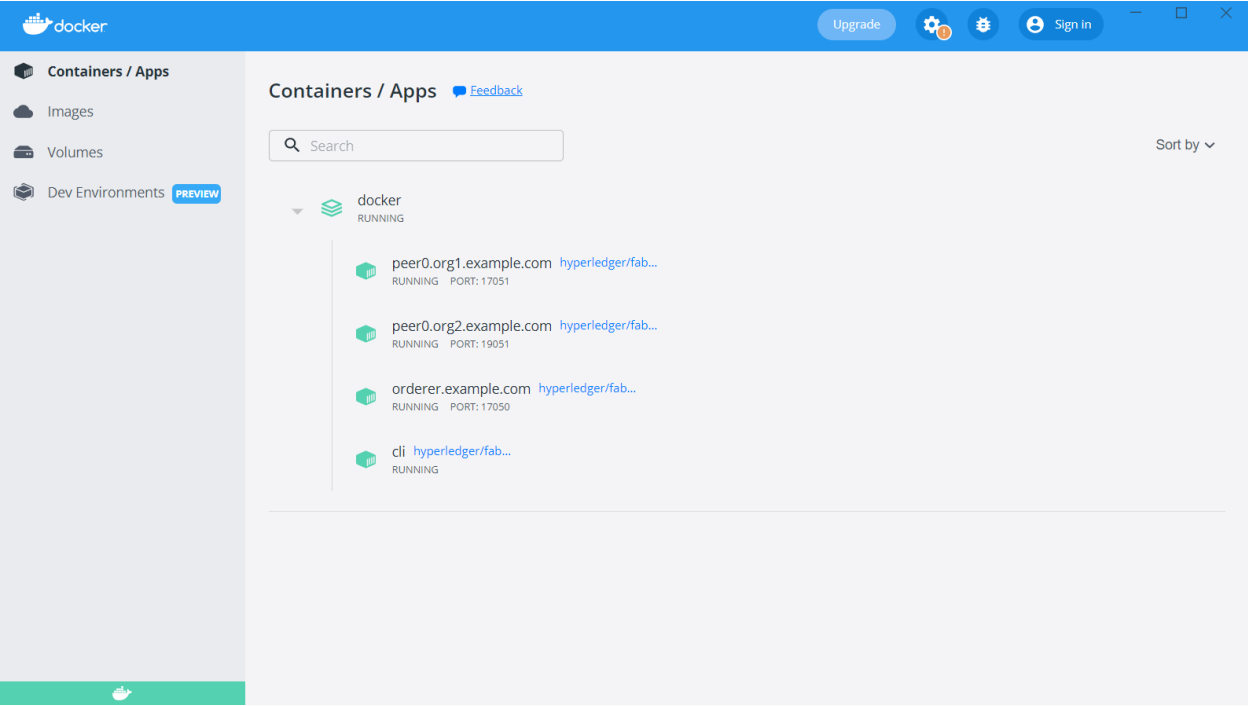


For the HLF installation on WSL I was using this [guide](#) with the exception of installing the docker-compose using common [installation](#) for linux and not one using the python pip as in the guide.

Bring up the test network

```
legion-11@DESKTOP-BT38P00:~$ cd fabric-samples/test-network/
legion-11@DESKTOP-BT38P00:~/fabric-samples/test-network$ ./network.sh up -verbose
Starting nodes with CLI timeout of '5' tries and CLI delay of '3' seconds and using database 'leveldb' with crypto from 'cryptogen'
LOCAL_VERSION=2.2.0
DOCKER_IMAGE_VERSION=2.2.0
/home/legion-11/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
/home/legion-11/fabric-samples/test-network/./bin/configtxgen
Generating Orderer Genesis block
+ configtxgen -profile TwoOrgsOrdererGenesis -channelID system-channel -outputBlock ./system-genesis-block/genesis.block
2022-04-19 16:47:22.039 EDT [common.tools.configtxgen] main -> INFO 001 Loading configuration
2022-04-19 16:47:22.110 EDT [common.tools.configtxgen.localconfig] completeInitialization -> INFO 002 orderer type: etcdraft
2022-04-19 16:47:22.111 EDT [common.tools.configtxgen.localconfig] completeInitialization -> INFO 003 Orderer.Etcdraft.Options unset, setting to tick_interval:"500ms" election_tick:10 heartbeat_t
max inflight blocks:5 snapshot interval size:16777216
2022-04-19 16:47:22.111 EDT [common.tools.configtxgen.localconfig] Load -> INFO 004 Loaded configuration: /home/legion-11/fabric-samples/test-network/configtx/configtx.yaml
2022-04-19 16:47:22.136 EDT [common.tools.configtxgen] doOutputBlock -> INFO 005 Generating genesis block
2022-04-19 16:47:22.138 EDT [common.tools.configtxgen] doOutputBlock -> INFO 006 Writing genesis block
+ res=0
Creating network "fabric-test" with the default driver
Creating volume "docker_orderer.example.com" with default driver
Creating volume "docker_peer0.org1.example.com" with default driver
Creating volume "docker_peer0.org2.example.com" with default driver
Creating peer0.org1.example.com ... done
Creating peer0.org2.example.com ... done
Creating orderer.example.com ... done
Creating cli ... done
CONTAINER ID        IMAGE                                     COMMAND                  CREATED             STATUS              PORTS
b1d6ec1c7c78       hyperledger/fabric-tools:latest        "/bin/bash"             4 seconds ago       Up less than a second    0.0.0.0:7050->7050/tcp, 0.0.0.0:17050->17050/tcp
e369b6d6dce6       hyperledger/fabric-orderer:latest      "orderer"               9 seconds ago       Up 3 seconds           0.0.0.0:19051->19051/tcp, 7051/tcp, 0.0.0.0:19051->19051/tcp
3f64791fa233       hyperledger/fabric-peer:latest         "peer node start"       9 seconds ago       Up 4 seconds           0.0.0.0:7051->7051/tcp, 0.0.0.0:17051->17051/tcp
0a96aaf4dddf       hyperledger/fabric-peer:latest         "peer node start"       9 seconds ago       Up 4 seconds           0.0.0.0:7051->7051/tcp, 0.0.0.0:17051->17051/tcp
legion-11@DESKTOP-BT38P00:~/fabric-samples/test-network$
```



Creating channel

[illegible]

