Configuration Manual

MSc Research Project

MSc in Cyber Security

Udith Ragav Saravana Kumar

Student ID: x23140348

School of Computing

National College of Ireland

Supervisor: Kamil Mahajan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Udith Ragav Saravana Kumar		
Student ID:	x23140348		
Programme:	MSc in Cyber Security	**Year:**	2024
Module:	MSc Research Project		
Supervisor:	Kamil Mahajan		
Submission Due Date:	29-01-2025		
Project Title:	Blockchain for Smart Home Data Privacy: Ethereum and Hyperledger Fabric.		
Word Count:	306 **Page Count** 6		

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Udith Ragav Saravana Kumar

Date: 29-01-2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Udith Ragav Saravana Kumar
Student ID : x23140348

1. Dependencies / Pre-requisites:

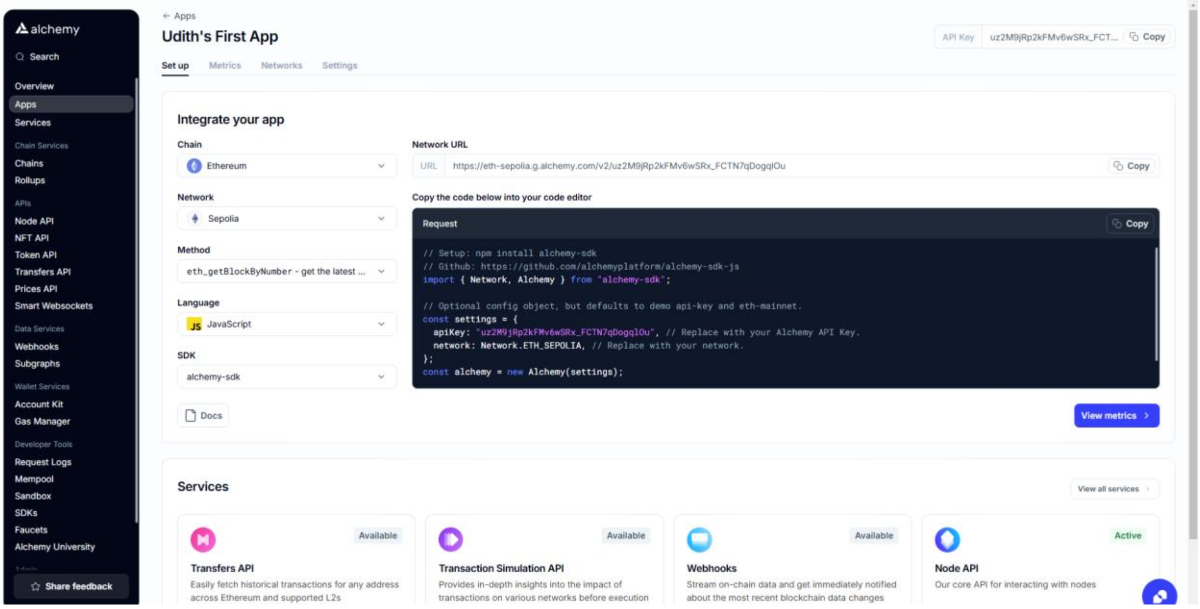
a) Visual Studio Code
b) Python 3.9
c) Jupyter Notebook
d) Alchemy Account
e) WSL (Ubuntu)
f) MetaMask web extension
g) Web3
h) IPFS Desktop
i) Hardhat
j) Nodejs
k) Go
l) cURL
m) Docker and Docker Compose (Desktop)

2.Ethereum Implementation:

To begin with we must install the required dependencies such as the IPFS desktop, Hardhat, Web3, NodeJS in the system.

2.1 Create an Alchemy account:

Ensure that your API is running in Alchemy.



Udith's First App

API Key uz2M9jRp2kFMv6wSRx_FCT...

Integrate your app

Chain: Ethereum

Network URL: https://eth-sepolia.g.alchemy.com/v2/uz2M9jRp2kFMv6wSRx_FCTN7qDogqlOu

Network: Sepolia

Method: eth_getBlockByNumber - get the latest ...

Language: JavaScript

SDK: alchemy-sdk

Copy the code below into your code editor

Request

```
// Setup: npm install alchemy-sdk
// Github: https://github.com/alchemyplatform/alchemy-sdk-js
import { Network, Alchemy } from "alchemy-sdk";

// Optional config object, but defaults to demo api-key and eth-mainnet.
const settings = {
  apiKey: "uz2M9jRp2kFMv6wSRx_FCTN7qDogqlOu", // Replace with your Alchemy API Key.
  network: Network.ETH_SEPOLIA, // Replace with your network.
};
const alchemy = new Alchemy(settings);
```

Services

Transfers API — Available
Easily fetch historical transactions for any address across Ethereum and supported L2s

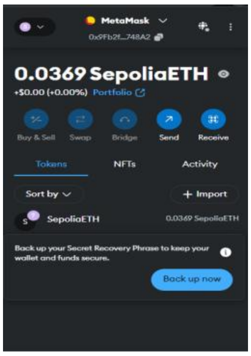
Transaction Simulation API — Available
Provides in-depth insights into the impact of transactions on various networks before execution

Webhooks — Available
Stream on-chain data and get immediately notified about the most recent blockchain data changes

Node API — Active
Our core API for interacting with nodes

2.2 Get the MetaMask extension:

Make sure that you network has enough Sepolia testnet currency to handle the transactions.



