

**Roll number – 19BCE123**

**Name – Jash Mavani**

**Batch - EL2**

**Course Code & Name – 2CSDE93 & Blockchain Technology**

## **Practical – 7**

**Aim :** To perform a thorough study of blockchain development on Hyperledger Fabric using Composer

## **Implementation Details :**

```
sanidhya@sanidhya-VirtualBox:~/fabric-samples/test-network$ sudo ./network.sh up createChannel -ca
Creating channel 'mychannel'.
If network is not up, starting nodes with CLI timeout of '5' tries and CLI delay of '3' seconds and using database 'leveldb with crypto from
'Certificate Authorities'
Bringing up network
LOCAL_VERSION=2.2.2
DOCKER_IMAGE_VERSION=2.2.2
CA_LOCAL_VERSION=1.4.9
CA_DOCKER_IMAGE_VERSION=1.4.9
Generating certificates using Fabric CA
Creating network "net_test" with the default driver
Creating ca_org1 ... done
Creating ca_orderer ... done
Creating ca_org2 ... done
Creating Org1 Identities
Enrolling the CA admin
+ fabric-ca-client enroll -u https://admin:adminpw@localhost:7054 --caname ca-org1 --tls.certfiles /home/sanidhya/fabric-samples/test-network
/organizations/fabric-ca/org1/tls-cert.pem
2021/10/29 20:52:37 [INFO] Created a default configuration file at /home/sanidhya/fabric-samples/test-network/organizations/peerOrganizations
/org1.example.com/fabric-ca-client-config.yaml
2021/10/29 20:52:37 [INFO] TLS Enabled
2021/10/29 20:52:37 [INFO] generating key: &{A:ecdsa S:256}
2021/10/29 20:52:37 [INFO] encoded CSR
2021/10/29 20:52:38 [INFO] Stored client certificate at /home/sanidhya/fabric-samples/test-network/organizations/peerOrganizations/org1.examp
le.com/msp/signcerts/cert.pem
2021/10/29 20:52:38 [INFO] Stored root CA certificate at /home/sanidhya/fabric-samples/test-network/organizations/peerOrganizations/org1.exam
ple.com/msp/cacerts/localhost-7054-ca-org1.pem
2021/10/29 20:52:38 [INFO] Stored Issuer public key at /home/sanidhya/fabric-samples/test-network/organizations/peerOrganizations/org1.examp
le.com/msp/IssuerPublicKey
2021/10/29 20:52:38 [INFO] Stored Issuer revocation public key at /home/sanidhya/fabric-samples/test-network/organizations/peerOrganizations/
org1.example.com/msp/IssuerRevocationPublicKey
Registering peer0
+ fabric-ca-client register --caname ca-org1 --id.name peer0 --id.secret peer0pw --id.type peer --tls.certfiles /home/sanidhya/fabric-samples
/test-network/organizations/fabric-ca/org1/tls-cert.pem
2021/10/29 20:52:38 [INFO] Configuration file location: /home/sanidhya/fabric-samples/test-network/organizations/peerOrganizations/org1.examp
le.com/fabric-ca-client-config.yaml
2021/10/29 20:52:38 [INFO] TLS Enabled
2021/10/29 20:52:38 [INFO] TLS Enabled
```

```

Fetching the most recent configuration block for the channel
+ peer channel fetch config config_block.pb -o orderer.example.com:7050 --ordererTLSHostnameOverride orderer.example.com -c mychannel --tls -
-cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tl
scacerts/tlsca.example.com-cert.pem
2021-10-29 15:23:02.442 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2021-10-29 15:23:02.446 UTC [cli.common] readBlock -> INFO 002 Received block: 1
2021-10-29 15:23:02.446 UTC [channelCmd] fetch -> INFO 003 Retrieving last config block: 1
2021-10-29 15:23:02.447 UTC [cli.common] readBlock -> INFO 004 Received block: 1
Decoding config block to JSON and isolating config to Org2MSPconfig.json
+ jq '.data.data[0].payload.data.config'
+ configtxlator proto_decode --input config_block.pb --type common.Block
+ jq '.channel_group.groups.Application.groups.Org2MSP.values += [{"AnchorPeers":{"mod_policy": "Admins","value":{"anchor_peers": [{"host": "p
eer0.org2.example.com","port": 9051}],"version": "0"}}]' Org2MSPconfig.json
Generating anchor peer update transaction for Org2 on channel mychannel
+ configtxlator proto_encode --input Org2MSPconfig.json --type common.Config
+ configtxlator proto_encode --input Org2MSPmodified_config.json --type common.Config
+ configtxlator compute_update --channel_id mychannel --original original_config.pb --updated modified_config.pb
+ configtxlator proto_decode --input config_update.pb --type common.ConfigUpdate
+ jq .
++ cat config_update.json
+ echo '{"payload":{"header":{"channel_header":{"channel_id":"mychannel", "type":2},"data":{"config_update":{"channel_id":"mychannel",
"isolated_data":{"groups":{"Application":{"groups":{"Org2MSP":{"groups":{"mod_po
licy":{"policies":{"Admins":{"mod_policy":{"policy": null, "version": "0"}}, "Endorsement":{"mod_po
licy":{"policy": null, "version": "0"}}, "Readers":{"mod_policy":{"policy": null, "version": "0"}}, "Wr
riters":{"mod_policy":{"policy": null, "version": "0"}}, "values":{"MSP":{"mod_policy":{"value":
null, "version": "0"}}, "version": "0"}}, "mod_policy":{"policies":{"values":{"version": "1"
}}}, "mod_policy":{"policies":{"values":{"write_set":{"groups":{"Applicati
on":{"groups":{"Org2MSP":{"groups":{"mod_policy":{"Admins","policies":{"Admins":{"mod_policy": "
","policy": 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":{"AnchorPeers":{"mod_policy":{"Admins","value":{"anchor_peers": [{"host": "peer0.or
g2.example.com","port": 9051}],"version": "0"}}, "MSP":{"mod_policy":{"value": null, "version": "0"}
},"version": "1"}}, "mod_policy":{"policies":{"values":{"version": "1"}}, "mod_policy":
"","policies":{"values":{"version": "0"}}}}}}}}'
+ configtxlator proto_encode --input config_update_in_envelope.json --type common.Envelope
2021-10-29 15:23:02.874 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2021-10-29 15:23:02.886 UTC [channelCmd] update -> INFO 002 Successfully submitted channel update
Anchor peer set for org 'Org2MSP' on channel 'mychannel'