Starting a chaincode on the channel

```
legion-11@DESKTOP-BT38P08:~/fabric-samples/test-network$ ./network.sh deployCC -ccn basic -ccp ../asset-transfer-basic/chaincode-go -ccl go
deploying chaincode on channel 'mychannel'
executing with the following
CHANNEL_NAME: mychannel
CC_NAME: basic
CC_SRC_PATH: ../asset-transfer-basic/chaincode-go
CC_SRC_LANGUAGE: go
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
Vendoring Go dependencies at ../asset-transfer-basic/chaincode-go
~/fabric-samples/asset-transfer-basic/chaincode-go ~/fabric-samples/test-network
go: downloading github.com/hyperledger/fabric-contract-api-go v1.1.1
go: downloading github.com/hyperledger/fabric-chaincode-go v0.0.0-20210718160520-38d29fabecb9
go: downloading github.com/hyperledger/fabric-protos-go v0.0.0-20201028172856-a3136dde2354
go: downloading github.com/stretchr/testify v1.5.1
go: downloading github.com/golang/protobuf v1.3.2
go: downloading google.golang.org/grpc v1.23.0
go: downloading github.com/xetpuuv/gojsonschema v1.2.0
go: downloading github.com/go-openapi/spec v0.19.4
go: downloading github.com/gobuffalo/packr v1.30.1
go: downloading github.com/davecgh/go-spew v1.1.1
go: downloading github.com/pmezard/go-difflib v1.0.0
go: downloading gopkg.in/yaml.v2 v2.2.8
go: downloading google.golang.org/genproto v0.0.0-20180831171423-11092d34479b
go: downloading golang.org/x/net v0.0.0-20190827160401-ba9fcec4b297
go: downloading github.com/xetpuuv/gojsonreference v0.0.0-20180127040603-bd5ef7bd5415
go: downloading github.com/gobuffalo/envy v1.7.0
go: downloading github.com/gobuffalo/packd v0.3.0
go: downloading github.com/go-openapi/jsonpointer v0.19.3
go: downloading github.com/go-openapi/jsonreference v0.19.2
go: downloading github.com/go-openapi/swag v0.19.5
go: downloading golang.org/x/sys v0.0.0-20190710143415-6ec70d6a5542
go: downloading github.com/xetpuuv/gojsonpointer v0.0.0-20180127040702-4e3ac2762d5f
go: downloading github.com/joho/godotenv v1.3.0
go: downloading github.com/rogpeppe/go-internal v1.3.0
go: downloading github.com/puerkitoBio/purell v1.1.1
go: downloading github.com/mailru/easyjson v0.0.0-20190626092158-b2ccc519800e
go: downloading github.com/puerkitoBio/urlesc v0.0.0-20170810143723-de5bf2ad4578
go: downloading golang.org/x/text v0.3.2
~/fabric-samples/test-network
Finished vendoring Go dependencies
+ peer lifecycle chaincode package basic.tar.gz --path ../asset-transfer-basic/chaincode-go --lang golang --label basic_1.0
+ res=0
Chaincode is packaged
Installing chaincode on peer0.org1...
Using organization 1
+ peer lifecycle chaincode install basic.tar.gz
+ res=0
2022-04-19 16:57:24.260 EDT [cli.lifecycle.chaincode] submitInstallProposal -> INFO 001 Installed remotely: response:<status:200 payload:"\nbasic_1.0:3cf6f67978d6b3f7c5e0375660c995b21db19c4330946079af
c3925ad7306881022\tbasic_1.0">
2022-04-19 16:57:24.273 EDT [cli.lifecycle.chaincode] submitInstallProposal -> INFO 002 Chaincode code package identifier: basic_1.0:3cf6f67978d6b3f7c5e0375660c995b21db19c4330946079afc3925ad7306881
Chaincode is installed on peer0.org1
Install chaincode on peer0.org2...
Using organization 2
+ peer lifecycle chaincode install basic.tar.gz
+ res=0
2022-04-19 16:58:49.732 EDT [cli.lifecycle.chaincode] submitInstallProposal -> INFO 001 Installed remotely: response:<status:200 payload:"\nbasic_1.0:3cf6f67978d6b3f7c5e0375660c995b21db19c4330946079af
c3925ad7306881022\tbasic_1.0">
2022-04-19 16:58:49.746 EDT [cli.lifecycle.chaincode] submitInstallProposal -> INFO 002 Chaincode code package identifier: basic_1.0:3cf6f67978d6b3f7c5e0375660c995b21db19c4330946079afc3925ad7306881
Chaincode is installed on peer0.org2
Using organization 1
+ peer lifecycle chaincode queryinstalled
+ res=0
Installed chaincodes on peer:
Package ID: basic_1.0:3cf6f67978d6b3f7c5e0375660c995b21db19c4330946079afc3925ad7306881, Label: basic_1.0
Query installed successful on peer0.org1 on channel
Using organization 1
+ peer lifecycle chaincode approvefromorg -o localhost:7050 --ordererTlsHostnameOverride orderer.example.com --tls --cafile /home/legion-11/fabric-samples/test-network/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlsacerts/tlsca.example.com-cert.pem --channelID mychannel --name basic --version 1.0 --package-id basic_1.0:3cf6f67978d6b3f7c5e0375660c995b21db19c43309
46079afc3925ad7306881 --sequence 1
+ res=0
2022-04-19 16:58:55.032 EDT [chaincodeCmd] ClientWait -> INFO 001 txid [da73e472ab74edc808e5a9f286ab5f8ef0af93117dacc444beca84857753d51] committed with status (VALID) at
Chaincode definition approved on peer0.org1 on channel 'mychannel'
Using organization 1
Checking the commit readiness of the chaincode definition on peer0.org1 on channel 'mychannel'...
Attempting to check the commit readiness of the chaincode definition on peer0.org1, Retry after 3 seconds.
+ peer lifecycle chaincode checkcommitreadiness --channelID mychannel --name basic --version 1.0 --sequence 1 --output json
+ res=0
{
  "approvals": {
    "Org1MSP": true,
    "Org2MSP": false
  }
}
```



```

Checking the commit readiness of the chaincode definition successful on peer0.org1 on channel 'mychannel'
Using organization 2
Checking the commit readiness of the chaincode definition on peer0.org2 on channel 'mychannel'...
Attempting to check the commit readiness of the chaincode definition on peer0.org2, Retry after 3 seconds.
+ peer lifecycle chaincode checkcommitreadiness --channelID mychannel --name basic --version 1.0 --sequence 1 --output json
+ res=0
{
  "approvals": {
    "Org1MSP": true,
    "Org2MSP": false
  }
}

Checking the commit readiness of the chaincode definition successful on peer0.org2 on channel 'mychannel'
Using organization 2
+ peer lifecycle chaincode approveformyorg -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile /home/legion-11/fabric-samples/test-network/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem --channelID mychannel --name basic --version 1.0 --package-id basic_1.0:3cf67978db63f7c5e0375660c995b21db19c4330946079afc3925ad7306881 --sequence 1
+ res=0
2022-04-19 16:59:05.265 EDT [chaincodeCmd] ClientWait -> INFO 001 txid [c7e4248a817f83e360370a37ce81d16e162cbc4f89f0951ab24e9729e40348a] committed with status (VALID) at
chaincode definition approved on peer0.org2 on channel 'mychannel'
Using organization 1
Checking the commit readiness of the chaincode definition on peer0.org1 on channel 'mychannel'...
Attempting to check the commit readiness of the chaincode definition on peer0.org1, Retry after 3 seconds.
+ peer lifecycle chaincode checkcommitreadiness --channelID mychannel --name basic --version 1.0 --sequence 1 --output json
+ res=0
{
  "approvals": {
    "Org1MSP": true,
    "Org2MSP": true
  }
}

Checking the commit readiness of the chaincode definition successful on peer0.org1 on channel 'mychannel'
Using organization 2
Checking the commit readiness of the chaincode definition on peer0.org2 on channel 'mychannel'...
Attempting to check the commit readiness of the chaincode definition on peer0.org2, Retry after 3 seconds.
+ peer lifecycle chaincode checkcommitreadiness --channelID mychannel --name basic --version 1.0 --sequence 1 --output json
+ res=0
{
  "approvals": {
    "Org1MSP": true,
    "Org2MSP": true
  }
}

Checking the commit readiness of the chaincode definition successful on peer0.org2 on channel 'mychannel'
Using organization 1
+ peer lifecycle chaincode commit -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile /home/legion-11/fabric-samples/test-network/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem --channelID mychannel --name basic --peerAddresses localhost:7051 --tlsRootCertFiles /home/legion-11/fabric-samples/test-network/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt --peerAddresses localhost:9051 --tlsRootCertFiles /home/legion-11/fabric-samples/test-network/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt --version 1.0 --sequence 1
+ res=0
2022-04-19 16:59:17.019 EDT [chaincodeCmd] ClientWait -> INFO 001 txid [c4dac8c2137562ca13970585f2a9f87cbdd0cd12952eb50155c96198b8c986] committed with status (VALID) at localhost:9051
2022-04-19 16:59:17.339 EDT [chaincodeCmd] ClientWait -> INFO 002 txid [c4dac8c2137562ca13970585f2a9f87cbdd0cd12952eb50155c96198b8c986] committed with status (VALID) at localhost:7051
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 basic
+ res=0
Committed chaincode definition for chaincode 'basic' on channel 'mychannel':
Version: 1.0, Sequence: 1, Endorsement Plugin: escv, Validation Plugin: vscv, 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 basic
+ res=0
Committed chaincode definition for chaincode 'basic' on channel 'mychannel':
Version: 1.0, Sequence: 1, Endorsement Plugin: escv, Validation Plugin: vscv, Approvals: [Org1MSP: true, Org2MSP: true]
Query chaincode definition successful on peer0.org2 on channel 'mychannel'
Chaincode initialization is not required
legion-11@DESKTOP-BT7J8P08:~/fabric-samples/test-network$

