# Fabric1.0带有TLS

../bin/cryptogen generate --config=./crypto-config.yaml

export FABRIC\_CFG\_PATH=$PWD

../bin/configtxgen -profile TwoOrgsOrdererGenesis -outputBlock ./channel-artifacts/genesis.block

export CHANNEL\_NAME=mychannel

../bin/configtxgen -profile TwoOrgsChannel -outputCreateChannelTx ./channel-artifacts/channel.tx -channelID $CHANNEL\_NAME

../bin/configtxgen -profile TwoOrgsChannel -outputAnchorPeersUpdate ./channel-artifacts/Org1MSPanchors.tx -channelID $CHANNEL\_NAME -asOrg Org1MSP

../bin/configtxgen -profile TwoOrgsChannel -outputAnchorPeersUpdate ./channel-artifacts/Org2MSPanchors.tx -channelID $CHANNEL\_NAME -asOrg Org2MSP

docker-compose -f docker-compose-cli.yaml up -d

CORE\_PEER\_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp

CORE\_PEER\_ADDRESS=peer0.org1.example.com:7051

CORE\_PEER\_LOCALMSPID="Org1MSP"

CORE\_PEER\_TLS\_ROOTCERT\_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt

docker exec -it cli bash

export CHANNEL\_NAME=mychannel

peer channel create -o orderer.example.com:7050 -c $CHANNEL\_NAME -f ./channel-artifacts/channel.tx --tls $CORE\_PEER\_TLS\_ENABLED --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem

peer channel join -b mychannel.block

peer chaincode install -n mycc -v 1.0 -p github.com/hyperledger/fabric/examples/chaincode/go/chaincode\_example02

peer chaincode instantiate -o orderer.example.com:7050 --tls $CORE\_PEER\_TLS\_ENABLED --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem -C $CHANNEL\_NAME -n mycc -v 1.0 -c '{"Args":["init","a", "100", "b","200"]}' -P "OR ('Org1MSP.member','Org2MSP.member')"

peer chaincode query -C $CHANNEL\_NAME -n mycc -c '{"Args":["query","a"]}'

peer chaincode invoke -o orderer.example.com:7050 --tls $CORE\_PEER\_TLS\_ENABLED --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem -C $CHANNEL\_NAME -n mycc -c '{"Args":["invoke","a","b","10"]}'

# Fabric1.0取消TLS

../bin/cryptogen generate --config=./crypto-config.yaml

export FABRIC\_CFG\_PATH=$PWD

../bin/configtxgen -profile TwoOrgsOrdererGenesis -outputBlock ./channel-artifacts/genesis.block

export CHANNEL\_NAME=mychannel

../bin/configtxgen -profile TwoOrgsChannel -outputCreateChannelTx ./channel-artifacts/channel.tx -channelID $CHANNEL\_NAME

../bin/configtxgen -profile TwoOrgsChannel -outputAnchorPeersUpdate ./channel-artifacts/Org1MSPanchors.tx -channelID $CHANNEL\_NAME -asOrg Org1MSP

../bin/configtxgen -profile TwoOrgsChannel -outputAnchorPeersUpdate ./channel-artifacts/Org2MSPanchors.tx -channelID $CHANNEL\_NAME -asOrg Org2MSP

docker-compose -f docker-compose-cli.yaml up -d

CORE\_PEER\_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp

CORE\_PEER\_ADDRESS=peer0.org1.example.com:7051

CORE\_PEER\_LOCALMSPID="Org1MSP"

CORE\_PEER\_TLS\_ROOTCERT\_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt

docker exec -it cli bash

export CHANNEL\_NAME=mychannel

peer channel create -o orderer.example.com:7050 -c $CHANNEL\_NAME -f ./channel-artifacts/channel.tx --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem

peer channel join -b mychannel.block

peer chaincode install -n mycc -v 1.0 -p github.com/hyperledger/fabric/examples/chaincode/go/chaincode\_example02

peer chaincode instantiate -o orderer.example.com:7050 --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem -C $CHANNEL\_NAME -n mycc -v 1.0 -c '{"Args":["init","a", "100", "b","200"]}' -P "OR ('Org1MSP.member','Org2MSP.member')"

peer chaincode query -C $CHANNEL\_NAME -n mycc -c '{"Args":["query","a"]}'

peer chaincode invoke -o orderer.example.com:7050 --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem -C $CHANNEL\_NAME -n mycc -c '{"Args":["invoke","a","b","10"]}'