```

⇒ It creates a channel named mychannel and also allows two peers, org1 & org2 to join mychannel. Now we can run the sample chaincode in this channel between the two peers. The chaincode is the ERC-20 token contract, previously deployed in Practical 4. This version is writing in Golang and can be deployed using the following command:

```

sudo ./network.sh deployCC -ccn token_erc20 -ccp
../token-erc20/chaincode-go/ -ccl go

```

```
sanidhya@sanidhya-VirtualBox:~/Fabric-samples/test-network$ sudo ./network.sh deployCC -ccn token_erc20 -ccp ../token-erc-20/chaincode-javascript
rpt/ -ccl javascript
deploying chaincode on channel 'mychannel'
executing with the following
- CHANNEL_NAME: mychannel
- CC_NAME: token_erc20
- CC_SRC_PATH: ../token-erc-20/chaincode-javascript/
- CC_SRC_LANGUAGE: javascript
- CC_VERSION: 1.0
- CC_SEQUENCE: 1
- CC_END_POLICY: NA
- CC_COLL_CONFIG: NA
- CC_INIT_FCN: NA
- DELAY: 3
- MAX_RETRY: 5
- VERBOSE: false
+ peer lifecycle chaincode package token_erc20.tar.gz --path ../token-erc-20/chaincode-javascript/ --lang node --label token_erc20_1.0
+ res=0
Chaincode is packaged
Installing chaincode on peer0.org1...
Using organization 1
+ peer lifecycle chaincode install token_erc20.tar.gz
+ res=0
2021-10-29 20:54:21.306 IST [cli.lifecycle.chaincode] submitInstallProposal -> INFO 001 Installed remotely: response:<status:200 payload:"\nP
token_erc20_1.0:1746c08cfe5481bd974b15fd878de212af65cc1698958dc2f5e4ae3f55b836b9\022\017token_erc20_1.0" >
2021-10-29 20:54:21.307 IST [cli.lifecycle.chaincode] submitInstallProposal -> INFO 002 Chaincode code package identifier: token_erc20_1.0:17
46c08cfe5481bd974b15fd878de212af65cc1698958dc2f5e4ae3f55b836b9
Chaincode is installed on peer0.org1
Install chaincode on peer0.org2...
Using organization 2
+ peer lifecycle chaincode install token_erc20.tar.gz
+ res=0
2021-10-29 20:54:41.481 IST [cli.lifecycle.chaincode] submitInstallProposal -> INFO 001 Installed remotely: response:<status:200 payload:"\nP
token_erc20_1.0:1746c08cfe5481bd974b15fd878de212af65cc1698958dc2f5e4ae3f55b836b9\022\017token_erc20_1.0" >
2021-10-29 20:54:41.481 IST [cli.lifecycle.chaincode] submitInstallProposal -> INFO 002 Chaincode code package identifier: token_erc20_1.0:17
46c08cfe5481bd974b15fd878de212af65cc1698958dc2f5e4ae3f55b836b9
Chaincode is installed on peer0.org2
Using organization 1
+ peer lifecycle chaincode queryinstalled
```

## Conclusion :

After completion of this practical, I learnt how HyperLedger Fabric works.