```

Interacting with the network

Adding asset by organization 1

```

legion-11@DESKTOP-BT7J8P08:~/fabric-samples/test-network$ export PATH=${PWD}/../bin:$PATH
legion-11@DESKTOP-BT7J8P08:~/fabric-samples/test-network$ export FABRIC_CFG_PATH=${PWD}/../config/
legion-11@DESKTOP-BT7J8P08:~/fabric-samples/test-network$ export CORE_PEER_TLS_ENABLED=true
legion-11@DESKTOP-BT7J8P08:~/fabric-samples/test-network$ export CORE_PEER_LOCALMSPID="Org1MSP"
legion-11@DESKTOP-BT7J8P08:~/fabric-samples/test-network$ export CORE_PEER_TLS_ROOTCERT_FILE=${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt
legion-11@DESKTOP-BT7J8P08:~/fabric-samples/test-network$ export CORE_PEER_MSPCONFIGPATH=${PWD}/organizations/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp
legion-11@DESKTOP-BT7J8P08:~/fabric-samples/test-network$ peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile ${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem -c mychannel -n basic --peerAddresses localhost:7051 --tlsRootCertFiles ${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt --peerAddresses localhost:9051 --tlsRootCertFiles ${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt -c '{"function": "InitLedger", "Args": []}'
2022-04-19 17:09:46.710 EDT [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 001 Chaincode invoke successful. result: status:200
legion-11@DESKTOP-BT7J8P08:~/fabric-samples/test-network$ peer chaincode query -c mychannel -n basic -c '{"Args":["GetAllAssets"]}'
[{"ID": "asset1", "color": "blue", "size": 15, "owner": "tomoko", "appraisedValue": 300}, {"ID": "asset2", "color": "red", "size": 15, "owner": "Brad", "appraisedValue": 400}, {"ID": "asset3", "color": "green", "size": 10, "owner": "Jin Soo", "appraisedValue": 500}, {"ID": "asset4", "color": "yellow", "size": 10, "owner": "Max", "appraisedValue": 600}, {"ID": "asset5", "color": "black", "size": 15, "owner": "Adriana", "appraisedValue": 700}, {"ID": "asset6", "color": "white", "size": 15, "owner": "Michel", "appraisedValue": 800}]
legion-11@DESKTOP-BT7J8P08:~/fabric-samples/test-network$ peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile ${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem -c mychannel -n basic --peerAddresses localhost:7051 --tlsRootCertFiles ${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt --peerAddresses localhost:9051 --tlsRootCertFiles ${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt -c '{"function": "TransferAsset", "Args":["asset6","Christopher"]}'
2022-04-19 17:10:06.102 EDT [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 001 Chaincode invoke successful. result: status:200
legion-11@DESKTOP-BT7J8P08:~/fabric-samples/test-network$ peer chaincode query -c mychannel -n basic -c '{"Args":["GetAllAssets"]}'
[{"ID": "asset1", "color": "blue", "size": 15, "owner": "tomoko", "appraisedValue": 300}, {"ID": "asset2", "color": "red", "size": 15, "owner": "Brad", "appraisedValue": 400}, {"ID": "asset3", "color": "green", "size": 10, "owner": "Jin Soo", "appraisedValue": 500}, {"ID": "asset4", "color": "yellow", "size": 10, "owner": "Max", "appraisedValue": 600}, {"ID": "asset5", "color": "black", "size": 15, "owner": "Adriana", "appraisedValue": 700}, {"ID": "asset6", "color": "white", "size": 15, "owner": "Christopher", "appraisedValue": 800}]