0x9Fb2f...748A2

0.0369 SepoliaETH

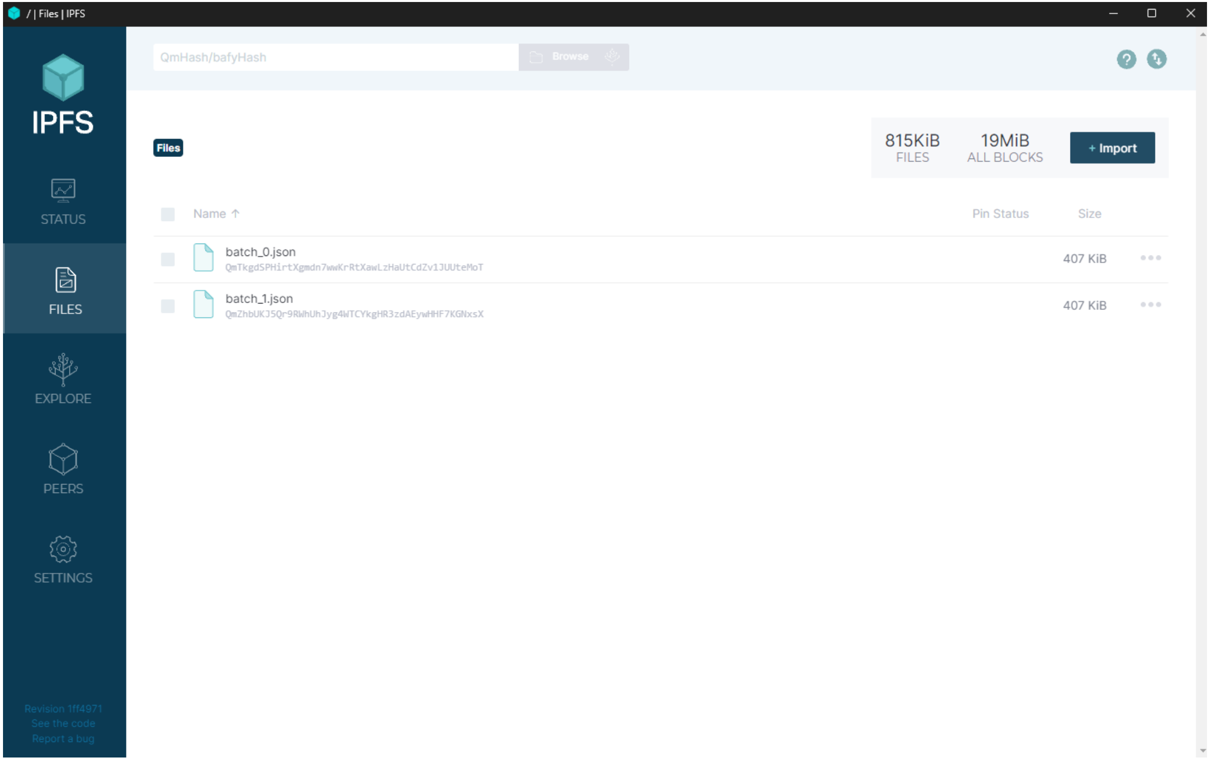
+$0.00 (+0.00%) Portfolio

SepoliaETH 0.0369 SepoliaETH

Back up your Secret Recovery Phrase to keep your wallet and funds secure.

2.3 Run the IPFS Desktop application:

Add the dataset files into the application.



815KiB FILES

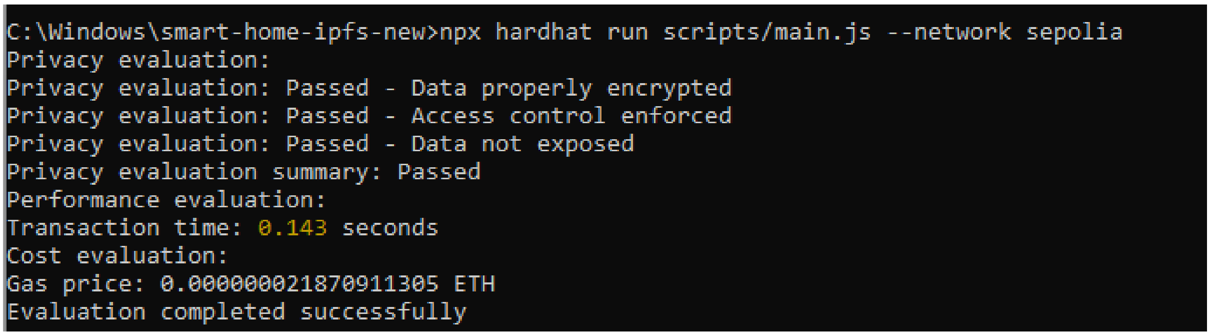
19MiB ALL BLOCKS

Name ↑	Pin Status	Size
batch_0.json QmTkgdSPHirtXgmdn7wwKrRtXawLzHaUtCdZv1JUUteMoT		407 KiB
batch_1.json QmZhbUKJ5Qr9RWhUhJyg4WTCYkgHR3zdAEywHHF7KGNxsX		407 KiB

2.4 Run the application:

Ensure that the hardhat network is running the machine and then run the application from the project directory using the script.