# Fabric1.1带有TLS

../bin/cryptogen generate --config=./crypto-config.yaml

export FABRIC\_CFG\_PATH=$PWD

../bin/configtxgen -profile TwoOrgsOrdererGenesis -outputBlock ./channel-artifacts/genesis.block

export CHANNEL\_NAME=mychannel

../bin/configtxgen -profile TwoOrgsChannel -outputCreateChannelTx ./channel-artifacts/channel.tx -channelID $CHANNEL\_NAME

../bin/configtxgen -profile TwoOrgsChannel -outputAnchorPeersUpdate ./channel-artifacts/Org1MSPanchors.tx -channelID $CHANNEL\_NAME -asOrg Org1MSP

../bin/configtxgen -profile TwoOrgsChannel -outputAnchorPeersUpdate ./channel-artifacts/Org2MSPanchors.tx -channelID $CHANNEL\_NAME -asOrg Org2MSP

docker-compose -f docker-compose-cli.yaml up -d

CORE\_PEER\_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp

CORE\_PEER\_ADDRESS=peer0.org1.example.com:7051

CORE\_PEER\_LOCALMSPID="Org1MSP"

CORE\_PEER\_TLS\_ROOTCERT\_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt

docker exec -it cli bash

export CHANNEL\_NAME=mychannel

peer channel create -o orderer.example.com:7050 -c $CHANNEL\_NAME -f ./channel-artifacts/channel.tx --tls $CORE\_PEER\_TLS\_ENABLED --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem

peer channel join -b mychannel.block

peer chaincode install -n mycc -v 1.0 -p github.com/chaincode/chaincode\_example02/go/

peer chaincode instantiate -o orderer.example.com:7050 --tls --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem -C $CHANNEL\_NAME -n mycc -v 1.0 -c '{"Args":["init","a", "100", "b","200"]}' -P "OR ('Org1MSP.peer','Org2MSP.peer')"

peer chaincode query -C $CHANNEL\_NAME -n mycc -c '{"Args":["query","a"]}'

peer chaincode invoke -o orderer.example.com:7050 --tls --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem -C $CHANNEL\_NAME -n mycc -c '{"Args":["invoke","a","b","10"]}'

peer chaincode query -C $CHANNEL\_NAME -n mycc -c '{"Args":["query","a"]}'

CORE\_PEER\_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org2.example.com/users/Admin@org2.example.com/msp

CORE\_PEER\_ADDRESS=peer0.org2.example.com:7051

CORE\_PEER\_LOCALMSPID="Org2MSP"

CORE\_PEER\_TLS\_ROOTCERT\_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt

export CHANNEL\_NAME=orgmychannel

../bin/configtxgen -profile OrgsChannel -outputCreateChannelTx ./channel-artifacts/orgchannel.tx -channelID $CHANNEL\_NAME

docker exec -it cli bash

CORE\_PEER\_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org3.example.com/users/Admin@org3.example.com/msp

CORE\_PEER\_ADDRESS=peer0.org3.example.com:7051

CORE\_PEER\_LOCALMSPID="Org3MSP"

CORE\_PEER\_TLS\_ROOTCERT\_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org3.example.com/peers/peer0.org3.example.com/tls/ca.crt

export CHANNEL\_NAME=mychannel

peer channel create -o orderer.example.com:7050 -c $CHANNEL\_NAME -f ./channel-artifacts/channel.tx --tls $CORE\_PEER\_TLS\_ENABLED --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem

export CHANNEL\_NAME=orgmychannel

peer channel create -o orderer.example.com:7050 -c $CHANNEL\_NAME -f ./channel-artifacts/orgchannel.tx --tls $CORE\_PEER\_TLS\_ENABLED --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem

peer channel join -b orgmychannel.block

peer chaincode install -n mycc -v 1.0 -p github.com/chaincode/chaincode\_example02/go/

peer chaincode instantiate -o orderer.example.com:7050 --tls --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem -C $CHANNEL\_NAME -n mycc -v 1.0 -c '{"Args":["init","a", "300", "b","600"]}' -P "OR ('Org1MSP.peer','Org3MSP.peer')"

peer chaincode query -C $CHANNEL\_NAME -n mycc -c '{"Args":["query","a"]}'

peer chaincode invoke -o orderer.example.com:7050 --tls --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem -C $CHANNEL\_NAME -n mycc -c '{"Args":["invoke","a","b","10"]}'

peer chaincode query -C $CHANNEL\_NAME -n mycc -c '{"Args":["query","a"]}'

# Fabric1.1取消TLS

../bin/cryptogen generate --config=./crypto-config.yaml

export FABRIC\_CFG\_PATH=$PWD

../bin/configtxgen -profile TwoOrgsOrdererGenesis -outputBlock ./channel-artifacts/genesis.block

export CHANNEL\_NAME=mychannel

../bin/configtxgen -profile TwoOrgsChannel -outputCreateChannelTx ./channel-artifacts/channel.tx -channelID $CHANNEL\_NAME

../bin/configtxgen -profile TwoOrgsChannel -outputAnchorPeersUpdate ./channel-artifacts/Org1MSPanchors.tx -channelID $CHANNEL\_NAME -asOrg Org1MSP

../bin/configtxgen -profile TwoOrgsChannel -outputAnchorPeersUpdate ./channel-artifacts/Org2MSPanchors.tx -channelID $CHANNEL\_NAME -asOrg Org2MSP

docker-compose -f docker-compose-cli.yaml up -d

CORE\_PEER\_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp

CORE\_PEER\_ADDRESS=peer0.org1.example.com:7051

CORE\_PEER\_LOCALMSPID="Org1MSP"

CORE\_PEER\_TLS\_ROOTCERT\_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt

docker exec -it cli bash

export CHANNEL\_NAME=mychannel

peer channel create -o orderer.example.com:7050 -c $CHANNEL\_NAME -f ./channel-artifacts/channel.tx --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem

peer channel join -b mychannel.block

peer chaincode install -n mycc -v 1.0 -p github.com/chaincode/chaincode\_example02/go/

peer chaincode instantiate -o orderer.example.com:7050 --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem -C $CHANNEL\_NAME -n mycc -v 1.0 -c '{"Args":["init","a", "100", "b","200"]}' -P "OR ('Org1MSP.peer','Org2MSP.peer')"

peer chaincode query -C $CHANNEL\_NAME -n mycc -c '{"Args":["query","a"]}'

peer chaincode invoke -o orderer.example.com:7050 --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem -C $CHANNEL\_NAME -n mycc -c '{"Args":["invoke","a","b","10"]}'

peer chaincode query -C $CHANNEL\_NAME -n mycc -c '{"Args":["query","a"]}'

# Fabrci-ca服务器搭建

fabric-ca-server init -b admin:adminpw

#fabric-ca-server start -b admin:adminpw

fabric-ca-server start -b admin:adminpw --cfg.affiliations.allowremove --cfg.identities.allowremove

export FABRIC\_CA\_CLIENT\_HOME=$HOME/fabric-ca/clients/admin

fabric-ca-client enroll -u http://admin:adminpw@localhost:7054

#export FABRIC\_CA\_CLIENT\_HOME=$HOME/fabric-ca/clients/admin

#fabric-ca-client register --id.name admin2 --id.type user --id.affiliation org1.department1 --id.attrs hf.Revoker=true

fabric-ca-client affiliation list

fabric-ca-client affiliation remove --force org1

fabric-ca-client affiliation remove --force org2

fabric-ca-client affiliation add com

fabric-ca-client affiliation add com.example

fabric-ca-client affiliation add com.example.org1

fabric-ca-client affiliation add com.example.org2

mkdir -p /root/fabric-ca-files/example.com/msp

fabric-ca-client getcacert -M /root/fabric-ca-files/example.com/msp

mkdir -p /root/fabric-ca-files/org1.example.com/msp

fabric-ca-client getcacert -M /root/fabric-ca-files/org1.example.com/msp

mkdir -p /root/fabric-ca-files/org2.example.com/msp

fabric-ca-client getcacert -M /root/fabric-ca-files/org2.example.com/msp

fabric-ca-client register -H /root/fabric-ca-client-configList/example/ --id.secret=password