```

Query asset by organization 2

```

legion-11@DESKTOP-BT7J8P08:~/fabric-samples/test-network$ export CORE_PEER_TLS_ENABLED=true
legion-11@DESKTOP-BT7J8P08:~/fabric-samples/test-network$ export CORE_PEER_LOCALMSPID="Org2MSP"
legion-11@DESKTOP-BT7J8P08:~/fabric-samples/test-network$ export CORE_PEER_TLS_ROOTCERT_FILE=${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt
legion-11@DESKTOP-BT7J8P08:~/fabric-samples/test-network$ export CORE_PEER_MSPCONFIGPATH=${PWD}/organizations/peerOrganizations/org2.example.com/users/Admin@org2.example.com/msp
legion-11@DESKTOP-BT7J8P08:~/fabric-samples/test-network$ export CORE_PEER_ADDRESS=localhost:9051
legion-11@DESKTOP-BT7J8P08:~/fabric-samples/test-network$ peer chaincode query -c mychannel -n basic -c '{"Args":["ReadAsset","asset6"]}'
{"ID": "asset6", "color": "white", "size": 15, "owner": "Christopher", "appraisedValue": 800}
legion-11@DESKTOP-BT7J8P08:~/fabric-samples/test-network$

```

Bring up the test network manually

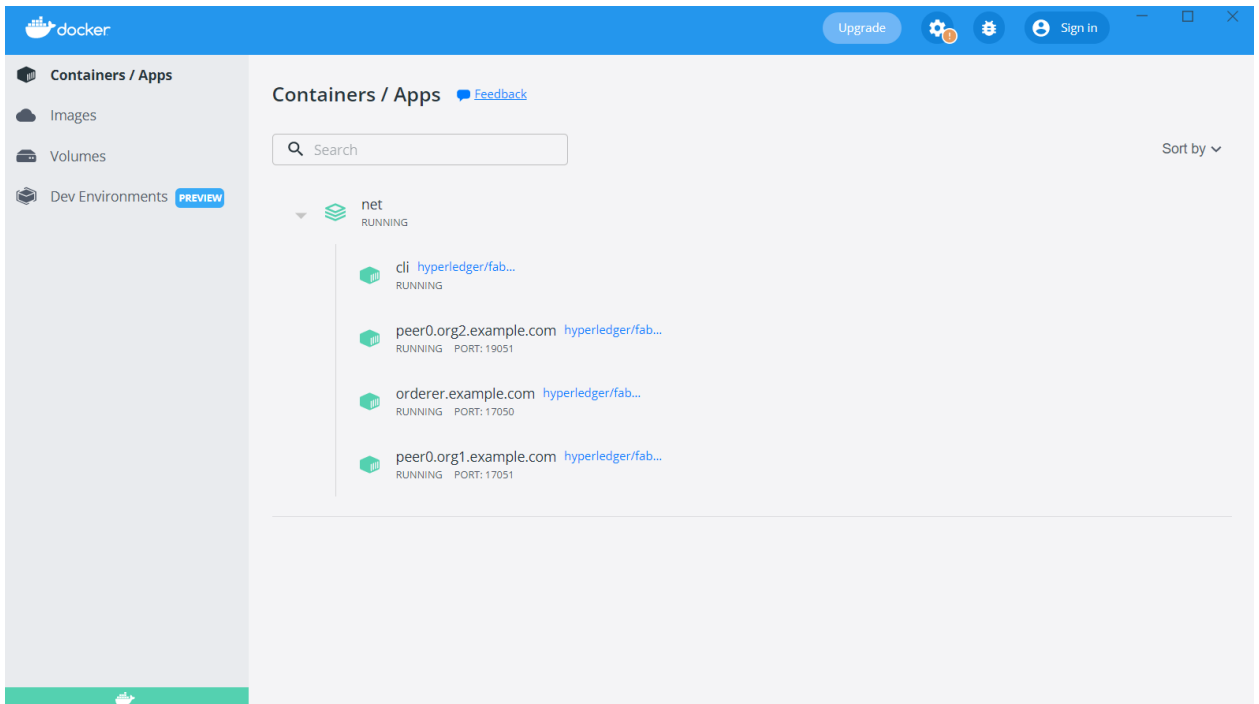
```
network.sh deployCC -cn mychaincode -ccp /user/mychaincode -ccv 1 -ccf javascript
legion-11@DESKTOP-BTJ8P00:~/new/network$ ./network.sh down
Stopping network
Removing network fabric_test
WARNING: Network fabric_test not found.
Removing volume net_orderer.example.com
WARNING: Volume net_orderer.example.com not found.
Removing volume net_peer0.org1.example.com
WARNING: Volume net_peer0.org1.example.com not found.
Removing volume net_peer0.org2.example.com
WARNING: Volume net_peer0.org2.example.com not found.
Removing network fabric_test
WARNING: Network fabric_test not found.
Removing volume net_peer0.org3.example.com
WARNING: Volume net_peer0.org3.example.com not found.
No containers available for deletion
No images available for deletion
```

Remove the old materials

```
legion-11@DESKTOP-BTJ8P00:~$ sudo rm -fr new/network/organizations/ordererOrganizations/*
peerOrganizations/*
sudo rm -fr new/network/system-genesis-block/*[sudo] password for legion-11:
!!Sorry, try again.
[sudo] password for legion-11:
legion-11@DESKTOP-BTJ8P00:~$ sudo rm -fr new/network/organizations/ordererOrganizations/*
genesis-block/*legion-11@DESKTOP-BTJ8P00:~$ sudo rm -fr new/network/organizations/peerOrganizations/*
legion-11@DESKTOP-BTJ8P00:~$ sudo rm -fr new/network/system-genesis-block/*
legion-11@DESKTOP-BTJ8P00:~$ cd new
```