Command line: npx hardhat run scripts/main.js --network sepolia



```
C:\Windows\smart-home-ipfs-new>npx hardhat run scripts/main.js --network sepolia
Privacy evaluation:
Privacy evaluation: Passed - Data properly encrypted
Privacy evaluation: Passed - Access control enforced
Privacy evaluation: Passed - Data not exposed
Privacy evaluation summary: Passed
Performance evaluation:
Transaction time: 0.143 seconds
Cost evaluation:
Gas price: 0.000000021870911305 ETH
Evaluation completed successfully
```

3.Hyperledger Fabric Implementation:

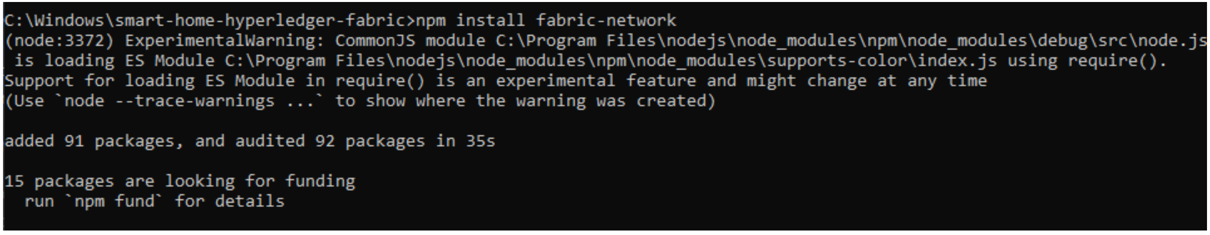
Install the pre-requisites for the Hyperledger implementation such as the Nodejs, Docker, Go, Python and cURL. We can do this with the help of Chocolatey in the Windows machine.

3.1 Download the Hyperledger Fabric Samples:

I have downloaded these from the GitHub link: https://github.com/hyperledger/fabric-samples.

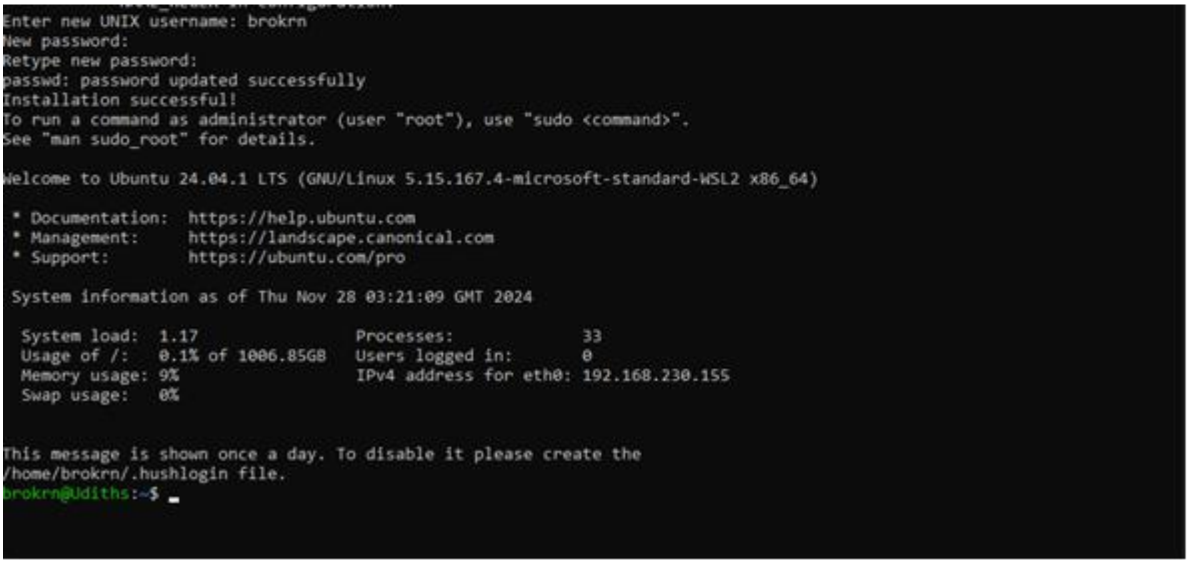
3.2 Install the fabric-network dependency on the project folder:

I used the npm command to install this dependency.



```
C:\Windows\smart-home-hyperledger-fabric>npm install fabric-network
(node:3372) ExperimentalWarning: CommonJS module C:\Program Files\nodejs\node_modules\npm\node_modules\debug\src\node.js
 is loading ES Module C:\Program Files\nodejs\node_modules\npm\node_modules\supports-color\index.js using require().
Support for loading ES Module in require() is an experimental feature and might change at any time
(Use `node --trace-warnings ...` to show where the warning was created)

added 91 packages, and audited 92 packages in 35s

15 packages are looking for funding
  run `npm fund` for details
```

