GENERATING CERTIFICATES **USING FABRIC-CA**

K B A

- Generating Certificates using docker-compose-ca.yaml file
- Creating registerEnroll.sh file

registerEnroll.sh file

- The process of creating the different identities and configurations for these organisations will be automated with the help of the script file.
- Create a new file inside the Fabric-network folder called registerEnroll.sh file.

Overview of registerEnroll.sh file

- Enroll the CA admin
- Define the NodeOUs
- Register the peer, user, org admin
- Enroll the peer, user, org admin
- Categorize the certificates by creating a separate folder for tls certs, ca certs

Overview of registerEnroll.sh file

- Command for enrolling an identity:
  - **fabric-ca-client enroll -u <serverPath> <options>**
- Command for registering an identity:
  - **fabric-ca-client register <options>**

registerEnroll.sh

- **#!/bin/bash**- Indicates it's a script file
- **function createOrg1()** - Creates the organization1's identities (peer, user, admin)
- **fabric-ca-client enroll**: This is the main command for obtaining the cryptographic materials for the registered identity.
- **-u:** Specifies the URL for the CA to connect
- **--caname ca-org1:** specifies the name of the CA
- **--tls.certfiles** flag specifies the location of the TLS certificate file of ca-org1.
- In summary it is used to enroll the CA admin. It communicates with the Fabric CA server running at localhost:7054.

```
fabric-ca-client enroll -u https://admin:adminpw@localhost:7054 --caname ca-org1
--tls.certfiles "${PWD}/organizations/fabric-ca/org1/ca-cert.pem"
```

Node Organizational Units

```
echo 'NodeOUs:
Enable: true
ClientOUIdentifier:
  Certificate: cacerts/localhost-8054-ca-org2.pem
  OrganizationalUnitIdentifier: client
PeerOUIdentifier:
  Certificate: cacerts/localhost-8054-ca-org2.pem
  OrganizationalUnitIdentifier: peer
AdminOUIdentifier:
  Certificate: cacerts/localhost-8054-ca-org2.pem
  OrganizationalUnitIdentifier: admin
OrdererOUIdentifier:
  Certificate: cacerts/localhost-8054-ca-org2.pem
  OrganizationalUnitIdentifier: orderer' > "${PWD}/organizations/peerOrganiz
```

registerEnroll.sh

- **NodeOUs**: Node Organizational Units (NodeOUs). NodeOUs are a way to classify and organize identities within an organization.
- **ClientOUIdentifier:** "Client" Organizational Unit (OU). Configures the organizational unit for client nodes .
- **Certificate: cacerts/localhost-7054-ca-org1.pem:** Specifies the ca file
- **OrganizationalUnitIdentifier**: client : Label for clients(We can name anything).
- **PeerOUIdentifier**: Configuration related to the "Peer" Organizational Unit (OU).
- **AdminOUIdentifier:** Configuration related to the “Admin” Organizational Unit.
- **OrdererOUIdentifier:** Configuration related to “Orderer” OU.

registerEnroll.sh

- **fabric-ca-client register -** It allows to interact with the fabric-CA and register identity
- **caname ca-org1** - Specifies the name of ca
- **id.name peer0:** Name for the identity being registered.
- **id.secret peer0pw:** Password for the identity registered.
- **id.type peer** - “peer" indicates that this registration is for a peer node.
- The identity is named "peer0," with the password "peer0pw," and it is designated as a peer node registered with fabric-ca.

```
fabric-ca-client register --caname ca-org1 --id.name peer0 --id.secret peer0pw
 --id.type peer --tls.certfiles "${PWD}/organizations/fabric-ca/org1/ca-cert.pem"
```

Registering Peer

registerEnroll.sh

- **fabric-ca-client register -** It allows to interact with the fabric-CA and register the identity
- **caname ca-org1** - Specifies the name of ca
- **id.name user1:** Name for the identity being registered.
- **id.secret user1pw:** Password for the identity registered.
- **id.type client** - "client" often signifies an end user.
- This command registers a new user of type client.

```
fabric-ca-client register --caname ca-org1 --id.name user1 --id.secret user1pw
 --id.type client --tls.certfiles "${PWD}/organizations/fabric-ca/org1/ca-cert.pem"
```

Registering User

registerEnroll.sh

- **fabric-ca-client register -** It allows to interact with the fabric-network and register identity
- **caname ca-org1** - Specifies the name of ca
- **id.name org1admin:** Name for the identity being registered.
- **id.secret org1adminpw:** Password for the identity registered.
- **id.type admin** - "admin" typically signifies a user with administrative privileges (network management)
- This command registers a new identity of type admin which has network priveledges.

```
fabric-ca-client register --caname ca-org1 --id.name org1admin --id.secret org1adminpw
 --id.type admin --tls.certfiles "${PWD}/organizations/fabric-ca/org1/ca-cert.pem"
```