mkdir -p /root/fabric-ca-files/example.com/admin

fabric-ca-client enroll -u http://Admin@example.com:password@localhost:7054 -H /root/fabric-ca-files/example.com/admin

mkdir /root/fabric-ca-files/example.com/msp/admincerts/

cp /root/fabric-ca-files/example.com/admin/msp/signcerts/cert.pem /root/fabric-ca-files/example.com/msp/admincerts/

mkdir -p /root/fabric-ca-files/org1.example.com/admin

fabric-ca-client register -H /root/fabric-ca-client-configList/org1/ --id.secret=password

fabric-ca-client enroll -u http://Admin@org1.example.com:password@localhost:7054 -H /root/fabric-ca-files/org1.example.com/admin

fabric-ca-client affiliation list -H /root/fabric-ca-files/org1.example.com/admin

mkdir /root/fabric-ca-files/org1.example.com/msp/admincerts/

cp /root/fabric-ca-files/org1.example.com/admin/msp/signcerts/cert.pem /root/fabric-ca-files/org1.example.com/msp/admincerts/

mkdir fabric-ca-files/org1.example.com/admin/msp/admincerts/

cp /root/fabric-ca-files/org1.example.com/admin/msp/signcerts/cert.pem /root/fabric-ca-files/org1.example.com/admin/msp/admincerts/

mkdir -p /root/fabric-ca-files/org2.example.com/admin

##修改fabric-ca-client-config.yaml文件

fabric-ca-client register -H /root/fabric-ca-client-configList/org2/ --id.secret=password

fabric-ca-client enroll -u http://Admin@org2.example.com:password@localhost:7054 -H /root/fabric-ca-files/org2.example.com/admin

fabric-ca-client affiliation list -H /root/fabric-ca-files/org2.example.com/admin

mkdir /root/fabric-ca-files/org2.example.com/msp/admincerts/

cp /root/fabric-ca-files/org2.example.com/admin/msp/signcerts/cert.pem /root/fabric-ca-files/org2.example.com/msp/admincerts/

mkdir fabric-ca-files/org2.example.com/admin/msp/admincerts/

cp /root/fabric-ca-files/org2.example.com/admin/msp/signcerts/cert.pem /root/fabric-ca-files/org2.example.com/admin/msp/admincerts/

mkdir /root/fabric-ca-files/example.com/orderer

fabric-ca-client register -H /root/fabric-ca-client-configList/orderer/ --id.secret=password

fabric-ca-client enroll -u http://orderer.example.com:password@localhost:7054 -H /root/fabric-ca-files/example.com/orderer

mkdir /root/fabric-ca-files/example.com/orderer/msp/admincerts

cp /root/fabric-ca-files/example.com/admin/msp/signcerts/cert.pem /root/fabric-ca-files/example.com/orderer/msp/admincerts/

mkdir /root/fabric-ca-files/org1.example.com/peer0

##修改fabric-ca-client-config.yaml文件

fabric-ca-client register -H /root/fabric-ca-client-configList/or1pe0/ --id.secret=password

fabric-ca-client enroll -u http://peer0.org1.example.com:password@localhost:7054 -H /root/fabric-ca-files/org1.example.com/peer0

mkdir /root/fabric-ca-files/org1.example.com/peer0/msp/admincerts

cp /root/fabric-ca-files/org1.example.com/admin/msp/signcerts/cert.pem /root/fabric-ca-files/org1.example.com/peer0/msp/admincerts/

mkdir /root/fabric-ca-files/org1.example.com/peer1

##修改fabric-ca-client-config.yaml文件

fabric-ca-client register -H /root/fabric-ca-client-configList/or1pe1/ --id.secret=password