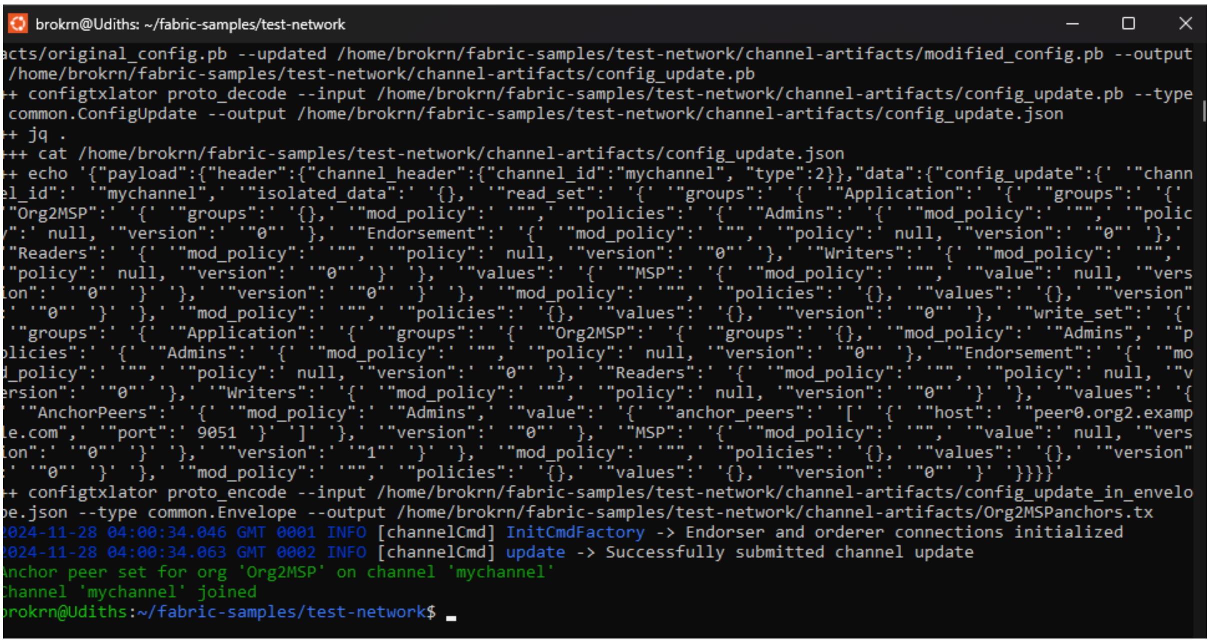
3.3 Install WSL in the Powershell(as Administrator) to use Linux commands in the Windows Machine



```
Enter new UNIX username: brokrn
New password:
Retype new password:
passwd: password updated successfully
Installation successful!
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 5.15.167.4-microsoft-standard-WSL2 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Thu Nov 28 03:21:09 GMT 2024

  System load:  1.17               Processes:             33
  Usage of /:   0.1% of 1006.85GB  Users logged in:       0
  Memory usage: 9%                 IPv4 address for eth0: 192.168.230.155
  Swap usage:   0%


This message is shown once a day. To disable it please create the
/home/brokrn/.hushlogin file.
brokrn@Udiths:~$
```