Generate crypto materials, create the genesis block, and deploy the network

```
legion-11@DESKTOP-BTJ8P00:~/new$ cd network/
legion-11@DESKTOP-BTJ8P00:~/new/network$ cryptogen generate --config=./organizations/cryptogen/crypto-config-org1.yaml --output="organizations"
ptogen/crypto-config-org2.yaml --output="organizations"
cryptogen generate --config=./organizations/cryptogen/crypto-config-orderer.yaml --output="organizations"org1.example.com
legion-11@DESKTOP-BTJ8P00:~/new/network$ cryptogen generate --config=./organizations/cryptogen/crypto-config-org2.yaml --output="organizations"
org2.example.com
legion-11@DESKTOP-BTJ8P00:~/new/network$ cryptogen generate --config=./organizations/cryptogen/crypto-config-orderer.yaml --output="organizations"
legion-11@DESKTOP-BTJ8P00:~/new/network$ export FABRIC_CFG_PATH=$PWD/configtx/
legion-11@DESKTOP-BTJ8P00:~/new/network$ configtxgen -profile TwoOrgsOrdererGenesis -channelID system-channel -outputBlock ./system-genesis-block/genes
is.block
2022-04-19 21:07:37.051 EDT [common.tools.configtxgen] main -> INFO 001 Loading configuration
2022-04-19 21:07:37.132 EDT [common.tools.configtxgen.localconfig] completeInitialization -> INFO 002 orderer type: etcdraft
2022-04-19 21:07:37.132 EDT [common.tools.configtxgen.localconfig] completeInitialization -> INFO 003 Orderer.EtcdRaft.Options unset, setting to tick_i
nterval:"500ms" election_tick:10 heartbeat_tick:1 max_inflight_blocks:5 snapshot_interval_size:16777216
2022-04-19 21:07:37.133 EDT [common.tools.configtxgen.localconfig] Load -> INFO 004 Loaded configuration: /home/legion-11/new/network/configtx/configtx
.yaml
2022-04-19 21:07:37.142 EDT [common.tools.configtxgen] doOutputBlock -> INFO 005 Generating genesis block
2022-04-19 21:07:37.145 EDT [common.tools.configtxgen] doOutputBlock -> INFO 006 Writing genesis block
legion-11@DESKTOP-BTJ8P00:~/new/network$ touch .env
legion-11@DESKTOP-BTJ8P00:~/new/network$ cp .env ./docker
legion-11@DESKTOP-BTJ8P00:~/new/network$ cd ~/new/network/docker
legion-11@DESKTOP-BTJ8P00:~/new/network/docker$ docker-compose -f docker-compose-test-net.yaml up -d
Creating network "fabric_test" with the default driver
Creating volume "net_orderer.example.com" with default driver
Creating volume "net_peer0.org1.example.com" with default driver
Creating volume "net_peer0.org2.example.com" with default driver
Creating peer0.org1.example.com ... done
Creating orderer.example.com ... done
Creating peer0.org2.example.com ... done
Creating cli ... done
legion-11@DESKTOP-BTJ8P00:~/new/network/docker$ █
```



The orderer system channel

```
legion-11@DESKTOP-BT3BP08: ~/new/network/docker$ cd ..
legion-11@DESKTOP-BT3BP08: ~/new/network$ export FABRIC_CFG_PATH=$(PWD)/configtx
legion-11@DESKTOP-BT3BP08: ~/new/network$ configtxgen -profile TwoOrgsOrdererGenesis -channelID system-channel -outputBlock ./system-genesis-block/genesis.block
2022-04-19 21:18:32.535 EDT [common.tools.configtxgen] main -> INFO 001 Loading configuration
2022-04-19 21:18:32.676 EDT [common.tools.configtxgen.localconfig] completeInitialization -> INFO 002 orderer type: etcdraft
2022-04-19 21:18:32.677 EDT [common.tools.configtxgen.localconfig] completeInitialization -> INFO 003 orderer.Etcdraft.Options unset, setting to tickInterval:"500ms" election_tick:10 heartbeat_tick:1
maxInflightBlocks:5 snapshotInterval:10777216
2022-04-19 21:18:32.677 EDT [common.tools.configtxgen] load -> INFO 004 Loaded configuration: /home/legion-11/new/network/configtx/configtx.yaml
2022-04-19 21:18:32.790 EDT [common.tools.configtxgen] doOutputBlock -> INFO 005 Generating genesis block
2022-04-19 21:18:32.793 EDT [common.tools.configtxgen] doOutputBlock -> INFO 006 Writing genesis block
```