fabric-ca-client enroll -u http://peer1.org1.example.com:password@localhost:7054 -H /root/fabric-ca-files/org1.example.com/peer1

mkdir /root/fabric-ca-files/org1.example.com/peer1/msp/admincerts

cp /root/fabric-ca-files/org1.example.com/admin/msp/signcerts/cert.pem /root/fabric-ca-files/org1.example.com/peer1/msp/admincerts/

mkdir /root/fabric-ca-files/org2.example.com/peer0

##修改fabric-ca-client-config.yaml文件

fabric-ca-client register -H /root/fabric-ca-client-configList/or2pe0/ --id.secret=password

fabric-ca-client enroll -u http://peer0.org2.example.com:password@localhost:7054 -H /root/fabric-ca-files/org2.example.com/peer0

mkdir /root/fabric-ca-files/org2.example.com/peer0/msp/admincerts

cp /root/fabric-ca-files/org2.example.com/admin/msp/signcerts/cert.pem /root/fabric-ca-files/org2.example.com/peer0/msp/admincerts/

mkdir /root/fabric-ca-files/org2.example.com/peer1

##修改fabric-ca-client-config.yaml文件

fabric-ca-client register -H /root/fabric-ca-client-configList/or2pe1/ --id.secret=password

fabric-ca-client enroll -u http://peer0.org2.example.com:password@localhost:7054 -H /root/fabric-ca-files/org2.example.com/peer1

mkdir /root/fabric-ca-files/org2.example.com/peer0/msp/admincerts

cp /root/fabric-ca-files/org2.example.com/admin/msp/signcerts/cert.pem /root/fabric-ca-files/org2.example.com/peer1/msp/admincerts/

# 网络动态添加组织

../bin/cryptogen generate --config=./crypto-config.yaml

export FABRIC\_CFG\_PATH=$PWD

../bin/configtxgen -profile TwoOrgsOrdererGenesis -outputBlock ./channel-artifacts/genesis.block

export CHANNEL\_NAME=orgmychannel

../bin/configtxgen -profile OrgsChannel -outputCreateChannelTx ./channel-artifacts/orgchannel.tx -channelID $CHANNEL\_NAME

../bin/configtxgen -printOrg Org3MSP -profile ./configtx.yaml > channel-artifacts/org3.json

docker-compose -f docker-compose-cli.yaml up -d

docker exec -it cli bash

CORE\_PEER\_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org3.example.com/users/Admin@org3.example.com/msp

CORE\_PEER\_ADDRESS=peer0.org3.example.com:7051

CORE\_PEER\_LOCALMSPID="Org3MSP"

CORE\_PEER\_TLS\_ROOTCERT\_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org3.example.com/peers/peer0.org3.example.com/tls/ca.crt

export CHANNEL\_NAME=orgmychannel

peer channel create -o orderer.example.com:7050 -c $CHANNEL\_NAME -f ./channel-artifacts/orgchannel.tx --tls $CORE\_PEER\_TLS\_ENABLED --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem

cd channel-artifacts

export CORE\_PEER\_TLS\_ROOTCERT\_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem

export CORE\_PEER\_LOCALMSPID="OrdererMSP"

export CORE\_PEER\_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp

export ORDERER\_CA=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem

peer channel fetch config config\_block.pb -o orderer.example.com:7050 -c testchainid --tls --cafile $ORDERER\_CA

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

jq .data.data[0].payload.data.config config\_old.json >config.json

jq -s '.[0] \* {"channel\_group":{"groups":{"Consortiums":{"groups": {"SampleConsortium":{"groups":{"Org3MSP":.[1]}}}}}}}' config.json org3.json > modified\_config.json

configtxlator proto\_encode --input config.json --type common.Config > original\_config.pb

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

configtxlator compute\_update --channel\_id testchainid --original original\_config.pb --updated modified\_config.pb > config\_update.pb