3.4 Set up the Fabric network:

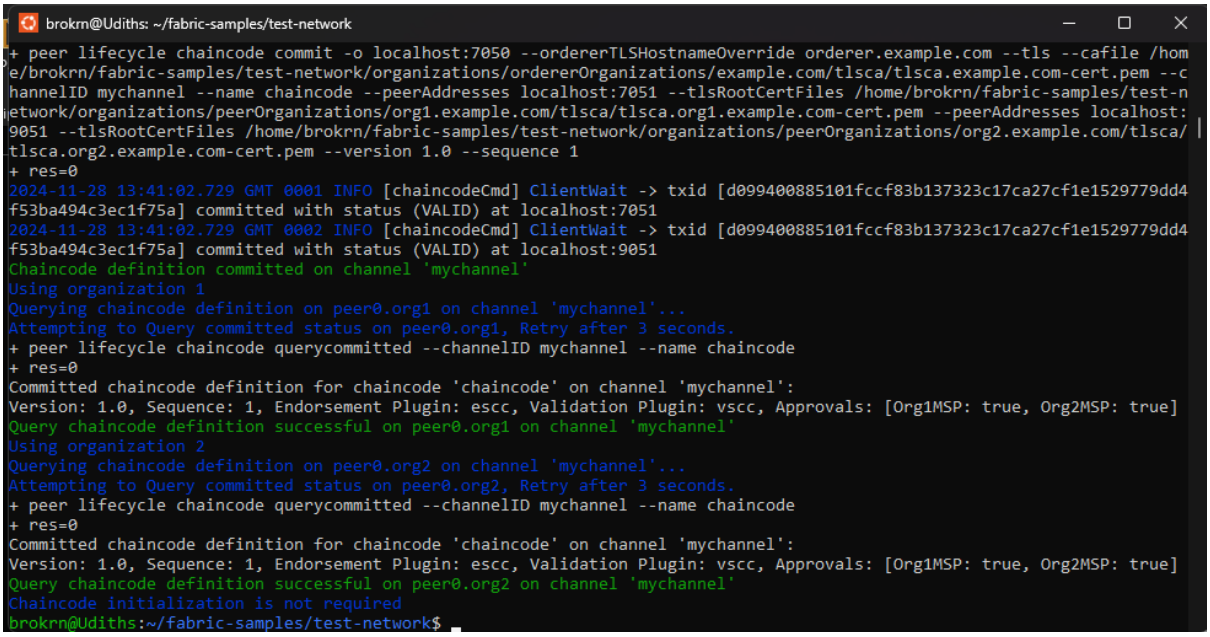
Command line: $ ./network.sh down $ ./network.sh up createChannel



```
cts/original_config.pb --updated /home/brokrn/fabric-samples/test-network/channel-artifacts/modified_config.pb --output
 /home/brokrn/fabric-samples/test-network/channel-artifacts/config_update.pb
+ configtxlator proto_decode --input /home/brokrn/fabric-samples/test-network/channel-artifacts/config_update.pb --type
 common.ConfigUpdate --output /home/brokrn/fabric-samples/test-network/channel-artifacts/config_update.json
+ jq .
++ cat /home/brokrn/fabric-samples/test-network/channel-artifacts/config_update.json
+ echo '{"payload":{"header":{"channel_header":{"channel_id":"mychannel", "type":2}},"data":{"config_update":{' '"chann
l_id":' '"mychannel",' '"isolated_data":' '{},' '"read_set":' '{' '"groups":' '{' '"Application":' '{' '"groups":' '{'
'"Org2MSP":' '{' '"groups":' '{},' '"mod_policy":' '"",' '"policies":' '{' '"Admins":' '{' '"mod_policy":' '"",' '"polic
":' null, '"version":' '"0"' '},' '"Endorsement":' '{' '"mod_policy":' '"",' '"policy":' null, '"version":' '"0"' '},'
'"Readers":' '{' '"mod_policy":' '"",' '"policy":' null, '"version":' '"0"' '},' '"Writers":' '{' '"mod_policy":' '"",'
'"policy":' null, '"version":' '"0"' '}' '},' '"values":' '{' '"MSP":' '{' '"mod_policy":' '"",' '"value":' null, '"vers
on":' '"0"' '}' '},' '"version":' '"0"' '}' '},' '"mod_policy":' '"",' '"policies":' '{},' '"values":' '{},' '"version"
' '"0"' '}' '},' '"mod_policy":' '"",' '"policies":' '{},' '"values":' '{},' '"version":' '"0"' '},' '"write_set":' '{'
'"groups":' '{' '"Application":' '{' '"groups":' '{' '"Org2MSP":' '{' '"groups":' '{},' '"mod_policy":' '"Admins",' '"p
licies":' '{' '"Admins":' '{' '"mod_policy":' '"",' '"policy":' null, '"version":' '"0"' '},' '"Endorsement":' '{' '"mo
_policy":' '"",' '"policy":' null, '"version":' '"0"' '},' '"Readers":' '{' '"mod_policy":' '"",' '"policy":' null, '"v
rsion":' '"0"' '},' '"Writers":' '{' '"mod_policy":' '"",' '"policy":' null, '"version":' '"0"' '}' '},' '"values":' '{
 '"AnchorPeers":' '{' '"mod_policy":' '"Admins",' '"value":' '{' '"anchor_peers":' '[' '{' '"host":' '"peer0.org2.examp
e.com",' '"port":' 9051 '}' ']' '},' '"version":' '"0"' '},' '"MSP":' '{' '"mod_policy":' '"",' '"value":' null, '"vers
on":' '"0"' '}' '},' '"version":' '"1"' '}' '},' '"mod_policy":' '"",' '"policies":' '{},' '"values":' '{},' '"version"
' '"0"' '}' '},' '"mod_policy":' '"",' '"policies":' '{},' '"values":' '{},' '"version":' '"0"' '}' '}}}}'
+ configtxlator proto_encode --input /home/brokrn/fabric-samples/test-network/channel-artifacts/config_update_in_envelo
e.json --type common.Envelope --output /home/brokrn/fabric-samples/test-network/channel-artifacts/Org2MSPanchors.tx
024-11-28 04:00:34.046 GMT 0001 INFO [channelCmd] InitCmdFactory -> Endorser and orderer connections initialized
024-11-28 04:00:34.063 GMT 0002 INFO [channelCmd] update -> Successfully submitted channel update
nchor peer set for org 'Org2MSP' on channel 'mychannel'
hannel 'mychannel' joined
rokrn@Udiths:~/fabric-samples/test-network$
```

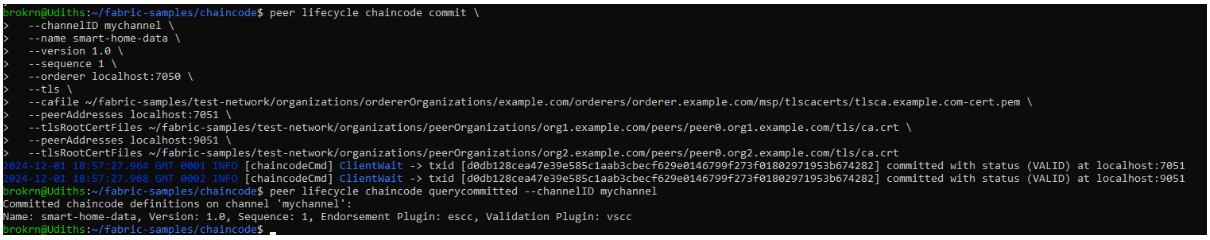
3.5 Deploy the chaincode:

