

## Practical 2

**Aim: Create and deploy a block chain network using Hyperledger Fabric SDK for Java.**

### Install Hyperledger (If not already done)

```
curl -sSL http://bit.ly/2ysb0FE | bash -s 2.2.0
```

Once its installed there will be a directory named fabric-samples

```
→ web3 ls
fabric-samples
→ web3 █
```

### Starting the hyperledger test network

```
cd fabric-samples/test-network
```

```
→ web3 cd fabric-samples/test-network
→ test-network git:(main) █
```

```
./network.sh up
```

```
→ test-network git:(main) ./network.sh up
Using docker and docker compose
Starting nodes with CLI timeout of '5' tries and CLI delay of '3' seconds and using database 'leveldb' with crypt
LOCAL_VERSION=v2.5.4
DOCKER_IMAGE_VERSION=v2.5.4
/home/hackerman/web3/fabric-samples/test-network/./bin/cryptogen
Generating certificates using cryptogen tool
Creating Org1 Identities
+ cryptogen generate --config=./organizations/cryptogen/crypto-config-org1.yaml --output=organizations
org1.example.com
+ res=0
Creating Org2 Identities
+ cryptogen generate --config=./organizations/cryptogen/crypto-config-org2.yaml --output=organizations
org2.example.com
+ res=0
Creating Orderer Org Identities
+ cryptogen generate --config=./organizations/cryptogen/crypto-config-orderer.yaml --output=organizations
+ res=0
Generating CCP files for Org1 and Org2
[+] Running 4/8
.: Network fabric_test Created
.: Volume "compose_orderer.example.com" Created
.: Volume "compose_peer0.org1.example.com" Created
.: Volume "compose_peer0.org2.example.com" Created
✓ Container peer0.org1.example.com Started
✓ Container peer0.org2.example.com Started
✓ Container orderer.example.com Started
✓ Container cli Started
```

## Creating HyperLedger channels

```
→ test-network git:(main) ./network.sh createChannel
Using docker and docker compose
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
Network Running Already
Using docker and docker compose
Generating channel genesis block 'mychannel.block'
Using organization 1
/home/hackerman/web3/fabric-samples/test-network/./bin/configtxgen
+ '[' 0 -eq 1 ']'
+ configtxgen -profile ChannelUsingRaft -outputBlock ./channel-artifacts/mychannel.block -channelID mychannel
2024-02-14 11:51:46.363 IST 0001 INFO [common.tools.configtxgen] main -> Loading configuration
2024-02-14 11:51:46.377 IST 0002 INFO [common.tools.configtxgen.localconfig] completeInitialization -> orderer type: etcdraft
2024-02-14 11:51:46.377 IST 0003 INFO [common.tools.configtxgen.localconfig] completeInitialization -> Orderer.EtcdRaft.Options unset, set
x_inflight_blocks:5 snapshot_interval_size:16777216
2024-02-14 11:51:46.377 IST 0004 INFO [common.tools.configtxgen.localconfig] Load -> Loaded configuration: /home/hackerman/web3/fabric-sam
2024-02-14 11:51:46.433 IST 0005 INFO [common.tools.configtxgen] doOutputBlock -> Generating genesis block
2024-02-14 11:51:46.433 IST 0006 INFO [common.tools.configtxgen] doOutputBlock -> Creating application channel genesis block
2024-02-14 11:51:46.435 IST 0007 INFO [common.tools.configtxgen] doOutputBlock -> Writing genesis block
+ res=0
Creating channel mychannel
Adding orderers
+ . scripts/orderer.sh mychannel
+ '[' 0 -eq 1 ']'
+ res=0
```

## Checking the configs

```
++ cat config_update.json
+ echo '{"payload":{"header":{"channel_header":{"channel_id":"mychannel", "type":2}},"data":{"config_update":{"channel_id":"' mychannel ',' isolated_data': '{}', 'read_set': {'groups':
{'Application': {'groups': {'Org2MSP': {'groups': {}, 'mod_policy': '',
'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': {'MSP': {'mod_policy':
', 'value': null, 'version': '0'}, 'version': '0'}, 'mod_policy':
'policies': {}, 'values': {}, 'version': '0'}, 'mod_policy':
', 'values': {}, 'version': '0'}, 'write_set': {'groups': {'Application
': {'groups': {'Org2MSP': {'groups': {}, 'mod_policy': 'Admins', 'policies':
{'Admins': {'mod_policy': '', 'policy': null, 'version': '0'}, 'Endorsement
': {'mod_policy': '', 'policy': null, 'version': '0'}, 'Readers': {'mod_polic
y': '', 'policy': null, 'version': '0'}, 'Writers': {'mod_policy':
', 'policy': null, 'version': '0'}, 'values': {'AnchorPeers': {'mod_policy': 'Admin
s', 'value': {'anchor_peers': [{'host': 'peer0.org2.example.com', 'port': 9051
'}] }, 'version': '0'}, 'MSP': {'mod_policy': '', 'value': null, 'version
': '0'}, 'version': '1'}, 'mod_policy': '', 'policies': {}, 'values':
, 'version': '0'}, 'mod_policy': '', 'policies': {}, 'values': {}, 've
rsion': '0'}}}}}'
+ configtxlator proto_encode --input config_update_in_envelope.json --type common.Envelope --output
Org2MSPanchors.tx
2024-02-14 06:22:01.434 UTC 0001 INFO [channelCmd] InitCmdFactory -> Endorser and orderer connectio
ns initialized
2024-02-14 06:22:01.501 UTC 0002 INFO [channelCmd] update -> Successfully submitted channel update
Anchor peer set for org 'Org2MSP' on channel 'mychannel'
Channel 'mychannel' joined
```

## Shutting down the network

```
→ test-network git:(main) ./network.sh down
Using docker and docker compose
Stopping network
[+] Running 12/12
 ✓ Container orderer.example.com      Removed      2.6s
 ✓ Container cli                       Removed      1.5s
 ✓ Container peer0.org1.example.com    Removed      2.5s
 ✓ Container peer0.org2.example.com    Removed      2.0s
 ✓ Volume compose_orderer3.example.com Removed       0.0s
 ✓ Network fabric_test                 Removed      0.5s
 ✓ Volume compose_orderer4.example.com Removed       0.0s
 ✓ Volume compose_peer0.org3.example.com Removed       0.0s
 ✓ Volume compose_orderer2.example.com Removed       0.0s
 ✓ Volume compose_peer0.org1.example.com Removed       0.1s
```