Creating an application channel

```
legion-11@DESKTOP-BT3BP08: ~/new/network$ export FABRIC_CFG_PATH=$(PWD)/../config/
legion-11@DESKTOP-BT3BP08: ~/new/network$ export CORE_PEER_TLS_ENABLED=true
legion-11@DESKTOP-BT3BP08: ~/new/network$ export CORE_PEER_LOCALMSPID="Org1MSP"
legion-11@DESKTOP-BT3BP08: ~/new/network$ export CORE_PEER_TLS_ROOTCERT_FILE=$(PWD)/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt
legion-11@DESKTOP-BT3BP08: ~/new/network$ export CORE_PEER_MSPCONFIGPATH=$(PWD)/organizations/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp
legion-11@DESKTOP-BT3BP08: ~/new/network$ export CORE_PEER_ADDRESS=localhost:7051
legion-11@DESKTOP-BT3BP08: ~/new/network$ peer channel create -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com -c channel1 -f ./channel-artifacts/channel1.tx --outputBlock ./channel-artifacts/channel1.block --tls --cafile $(PWD)/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem
2022-04-19 21:20:52.926 EDT [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2022-04-19 21:20:53.471 EDT [cli.common] readBlock -> INFO 002 Expect block, but got status: &(NOT_FOUND)
2022-04-19 21:20:53.527 EDT [channelCmd] InitCmdFactory -> INFO 003 Endorser and orderer connections initialized
2022-04-19 21:20:53.736 EDT [cli.common] readBlock -> INFO 004 Expect block, but got status: &(NOT_FOUND)
2022-04-19 21:20:53.760 EDT [channelCmd] InitCmdFactory -> INFO 005 Endorser and orderer connections initialized
2022-04-19 21:20:55.987 EDT [cli.common] readBlock -> INFO 006 Expect block, but got status: &(SERVICE_UNAVAILABLE)
2022-04-19 21:20:56.017 EDT [channelCmd] InitCmdFactory -> INFO 007 Endorser and orderer connections initialized
2022-04-19 21:20:56.226 EDT [cli.common] readBlock -> INFO 008 Expect block, but got status: &(SERVICE_UNAVAILABLE)
2022-04-19 21:20:56.253 EDT [channelCmd] InitCmdFactory -> INFO 009 Endorser and orderer connections initialized
2022-04-19 21:20:56.474 EDT [cli.common] readBlock -> INFO 00a Expect block, but got status: &(SERVICE_UNAVAILABLE)
2022-04-19 21:20:56.583 EDT [channelCmd] InitCmdFactory -> INFO 00b Endorser and orderer connections initialized
2022-04-19 21:20:56.707 EDT [cli.common] readBlock -> INFO 00c Expect block, but got status: &(SERVICE_UNAVAILABLE)
2022-04-19 21:20:56.725 EDT [channelCmd] InitCmdFactory -> INFO 00d Endorser and orderer connections initialized
2022-04-19 21:20:56.942 EDT [cli.common] readBlock -> INFO 00e Expect block, but got status: &(SERVICE_UNAVAILABLE)
2022-04-19 21:20:56.961 EDT [channelCmd] InitCmdFactory -> INFO 00f Endorser and orderer connections initialized
2022-04-19 21:20:57.191 EDT [cli.common] readBlock -> INFO 010 received block: 0
```

Join peers to the channel

```
legion-11@DESKTOP-BT3BP08: ~/new/network$ peer channel join -b ./channel-artifacts/channel.block
2022-04-19 21:22:41.483 EDT [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2022-04-19 21:22:43.264 EDT [channelCmd] executeJoin -> INFO 002 Successfully submitted proposal to join channel
legion-11@DESKTOP-BT3BP08: ~/new/network$ peer channel getinfo -c channel1
2022-04-19 21:22:51.442 EDT [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
blockchain info: {"height":1,"currentBlockhash":"g4db5f9kypg/HY4R/GtsITk8B85CESZAXNne03dM8+"}
legion-11@DESKTOP-BT3BP08: ~/new/network$ export CORE_PEER_TLS_ENABLED=true
legion-11@DESKTOP-BT3BP08: ~/new/network$ export CORE_PEER_LOCALMSPID="Org2MSP"
legion-11@DESKTOP-BT3BP08: ~/new/network$ export CORE_PEER_TLS_ROOTCERT_FILE=$(PWD)/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt
legion-11@DESKTOP-BT3BP08: ~/new/network$ export CORE_PEER_MSPCONFIGPATH=$(PWD)/organizations/peerOrganizations/org2.example.com/users/Admin@org2.example.com/msp
legion-11@DESKTOP-BT3BP08: ~/new/network$ export CORE_PEER_ADDRESS=localhost:9051
legion-11@DESKTOP-BT3BP08: ~/new/network$ peer channel fetch 0 ./channel-artifacts/channel.org2.block -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com -c channel1 --tls --cafile $(PWD)/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem
2022-04-19 21:23:15.119 EDT [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2022-04-19 21:23:15.176 EDT [cli.common] readBlock -> INFO 002 received block: 0
legion-11@DESKTOP-BT3BP08: ~/new/network$ peer channel join -b ./channel-artifacts/channel.org2.block
2022-04-19 21:23:20.300 EDT [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2022-04-19 21:23:23.110 EDT [channelCmd] executeJoin -> INFO 002 Successfully submitted proposal to join channel
legion-11@DESKTOP-BT3BP08: ~/new/network$
```