Command line: $ ./network.sh deployCC -ccn chaincode -ccp ../chaincode -ccl javascript -c mychannel



```
+ peer lifecycle chaincode commit -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile /hom
e/brokrn/fabric-samples/test-network/organizations/ordererOrganizations/example.com/tlsca/tlsca.example.com-cert.pem --c
hannelID mychannel --name chaincode --peerAddresses localhost:7051 --tlsRootCertFiles /home/brokrn/fabric-samples/test-n
etwork/organizations/peerOrganizations/org1.example.com/tlsca/tlsca.org1.example.com-cert.pem --peerAddresses localhost:
9051 --tlsRootCertFiles /home/brokrn/fabric-samples/test-network/organizations/peerOrganizations/org2.example.com/tlsca/
tlsca.org2.example.com-cert.pem --version 1.0 --sequence 1
+ res=0
2024-11-28 13:41:02.729 GMT 0001 INFO [chaincodeCmd] ClientWait -> txid [d099400885101fccf83b137323c17ca27cf1e1529779dd4
f53ba494c3ec1f75a] committed with status (VALID) at localhost:7051
2024-11-28 13:41:02.729 GMT 0002 INFO [chaincodeCmd] ClientWait -> txid [d099400885101fccf83b137323c17ca27cf1e1529779dd4
f53ba494c3ec1f75a] committed with status (VALID) at localhost:9051
Chaincode definition committed on channel 'mychannel'
Using organization 1
Querying chaincode definition on peer0.org1 on channel 'mychannel'...
Attempting to Query committed status on peer0.org1, Retry after 3 seconds.
+ peer lifecycle chaincode querycommitted --channelID mychannel --name chaincode
+ res=0
Committed chaincode definition for chaincode 'chaincode' on channel 'mychannel':
Version: 1.0, Sequence: 1, Endorsement Plugin: escc, Validation Plugin: vscc, Approvals: [Org1MSP: true, Org2MSP: true]
Query chaincode definition successful on peer0.org1 on channel 'mychannel'
Using organization 2
Querying chaincode definition on peer0.org2 on channel 'mychannel'...
Attempting to Query committed status on peer0.org2, Retry after 3 seconds.
+ peer lifecycle chaincode querycommitted --channelID mychannel --name chaincode
+ res=0
Committed chaincode definition for chaincode 'chaincode' on channel 'mychannel':
Version: 1.0, Sequence: 1, Endorsement Plugin: escc, Validation Plugin: vscc, Approvals: [Org1MSP: true, Org2MSP: true]
Query chaincode definition successful on peer0.org2 on channel 'mychannel'
Chaincode initialization is not required
brokrn@Udiths:~/fabric-samples/test-network$
```

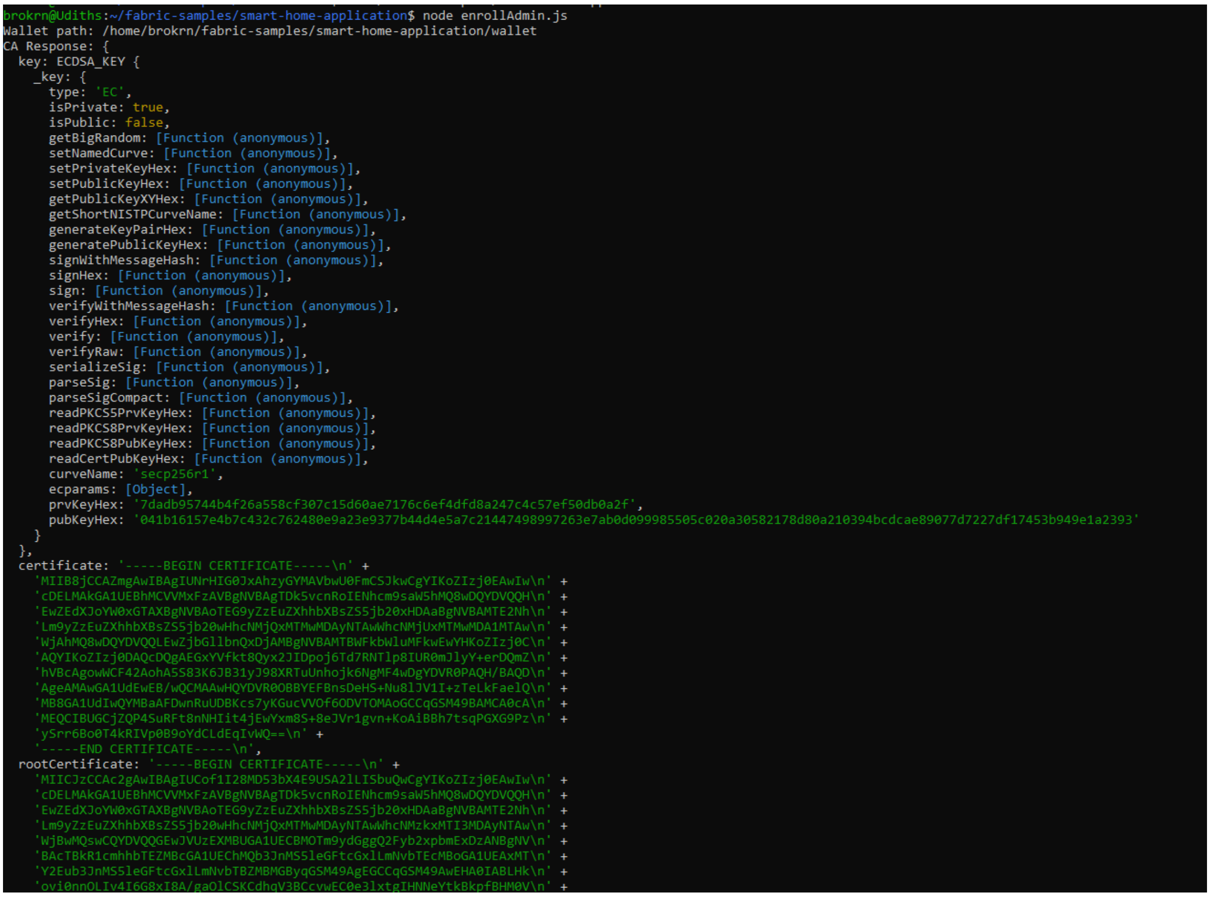
3.6 Commit and Verify the chaincode:



```
brokrn@Udiths:~/fabric-samples/chaincode$ peer lifecycle chaincode commit \
>   --channelID mychannel \
>   --name smart-home-data \
>   --version 1.0 \
>   --sequence 1 \
>   --orderer localhost:7050 \
>   --tls \
>   --cafile ~/fabric-samples/test-network/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem \
>   --peerAddresses localhost:7051 \
>   --tlsRootCertFiles ~/fabric-samples/test-network/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt \
>   --peerAddresses localhost:9051 \
>   --tlsRootCertFiles ~/fabric-samples/test-network/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt
2024-12-01 18:57:27.964 GMT 0001 INFO [chaincodeCmd] ClientWait -> txid [d0db128cea47e39e585c1aab3cbecf629e0146799f273f01802971953b674282] committed with status (VALID) at localhost:7051
2024-12-01 18:57:27.968 GMT 0002 INFO [chaincodeCmd] ClientWait -> txid [d0db128cea47e39e585c1aab3cbecf629e0146799f273f01802971953b674282] committed with status (VALID) at localhost:9051
brokrn@Udiths:~/fabric-samples/chaincode$ peer lifecycle chaincode querycommitted --channelID mychannel
Committed chaincode definitions on channel 'mychannel':
Name: smart-home-data, Version: 1.0, Sequence: 1, Endorsement Plugin: escc, Validation Plugin: vscc
brokrn@Udiths:~/fabric-samples/chaincode$
```

