High-level Walk-through

1.   Generating crypto-material from the organizations and orderer (Org1, Org2, and Orderer) using Fabric CA server.

2.   Generating channel artifacts eg. system channel genesis block, channel transaction, channel genesis block, etc,).

3.   Updating anchor peers.

4.   Creating a channel.

5.   Joining the channel.

6.   Deploying chaincode on the channel.

7.   Invoke or query the chaincode.

Prerequisites

1.   Golang

2.   Docker

3.   Nodejs

Bring Up Network

Here we first download the HLF (Hyperledger Fabric Network) binaries along with fabric examples. We will be using the Test network provided in the fabric examples. Our network will consist of Two Organization with one peer each, One orderer node, Three CA server (one for each organization and third for orderer), Two couchdb instances for state database(one for each peer)

1. Download the fabric binaries.

curl -sSL https://bit.ly/2ysbOFE | bash -s -- 2.3.0 1.4.9

2. Set the environment variable

cd fabric-samples && export PATH=$PWD/bin:$PATH

3. Once step 2 is completed, go to the test-network folder.

cd test-network

4. Run the command and pass the options according to your need. To check all the available options run ./network.sh help

help command

5. To start the network with Fabric CA Servers, CouchDB and creating a channel mychannel.

- -ca: Use Certificate Authorities to generate network crypto material

- -c: Name of channel to create (defaults to "mychannel")

- -s: Peer state database to deploy: goleveldb (default) or couchdb

- up — Bring up Fabric orderer and peer nodes. No channel is created.

- up createChannel: Bring up fabric network with one channel.

- createChannel: Create and join a channel after the network is created

./network.sh up createChannel -ca -c mychannel -s couchdb

if everything goes well that at the end you can see something like this.

we have completed step 1 to step 5 in the High-level Walk-through section of this article.

6. Next step to deploy chaincode, in the example repository there are multiple chaincode

- -ccn: Chaincode name. This flag can be used to deploy one of the asset transfer samples to a channel. Sample options: basic (default), ledger, private, sbe, secured

- -ccl: Programming language of the chaincode to deploy: go (default), java, javascript, typescript

- -verbose: Verbose mode

You can explore all the available options in the help section

./network.sh deployCC -ccn basic -ccl go -verbose

once the chaincode deployment is successful you will get output like this

7. Now over chaincode deployment is done and we will now invoke and query the chaincode. we will be using nodejs SDK to interact with the blockchain network.

cd ../asset-transfer-basic/application-javascript && npm install

node app.js

This script will register the user appUser into the FileSystemWallet and will invoke the InitLedger function of the chaincode and this function will create 6 records into the ledger. After this it call the function GetAllAssets and this function returns all the records stored in the ledger, we can access the

CouchDB via the fauxton UI (peer0.org1=> http://localhost:5984/_utils/# and peer0.org2=> http://localhost:7984/_utils/#). After that it call CreateAsset which add a new record into the ledger.

And we have out HLF network up and running.