Set anchor peers

```
legion-11@DESKTOP-BT73P00:~/new/network$ export FABRIC_CFG_PATH=$(pwd)/../config/
legion-11@DESKTOP-BT73P00:~/new/network$ export CORE_PEER_TLS_ENABLED=true
legion-11@DESKTOP-BT73P00:~/new/network$ export CORE_PEER_LOCALMSPID="Org1esp"
legion-11@DESKTOP-BT73P00:~/new/network$ export CORE_PEER_TLS_ROOTCERT_FILE=$(pwd)/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt
legion-11@DESKTOP-BT73P00:~/new/network$ export CORE_PEER_MSPCONFIGPATH=$(pwd)/organizations/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp
legion-11@DESKTOP-BT73P00:~/new/network$ export CORE_PEER_ADDRESS=localhost:7051
legion-11@DESKTOP-BT73P00:~/new/network$ peer channel fetch config channel-artifacts/config_block.pb -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com -c channel1 --tls --cafile $(pwd)/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem
2022-04-19 21:24:39.534 EDT [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2022-04-19 21:24:39.565 EDT [cli.common] readBlock -> INFO 002 Received block: 0
2022-04-19 21:24:39.565 EDT [channelCmd] fetch -> INFO 003 Retrieving last config block: 0
2022-04-19 21:24:39.586 EDT [cli.common] readBlock -> INFO 004 Received block: 0
legion-11@DESKTOP-BT73P00:~/new/network$ cd channel-artifacts
legion-11@DESKTOP-BT73P00:~/new/network/channel-artifacts$ configtxlator proto_decode --input config_block.pb --type common.Block --output config_block.json
{ "data":{ "data":[ { "payload":{ "data":{ "config":{ "config_block.json" } } } } } }
legion-11@DESKTOP-BT73P00:~/new/network/channel-artifacts$ cp config_block.json config_copy.json
legion-11@DESKTOP-BT73P00:~/new/network/channel-artifacts$ jq '.channel_group.groups.Application.groups.Org1MSP.values += [{"AnchorPeers":{"mod_policy": "Admins", "value":{"anchor_peers": [{"host": "peer0.org1.example.com", "port": 7051}], "version": "0"} }]' config_copy.json > modified_config.json
legion-11@DESKTOP-BT73P00:~/new/network/channel-artifacts$ configtxlator proto_encode --input config.json --type common.Config --output config.pb
proto_encode --input modified_config.json --type common.Config --output modified_config.pb
configtxlator compute_update --channel_id channel1 --original config.pb --updated modified_config.pb --output config_update.pb
legion-11@DESKTOP-BT73P00:~/new/network/channel-artifacts$ configtxlator proto_encode --input config_update.pb --type common.ConfigUpdate --output config_update.pb
legion-11@DESKTOP-BT73P00:~/new/network/channel-artifacts$ configtxlator proto_decode --input config_update.pb --type common.ConfigUpdate --output config_update.json
{ "channel_id": "channel1", "type": "2", "data": { "config_update": { "cat": "config_update.json" } } } | jq -> config_update_in_envelope.json
configtxlator proto_encode --input config_update_in_envelope.json --type common.Envelope --output config_update_in_envelope.pb
legion-11@DESKTOP-BT73P00:~/new/network/channel-artifacts$ echo '{"payload":{"header":{"channel_header":{"channel_id":"channel1", "type":"2"},"data":{"config_update":{"cat":"config_update.json"}}}}}' | jq -> config_update_in_envelope.json
legion-11@DESKTOP-BT73P00:~/new/network/channel-artifacts$ configtxlator proto_encode --input config_update_in_envelope.pb --type common.Envelope --output config_update_in_envelope.pb
legion-11@DESKTOP-BT73P00:~/new/network/channel-artifacts$ cd ..
legion-11@DESKTOP-BT73P00:~/new/network/channel-artifacts$ configtxlator compute_update --channel_id channel1 --original config.pb --updated modified_config.pb --output config_update.pb
legion-11@DESKTOP-BT73P00:~/new/network/channel-artifacts$ configtxlator proto_encode --input config_update.pb --type common.ConfigUpdate --output config_update.pb
legion-11@DESKTOP-BT73P00:~/new/network/channel-artifacts$ configtxlator proto_decode --input config_update.pb --type common.ConfigUpdate --output config_update.json
{ "channel_id": "channel1", "type": "2", "data": { "config_update": { "cat": "config_update.json" } } } | jq -> config_update_in_envelope.json
legion-11@DESKTOP-BT73P00:~/new/network/channel-artifacts$ configtxlator proto_encode --input config_update_in_envelope.json --type common.Envelope --output config_update_in_envelope.pb
legion-11@DESKTOP-BT73P00:~/new/network/channel-artifacts$ cd ..
legion-11@DESKTOP-BT73P00:~/new/network/channel-artifacts$ peer channel update -f channel-artifacts/config_update_in_envelope.pb -c channel1 -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile $(pwd)/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem
2022-04-19 21:26:21.805 EDT [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2022-04-19 21:26:22.082 EDT [channelCmd] update -> INFO 002 Successfully submitted channel update
legion-11@DESKTOP-BT73P00:~/new/network$ export CORE_PEER_TLS_ENABLED=true
legion-11@DESKTOP-BT73P00:~/new/network$ export CORE_PEER_LOCALMSPID="Org2MSP"
legion-11@DESKTOP-BT73P00:~/new/network$ export CORE_PEER_TLS_ROOTCERT_FILE=$(pwd)/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt
legion-11@DESKTOP-BT73P00:~/new/network$ export CORE_PEER_MSPCONFIGPATH=$(pwd)/organizations/peerOrganizations/org2.example.com/users/Admin@org2.example.com/msp
legion-11@DESKTOP-BT73P00:~/new/network$ export CORE_PEER_ADDRESS=localhost:7051
legion-11@DESKTOP-BT73P00:~/new/network$ peer channel fetch config channel-artifacts/config_block.pb -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com -c channel1 --tls --cafile $(pwd)/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem
2022-04-19 21:27:46.299 EDT [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2022-04-19 21:27:46.410 EDT [cli.common] readBlock -> INFO 002 Received block: 1
2022-04-19 21:27:46.416 EDT [channelCmd] fetch -> INFO 003 Retrieving last config block: 1
2022-04-19 21:27:46.437 EDT [cli.common] readBlock -> INFO 004 Received block: 1
legion-11@DESKTOP-BT73P00:~/new/network$ cd channel-artifacts
legion-11@DESKTOP-BT73P00:~/new/network/channel-artifacts$ configtxlator proto_decode --input config_block.pb --type common.Block --output config_block.json
{ "data":{ "data":[ { "payload":{ "data":{ "config":{ "config_block.json" } } } } } }
legion-11@DESKTOP-BT73P00:~/new/network/channel-artifacts$ cp config_block.json config_copy.json
legion-11@DESKTOP-BT73P00:~/new/network/channel-artifacts$ jq '.channel_group.groups.Application.groups.Org2MSP.values += [{"AnchorPeers":{"mod_policy": "Admins", "value":{"anchor_peers": [{"host": "peer0.org2.example.com", "port": 7051}], "version": "0"} }]' config_copy.json > modified_config.json
legion-11@DESKTOP-BT73P00:~/new/network/channel-artifacts$ configtxlator proto_encode --input config.json --type common.Config --output config.pb
proto_encode --input modified_config.json --type common.Config --output modified_config.pb
configtxlator compute_update --channel_id channel1 --original config.pb --updated modified_config.pb --output config_update.pb
legion-11@DESKTOP-BT73P00:~/new/network/channel-artifacts$ configtxlator proto_encode --input config_update.pb --type common.ConfigUpdate --output config_update.pb
legion-11@DESKTOP-BT73P00:~/new/network/channel-artifacts$ configtxlator proto_decode --input config_update.pb --type common.ConfigUpdate --output config_update.json
{ "channel_id": "channel1", "type": "2", "data": { "config_update": { "cat": "config_update.json" } } } | jq -> config_update_in_envelope.json
legion-11@DESKTOP-BT73P00:~/new/network/channel-artifacts$ configtxlator proto_encode --input config_update_in_envelope.json --type common.Envelope --output config_update_in_envelope.pb
legion-11@DESKTOP-BT73P00:~/new/network/channel-artifacts$ echo '{"payload":{"header":{"channel_header":{"channel_id":"channel1", "type":"2"},"data":{"config_update":{"cat":"config_update.json"}}}}}' | jq -> config_update_in_envelope.json
legion-11@DESKTOP-BT73P00:~/new/network/channel-artifacts$ configtxlator proto_encode --input config_update_in_envelope.pb --type common.Envelope --output config_update_in_envelope.pb
legion-11@DESKTOP-BT73P00:~/new/network/channel-artifacts$ cd ..
legion-11@DESKTOP-BT73P00:~/new/network/channel-artifacts$ peer channel update -f channel-artifacts/config_update_in_envelope.pb -c channel1 -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile $(pwd)/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem
2022-04-19 21:29:11.045 EDT [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2022-04-19 21:29:11.159 EDT [channelCmd] update -> INFO 002 Successfully submitted channel update
legion-11@DESKTOP-BT73P00:~/new/network$ peer channel getinfo -c channel1
2022-04-19 21:29:18.702 EDT [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
Blockchain info: { "height": 3, "currentBlockHash": "c2g2Uoql8Z9LSQ3GvQmU4SHWqnrhB7Tt0Tp3qymA=", "previousBlockHash": "0Fa16rBUYyGPRUjmwEaqIDLtXn2AL/zSooZhmME4qI=" }
legion-11@DESKTOP-BT73P00:~/new/network$ }
```