3.7 Enrol the admin so that the application can run.

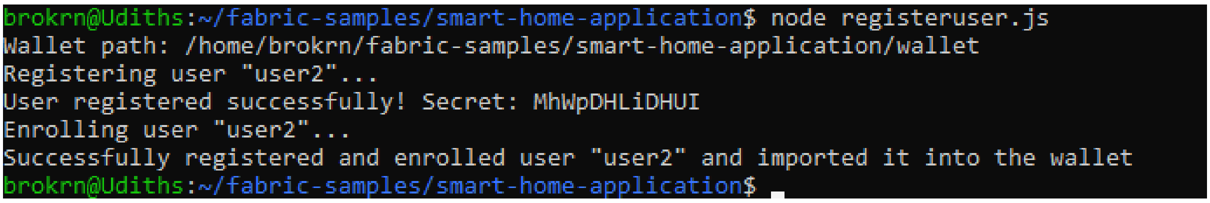
Command line: $ node enrollAdmin.js



```
brokrn@Udiths:~/fabric-samples/smart-home-application$ node enrollAdmin.js
Wallet path: /home/brokrn/fabric-samples/smart-home-application/wallet
CA Response: {
  key: ECDSA_KEY {
    _key: {
      type: 'EC',
      isPrivate: true,
      isPublic: false,
      getBigRandom: [Function (anonymous)],
      setNamedCurve: [Function (anonymous)],
      setPrivateKeyHex: [Function (anonymous)],
      setPublicKeyHex: [Function (anonymous)],
      getPublicKeyXYHex: [Function (anonymous)],
      getShortNISTPCurveName: [Function (anonymous)],
      generateKeyPairHex: [Function (anonymous)],
      generatePublicKeyHex: [Function (anonymous)],
      signWithMessageHash: [Function (anonymous)],
      signHex: [Function (anonymous)],
      sign: [Function (anonymous)],
      verifyWithMessageHash: [Function (anonymous)],
      verifyHex: [Function (anonymous)],
      verify: [Function (anonymous)],
      verifyRaw: [Function (anonymous)],
      serializeSig: [Function (anonymous)],
      parseSig: [Function (anonymous)],
      parseSigCompact: [Function (anonymous)],
      readPKCS5PrvKeyHex: [Function (anonymous)],
      readPKCS8PrvKeyHex: [Function (anonymous)],
      readPKCS8PubKeyHex: [Function (anonymous)],
      readCertPubKeyHex: [Function (anonymous)],
      curveName: 'secp256r1',
      ecparams: [Object],
      prvKeyHex: '7dadb95744b4f26a558cf307c15d60ae7176c6ef4dfd8a247c4c57ef50db0a2f',
      pubKeyHex: '041b16157e4b7c432c762480e9a23e9377b44d4e5a7c21447498997263e7ab0d099985505c020a30582178d80a210394bcdcae89077d7227df17453b949e1a2393'
    }
  },
  certificate: '-----BEGIN CERTIFICATE-----\n' +
    'MIIB8jCCAZmgAwIBAgIUNrHIG0JxAhzyGYMAVbwU0FmCSJkwCgYIKoZIzj0EAwIw\n' +
    'cDELMAkGA1UEBhMCVVMxFzAVBgNVBAgTDk5vcnRoIENhcm9saW5hMQ8wDQYDVQQH\n' +
    'EwZEdXJoYW0xGTAXBgNVBAoTEG9yZzEuZXhhbXBsZS5jb20xHDAaBgNVBAMTE2Nh\n' +
    'Lm9yZzEuZXhhbXBsZS5jb20wHhcNMjQxMTMwMDAyNTAwWhcNMjUxMTMwMDA1MTAw\n' +
    'WjAhMQ8wDQYDVQQLEwZjbGllbnQxDjAMBgNVBAMTBWFkbWluMFkwEwYHKoZIzj0C\n' +
    'AQYIKoZIzj0DAQcDQgAEGxYVfkt8Qyx2JIDpoj6Td7RNTlp8IUR0mJlyY+erDQmZ\n' +
    'hVBcAgowWCF42AohA5S83K6JB31yJ98XRTuUnhojk6NgMF4wDgYDVR0PAQH/BAQD\n' +
    'AgeAMAwGA1UdEwEB/wQCMAAwHQYDVR0OBBYEFBnsDeHS+Nu8lJV1I+zTeLkFaelQ\n' +
    'MB8GA1UdIwQYMBaAFDwnRuUDBKcs7yKGucVVOf6ODVTOMAoGCCqGSM49BAMCA0cA\n' +
    'MEQCIBUGCjZQP4SuRFt8nNHIit4jEwYxm8S+8eJVr1gvn+KoAiBBh7tsqPGXG9Pz\n' +
    'ySrr6Bo0T4kRIVp0B9oYdCLdEqIvWQ==\n' +
    '-----END CERTIFICATE-----\n',
  rootCertificate: '-----BEGIN CERTIFICATE-----\n' +
    'MIICJzCCAc2gAwIBAgIUCof1I28MD53bX4E9USA2lLISbuQwCgYIKoZIzj0EAwIw\n' +
    'cDELMAkGA1UEBhMCVVMxFzAVBgNVBAgTDk5vcnRoIENhcm9saW5hMQ8wDQYDVQQH\n' +
    'EwZEdXJoYW0xGTAXBgNVBAoTEG9yZzEuZXhhbXBsZS5jb20xHDAaBgNVBAMTE2Nh\n' +
    'Lm9yZzEuZXhhbXBsZS5jb20wHhcNMjQxMTMwMDAyNTAwWhcNMzkxMTI3MDAyNTAw\n' +
    'WjBwMQswCQYDVQQGEwJVUzEXMBUGA1UECBMOTm9ydGggQ2Fyb2xpbmExDzANBgNV\n' +
    'BAcTBkR1cmhhbTEZMBcGA1UEChMQb3JnMS5leGFtcGxlLmNvbTEcMBoGA1UEAxMT\n' +
    'Y2Eub3JnMS5leGFtcGxlLmNvbTBZMBMGByqGSM49AgEGCCqGSM49AwEHA0IABLHk\n' +
    'ovi0nnOLIv4I6G8xI8A/gaOlCSKCdhqV3BCcvwEC0e3lxtgIHNNeYtkBkpfBHM0V\n' +
```