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

echo '{"payload":{"header":{"channel\_header":{"channel\_id":"testchainid", "type":2}},"data":{"config\_update":'$(cat config\_update.json)'}}}' > config\_update\_in\_envelope.json

configtxlator proto\_encode --input config\_update\_in\_envelope.json --type common.Envelope > org3\_update\_in\_envelope.pb

export CORE\_PEER\_LOCALMSPID="OrdererMSP"

export CORE\_PEER\_TLS\_ROOTCERT\_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem

export CORE\_PEER\_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/users/Admin@example.com/msp

export ORDERER\_CA=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem

peer channel signconfigtx -f org3\_update\_in\_envelope.pb

peer channel update -f org3\_update\_in\_envelope.pb -c testchainid -o orderer.example.com:7050 --tls --cafile $ORDERER\_CA

CORE\_PEER\_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org3.example.com/users/Admin@org3.example.com/msp

CORE\_PEER\_ADDRESS=peer0.org3.example.com:7051

CORE\_PEER\_LOCALMSPID="Org3MSP"

CORE\_PEER\_TLS\_ROOTCERT\_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org3.example.com/peers/peer0.org3.example.com/tls/ca.crt

export CHANNEL\_NAME=orgmychannel

peer channel create -o orderer.example.com:7050 -c $CHANNEL\_NAME -f ./channel-artifacts/orgchannel.tx --tls $CORE\_PEER\_TLS\_ENABLED --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem

# Channel动态添加组织

../bin/cryptogen generate --config=./crypto-config.yaml

export FABRIC\_CFG\_PATH=$PWD

../bin/configtxgen -profile TwoOrgsOrdererGenesis -outputBlock ./channel-artifacts/genesis.block

export CHANNEL\_NAME=mychannel

../bin/configtxgen -profile TwoOrgsChannel -outputCreateChannelTx ./channel-artifacts/channel.tx -channelID $CHANNEL\_NAME

../bin/configtxgen -profile TwoOrgsChannel -outputAnchorPeersUpdate ./channel-artifacts/Org1MSPanchors.tx -channelID $CHANNEL\_NAME -asOrg Org1MSP

../bin/configtxgen -profile TwoOrgsChannel -outputAnchorPeersUpdate ./channel-artifacts/Org2MSPanchors.tx -channelID $CHANNEL\_NAME -asOrg Org2MSP

../bin/configtxgen -printOrg Org3MSP -profile ./configtx.yaml > channel-artifacts/org3.json

docker-compose -f docker-compose-cli.yaml up -d

CORE\_PEER\_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp

CORE\_PEER\_ADDRESS=peer0.org1.example.com:7051

CORE\_PEER\_LOCALMSPID="Org1MSP"

CORE\_PEER\_TLS\_ROOTCERT\_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt

docker exec -it cli bash

export CHANNEL\_NAME=mychannel

peer channel create -o orderer.example.com:7050 -c $CHANNEL\_NAME -f ./channel-artifacts/channel.tx --tls $CORE\_PEER\_TLS\_ENABLED --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem

peer channel join -b mychannel.block

peer chaincode install -n mycc -v 1.0 -p github.com/chaincode/chaincode\_example02/go/

peer chaincode instantiate -o orderer.example.com:7050 --tls --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem -C $CHANNEL\_NAME -n mycc -v 1.0 -c '{"Args":["init","a", "100", "b","200"]}' -P "OR ('Org1MSP.peer','Org2MSP.peer')"

peer chaincode query -C $CHANNEL\_NAME -n mycc -c '{"Args":["query","a"]}'

cd channel-artifacts

export ORDERER\_CA=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem

peer channel fetch config config\_block.pb -o orderer.example.com:7050 -c mychannel --tls --cafile $ORDERER\_CA

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

jq .data.data[0].payload.data.config config\_old.json >config.json

jq -s '.[0] \* {"channel\_group":{"groups":{"Application":{"groups": {"Org3MSP":.[1]}}}}}' config.json org3.json > modified\_config.json

configtxlator proto\_encode --input config.json --type common.Config > original\_config.pb

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

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

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

echo '{"payload":{"header":{"channel\_header":{"channel\_id":"mychannel", "type":2}},"data":{"config\_update":'$(cat config\_update.json)'}}}' > config\_update\_in\_envelope.json

configtxlator proto\_encode --input config\_update\_in\_envelope.json --type common.Envelope > org3\_update\_in\_envelope.pb

