────────────────────────────────────
──────────┐
│               File               │                      Purpose
│
├──────────────────────────────────┼──────────────────────────────────────────────
──────────┤
│ ca.pem                           │ CA certificate (needed by both server
and client) │
├──────────────────────────────────┼──────────────────────────────────────────────
──────────┤
│ server-cert.pem / server-key.pem │ Server certificate and key
│
├──────────────────────────────────┼──────────────────────────────────────────────
──────────┤
```

```
| cert.pem / key.pem                | Client certificate and key
|                                   |
└───────────────────────────────────┴──────────────────────────────────────
──────────┘
```

References

Docker Exec Command With Examples - https://devconnected.com/docker-exec-command-with-examples/

Exploring Docker container's file system - https://stackoverflow.com/questions/20813486/exploring-docker-containers-file-system

[Docker]