Registering Admin

registerEnroll.sh

- **fabric-ca-client enroll**: This is the main command for generating certificates for a new identity with the help of Fabric CA
- **-u:** Specifies the URL
- **--caname ca-org1:** Specifies the name of the CA
- **-M :** Specifies the directory in which to store the Membership Service Provider information for the enrolled peer.
- **--tls.certfiles:** Specifies the location of the TLS certificate file.
- This line of code is used to enroll a peer node and generate MSP for the peer.

```
fabric-ca-client enroll -u https://peer0:peer0pw@localhost:7054 --caname ca-org1
 -M "${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/msp"
  --tls.certfiles "${PWD}/organizations/fabric-ca/org1/ca-cert.pem"
```

Generating the peer MSP

registerEnroll.sh

- **fabric-ca-client enroll**: It is used for generating certificates for a new identity with the Fabric CA (in this case, it's focused on obtaining TLS certificates for secure communication)
- **-u :** Specifies the URL
- **--caname ca-org1:** specifies the name of the CA
- **-M :** Specifies the directory in which to store the TLS certificate for the enrolled peer.
- **--enrollment.profile tls**: Specifies the enrollment profile should be set to tls.
- **--csr.hosts peer0.org1.example.com --csr.hosts localhost :** Allows the certificates to be valid for multiple hostnames. These hostnames are the ones for which the TLS certificate will be valid.
- **--tls.certfiles** flag specifies the location of the TLS certificate file.
- This line of code is used to enroll a peer node and generate MSP for the peer.

```
fabric-ca-client enroll -u https://peer0:peer0pw@localhost:7054 --caname ca-org1
-M "${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls"
--enrollment.profile tls --csr.hosts peer0.org1.example.com --csr.hosts localhost
--tls.certfiles "${PWD}/organizations/fabric-ca/org1/ca-cert.pem"
```

Generating the peer TLS certificate ➡

registerEnroll.sh

- **fabric-ca-client enroll**: This is the main command for generating certificates for a new identity with the help of Fabric CA
- **-u:** Specifies the URL
- **--caname ca-org1:** Specifies the name of the CA
- **-M :** Specifies the directory in which to store the Membership Service Provider information for the enrolled peer.
- **--tls.certfiles:** Specifies the location of the TLS certificate file.
- This line of code is used to enroll a user and generate MSP for the user.

```
fabric-ca-client enroll -u https://user1:user1pw@localhost:7054 --caname ca-org1
-M "${PWD}/organizations/peerOrganizations/org1.example.com/users/User1@org1.example.com/msp"
--tls.certfiles "${PWD}/organizations/fabric-ca/org1/ca-cert.pem"
```

Generating the user MSP

registerEnroll.sh

- **fabric-ca-client enroll**: This is the main command for generating certificates for a new identity with the help of Fabric CA
- **-u:** Specifies the URL
- **--caname ca-org1:** Specifies the name of the CA
- **-M :** Specifies the directory in which to store the Membership Service Provider information for the enrolled peer.
- **--tls.certfiles:** Specifies the location of the TLS certificate file.
- This line of code is used to enroll an admin and generate MSP for the admin identity.

```
fabric-ca-client enroll -u https://org1admin:org1adminpw@localhost:7054 --caname ca-org1
-M "${PWD}/organizations/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp"
--tls.certfiles "${PWD}/organizations/fabric-ca/org1/ca-cert.pem"
```

Generating the admin MSP

Generating the orderer identities

```
echo "Generating the orderer msp"
set -x
fabric-ca-client enroll -u https://orderer:ordererpw@localhost:9054 --caname ca-orderer
-M "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp"
--tls.certfiles "${PWD}/organizations/fabric-ca/ordererOrg/ca-cert.pem"
```

```
echo "Generating the orderer-tls certificates, use --csr.hosts to specify Subject Alternative Names"
set -x
fabric-ca-client enroll -u https://orderer:ordererpw@localhost:9054 --caname ca-orderer
-M "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/tls"
--enrollment.profile tls --csr.hosts orderer.example.com --csr.hosts localhost
--tls.certfiles "${PWD}/organizations/fabric-ca/ordererOrg/ca-cert.pem"
```

```
echo "Generating the admin msp"
set -x
fabric-ca-client enroll -u https://ordererAdmin:ordererAdminpw@localhost:9054 --caname ca-orderer
-M "${PWD}/organizations/ordererOrganizations/example.com/users/Admin@example.com/msp"
--tls.certfiles "${PWD}/organizations/fabric-ca/ordererOrg/ca-cert.pem"
```