export PATH=${PWD}/../bin:$PATH

export FABRIC\_CFG\_PATH=${PWD}/configtx

For the test network, the configtxgen tool uses the channel profiles that are defined in the configtxt\configtx.yaml file to create the channel configuration and write it to the [protobuf format](https://developers.google.com/protocol-buffers) that can be read by Fabric.

This configtx.yaml file contains the following information that we will use to create our new channel:

configtxgen -profile TwoOrgsOrdererGenesis -channelID system-channel -outputBlock ./system-genesis-block/genesis.block

creating application channel

configtxgen -profile TwoOrgsChannel -outputCreateChannelTx ./channel-artifacts/mychannel.tx -channelID mychannel

export FABRIC\_CFG\_PATH=$PWD/../config/

Admin Credentials of org1

export CORE\_PEER\_TLS\_ENABLED=true

export CORE\_PEER\_LOCALMSPID="Org1MSP"

export CORE\_PEER\_TLS\_ROOTCERT\_FILE=${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt

export CORE\_PEER\_MSPCONFIGPATH=${PWD}/organizations/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp

export CORE\_PEER\_ADDRESS=localhost:7051

Creating channel

peer channel create -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com -c mychannel -f ./channel-artifacts/mychannel.tx --outputBlock ./channel-artifacts/mychannel.block --tls --cafile "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem"

Joining Channel from org1

peer channel join -b ./channel-artifacts/mychannel.block

Joining channel from another org2

Setting up the org2 env

export CORE\_PEER\_TLS\_ENABLED=true

export CORE\_PEER\_LOCALMSPID="Org2MSP"

export CORE\_PEER\_TLS\_ROOTCERT\_FILE=${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt

export CORE\_PEER\_MSPCONFIGPATH=${PWD}/organizations/peerOrganizations/org2.example.com/users/Admin@org2.example.com/msp

export CORE\_PEER\_ADDRESS=localhost:9051

Fetching genesis block

peer channel fetch 0 ./channel-artifacts/channel\_org2.block -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com -c mychannel --tls --cafile "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem"

peer channel join -b ./channel-artifacts/channel\_org2.block

## Setting up the anchor peer

export FABRIC\_CFG\_PATH=$PWD/../config/

export CORE\_PEER\_TLS\_ENABLED=true

export CORE\_PEER\_LOCALMSPID="Org1MSP"

export CORE\_PEER\_TLS\_ROOTCERT\_FILE=${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt

export CORE\_PEER\_MSPCONFIGPATH=${PWD}/organizations/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp

export CORE\_PEER\_ADDRESS=localhost:7051

peer channel fetch config channel-artifacts/config\_block.pb -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com -c mychannel --tls --cafile "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem"

cd channel-artifacts

configtxlator proto\_decode --input config\_block.pb --type common.Block --output config\_block.json

jq '.data.data[0].payload.data.config' config\_block.json > config.json

These commands convert the channel configuration block into a streamlined JSON, config.json, that will serve as the baseline for our update. Because we don’t want to edit this file directly, we will make a copy that we can edit. We will use the original channel config in a future step

cp config.json config\_copy.json

You can use the jq tool to add the Org1 anchor peer to the channel configuration

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

After this step, we have an updated version of channel configuration in JSON format in the modified\_config.json file. We can now convert both the original and modified channel configurations back into protobuf format and calculate the difference between them.

proto\_encode --input config.json --type common.Config --output config.pb

configtxlator proto\_encode --input modified\_config.json --type common.Config --output modified\_config.pb

configtxlator compute\_update --channel\_id mychannel --original config.pb --updated modified\_config.pb --output config\_update.pb

The new protobuf named channel\_update.pb contains the anchor peer update that we need to apply to the channel configuration. We can wrap the configuration update in a transaction envelope to create the channel configuration update transaction.

configtxlator proto\_decode --input config\_update.pb --type common.ConfigUpdate --output config\_update.json

echo '{"payload":{"header":{"channel\_header":{"channel\_id":"mychannel", "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

cd ..

We can add the anchor peer by providing the new channel configuration to the peer channel update command. Because we are updating a section of the channel configuration that only affects Org1, other channel members do not need to approve the channel update.

export CORE\_PEER\_TLS\_ENABLED=true

export CORE\_PEER\_LOCALMSPID="Org1MSP"

export CORE\_PEER\_TLS\_ROOTCERT\_FILE=${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt

export CORE\_PEER\_MSPCONFIGPATH=${PWD}/organizations/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp

export CORE\_PEER\_ADDRESS=localhost:7051

peer channel update -f channel-artifacts/config\_update\_in\_envelope.pb -c mychannel -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"

export CORE\_PEER\_TLS\_ENABLED=true

export CORE\_PEER\_LOCALMSPID="Org2MSP"

export CORE\_PEER\_TLS\_ROOTCERT\_FILE=${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt

export CORE\_PEER\_MSPCONFIGPATH=${PWD}/organizations/peerOrganizations/org2.example.com/users/Admin@org2.example.com/msp

export CORE\_PEER\_ADDRESS=localhost:9051

peer channel fetch config channel-artifacts/config\_block.pb -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com -c mychannel --tls --cafile "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem"

cd channel-artifacts/

configtxlator proto\_decode --input config\_block.pb --type common.Block --output config\_block.json

jq '.data.data[0].payload.data.config' config\_block.json > config.json

cp config.json config\_copy.json

jq '.channel\_group.groups.Application.groups.Org2MSP.values += {"AnchorPeers":{"mod\_policy": "Admins","value":{"anchor\_peers": [{"host": "peer0.org2.example.com","port": 9051}]},"version": "0"}}' config\_copy.json > modified\_config.json

configtxlator proto\_encode --input config.json --type common.Config --output config.pb

configtxlator proto\_encode --input modified\_config.json --type common.Config --output modified\_config.pb

configtxlator compute\_update --channel\_id mychannel --original config.pb --updated modified\_config.pb --output config\_update.pb

configtxlator proto\_decode --input config\_update.pb --type common.ConfigUpdate --output config\_update.json

echo '{"payload":{"header":{"channel\_header":{"channel\_id":"mychannel", "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

cd ..

peer channel update -f channel-artifacts/config\_update\_in\_envelope.pb -c mychannel -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"

peer channel getinfo -c mychannel