CORE\_PEER\_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp

CORE\_PEER\_ADDRESS=peer0.org1.example.com:7051

CORE\_PEER\_LOCALMSPID="Org1MSP"

CORE\_PEER\_TLS\_ROOTCERT\_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt

peer channel signconfigtx -f org3\_update\_in\_envelope.pb

CORE\_PEER\_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org2.example.com/users/Admin@org2.example.com/msp

CORE\_PEER\_ADDRESS=peer0.org2.example.com:7051

CORE\_PEER\_LOCALMSPID="Org2MSP"

CORE\_PEER\_TLS\_ROOTCERT\_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt

peer channel signconfigtx -f org3\_update\_in\_envelope.pb

export ORDERER\_CA=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem

peer channel update -f org3\_update\_in\_envelope.pb -c mychannel -o orderer.example.com:7050 --tls --cafile $ORDERER\_CA

docker exec -it cli bash

CORE\_PEER\_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org3.example.com/users/Admin@org3.example.com/msp

CORE\_PEER\_ADDRESS=peer0.org3.example.com:7051

CORE\_PEER\_LOCALMSPID="Org3MSP"

CORE\_PEER\_TLS\_ROOTCERT\_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org3.example.com/peers/peer0.org3.example.com/tls/ca.crt

peer channel join -b mychannel.block

peer chaincode install -n mycc -v 1.0 -p github.com/chaincode/chaincode\_example02/go/

export CHANNEL\_NAME=mychannel

peer chaincode query -C mychannel -n mycc -c '{"Args":["query","a"]}'

export CHANNEL\_NAME=orgmychannel

peer channel create -o orderer.example.com:7050 -c $CHANNEL\_NAME -f ./channel-artifacts/orgchannel.tx --tls $CORE\_PEER\_TLS\_ENABLED --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem

docker stop $(docker ps -a -q)

docker rm $(docker ps -a -q)

docker-compose -f docker-compose-zookeeper.yaml up -d

docker-compose -f docker-compose-kafka.yaml up -d

docker-compose -f docker-compose-orderer-kafka.yaml up -d

docker-compose -f docker-compose-peer-kafka.yaml up -d

CORE\_PEER\_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp

CORE\_PEER\_ADDRESS=peer0.org1.example.com:7051

CORE\_PEER\_LOCALMSPID="Org1MSP"

CORE\_PEER\_TLS\_ROOTCERT\_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt

docker exec -it cli bash

export CHANNEL\_NAME=mychannel

peer channel create -o orderer3.example.com:7050 -c $CHANNEL\_NAME -f ./channel-artifacts/channel.tx --tls $CORE\_PEER\_TLS\_ENABLED --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer3.example.com/msp/tlscacerts/tlsca.example.com-cert.pem

export ORDERER\_CA=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer2.example.com/msp/tlscacerts/tlsca.example.com-cert.pem

export CHANNEL\_NAME=mychannel

peer channel fetch 0 mychannel.block -o orderer2.example.com:7050 -c $CHANNEL\_NAME --tls --cafile $ORDERER\_CA

peer channel join -b mychannel.block

peer chaincode install -n mycc -v 1.0 -p github.com/chaincode/chaincode\_example02/go/

peer chaincode instantiate -o orderer3.example.com:7050 --tls --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer3.example.com/msp/tlscacerts/tlsca.example.com-cert.pem -C $CHANNEL\_NAME -n mycc -v 1.0 -c '{"Args":["init","a", "100", "b","200"]}' -P "OR ('Org1MSP.peer','Org2MSP.peer')"

peer chaincode instantiate -o orderer1.example.com:7050 --tls --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer1.example.com/msp/tlscacerts/tlsca.example.com-cert.pem -C $CHANNEL\_NAME -n mycc -v 1.0 -c '{"Args":["init","a", "100", "b","200"]}' -P "OR ('Org1MSP.peer','Org2MSP.peer')"

peer chaincode query -C $CHANNEL\_NAME -n mycc -c '{"Args":["query","a"]}'