3.8 Register the user.

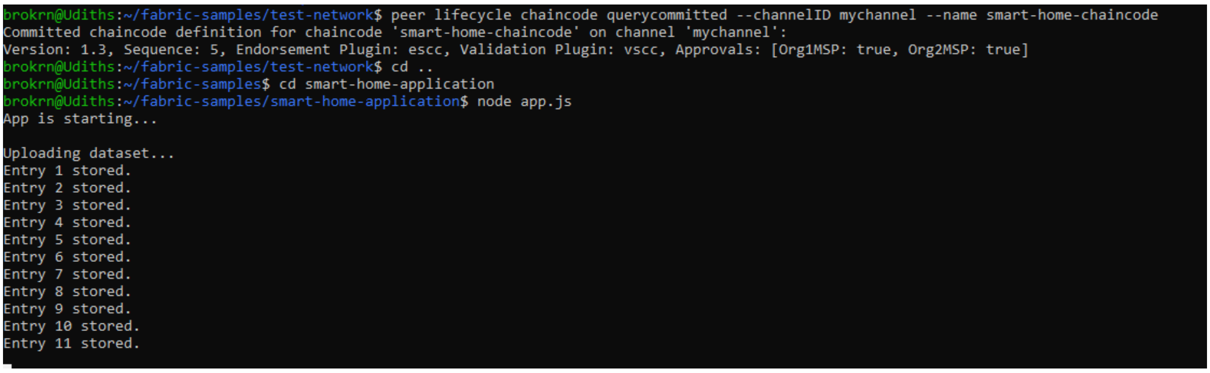
Command line: $ node registeruser.js



```
brokrn@Udiths:~/fabric-samples/smart-home-application$ node registeruser.js
Wallet path: /home/brokrn/fabric-samples/smart-home-application/wallet
Registering user "user2"...
User registered successfully! Secret: MhWpDHLiDHUI
Enrolling user "user2"...
Successfully registered and enrolled user "user2" and imported it into the wallet
brokrn@Udiths:~/fabric-samples/smart-home-application$
```

3.9 Run the application script

Command line: $ node app.js



```
brokrn@Udiths:~/fabric-samples/test-network$ peer lifecycle chaincode querycommitted --channelID mychannel --name smart-home-chaincode
Committed chaincode definition for chaincode 'smart-home-chaincode' on channel 'mychannel':
Version: 1.3, Sequence: 5, Endorsement Plugin: escc, Validation Plugin: vscc, Approvals: [Org1MSP: true, Org2MSP: true]
brokrn@Udiths:~/fabric-samples/test-network$ cd ..
brokrn@Udiths:~/fabric-samples$ cd smart-home-application
brokrn@Udiths:~/fabric-samples/smart-home-application$ node app.js
App is starting...

Uploading dataset...
Entry 1 stored.
Entry 2 stored.
Entry 3 stored.
Entry 4 stored.
Entry 5 stored.
Entry 6 stored.
Entry 7 stored.
Entry 8 stored.
Entry 9 stored.
Entry 10 stored.
Entry 11 stored.
```