Bring down the network

```
legion-11@DESKTOP-BT38P08: ~/fabric-samples/test-networks ./network.sh down
Stopping network
Stopping cli ... done
Stopping orderer.example.com ... done
Stopping peer0.org2.example.com ... done
Stopping peer0.org1.example.com ... done
Removing cli ... done
Removing orderer.example.com ... done
Removing peer0.org2.example.com ... done
Removing peer0.org1.example.com ... done
Removing network fabric_test
Removing volume docker_orderer.example.com
Removing volume docker_peer0.org1.example.com
Removing volume docker_peer0.org2.example.com
Removing network fabric_test
WARNING: network fabric_test not found.
Removing volume docker_peer0.org1.example.com
WARNING: Volume docker_peer0.org1.example.com not found.
No containers available for deletion
Untagged: dev-peer0.org2.example.com-basic_1.0-3cfcf67978dbb3f7c5e037566c995b21db19c4330946079afc3925ad7306881-6a8fb30d4a694b406bf4f255c7e84f22faeb3190e416cfb1d742ae192a406805:latest
Deleted: sha256:4d69ccd9170434f7690fb32a66e0994c2d2724ce8b4c033564cae3d87851eb8
Deleted: sha256:e2766330faa7a81ef5cab103b06c7b2a684373bca8bfbdb39137ff2d5b187e0
Deleted: sha256:93055171d074f1307520e3d84c1bcb94954680a315410482c2432511aaac
Deleted: sha256:64148d20b767958c954acbf5dc4c11b9e20c6e1703ef4688674a64ec58521
Untagged: dev-peer0.org1.example.com-basic_1.0-3cfcf67978dbb3f7c5e037566c995b21db19c4330946079afc3925ad7306881-c6c446ce0bcf4e0229bc2660312ba359cb00358628cf199bfc1ae60f85b825a7:latest
Deleted: sha256:66e008e2042ce1eb22bc85571fcd8ff0cada365d763ece21acd6795e9987a6a
Deleted: sha256:5283eb8c37a0b00d41e8901edfbbazc4bda2b678bea0fa9d19b498cd5cd361
Deleted: sha256:8de766076a313d56c2b7185b09451539fdc3716bc15015939437faa12c5aaf4
Deleted: sha256:7ab2ef88ef5dffc3a04af7c932a0fcb22e09307383ad0b30a853ea0819cefe
```

What is the main difference between Hyperledger applications and applications such as Ethereum and Bitcoin

Hyperledger is a permissioned blockchain, transaction processing is performed by predefined users issued by Certificate Authorities. The MSP on the ordering service contains the peer's public key which is then used to verify that the signature attached to the transaction is valid.

In case of Bitcoin and Ethereum used permissionless blockchains: There are no restrictions on the identities of processors, thus everyone can start mining to create blocks.

Hyperledger allows nodes to choose between No-op (no consensus needed) and an agreement protocol (PBFT) whereby two or more parties can agree on a key in such a way that both influence the outcome while Ethereum/Bitcoin uses Proof-of-Work consensus mechanism.

Since HLF is permissioned network, it doesn't necessary require cryptocurrencies for transactions. Thus, *Hyperledger* allows confidential transactions, gives security to make transactions visible to select parties having correct encryption keys.

Briefly describe the process by which a new transaction is recorded on the ledger.

1. Client propose the transaction to the peers
2. Each of these endorsement peers will execute the chain code and generate transaction proposal response and endorsement (signatures).
3. An application (client) receives a sufficient number of signed proposal responses
4. Transaction proposal, proposal response with endorsement are sent to the orderer to order transactions of block.
5. Each peer transaction within a block is validated to ensure that it has been consistently endorsed by all relevant organizations before it is applied to the ledger
6. The copy of the block is sent to all the other peers to update the ledger by committing this block.

Briefly describe how security is achieved in the Fabric framework.

Hyperledger Fabric is a permissioned blockchain where each component and actor has an identity, and policies define access control and governance.

First of all because the network is permissioned, not everyone is allowed to update the blockchain, each of the blockchain actors needs its digital identity provided by Certificate Authority, and they define the permissions over resources and access to information that actors have in a blockchain network.

These identities are verified by the Membership Service Providers because MSP on the ordering service contains the peer's public key which is then used to verify that the signature attached to the transaction is valid.

For members to agree on accepting or rejecting changes to the network the set of policies are used (they are configured for channel). That way members can describe the criteria for adding or removing members from a channel change how blocks are formed, or specify the number of organizations required to endorse a smart contract.

To agree on the transaction ordering the Ordering service nodes are used. They order transactions into blocks and then distribute blocks to connected peers for validation and committing and managed by an administrator of an organization.

What is TLS and what's its use in Fabric? Briefly describe it

Transport Layer Security is a cryptographic protocol designed to provide communications security over a computer network. It is just a common protocol for secure information transfer. In fabric it is used for secure communication between nodes. TLS communication can use both one-way (server only) and two-way (server and client) authentication.

Once the client and server have agreed to use TLS, they negotiate a stateful connection by using a handshaking procedure. The protocols use a handshake with an asymmetric cipher to establish not only cipher settings but also a session-specific shared key with which further communication is encrypted using a symmetric cipher.

During this handshake, the client and server agree on various parameters used to establish the connection's security, main of them are: session keys (exchange of public keys for encryption), digital certificate provided by certificate authority, and hash function that will be used by both parties.

This concludes the handshake and begins the secured connection, which is encrypted and decrypted with the session key until the connection closes.