（1）配置和启动fabric网络

1，清理工作

删掉crypto-config 文件夹，并且删掉channel-artifacts文件夹里面的文件

再把docker中运行的容器都删掉（docker镜像相当于类，容器相当于类）

sudo docker rm $(sudo docker ps -aq)

2，利用cryptogen工具和crypto-config.yaml配置文件生成peer和order节点的证书文件

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

利用configtxgen工具和configtx.yaml配置文件生成order创世区块genesis.block，通道配置文件channel.tx和组织锚节点Org1MSPanchors.tx，Org2MSPanchors.tx

$CHANNEL_NAME是命令行传入的通道名参数

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

.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

3，编写或者用docker-compose-e2e-template.yaml自动生成docker-compose-e2e.yaml文件，用于启动各种容器, 若启动了CA容器，则需要替换文件中的公私钥，然后

docker-compose -f docker-compose-e2e.yaml up -d

最后的-d是不查看日志的意思，up换为down是停止，移除容器

docker ps查看已经启动的容器

4，连接peer进行各种操作，cli连接哪个peer在容器配置文件docker-compose-e2e.yaml中已经配置好该cli 的 peer的环境. 只要进入cli容器就行了

docker exec -it cli bash

退出cli容器 exit

5，使用channel.tx创建通道

peer channel create -o orderer.example.com:7050 -c $CHANNEL_NAME -f ./channel-artifacts/channel.tx

然后 ls 就能看到该通道的创世区块 mychannel.block. mychannel是$CHANNEL_NAME中假定输入的通道名称

6，使用mychannel.block 使当前节点加入通道中

peer channel join -b my channel.block

若要查看该节点进入哪些通道使用peer channel list

切换cli的连接节点，直接在命令行中改变节点的环境

CORE_PEER_ADDRESS=peer1.org1.example.com:7051

CORE_PEER_LOCALMSPID="Org1MSP"

CORE_PEER_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp //用户节点还是admin节点都设置Admin那个

//如果没有TLS则不用设置下面的

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

7，更新锚节点，需要进入锚节点内更新

peer channel update -o orderer.example.com:7050 -c $CHANNEL_NAME -f ./channel-artifacts/Org1MSPanchors.tx

8，安装智能合约，安装时不需要指定orderer，mycc是智能合约名称， -p 后面是地址。每个需要安装智能合约的节点都要安装一次，一般是背书节点

peer chaincode install -n mycc -v 1.0 -p github.com/hyperledger/fabric/examples/chaincode/go/chaincode_example02

9，实例化智能合约，只需要一次，可以在实例化时指定初始化参数和背书策略。实例化后会有一个运行链码的独立的docker容器，而实例化策略一般是链码打包时指定的，实例化策略默认是任何msp的admin来进行链码的实例化。

peer chaincode instantiate -o orderer.example.com:7050 -C $CHANNEL_NAME -n mycc -v 1.0 -c '{"Args":["init","a","100","b","200"]}' -P "OR('Org1MSP.peer','Org2MSP.peer')"

10，调用链码

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

peer chaincode invoke -o orderer.example.com:7050 -C $CHANNEL_NAME -n mycc -c '{"Args":["invoke","a","b","10"]}'

（2）链码的打包和安装

若链码拥有多个所有者，需要多个所有者签名

1，创建一个已经签名的包，-i后面指定的实例化策略，下面意思是只能由Org1MSP中的admin实例化。实例化策略默认是任何msp的admin来进行链码的实例化。

peer chaincode package -n mycc -p github.com/hyperledger/fabric/examples/chaincode/go/chaincode_example02 -v 1.0 -s -S -i "AND('Org1MSP.admin')" ccpack.out

2，其他所有者签名，先设置环境连接其他peer节点，然后依此签名，输出一个包

peer chaincode signpackage ccpack.out signedccpack.out

3，安装链码

peer chaincode install signedccpack.out

4，实例化时进入实例化策略指定的那个peer下面实例化

查看peer下面有没有安装链码

peer chaincode list --installed

（3）链码升级

在之前的链码文件上做了改动，进行链码升级不影响之前的账本数据，若重新安装，那么就查不到之前的账本数据

1，需要在安装链码的节点上都安装一遍新的链码，链码名称不变，版本号升级，由原来的1.0变为1.1

peer chaincode install -n mycc -v 1.1 -p github.com/hyperledger/fabric/examples/chaincode/go/chaincode_example02

2，升级链码，与instantiate一样，需要init函数，init函数可以是空函数，就不用写参数了

peer chaincode upgrade -o orderer.wisedu.com:7050 -C mychannel -n mycc -v 1.1 -c '{"Args":["init"]}' -P "OR('Org1MSP.peer','Org2MSP.peer','Org3MSP.peer')"

（4）创建多通道

1，使用configtxgen工具与configtx.yaml配置文件中的TwoOrgsChannel模板, 来生成新建通道的配置交易文件，注意新创建的通道名不要与原来的通道名重合

./bin/configtxgen -profile TwoOrgsChannel -outputCreateChannelTx ./channel-artifacts/yourchannel.tx -channelID yourchannel

2.使用configtxgen工具与configtx.yaml配置文件生成锚节点配置更新文件，注意每个组织都需要分别生成且注意指定对应的组织名称

./bin/configtxgen -profile TwoOrgsChannel -outputAnchorPeersUpdate ./channel-artifacts/yourOrg2MSPanchors.tx -channelID yourchannel -asOrg Org2MSP

3.进入cli容器：

sudo docker exec -it cli bash

4.根据yourchannel.tx文件创建通道：

peer channel create -o orderer.wisedu.com:7050 -c yourchannel -f ./channel-artifacts/yourchannel.tx

5.依次将该应用通道中所包含的成员节点加入到通道中：

peer channel join -b yourchannel.block

6.更新各锚节点配置：

peer channel update -o orderer.wisedu.com:7050 -c yourchannel -f ./channel-artifacts/yourOrg2MSPanchors.tx

7 安装，初始化链码